sensors

Special Issue Reprint

IoT Network Security

Edited by
Jian Li

mdpi.com/journal/sensors

MDPI

IoT Network Security

IoT Network Security

Guest Editor

Jian Li

MDPI

Basel • Beijing • Wuhan • Barcelona • Belgrade • Novi Sad • Cluj • Manchester

Guest Editor
Jian Li
School of Cyberspace Security
Beijing University of Posts
and Telecommunications
Beijing
China

Editorial Office
MDPI AG
Grosspeteranlage 5
4052 Basel, Switzerland

This is a reprint of the Special Issue, published open access by the journal *Sensors* (ISSN 1424-8220), freely accessible at: https://www.mdpi.com/journal/sensors/special_issues/94WQYD6IAL.

For citation purposes, cite each article independently as indicated on the article page online and as indicated below:

Lastname, A.A.; Lastname, B.B. Article Title. *Journal Name* **Year**, *Volume Number*, Page Range.

ISBN 978-3-7258-3837-0 (Hbk)
ISBN 978-3-7258-3838-7 (PDF)
https://doi.org/10.3390/books978-3-7258-3838-7

Contents

sensors

MDPI

Article

A Blockchain Copyright Protection Scheme Based on CP-ABE Scheme with Policy Update

Jian Jiang, Yulong Gao *, Yufei Gong and Zhengtao Jiang

State Key Laboratory of Media Convergence and Communication, Communication University of China, Beijing 100024, China; jiangjian@cdcgc.com.cn (J.J.); 2021212063037@cuc.edu.cn (Y.G.); z.t.jiang@163.com (Z.J.)
* Correspondence: ylgao@cuc.edu.cn

Abstract: Although the copyright protection schemes supported by blockchain have significantly changed traditional copyright data management, there are still some data security challenges that cannot be ignored, especially the secure access and controllable management of copyright data. Quantum computing attacks also pose a threat to its security. Targeting these issues, we design and propose a blockchain copyright protection scheme based on attribute-based encryption (ABE). In this scheme, the security advantages of blockchain technology are utilized to ensure the authenticity and integrity of copyright data. Based on lattice cryptography and the decision ring learning with errors (R-LWE) problem, a new ABE algorithm that supports searchable ciphertext and policy updates is designed. Then, we introduce it into the blockchain copyright protection scheme, which enables secure access to copyright data and fine-grained control. In addition, the lattice cryptography can strengthen this scheme against quantum attacks. Through security analysis, our scheme can prove to be secure against adaptive chosen keyword attacks, selective chosen plaintext attacks, and adaptive chosen policy attacks in the random oracle model. More importantly, the comparison analysis and experimental results show that our proposed approach has lower computation costs and storage costs. Therefore, our scheme has better security and performance in copyright protection.

Keywords: blockchain; copyright protection; attribute-based encryption; lattice; decision R-LWE

Citation: Jiang, J.; Gao, Y.; Gong, Y.; Jiang, Z. A Blockchain Copyright Protection Scheme Based on CP-ABE Scheme with Policy Update. *Sensors* **2024**, *24*, 4493. https://doi.org/10.3390/s24144493

Academic Editor: Alessandra Rizzardi

Received: 5 June 2024
Revised: 3 July 2024
Accepted: 7 July 2024
Published: 11 July 2024

1. Introduction

Traditional digital copyright protection faces multiple challenges, with frequent piracy and infringement. In the digital era, the reproduction and dissemination of works become extremely easy, leading to the prosperity of the piracy market, causing huge economic losses to copyright owners, and it is still very difficult to trace and maintain copyright on the Internet. At the same time, the vulnerabilities and risks of the centralized platform cannot be ignored. The centralized copyright protection platform is vulnerable to hacker attacks, resulting in copyright information being tampered with or disclosed [1]. In view of the many problems existing in copyright protection, blockchain technology, as an emerging technical means, is widely used in the field of digital copyright protection. Blockchain technology provides a new solution for digital copyright protection through the characteristics of decentralization, tamper proof, and transparency [2]. First, blockchain technology can ensure the uniqueness and tamper resistance of digital content. By storing copyright information on distributed ledgers and using encryption algorithms for verification, the ownership and integrity of works can be effectively protected [3]. Secondly, blockchain can establish smart contracts to manage and protect copyright. Smart contracts can automate the implementation of copyright use rules and trading conditions, reducing the risk of copyright disputes. In addition, blockchain technology can also realize the transparency and traceability of copyright information, providing a fairer and more reliable environment for copyright transactions [4]. In 2019, Zhang et al. proposed a decentralized digital rights management and transaction system architecture based on blockchain technology [5], which is built in a way that consumes fewer resources and throughput.

Although blockchain has shown great potential in digital copyright protection, there are also some new data security challenges, especially data privacy protection [6]. Consensus and storage of copyright data in the blockchain public ledger while ensuring sensitive privacy is not compromised and resisting quantum computing attack is an urgent issue that needs to be addressed [7,8]. Attribute-based encryption (ABE) is an extension of identity-based encryption and is applied in areas such as the Internet of Things and cloud computing [9]. It has encryption advantages and can achieve user privacy and security management in access control. More specifically, in the ABE scheme, the encryptor formulates access policies based on the user's characteristic information. The user determines whether they can decrypt the ciphertext based on their own characteristics, and the user's characteristic information is defined as an attribute. Wang et al. proposed a special ABE scheme with revocation support and flexible access structure on the lattice by using a binary tree [10]. Meanwhile, public-key encryption with keyword search (PEKS) enables users to search target encrypted data by keywords, which increases data privacy [11]. In 2019, Liu et al. proposed a KS-ABE scheme to enhance the security of ciphertext search in cloud storage. Meanwhile, this scheme can resist quantum attacks under the LWE assumption [12]. In 2020, Behnia et al. proposed two PEKS schemes using the NTRU lattice and the LWE lattice and implemented these schemes on the Amazon Web Services cloud infrastructure [13]. In 2021, Zuo et al. proposed a blockchain-based ciphertext-policy attribute-based encryption (CP-ABE) scheme for cloud data secure sharing without relying on any trusted third parties [14]. Then, Zhang et al. proposed a new lattice-based ABE algorithm, which is used as an access control method for blockchain data to protect its security [15].

In practical applications, CP-ABE is more suitable for scenarios of distributed storage and update policy, as it allows data owners to define authorized attribute sets as access control policies embedded in ciphertext [16]. Each user can obtain a key corresponding to their attribute set, and the condition for successful decryption is that the attribute set meets the access policy. Therefore, CP-ABE is often combined with blockchain technology to achieve data security [17]. Moreover, the policy updates of decision-makers on user attributes are constantly changing, so policy updates are particularly important in the ABE scheme's practical application. This approach can ensure the privacy and security of blockchain-based copyright data and update user access policies to ensure data security [18]. Therefore, in this study, we design a blockchain copyright protection scheme based on CP-ABE. In our scheme, the security advantages of blockchain technology are utilized to ensure the authenticity and integrity of copyright data. Based on lattice cryptography and the decision ring learning with errors (R-LWE) problem, we propose a new CP-ABE algorithm that supports searchable ciphertext and policy updates. Then, we introduce it into the blockchain copyright protection scheme. This scheme enables secure access to copyright data and fine-grained control. The security of the proposed scheme can be reduced to the $\gamma - SVP$ problem and significantly improve users' privacy and system data security in blockchain-enabled systems and applications.

Then, the main contributions of this paper are summarized as follows:

- We propose a lattice-based CP-ABE scheme to improve anti-quantum security for blockchain copyright protection. The lattice assumption can make our scheme more secure against quantum attacks.
- We construct a new blockchain copyright protection scheme based on the CP-ABE scheme with a policy update, which enables secure access to copyright data and fine-grained control. We provide detailed descriptions of the copyright protection processes. The proposed scheme can significantly protect the copyright and system security.
- We prove our scheme is secure against adaptive chosen keyword attacks, selective chosen plaintext attacks, and adaptive chosen policy attacks in the random oracle model. Furthermore, we compare the proposed scheme with similar literature that shows that it has lower computation costs and storage costs.

The rest of the paper is organized as follows. In Section 2, some definitions and lemmas of the lattice theories are presented. In Section 3, we propose a CP-ABE scheme with policy update. The security proof is presented in Section 4. In Section 5, we give the performance analysis and efficiency comparison of our scheme with other schemes. In Section 6, a blockchain copyright protection scheme based on CP-ABE is designed. The conclusions are provided in Section 7.

2. Preliminaries

Some definitions and lemmas of the lattice theories are presented in this section, which are in relation to our scheme.

$\mathbb{R}, \mathbb{Z}$ denote the set of all reals and the set of positive integers, respectively. Let $\mathbb{R}^m$ be the m-dimensional Euclidean vector space with its usual topology. $m \in \mathbb{Z}, n \in \mathbb{Z}, m \geq n$ and L denote the Λ lattice; the orthogonal lattice corresponding to Λ is represented by $\Lambda^{\perp}$, vector $\mathbf{x} = (x_1, x_2, \cdots, x_{n-1}, x_n)^T$ in the space $\mathbb{R}^m$, and its Euclidean norm $\|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2 + \cdots x_{n-1}^2 + x_n^2}$.

Definition 1. *(Lattice) Given n-linearly independent vectors, lattice L generated by them is the set of vectors.*

$$L(\mathbf{v_1}, \mathbf{v_2}, \cdots, \mathbf{v_n}) = \left\{ \sum_{i=1}^{n} a_i \mathbf{v_i} | a_i \in \mathbb{Z}, i = 1, \cdots, n \right\} \tag{1}$$

$V = [\mathbf{v_1}, \mathbf{v_2}, \cdots, \mathbf{v_n}]$ is known as the basis of the lattice L. The same lattice can be represented by different lattice bases. Given a prime number q, a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ defines:

$$\Lambda_q\Big(\mathbf{A}\Big) = \Big\{ y \in \mathbb{Z}^m \Big| y = \mathbf{A}^T \mathbf{x} \bmod q, \mathbf{x} \in \mathbb{Z}^n \Big\}, \tag{2}$$

$$\Lambda_q{}^{\perp}(\mathbf{A}) = \{ y \in \mathbb{Z}^m | \mathbf{A}y = 0 \bmod q \} \tag{3}$$

Definition 2. *(Lattice SIS problem) Given an integer q, a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, and a real constant $v > 0$, find a nonzero vector $\mathbf{x} \in \mathbb{Z}^m$ such that $\mathbf{A}\mathbf{x} \equiv 0 \bmod q$ and $\|\mathbf{x}\| \leq v$.*

Based on the hardness of the SIS problem, for any polynomial-bounded m, v, and any prime $q \geq v \cdot \omega \sqrt{n \log n}$, solving SIS on the average is as hard as approximating the shortest independent vector problem (SIVP) in the worst case.

Ring learning with errors (R-LWE) was proposed by Lyubashevsky [19]. It operates on the ring $Z_q[x]/(f)$, where f is an irreducible polynomial, and q is a prime. In most cases, $f = x^n + 1$, where n is a power of 2.

Definition 3. *(Decision R-LWE problem) For $R = Z_q[x]/(x^n + 1)$, $n = 2^k, k \geq 1$, $q = 1 \bmod 2n$, $\mathbf{a} \in R_q$, $\mathbf{a}$ and $\mathbf{S}$ are uniformly and randomly selected, $\mathbf{e} \in R$ is an error vector that follows a discrete Gaussian distribution Ψ_a. Let $\mathbf{b} = \mathbf{a} \cdot \mathbf{s} + \mathbf{e}$, $\mathbf{b} \in R_q$. The decision R-LWE problem is to distinguish between vector group ($\boldsymbol{a}$,$\boldsymbol{b}$) and vector group uniformly and randomly selected on R_q^2.*

Definition 4. *(Random oracle model based on decision R-LWE problem) Random oracle O will be sampled with equal probability through two samplers defined as follows:*

Pseudo-random oracle O_1 outputs pseudo-random sampling $(\mathbf{a}, \mathbf{b}) = (\mathbf{a}, \mathbf{a} \cdot \mathbf{x} + \mathbf{e}) \in R_q^2$, where $\mathbf{a} \in R_q$ is a uniform random vector, $\mathbf{x} \in R_q$, the small error term $\mathbf{e}$ satisfies discrete Gaussian distribution.

True random oracle O_2 outputs uniformly random and mutually independent sample $(\mathbf{a}, \mathbf{b}) \in R_q^2$ on the domain R_q^2, which are truly completely random.

The decision R-LWE problem allows adversaries to query the oracle many times and make guesses O_r, $r \in \{1, 2\}$ about the sampler based on the obtained samples. Due to

the difficulty of the decision R-LWE problem, the advantages $\Pr[r = 1] - \Pr[r = 2]$ of any adversary's guess O_r on O are negligible.

Lemma 1. *For a lattice L with dimensional m and rank n,* $\mathbf{c} \in \mathbb{R}^m$*, positive real* $\varepsilon < \exp(-4\pi)$ *and* $s \geq \eta_\varepsilon(L)$ *for random* $\mathbf{x} \in L$ *such that* $D_{L,s,\mathbf{c}}(\mathbf{x}) \leq 1 + \varepsilon/(1-\varepsilon)2^{-n}$.

Lemma 2. *Let* $q > 2$*, a matrix* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$*,* $\boldsymbol{B}$ *is the basis of* $\Lambda_q^{\perp}(\mathbf{A})$*, and Gaussian parameter* $s \geq \left\|\widetilde{\mathbf{B}}\right\| \omega(logm)$*. Then, any vector* $\mathbf{y} \in \mathbb{Z}_q^n$ *algorithm SamplePre(*$\boldsymbol{A}$*,* $\boldsymbol{B}$*,* $\boldsymbol{y}$*, s) outputs a vector* $\mathbf{e} \in Z_q^m$ *from a distribution that is statistically close to* $D_{\Lambda_q^{\perp}(\mathbf{A}),s}(\mathrm{x})$.

Lemma 3. *For any prime* $q = poly(n)$ *and any* $m \geq 5n\lg q$*, there is a probabilistic polynomial-time algorithm* $TrapGen(1^n)$ *that outputs a matrix* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *and a full-rank set* $\mathbf{S} \subset \Lambda^{\perp}(\mathbf{A}, q)$*. The distribution of A is statistically close to uniform over* $\mathbb{Z}_q^{n \times m}$ *and the length* $\|\mathbf{S}\| \leq L = m^{1+\varepsilon} \wedge \varepsilon > 0$.

Lemma 4. *Given a matrix* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *and an m-dimensional lattice* $\Lambda_q^{\perp}(\mathbf{A})$*, input a basis T of the lattice* $\Lambda_q^{\perp}(\mathbf{A})$*, which has a nonsingular matrix* $\mathbf{R} = \mathbf{T}^{-1}$ *and* $\mathbf{R} \in \mathbb{Z}^{m \times m}$*, then input Gaussian parameter* $s \geq \left\|\widetilde{\mathbf{T}}\right\| m^d \omega(\lg^{d+1}(m))$*, BasisDel(*$\boldsymbol{A}$*,* $\boldsymbol{R}$*,* $\boldsymbol{T}$*, s) can output a basis* $\boldsymbol{B}$ *of* $\Lambda^{\perp}(\mathbf{A}\mathbf{R}^{-1})$ *with overwhelming probability* $\left\|\widetilde{\mathbf{B}}\right\| \leq s\sqrt{m}$.

3. CP-ABE Scheme with Policy Update

3.1. Formal Definition

Our ABE scheme consists of six probabilistic polynomial time (PPT) algorithms, such as Setup, Index sharing, KeyGen, Encrypt, Decrypt, and Update, as follows:

(1) **Setup.** The algorithm takes security parameters as input, and the system generates public key *PK* and master key *MK*. Among them, the *MK* is kept by the system.
(2) **Index sharing.** The algorithm mainly includes Index-generation, Trapdoor algorithm, and Test algorithm, which returns a result that stores data and data indexes.
(3) **KeyGen.** The algorithm takes *PK*, *MK*, and user access control policy *Plc* as input. The system generates secret key *sk* for users according to attribute policy *T*.
(4) **Encrypt.** The algorithm takes public key *PK*, attribute policy *T*, and message *M* as input and outputs ciphertext C_1.
(5) **Decrypt.** The algorithm takes the public key *PK*, secret key *sk*, and ciphertext *C* as inputs. Only if the access control policy *Plc* matches the user attribute policy *T* does the algorithm output plaintext *M*.
(6) **Update.** Input the main public key, update the access policy, ciphertext, and random trapdoors used in the encryption algorithms, and output new ciphertext C_2.

3.2. Our Proposed Scheme

According to the algorithm definition in Section 3.1, the specific process of our proposed CP-ABE scheme with policy update is as follows.

Setup.

(1) Set user attribute set $\mathrm{att}_i \in S, i = 1, 2, ..., N$ and the access policy Plc_i corresponding to its attribute $\mathrm{att}_i \in S$.
(2) Calculate $\sigma_f \leftarrow 1.13\sqrt{q/2N}$, select parameters $f, g \leftarrow D_{N,\sigma_f}$. Then, calculate the Gram–Schmidt norm as follows.

$$||B_{\mathrm{f},g}|| \leftarrow MAX(|||(g, -f)||, ||(\frac{q\overline{f}}{f * \overline{f} + g * \overline{g}}, \frac{q\overline{g}}{f * \overline{f} + g * \overline{g}})||) \tag{4}$$

If $|B_{\mathrm{f},g}|| \leq 1.13\sqrt{q}$, continue; otherwise, go back to select parameters $f, g \leftarrow D_{N,\sigma_f}$ again.

(3) Calculate ρ_f, ρ_g, R_f, R_g by using the extended Euclidean algorithm as follows. $\rho_f \cdot f = R_f \bmod (x_N + 1), \rho_g \cdot g = R_g \bmod (x_N + 1)$, where $\rho_f, \rho_g \in \Re, R_f, R_g \in Z$. If $\gcd(R_f, R_g) \neq 1$ or $\gcd(R_f, q) \neq 1$, go back to select parameters; otherwise, continue.

(4) Run *extended Euclidean* algorithm to get u, v, and satisfying $u \cdot R_f + v \cdot R_g = 1$ and $u, v \in Z$. Calculate $F = q \cdot v \cdot \rho_g$, $G = q \cdot u \cdot \rho_f$, $k = \left\lceil (F * \overline{g} + G * \overline{g}) / (f * \overline{f} + g * \overline{g}) \right\rceil \in \Re_q$. Thus, reduce F and G as follows: $F \leftarrow F - k * f, G \leftarrow G - k * g$.

(5) Calculate $h = g * f^{-1} \bmod q$, $h \in \Re_q$, $B = \begin{pmatrix} A(g) & -A(f) \\ A(G) & -A(F) \end{pmatrix}$, $B \in Z_q^{2N \times 2N}$. Output public key $PK = \{h\}$ and master key $MK = \{B\}$.

Index sharing. Index sharing includes the *Index-generation* algorithm, *Trapdoor* algorithm, and *Test* algorithm. During the index-sharing process, the data owner first establishes an index trapdoor to generate a data index and records the data index in the information storage. When data users need to obtain data, they send a request to search the information storage and extract the corresponding data index to obtain the corresponding search results.

(1) **Index-generation.** N is a power-of-two integer, and the Trapdoor algorithm corresponds to the encrypted keywords $k \in \{0,1\}^*$ and $t_1 = H_1(k)$, $H_1 : \{0,1\}^* \to Z_q^N$. Then, randomly select parameters r_1, d_1, d_2, ω_1, where $r_1, d_1, d_2 \leftarrow \{-1, 0, 1\}$, $\omega_1 \leftarrow \{0,1\}^N$. At last, calculate $I_1 = r_1 * h + d_1 \in \Re_q$, $I_2 = r_1 * t_1 + d_2 + \lfloor q/2 \rfloor \omega_1 \in \Re_q$ and generate index results $S_k = \{I_1, I_2, H_1(\omega_1, I_2)\}$.

(2) **Trapdoor.** Run *SamplePre* algorithm $(s, T_k) \leftarrow SamplePre(B.\sigma, (t_1, 0))$, which satisfies $s + T_k * h = t_1$. Output corresponding trapdoor T_k for keywords k.

(3) **Test.** Calculate $y = \lfloor 2(I_2 - I_1 * T_k)/q \rfloor$; If $H_1(y, I_2) = H_1(\omega_1, I_2)$, and return d = 1; otherwise, return d = 0.

Keygen. Calculate the hash value of the attribute access control policy $t_2 \leftarrow H_3(Plc_i) \in Z_q^N$. Run *SamplePre* algorithm $(s_1, s_2) \leftarrow SamplerPre(B, \delta, (t_2, 0))$, which satisfies $s_1 + s_2 * h = t_2$. In this way, output the secret key $sk = s_2$.

Encrypt. Randomly select $r_2, d_3, d_4 \leftarrow \{-1, 0, 1\}, \omega_2 \leftarrow \{0,1\}^N$. Calculate $u = r_2 * h + d_3$, $v = r_2 * t_2 + d_4 + \lfloor q/2 \rfloor \omega_2$. Then, calculate $c = m \oplus H_4(\omega_2, att_i, v)$, and return the ciphertext $\{u, v, c\}$.

Decrypt. Calculate $\chi = v - u \cdot s_2$, $\omega_2 \leftarrow \lceil 2\chi/q \rceil$. Output the original message $m = c \oplus H_1(\omega_2, att_i, v)$.

Update. Define access structure $\hat{W} = \hat{W}^+ \cup \hat{W}^-$, message $m = \{m_0, m_1, ..., m_{n-1}\} \in \{0,1\}^n$, which is expressed as polynomial $m(x) = m_0 + m_1 x + ... + m_{n-1} x^{n-1} \in R_q$; randomly and uniformly sample (B_i^+, B_i^-) each attribute $x_i \in S$, and $B_i^+, B_i^- \leftarrow R_q^{l \times m}$; select random numbers $s, \beta \leftarrow R_q$ and sample $e \leftarrow D_{R,\sigma}$, $e_A \leftarrow D_{R_q^m,\sigma}$, calculate $c_1 = \beta s + e = m[q/2]$ and $c_A = A^T s + e_A \in R_q^m$; for each attribute $x_i \in S$, calculate separately based on the following conditions.

(1) If $x_i \in \hat{W}^+$, sample $e_i \leftarrow D_{R_q^m,\sigma}$ and calculate $c_i = (B_i^+)^T s + e_i \in R_q^m$;

(2) If $x_i \in \hat{W}^-$, sample $e_i \leftarrow D_{R_q^m,\sigma}$ and calculate $c_i = (B_i^-)^T s + e_i \in R_q^m$;

(3) If $x_i \in \hat{W}$, sample $e_i{}^+, e_i^- \leftarrow D_{R_q^m,\sigma}$, calculate $c_i^+ = (B_i^+)^T s + e_i^+ \in R_q^m$ and $c_i^- = (B_i^-)^T s + e_i^- \in R_q^m$;

At last, return the updated ciphertext $C' = (\hat{W}, c_A, \{c_i\}_{x_i \in \hat{W}}, (c_i^+, c_i^-)_{x_i \in X \setminus \hat{W}}, c_1)$.

4. Security Analysis

4.1. Correctness

Theorem 1. *The proposed CP-ABE scheme satisfies the correctness of keyword indexing.*

Proof. Given the master key $MK = \{B\}$, public key $PK = \{h\}$, data index structure $s_k = \{I_1, I_2, H_1(\omega_1, I_2)\}$, and keyword trapdoor T_k, this scheme is the output of the correct

dependency test algorithm $d = 1$. In this calculation process, the parameter r_1, d_1, d_2, T_k are short vectors, so $r_1 * s + d_2 - T_k * d_1$ in the interval $(-q/4, q/4)$. And

$$\begin{aligned} I_2 - I_1 * T_k &= (r_1 * t_1 + d_2 + \lfloor q/2 \rfloor \omega_1) - (r_1 * h + d_1) * T_k \\ &= r_1 * s + d_2 + \lfloor q/2 \rfloor \omega_1 - T_k * d_1 \end{aligned} \tag{5}$$

According to the Formula (5), we have $\lceil 2(I_2 - I_1 * T_k)/q \rceil = \omega_i$. Thus, it is proven that the keyword index in our ABE scheme satisfies correctness. □

Theorem 2. *The proposed ABE scheme satisfies the correctness of the policy update.*

Proof. If the access policy formulated by the decision-maker is consistent with the user's attribute set $x_i \in S$, then there is $S \cap W^+ = W^+$, $S \cap W^- = \varnothing$. If there is $\widetilde{B}_i \in \{B_i^+, B_i^-\}$

$$\begin{aligned} a &= w_A^T (A^T s) + w_A^T e_A + \sum_{i \in [l]} w_i^T (\widetilde{B}_i^T s) + \sum_{j \in [l]} w_j^T e_j \\ &= (A w_A)^T \mathrm{s} + w_A^T e_A + \sum_{i \in [l]} ((\widetilde{B}_i w_i)^T s) + \sum_{j \in [l]} w_j^T e_j \\ &= \beta s + w_A^T e_A + \sum_{i \in [l]} w_i^T e_i \end{aligned} \tag{6}$$

$\mu' = c_1 - a$, so we can have

$$\mu' = c_1 - (\beta s + w_A^T e_A + \sum_{i \in [l]} w_i^T e_i) \approx \mu[q/2] \tag{7}$$

According to the above Formulas (6) and (7), it is proven that the proposed ABE scheme satisfies the correctness of the policy update. □

4.2. ABE Keyword Index Security

Theorem 3. *The proposed ABE scheme satisfies keyword index security, which satisfies the indistinguishability against adaptive chosen keyword attack (IND-CKA) in the random oracle model.*

Proof. Assume Eve is a polynomial-time malicious adversary, and Charlie is a challenger who wants to solve the hard problem with the query results from Eve. Adversary Eve first specifies the challenge keywords k_0 and k_1. Adversary Eve and challenger Charlie conduct the following query response game. □

(1) **Setup.** Challenger Charlie sets the algorithm to generate a public key $PK = \{h\}$ and a master key $MK = \{B\}$; that is, Challenger Charlie saves the master key and sends the public key to adversary Eve.
(2) **Queries 1.** Adversary Eve queries about hash query, index generation query, and trapdoor query with polynomial time. The specific process is as follows.

***H_1*-query.** Challenger Charlie initializes two empty lists, L_1 and L_2, to save the query results. Adversary Eve queries the hash function H_1 of the non-target access control policy. Challenger Charlie initializes empty lists and uses them to store the query results of the hash function. If the result is already on the lists L_1 and L_2, Charlie will return the result and give it to Eve; if the result does not appear on the list, Challenger Charlie will randomly select parameters r'_2, d'_4, ω'_2, where $r'_2, d'_4 \leftarrow \{-1, 0, 1\}$, $\omega'_2 \leftarrow \{0, 1\}^N$, and calculate $h' = H_1(\omega'_2, att'_i, r'_2 * t'_2 + d'_4 + \lfloor q/2 \rfloor \omega'_2)$. Challenger Charlie sends the $t'_2 = H_1(Plc_i)$ and h' to adversary Eve and respectively stores the new result (k_i, t'_2) and $(r'_2, d'_4, \omega'_2, h')$ in list L_1 and list L_2.

Index generation query. Adversary Eve queries the Index-generation algorithm for keyword k_i. Challenger Charlie initializes a new list L_3 to store the index-generated query

results. When the generated result already exists in L_1, Challenger Charlie directly gives the result s'_{k_i} to adversary Eve. Otherwise, Challenger Charlie first performs a hash H_1 algorithm to get I'_2 and h', then randomly selects $d'_1 \leftarrow \{-1, 0, 1\}$ and calculates $I'_1 = r'_1 * h' + d'_1$, and finally gives the result s'_{k_i} to adversary Eve and stores $(k_i, I'_1, I'_2, t'_1, h', s'_{k_i})$ in L_3.

Trapdoor query. Adversary Eve performs a polynomial time inquiry on the Trapdoor algorithm of the attribute set S'. Assuming that adversary Eve has already inquired about the keyword, challenger Charlie runs the Trapdoor algorithm and generates T'_k using the sampling algorithm $(S', T'_k) \leftarrow SamplerPre(B, \sigma, (t'_1, 0))$. Afterward, Charlie returns T'_k to adversary Eve.

(3) **Challenge.** Adversary Eve randomly selects keywords k_1 and k_2. Challenger Charlie randomly selects keywords k_i ($i = \{1, 2\}$). If i = 1, challenger Charlie executes the Index-generation algorithm and Trapdoor algorithm to regenerate a new index, which is returned to adversary Eve. If i = 2, challenger Charlie returns the searchable ciphertext to adversary Eve.

(4) **Queries 2.** Adversary Eve repeats the Query 1 operations multiple times, sets the hash function H_1-query h times, and queries the Trapdoor algorithms for non-target sets.

(5) **Guess.** Based on the query results, adversary Eve gives a guess $i' = \{1, 2\}$ about i. If $i' = i$ return 1. Otherwise, return 0. If adversary Eve can give a correct conjecture i with an undeniable probability $\varepsilon > 0$, challenger Charlie can solve the R-LWE difficulty problem with probability ε/h. As the h times of query increases, challenger Charlie needs to solve the R-LWE difficulty problem less frequently. Therefore, given the difficulty of the R-LWE problem, the proposed ABE scheme satisfies the keyword index security.

4.3. ABE Ciphertext Security

Theorem 4. *The proposed ABE scheme satisfies ciphertext security, which satisfies the indistinguishability under selective chosen plaintext attack (IND-sCPA) in the random oracle model.*

Proof. As described in the previous subsection, suppose that Eve is a polynomial-time malicious adversary who can successfully attack the proposed scheme with non-negligible probability ε, and Charlie is a challenger who wants to solve the SVP problem on the NTRU lattice with the query results from Eve. Adversary Eve and challenger Charlie conduct the following query response game. □

(1) **Setup.** Adversary Eve assigns the challenge attribute set S'. Challenger Charlie sets the algorithm to generate a public key $PK = \{h\}$ and a master key $MK = \{B\}$; that is, Challenger Charlie saves the master key and sends the public key to adversary Eve.

(2) **Queries 1.** Adversary Eve queries about hash query, private key query, and ciphertext query with polynomial time. The specific process is as follows.

***H_1*-query.** Challenger Charlie initializes two empty lists, L_3 and L_4, to save the query results. Eve queries the hash function H_1 of the non-target access control policy Plc_i. Challenger Charlie initializes empty lists and uses them to store the query results of the hash function. If the result is already on list L_3 and list L_4, Charlie returns the result t'_2 and gives it to Eve. Otherwise, Charlie randomly selects parameters r'_2, d'_4, ω'_2, where $r'_2, d'_4 \leftarrow \{-1, 0, 1\}, \omega'_2 \leftarrow \{0, 1\}^N$, and calculates $h' = H_1(\omega'_2, att'_i, r'_2 * t'_2 + d'_4 + \lfloor q/2 \rfloor \omega'_2)$. Charlie sends the $t'_2 = H_1(Plc_i)$ and h' to Eve and respectively stores the new result (t'_2) and $(r'_2, d'_4, \omega'_2, h')$ in list L_1 and list L_2.

Private key query. Eve queries the private key related to access control policies Plc_i for message m_i. Charlie initializes a new list L_3 to store the query results. If the attribute S' satisfies the access control policy Plc_i, Charlie's returns continue. Otherwise, Charlie

executes the Keygen algorithm and uses $(s'_1, s'_2) \leftarrow SamplerPre(B, \delta, (t'_2, 0))$ to obtain s'_2 and return it to Eve.

Ciphertext query. Eve performs a polynomial time inquiry on the Trapdoor algorithm of the attribute set S'. If the attribute S' satisfies the access control policy Plc_i, Charlie's returns continue. Otherwise, according to the encrypt algorithm, Charlie executes $c_i = m_i \oplus H_4(\omega'_2, att'_i, v')$ to obtain ciphertext (u', v', c_i) and return it to Eve.

(3) **Challenge.** Adversary Eve randomly selects messages m_1 and m_2. Charlie randomly selects messages m_{i} ($i = \{1, 2\}$). If i = 1, challenger executes the encrypt algorithm and sends the corresponding ciphertext to adversary. If i = 2, Charlie randomly selects one ciphertext from the ciphertext set to Eve.

(4) **Queries 2.** Eve repeats the Query 1 operations n times and queries the private key query and ciphertext query for non-target attribute sets S'.

(5) **Guess.** Based on these query results, Eve gives a guess $i' = \{1, 2\}$ about i. If $i' = i$, return 1. Otherwise, return 0. If the Adversary can give a correct guess with an undeniable probability $\varepsilon > 0$, the challenger can solve the $\gamma - SVP$ problem with probability ε/n. Therefore, under the difficulty of the $\gamma - SVP$ problem, the proposed ABE scheme satisfies the keyword index security.

4.4. ABE Update Policy Security

Theorem 5. *Under the assumption of the R-LWE decision, the proposed ABE scheme satisfies the updated policy security, which satisfies the IND-sCPA in the random oracle model.*

Proof. As described in the previous subsection, assuming there is an adversary Eve who wins the IND-sCPA game with an undeniable probability $\varepsilon > 0$, the probability of challenger Charlie solving the R-LWE problem is $\varepsilon/2$. Eve and Charlie conduct the following query response game. □

(1) **Setup.** Eve declares a query access structure $W^* = W^+ \cup W^-$ and sends it to Charlie. Charlie interacts under the oracle machine after receiving the access structure W^*, and the oracle machine randomly selects uniform random samples and pseudo-random samples. Thus, Charlie obtains $(A, V_A) \in R_q^{l \times m} R_q^m$.

(2) **Private key query.** Eve queries the private key related to access control policies Plc_i for message m_i. This step is the same as the step shown in the proof of Theorem 4.

(3) **Challenge.** Charlie randomly selects messages $m_0, m_1 \in R_q$ from those submitted by Eve. And according to the Update algorithm, he calculates $c_0 = v + \mu_\beta \lceil q/2 \rceil$, $c_A = V_A$. If $x_i \in W^+$, $c_i = V_i^+$. If $x_i \in W^-$, $c_i = V_i^-$. If $x_i \in x/W^*$, $c_i^+ = V_i^+$ and $c_i^- = V_i^-$. Afterward, Charlie sends $CT^* = (W^*, c_A, \{c_i\}_{x_i \in W^+}, c_1)$ to Eve.

(4) **Guess.** Eve repeats the private key query as many times as before. Then, Eve outputs a guess β' on β. If $\beta = \beta'$, the oracle performs pseudo-random sampling on R-LWE. Otherwise, the oracle implements true random sampling.

The results can be divided into two situations. Assuming that the oracle is an R-LWE sampler, the updatable strategy is an effective challenge to ciphertext, and the distribution of ciphertext is consistent with the distribution of the challenger's and adversary's games. It can be inferred that the advantage of the adversary is in making correct guesses; otherwise, if the oracle samples are truly random instances, it proves that the updatable strategy is uniformly distributed, so the adversary can obtain a correct guess with a probability of 1/2. Therefore, according to Definition 3 and Definition 4 in Section 2, the proposed ABE scheme satisfies the updated policy security, which satisfies the IND-sCPA in the random oracle model.

5. Efficiency

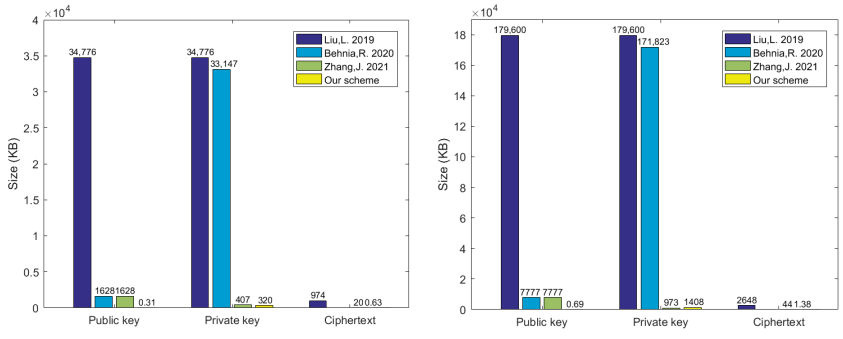
In this section, we analyze the privacy protection scheme for blockchain data based on ABE. The specific analysis mainly includes efficiency analysis and performance analysis. Efficiency analysis mainly focuses on parameter size analysis and calculation time analysis. Efficiency analysis and performance analysis demonstrate the excellent efficiency and performance of this scheme by comparing it with similar schemes.

The parameter sizes of the proposed scheme are compared with similar schemes, mainly including the sizes of public key, private key, index, trapdoor, and ciphertext. The specific comparison results are shown in Table 1.

Table 1. Comparison with other schemes based on lattice.

Scheme	ABE-PEKS [12]	LWE-PEKS [13]	Ref. [10]	Ref. [15]	Our Scheme
Public key	$(m^2 + mn + n)\log q$	$mn\log q$	$(5mn + 2n)\log q$	$mn\log q$	$n\log q$
Private key	$(m^2 + mn + n)\log q$	$m^2\log q$	$2nlm \log q$	$(n + lm)\log q$	$4n^2 \log q$
Index	$3mn \log q$	$(2mn + n)\log q$	—	—	$2n \log q$
Trapdoor	$2mn \log q$	$3mn\log q$	—	—	$2n^2 \log q$
Ciphertext	$(2n^2 + 2lm + 1) \log q$	—	$(3nml + 2)\log q$	$(nl + 1)\log q$	$(2n + 1) \log q$

Furthermore, the performance and efficiency of these schemes are also analyzed and verified by simulation experiments. The secret key size and ciphertext size of our proposed scheme and those in ABE-PEKS (Liu, L. 2019) [12], LWE-PEKS (Behnia, R. 2020) [13], and Zhang, J. 2021 [15] are compared, respectively. According to the actual requirements in these schemes, under reasonable parameters, we set $n = 256$, $q = 210$, and $m = 5211$ ($m \geq 2n\lceil \log q \rceil$) with 80-bit security and n = 512, q = 211, m = 11312 ($m \geq 2n\lceil \log q \rceil$) with 192-bit security. The message length is set to $l = 64$ bit. The public key, private key, and ciphertext sizes of each scheme are compared. The results of the simulation experiments are shown in Figures 1a and 1b, respectively. Here, LWE-PEKS's ciphertext size is not shown in Figure 1 because the encryption phase does not exist. Through the comparison results, under the same conditions, the public key, private key, and ciphertext sizes in our scheme are significantly reduced, which is conducive to saving the computational overhead in blockchain and optimizing the operation efficiency for our scheme.



(**a**) (**b**)

Figure 1. Comparison sizes with other schemes. (**a**) 80-bit security; (**b**) 192-bit security.

In this section, the computational costs of the proposed scheme and similar schemes are analyzed and compared, mainly considering the computational costs of the proposed *Index-generation*, *Trapdoor*, *Test*, *Keygen*, *Encrypt*, and *Decrypt*. As shown in Table 2, T_h, T_{gs}, T_{sp}, T_{mul}, T_m, and T_e are set to represent the average consumption time of the following algorithms: *Hash*, *Gaussain-samplepre*, *Samplepre*, *Modulo-multiplication*, *Matrix operation*, and *Encode operation*, respectively. According to the comparison results in Table 2, it can be concluded that compared with the literature, our proposed scheme has a lower computational time in

terms of index generation, encryption algorithm, decryption algorithm, etc. Therefore, our scheme has lower overall computational time costs and higher efficiency.

Table 2. Time cost comparison with other schemes.

Scheme	ABE-PEKS	LWE-PEKS	Our Scheme
Index-generation	$T_h + 2T_{mul} + T_m$	$T_h + 2T_{mul} + 2T_m$	$2T_h + 2T_{mul}$
Trapdoor	$tT_h + 2T_{mul} + T_m$	T_{gs}	T_{sp}
Test	$kT_{mul} + T_{gs}$	T_{mul}	$T_h + T_{mul}$
Keygen	-	$2T_{gs}$	$T_h + T_{sp}$
Encrypt	-	$2T_{mul} + 2T_m$	$T_h + 2T_{mul}$
Decrypt	-	$2T_{mul} + 2T_m$	$T_h + T_{mul}$

6. Novel Copyright Protection Scheme

6.1. Scheme Design

In this section, a blockchain copyright protection scheme based on the CP-ABE scheme with policy update is designed. As shown in Figure 2, this scheme combines the immutability of blockchain with the flexibility of ABE, aiming to provide a secure and efficient digital copyright protection mechanism. The detailed working process of our new scheme is as follows:

(1) **Initialization.** The blockchain platform serves as the infrastructure for digital copyright protection. There is a secure and reliable key management center in the system, which is responsible for generating and managing the required public key *PK* and master key *MK* for ABE through the *Setup* algorithm and defining the attribute sets and access policies. The attribute set includes various features of digital content, such as author, work name, and release date. The access policy defines which combinations of attributes allow users to access digital content. In addition, through the index sharing algorithm, after the three processes of index generation, trapdoor, and testing mentioned above, a data index structure containing storage addresses and keywords for digital works can be constructed.

(2) **Copyright registration.** Copyright owner registers digital copyright information on the blockchain, including work content, attribute sets, and access policies. By using the *Keygen* algorithm in our ABE scheme, the Key management center generates key *sk* based on registration information and securely distributes them to the corresponding content creators.

(3) **Encryption.** Copyright owner uses attribute-based encryption algorithms to encrypt digital content and embed access policies into ciphertext $c = m \oplus H_4(\omega_2, att_i, v)$. The encrypted digital content is uploaded to the cloud server system, and it is constructed as a transaction record in the blockchain for other users to verify.

(4) **Access control and decryption.** By using the *Index sharing* algorithm, the copyright owner establishes an index trapdoor to generate an index $S_k = \{I_1, I_2, H_1(\omega_1, I_2)\}$ that is recorded in the blockchain. Thus, users can find the desired digital work index through keyword indexing and then request the corresponding digital content from the system. Among them, users need to provide their own attribute set proof. Smart contracts on the blockchain are proven based on access policies and user attribute sets to determine whether the user meets the access conditions. If the user meets the access conditions, the smart contract will obtain the corresponding public key from the blockchain and send it to the user. Users use the secret key to decrypt encrypted digital content and obtain the original content.

(5) **Policy Update.** When the copyright owner needs to update the access policy, they execute the *Update* algorithm based on the new access structure, generate new encrypted files, and store them in the system. Smart contracts on the blockchain will automatically update access policies, ensuring that only users who meet the new policies can access digital works.

(6) **Copyright tracing and protection.** If infringement is discovered, the copyright owner can search for registration information and transaction records of digital copyright on the blockchain to prove their copyright ownership. The consensus mechanism of the blockchain ensures the authenticity and integrity of copyright information and provides strong evidence for rights protection.

(7) **Audit and regulation.** Regulatory authorities can regularly audit digital copyright information on the blockchain to ensure the compliance and effectiveness of copyright protection schemes. For discovered violations, regulatory authorities can take corresponding punitive measures to maintain order in the digital copyright market.

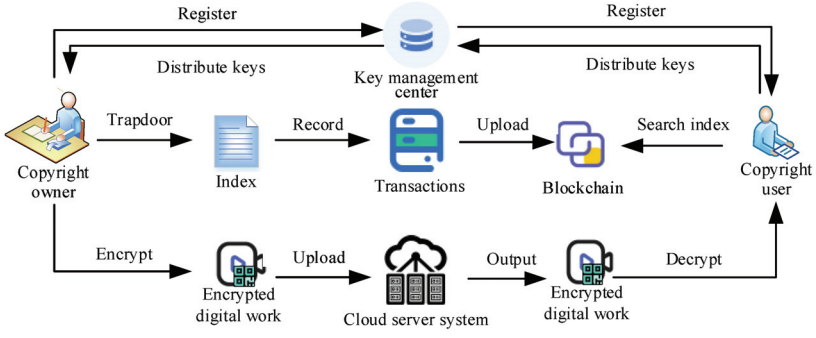


Figure 2. Overview of novel copyright protection scheme.

Through the above seven stages of work, the blockchain digital copyright protection scheme based on attribute-based encryption with updatable policies can achieve secure, efficient, and flexible digital copyright protection. This not only protects the legitimate rights and interests of creators but also promotes the legitimate dissemination and sharing of digital content.

6.2. Advantages of CP-ABE and Blockchain

We design a lightweight blockchain-based copyright protection scheme by recording copyrights on-chain while storing the actual works off-chain. Specifically, the index structure, which contains the storage address and keywords of digital copyright data, is uploaded to the on-chain ledger. Meanwhile, the digital content data, which occupies a substantial storage space, is stored on local servers or cloud servers of various digital media. This approach, combined with CP-ABE and searchable ciphertext, achieves efficient and secure solutions. The specific advantages are as follows:

(1) Unlike traditional copyright protection systems, in our scheme, the on-chain ledger records only lightweight information, such as data storage, verification, operation, and transaction records. Upon generation of the original copyright data, it forms an index structure that is associated with real data, encompassing keywords, storage addresses, and signatures of digital content. This approach enables the proposed scheme to reduce the cost of storage space and enhance the efficiency of blockchain copyright data recording. In addition, the data storage method of blockchain enhances the copyright data's traceability and integrity.

(2) After encrypting data using the proposed CP-ABE scheme with a searchable encryption algorithm, a trapdoor containing copyright data keywords is generated. Subsequently, the index is recorded as a transaction in the blockchain ledger. Users can search for the desired digital works by embedding keywords in the data index structure beforehand. Upon passing the preset trapdoor verification, access to the data is granted, and the actual digital works can be downloaded from the cloud server through the address information in the index. Additionally, the keyword search functionality of this scheme is highly suitable for cross-institutional sharing

of copyright data. On public cloud platforms, this indexing method can establish an effective channel to find the required digital work data through keywords. Furthermore, this approach prevents damage caused by direct human contact with data, thereby effectively protecting the security of digital works.

(3) Due to the large size of digital work files, copyright protection systems have storage requirements for a large amount of audio and video data. However, placing this massive amount of data on the blockchain is impractical. Therefore, utilizing local servers and cloud storage technology, as described in this article, is a more promising and resource-efficient way to address this data storage challenge.

(4) For copyright data, it is first encrypted by CP-ABE, and the encrypted data are stored on the blockchain. Incorporating the attributes of data into the secret key enables users to control their respective copyright data in a more granular manner. Only by verifying the correct attribute access policy can permission to view real data be obtained, thereby ensuring copyright data security.

6.3. Scheme Security Analysis

For the copyright protection scheme's security, firstly, it is necessary to target complex user access and achieve fine-grained management of data access. Secondly, it is extremely important to ensure the copyright data's security, including copyright data encryption security and privacy security. In addition, the data stored within the system cannot be maliciously tampered with, and the copyright protection system resists single points of failure and data silos. Based on the above security requirements, in this subsection, we conduct a detailed security analysis of the proposed copyright protection scheme.

6.3.1. Fine-Grained Access Control

Our CP-ABE scheme focuses on access control for attribute encryption in copyright protection. The CP-ABE scheme allows for the association of access control policies with encrypted ciphertext data and uses user attributes to define access rules. Only users who meet these attribute conditions can decrypt and access copyright data and conduct copyright transactions. Meanwhile, through the previous security analysis of the CP-ABE scheme, it is proven that the CP-ABE scheme satisfies the correctness of encryption. This CP-ABE technology provides fine-grained access control capabilities while protecting data privacy and ensuring data security. Therefore, our scheme satisfies the security requirements for copyright data's fine-grained management in copyright protection systems.

6.3.2. Data Security and Privacy Protection

On the one hand, in this paper, CP-ABE is applied to copyright data so that user who satisfies the attributes can decrypt the data, ensuring the privacy and security of the data. Only when the user's attribute satisfies the access policy can the data be decrypted and obtained. Otherwise, there is no other way to obtain copyright information or complete copyright transactions. The correctness and ciphertext security analysis of the attribute encryption scheme proven in Section 4 proves that the scheme can ensure the privacy and security of copyright data. On the other hand, the proposed CP-ABE scheme is based on lattice. Lattice-based cryptography is generally considered to have the advantage of resisting quantum computing attacks, and it can deal with the threat of quantum computers in the future [20]. Researchers have advocated for the development of novel strategies to include data encryption in the post-quantum era. Subsequently, lattice-based cryptography has garnered significant attention and focus in recent years, with its applications expanding rapidly across multiple domains [21,22]. Therefore, by using the theory of random lattice and the corresponding lattice basis, in this paper, we propose a new lattice-based CP-ABE scheme to improve anti-quantum security for blockchain copyright protection, which can protect data security in the future.

6.3.3. Resist Tampering Attack

If an adversary intends to tamper with existing copyright data and transaction information on the blockchain through tampering attacks, the blockchain can effectively resist tampering attacks through timestamp proof, distributed ledger, digital signature, SHA-256 algorithm, consensus mechanism, etc. Only if the consensus is successful can this transaction be verified and submitted to the blockchain distributed ledger. Therefore, by introducing blockchain technology into this copyright protection scheme, it can effectively prevent individual malicious nodes from tampering with transaction data.

6.3.4. Resist Single Point of Failure and Data Island

By harnessing the power of blockchain technology for decentralized distributed storage and robust authentication of copyright data, this innovative copyright protection scheme addresses the critical issues of single-point failure and data isolation that plague traditional, centralized copyright protection systems, thereby ensuring a more secure and reliable approach. This scheme utilizes the distributed ledger feature of blockchain technology to back up copyright data in a distributed manner on multiple nodes. In this way, even if some nodes fail or are attacked, other nodes can still work normally, ensuring the safe operation of the system and copyright data's integrity. At the same time, the advantages of secure sharing of blockchain data can improve data island problems and achieve interconnectivity of copyright data. Therefore, by using blockchain for decentralized distributed storage and secure authentication of copyright data, this new copyright protection scheme can effectively solve the single point of failure and data island problems faced by traditional centralized copyright protection systems. In addition, the transparency and traceability of blockchain enhance the transparency of copyright usage and the verifiability of historical change records, providing strong evidence support for copyright protection.

7. Conclusions

The application of blockchain in fields such as the Internet of Things and digital copyright has become a trend. This study focuses on the urgent challenges of data privacy security and fine-grained access control that need to be addressed. We have studied and designed an innovative blockchain copyright protection scheme based on CP-ABE, which effectively addresses the data security and privacy challenges in traditional copyright data management, particularly in defense against quantum computing attacks. By utilizing the security and lattice cryptography of blockchain technology, this scheme not only ensures the authenticity and integrity of copyright data but also achieves secure access and fine-grained control of copyright data. In addition, the use of lattice-based cryptography can significantly enhance the scheme's ability to resist quantum attacks. After security analysis and experimental verification, this scheme demonstrates resistance to multiple attacks under the random oracle model while having lower computational and storage costs, providing an efficient and secure new method for copyright protection.

Author Contributions: Conceptualization, J.J.; methodology, J.J. and Y.G. (Yulong Gao); formal analysis, J.J. and Y.G. (Yulong Gao); writing—original draft preparation, J.J., Y.G. (Yulong Gao) and Y.G. (Yufei Gong); writing—review and editing, Y.G. (Yulong Gao) and Z.J.; J.J. and Y.G. (Yulong Gao) contributed equally to this work and should be considered joint first authors. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported in part by the National Key R&D Program of China (2021ZD0111404), the Fundamental Research Funds for the Central Universities (CUC24QT04, CUC230D016), the National Natural Science Foundation of China (62172005), the National Key Research and Development Program of China (2022YFC3302103-01), and the Strategic Research Program of Science and Technology Commission of the Ministry of Education of China (JYB2022-01).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors extend their appreciation to the State Key Laboratory of Media Convergence and Communication in the Communication University of China for funding this research work through Fundamental Research Funds for the Central Universities.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Aberna, P.; Agilandeeswari, L. Digital image and video watermarking: Methodologies, attacks, applications, and future directions. *Multimed. Tools Appl.* **2024**, *83*, 5531–5591. [CrossRef]
2. Hwang, J.S.; Kim, H.G. Blockchain-based copyright management system capable of registering creative ideas. *J. Internet Comput. Serv.* **2019**, *20*, 57–65.
3. Lu, Z.; Shi, Y.; Tao, R.; Zhang, Z. Blockchain for digital rights management of design works. In Proceedings of the 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 18–20 October 2019.
4. Ma, Z.F.; Ming, J.; Gao, H.M.; Zhen, W. Blockchain for digital rights management. *Future Gener. Comput. Syst.* **2018**, *89*, 746–764. [CrossRef]
5. Zhang, X.; Yin, Y. Research on Digital Copyright Management System Based on Blockchain Technology. In Proceedings of the 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 15–17 March 2019.
6. Feng, Q.; He, D.B.; Zeadally, S.; Khan, M.K.; Kumar, N. A survey on privacy protection in blockchain system. *J. Netw. Comput. Appl.* **2019**, *126*, 45–58. [CrossRef]
7. Fernández-Caramès, T.M.; Fraga-Lamas, P. Towards Post-Quantum Blockchain: A Review on Blockchain Cryptography Resistant to Quantum Computing Attacks. *IEEE Access.* **2020**, *8*, 21091–21116. [CrossRef]
8. Gao, Y.L.; Chen, X.B.; Chen, Y.L.; Sun, X.; Niu, X.X.; Yang, Y.X. A Secure Cryptocurrency Scheme Based on Post-Quantum Blockchain. *IEEE Access.* **2018**, *6*, 27205–27213. [CrossRef]
9. Yu, G.S.; Zha, X.; Wang, X.; Ni, W.; Yu, K.; Yu, P.; Zhang, J.A.; Liu, R.P.; Guo, Y.J. Enabling Attribute Revocation for Fine-Grained Access Control in Blockchain-IoT Systems. *IEEE Trans. Eng. Manag.* **2020**, *67*, 1213–1230. [CrossRef]
10. Wang, S.; Zhang, X.; Zhang, Y. Efficient revocable and grantable attribute-based encryption from lattices with fine-grained access control. *IET Inf. Secur.* **2018**, *12*, 141–149. [CrossRef]
11. Zhang, X.; Xu, C.; Wang, H.; Zhang, Y.; Wang, S. FS-PEKS: Lattice-Based Forward Secure Public-Key Encryption with Keyword Search for Cloud-Assisted Industrial Internet of Things. *IEEE Trans. Dependable Secur. Comput.* **2021**, *18*, 1019–1032. [CrossRef]
12. Liu, L.; Wang, S.; He, B.; Zhang, D. A Keyword-Searchable ABE Scheme from Lattice in Cloud Storage Environment. *IEEE Access* **2019**, *7*, 109038–109053. [CrossRef]
13. Behnia, R.; Ozmen, M.O.; Yavuz, A.A. Lattice-Based Public Key Searchable Encryption from Experimental Perspectives. *IEEE Trans. Dependable Secur. Comput.* **2020**, *17*, 1269–1282. [CrossRef]
14. Zuo, Y.; Kang, Z.; Chen, Z. BCAS: A blockchain-based ciphertext-policy attribute-based encryption scheme for cloud data security sharing. *Int. J. Distrib. Sens. Netw.* **2021**, *17*, 1–16. [CrossRef]
15. Zhang, J.; Xin, Y.; Gao, Y.; Lei, X.; Yang, Y. Secure ABE Scheme for Access Management in Blockchain-Based IoT. *IEEE Access* **2021**, *9*, 54840–54849. [CrossRef]
16. Li, J.; Yao, W.; Han, J.; Zhang, Y.; Shen, J. User Collusion Avoidance CP-ABE with Efficient Attribute Revocation for Cloud Storage. *IEEE Syst. J.* **2017**, *12*, 1767–1777. [CrossRef]
17. Zhang, Z.; Ren, X. Data security sharing method based on CP-ABE and blockchain. *J. Intell. Fuzzy Syst.* **2021**, *40*, 2193–2203. [CrossRef]
18. Liu, S.; Yu, J.; Chen, L.; Chai, B. Blockchain-Assisted Comprehensive Key Management in CP-ABE for Cloud-Stored Data. *IEEE Trans. Netw. Serv. Manag.* **2023**, *20*, 1745–1758. [CrossRef]
19. Lyubashevsky, V.; Peikert, C.; Regev, O. On ideal lattices and learning with errors over rings. In Proceedings of the Advances in Cryptology–EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Riviera, French, 30 May–3 June 2010.
20. Nejatollahi, H.; Dutt, N.; Ray, S.; Regazzoni, F.; Banerjee, I.; Cammarota, R. Post-quantum lattice-based cryptography implementations: A survey. *ACM Comput. Surv.* **2019**, *51*, 1–41. [CrossRef]
21. Asif, R. Post-quantum cryptosystems for Internet-of-Things: A survey on lattice-based algorithms. *IoT* **2021**, *2*, 71–91. [CrossRef]
22. Liu, Z.; Choo, K.; Grossschadl, J. Securing edge devices in the post-quantum internet of things using lattice-based cryptography. *IEEE Commun. Mag.* **2018**, *56*, 158–162. [CrossRef]

sensors

MDPI

Article

Post-Quantum Secure Identity-Based Signature Scheme with Lattice Assumption for Internet of Things Networks

Yang Zhang [1], Yu Tang [2], Chaoyang Li [2,*], Hua Zhang [1] and Haseeb Ahmad [3]

[1] College of Food and Bioengineering, Zhengzhou University of Light Industry, Zhengzhou 450001, China
[2] College of Software Engineering, Zhengzhou University of Light Industry, Zhengzhou 450001, China
[3] Department of Computer Science, National Textile University, Faisalabad 37610, Pakistan
* Correspondence: lichaoyang@zzuli.edu.cn

Abstract: The Internet of Things (IoT) plays an essential role in people's daily lives, such as healthcare, home, traffic, industry, and so on. With the increase in IoT devices, there emerge many security issues of data loss, privacy leakage, and information temper in IoT network applications. Even with the development of quantum computing, most current information systems are weak to quantum attacks with traditional cryptographic algorithms. This paper first establishes a general security model for these IoT network applications, which comprises the blockchain and a post-quantum secure identity-based signature (PQ-IDS) scheme. This model divides these IoT networks into three layers: perceptual, network, and application, which can protect data security and user privacy in the whole data-sharing process. The proposed PQ-IDS scheme is based on lattice cryptography. Bimodal Gaussian distribution and the discrete Gaussian sample algorithm are applied to construct the fundamental difficulty problem of lattice assumption. This assumption can help resist the quantum attack for information exchange among IoT devices. Meanwhile, the signature mechanism with IoT devices' identity can guarantee non-repudiation of information signatures. Then, the security proof shows that the proposed PQ-IDS can obtain the security properties of unforgeability, non-repudiation, and non-transferability. The efficiency comparisons and performance evaluations show that the proposed PQ-IDS has good efficiency and practice in IoT network applications.

Keywords: Internet of Things; identity-based signature; post-quantum; lattice cryptography

Citation: Zhang, Y.; Tang, Y.; Li, C.; Zhang, H.; Ahmad, H. Post-Quantum Secure Identity-Based Signature Scheme with Lattice Assumption for Internet of Things Networks. *Sensors* **2024**, *24*, 4188. https://doi.org/10.3390/s24134188

Academic Editor: Klaus Moessner

Received: 27 May 2024
Revised: 17 June 2024
Accepted: 26 June 2024
Published: 27 June 2024

1. Introduction

The Internet of Things (IoT) plays a very essential role in people's daily production and life [1]. As daily travel, consumption, production, medical treatment, etc., are gradually becoming digital and intelligent, a variety of smart IoT devices surround people's lives. Although these IoT devices provide convenience, they also bring many security threats to people's lives and property safety. IoT network security is gaining much attention with the IoT network development [2].

The security issue is the most essential part of data sharing among different IoT devices through the public network. As shown in Figure 1, data security and user identity privacy should be given more consideration in IoT network applications, such as the Internet of Medical Things (IoMT) [3], the Home Internet of Things (Home IoT) [4], the Internet of Vehicles (IoV) [5], and the Industrial Internet of Things (IIoT) [6]. In IoMT systems, medical data are essential in medical diagnosis, drug development, disease prediction, and medical device creation. The security of medical data and user privacy is the guarantee for data value play. In the Home IoT, smart home devices collect data about the user's daily life, health condition, living habits, and personal privacy. These data are related to the safety of users' lives and property. In IoV systems, smart driving cars, cameras, and traffic lights collect information about roads, traffic, logistics, and more in real time. Infrastructure information, such as geography and transportation, is relevant to

national security. In IIoT systems, people, machines, materials, materials, law, environment, enterprise upstream and downstream, products, users, and other elements are connected to achieve data transmission between elements. It provides identity data acquisition, label management, identity registration, identity analysis, data processing, and identity data modeling functions to achieve the marking, management, and positioning of elements, and the industrial Internet identity resolution security mechanism has become the basis of industrial Internet applications. IoT data are becoming a more important asset for IoT network applications, which plays an essential role in process improvement, industrial upgrading, and social progress. Data security and user privacy security are two main parts of most IoT network applications that should be given more consideration.

Figure 1. Security issues for IoT network.

Blockchain is a distributed data management technology that utilizes the consensus mechanism and cryptographic algorithm to achieve secure data sharing in different and strange modes [7]. The consensus mechanism helps the system achieve distributed consensus with a uniform ledger for the whole network, for example, proof of work (PoW) [8], proof of stake (PoS) [9]. The cryptographic algorithm helps the system achieve, for example, DES [10], ECC [11]. With the deepening of the application of blockchain technology in IoT network applications, there emerge many different kinds of chains with different distributed structures, consensus protocols, and cryptographic algorithms. These chains bring the IoT networks back to a centralized model as the IoT data cannot be freely shared among different IoT devices. So, cross-chain technology has gained more attention in recent years.

Cryptography is a general technology that utilizes difficult mathematical problems to achieve secure data sharing among public Internet work [12]. However, the development of quantum algorithms threatens current classical cryptographic algorithms-based IoT network applications [13]. Lattice cryptography is a promising anti-quantum theory that utilizes lattice assumptions to improve post-quantum security [14]. For example, the Number Theory Research Unit (NTRU) consists of the convolution of polynomials, which is based on the shortest vector problem (SVP) on a lattice theory [15]. It uses truncated polynomial rings to encrypt and decrypt data. GGH is an asymmetric cryptographic algorithm constructed by the nearest vector problem [16]. Learning With Errors (LWE) is an unsolvable problem in machine learning that can be reduced to a SIVP lattice problem [17]. There is a variant of ring-based LWE where polynomials replace the integer vectors in the LWE problem. The identity-based signature mechanism utilizes the identity of IoT devices to establish the traceability mechanism for IoT data sharing.

This paper focuses on the security issues of data loss, privacy leakage, and anti-quantum weakness in most current IoT network applications, and proposes a PQ-IDS to

improve the data-sharing security among different IoT devices. The main contributions of this paper are summarized as follows.

1. A security model is constructed with blockchain and the PQ-IDS algorithm for IoT network applications. This model divides the IoT network into three layers: perceptual, network, and application, which help to establish a distributed and secure data-sharing mechanism for IoT network applications.
2. A PQ-IDS scheme is proposed based on lattice assumption, and it inserts the IoT device's identity into the signature scheme to achieve the traceability of IoT data sharing among the IoT network. This scheme helps to achieve anti-quantum attack security with the lattice cryptography theories.
3. Security analysis and proof about the correctness, unforgeability, non-repudiation, traceability, and post-quantum security are presented. The performance evaluations of key-size data-sharing transactions are presented. These results show the efficiency and practicality of the proposed security model and PQ-IDS scheme.

Next, Section 2 gives some related work, Section 3 gives some preliminaries, Section 4 gives the PQ-IDS scheme, Section 5 gives the security proof of the PQ-IDS scheme, Section 6 gives the efficiency comparison and performance, and Section 7 gives the conclusion.

2. Related Work

2.1. IoT Network Security

In IoT network applications, there exist a lot of attacks, such as deception, information leakage, tampering, denial of service (DoS), and privilege escalation, which threaten the personal and property safety of the system users. For the deception attack, Rehman et al. [18] utilized the model-based method to construct a proactive defense framework, which aggregated the technologies of proactive defense mechanisms and cyber deception to resist the deception attack in IoT networks. For the information leakage, Xu et al. [19] established a data-sharing framework with blockchain technology for IIoT, and inserted federated learning into this scheme to improve the tamper-proof and decentralized capabilities. Singh et al. [20] constructed a privacy-preserving model for IoMT with blockchain and zero-knowledge proof, which strengthened the adaptability in lightweight computer devices. For the tampering, Zhang et al. [21] introduced a malware traffic classification model dependent on neural architecture search, which could improve the searchability of the optimal model in a realistic IoT environment. Li et al. [22] designed a hierarchical and multi-group data-sharing scheme for IIoT, which could improve the data access control ability against the key leakage resilience attack. For the DoS, Malik et al. [23] introduced a detect model based on feature engineering and machine learning, which achieved resisting distributed DoS in standardized IoT. Khanday et al. [24] proposed an intrusion detection platform based on machine learning and deep learning classifiers for IoT networks, which could improve the detection accuracy and efficiency of DDoS attacks. For the privilege escalation, Mehmood et al. [25] constructed an insider threat detection and classification framework based on machine learning, and introduced a systematic approach to resist the privilege escalation in relation to the anomalies and security problems. However, the methods used in these references are all classical cryptographic algorithms, which cannot resist quantum attacks. This paper plans to utilize blockchain technology and a post-quantum signature algorithm to provide anti-quantum security for data-sharing in IoT networks.

2.2. Signatures for IoT Networks

Digital signatures play an essential role in the data-sharing processes in different IoT network applications. In particular, the identity-based signature (IDS) has widespread application due to its identification mechanisms and traceability. Liu et al. [26] designed a distributed multi-signature scheme with discrete logarithms to improve the security and efficiency of IoT identification with centralized signature schemes. Jia et al. [27] designed a certificateless signature protocol with an ECC-based discrete logarithm problem, and

stated that it was more suitable for resource-limited IoT devices. Du et al. [28] figured out the security weakness of the scheme in Ref. [27], and presented a new signature scheme with the ECC cryptosystem, which could resist the forger attack from the super adversaries. Li et al. [29] gave an aggregate signature scheme to strengthen the security of large-scale data transmission in IIoT, which also utilized blockchain technology to achieve secure, distributed, and autonomous data management. Bao et al. [30] presented an identity management scheme to protect identity privacy for IIoT, and took the blockchain technology together to achieve distributed privacy-preserving.

In most current IoT network applications, the IDS schemes constructed by large integer factorization and discrete logarithm problems have good applicability to satisfy the needs of IoT devices with high throughput and low latency. However, these IDS schemes cannot resist the quantum attack [31]. Therefore, post-quantum cryptography has gained more attention in current academia and industry to improve the security properties of anti-quantum attacks. Lattice cryptography is one promising post-quantum algorithm. Srivastava et al. [32] presented an identity-based multi-signature scheme with a multivariate quadratic problem, which could achieve anti-quantum security for real-time information exchange in IoV. Wang et al. [33] introduced a proxy signature scheme with lattice assumption, which utilized the lattice basis delegation and preimage sample technologies to achieve post-quantum secure IoT. Wu et al. [34] designed an ID-based proxy signature based on the NTRU lattice and utilized message recovery technology to guarantee secure message exchange among different IoT devices. Prajapat et al. [35] gave an ID-based aggregate signature based on lattice cryptography, which could improve the messages promptly, vehicle identity confidentiality, and authentication swiftly in the vehicle ad hoc networks (VANETs). Sun et al. [36] utilized the path signature to construct an XAI-enabled network, which could achieve efficient time series classification. The simple comparisons of these former schemes are shown in the following Table 1. Although these lattice-based IDS schemes strengthen the anti-quantum ability of IoT networks, there still exist some security issues of privacy leakage, big key size, and high-performance consumption. This paper plans to design a post-quantum secure IDS for general IoT network applications.

Table 1. Cryptographic algorithm comparisons for IoT network.

Scheme	IoT Network	Main Technology	Limitation	Anti-Quantum
Liu et al. [26]	IoT	ID multi-signature; Threshold Merkle tree; Consortium blockchain	Identity traceability lack	No
Jia et al. [27]	IoT	Certificateless signature; ECC	Identity traceability lack; Centralized management	No
Du et al. [28]	IoT	Certificateless signature; ECC; Security authentication technology	Identity traceability lack; Centralized management	No
Li et al. [29]	IIoT	Aggregate signature; Permissioned blockchain; Smart contract	Identity traceability lack; Data openness; Centralized like management	No
Bao et al. [30]	IIoT	Identity management; Permissioned blockchain; Smart contract	Data openness; Centralized like management	No
Srivastava et al. [32]	IoV	ID multi-signature; Blockchain; Cloud server network	Efficiency to be improved	No
Wang et al. [33]	IoT	Proxy signature; Fixed dimension lattice basis delegation; Preimage sample	Identity traceability lack; Centralized management; Efficiency to be improved	No

Table 1. *Cont.*

Scheme	IoT Network	Main Technology	Limitation	Anti-Quantum
Wu et al. [34]	IoT	ID proxy signature; NTRU lattice; Message recovery; Blockchain	Efficiency to be improved	Yes
Prajapat et al. [35]	VANETs	ID aggregate signature; Vehicle pseudo-identities; Cloud storage	Centralized management; Efficiency to be improved	Yes
Sun et al. [36]	XAI network	Path signature; Time series classification	Identity traceability lack; Centralized management	No
This paper	General IoT	ID signature; LWE lattice; Blockchain; Gaussian distribution	Efficiency to be improved	Yes

3. Preliminaries

3.1. Lattice Theory

Some definitions of the lattice theories are presented, which are in relation to the construction of the DVS scheme. First and foremost, some notations of parameters used in this paper are explained in Table 2.

Table 2. System parameter notations.

Notation	Meaning
n	Security parameter
q	A prime number
m	A positive integer with $m \geq 2n\lceil log\ q \rceil$
L	Threshold parameter
σ	The standard deviation
pp	Public parameter
msk	Muster secret key
ID_i	User i's identity
D_σ^m	The bimodal Gaussian distribution
pk	User's public key
sk	User's private key
H_1, H_2	The cryptographic Hash function
μ	The message

Definition 1. ***Lattice*** **[14]***: Given a set of linearly independent vectors $v_1, \cdots, v_n \in \mathbb{R}^m$, the lattice Λ_L is defined as Equation (1) as follows:*

$$\Lambda_L = \{a_1 v_1 + a_2 v_2 + \cdots + a_n v_n : a_1, a_2, \cdots, a_n \in \mathbb{Z}\} \tag{1}$$

The matrices $A = (a_1, \cdots, a_m) \subset \mathbb{R}^{n \times m}$ establish a basis for the lattice Λ with dimension n and rank m. In general, n and m satisfy $m = O(n \log q)$.

Definition 2. ***q-ary Lattice*** **[14]** *"q-ary" lattice is defined with a prime number q, matrix $\mathcal{A} \in \mathbb{Z}_q^{n \times m}$, which is shown in Equation (2) as follows:*

$$\begin{aligned} \mathcal{L}^{\perp}(A) &= \{x \in \mathbb{Z}^m | Ax = 0\ mod q\} \\ \mathcal{L}_y^{\perp}(A) &= \{x \in \mathbb{Z}^m | A^T \mathbf{y} = x\ mod\ q\ for\ y \in \mathbb{Z}^n\} \end{aligned} \tag{2}$$

Definition 3. ***Gaussian distribution*** **[37]***: With standard deviation $\sigma \in \mathbb{R}$ and center $c \in \mathbb{R}$ evaluated at $x \in \mathbb{R}$, the Gaussian distribution is defined by $\rho_{c,\sigma}(x) = exp(\frac{-(x-c)^2}{2\sigma^2})$, and more generally by $\rho_{c,\sigma}(x) = exp(\frac{-||x-c||^2}{2\sigma^2})$ for $x, c \in \mathbb{R}^n$. Here, $\rho_\sigma(x)$ represents the Gaussian distribution when the center $c = 0$. $D_\sigma(x) = \rho_\sigma(x)/\rho_\sigma(\mathbb{Z})$ represents the discrete Gaussian distribution over $\mathbb{Z}$ with center $c = 0$. $D_\sigma(x) = \rho_\sigma(x)/\rho_\sigma(\mathbb{Z}^m)$ represents the more general situation over $\mathbb{Z}^m$ with center $c = 0$.*

Definition 4. $\mathfrak{R} - SIS^{\kappa}_{q,n,m,\beta}$ ***problem*** **[37]***: Given ring $\Re$, a distribution κ over $\Re_q^{n*m}$, $\mathfrak{R} - SIS^{\kappa}_{q,n,m,\beta}$ is a problem to find non-zero $v \in \Re_q^m$ to achieve Equation (3), as follows:*

$$Av = 0 \tag{3}$$

where $A \in \Re_q^{n*m}$, and $||v||_2 \leq \beta$.

3.2. Model Definitions

(1) Scheme model

For an ID-based signature, there are four probabilistic polynomial time algorithms: Setup, KeyExt, Sign, and Verify.

- **Setup(1^n):** Given a security parameter n and parameters q and m, KGC derives some public parameters pp and the master key msk.
- **KeyExt(pp, msk):** Given the public parameters pp and the master key msk, KGC generates public/private key pairs (pk, sk) for the system user with identity ID_i.
- **Sign(pp, sk, μ):** The signer generates a signature e in relation to a with message μ with his own private key sk.
- **Verify(pp, pk, μ, e):** The verifier verifies the e's legality with pp and pk, and output $Accept/Reject$.

(2) Security model

To prove the security of this ID-based signature, it generally establishes a query–response game in a random oracle model. By making reasonable assumptions about the adversary A in the game, it can derive a contradiction that does not correspond to reality by utilizing the proof by contradiction. Then, it will derive that the ID-based signature is secure. Next are the detailed descriptions of this security model.

- *Initialize:* C executes the **Setup** algorithm to derive the system pp.
- *Query*: A queries the Hash, secret key, and signature algorithms with enough times in polynomial time, respectively.
  - *Hash query*: A queries all hash algorithms about the user ID_i or message μ_j (not the target user ID_i^* and message μ_j^*).
  - *Secret key query*: A queries the private key about the user ID_i (not the target user ID_i^*).
  - *Signature query*: A A queries the signature about the user ID_i or message μ_j (not the target user ID_i^* and message μ_j^*).
- *Forge*: A generates a signature (e, μ) about the target user ID_i^* and message μ_j^*, and sends it to C.
- *Challenge*: C generates another signature (e', μ') about the target user ID_i^* and message μ_j^* based on the hypothesis. Then, C attempts to derive a solution for the $\mathbb{Z} - SIS^{\kappa}_{q,n,m,\beta}$ instance by utilizing these two signatures (e, μ) and (e', μ').
- *Analyze*: The lattice assumption SIS cannot be solved with the current highest computing power. This implies that the hypothesis of the former adversary A is invalid.

4. PQ-IDS for IoT Network Applications

This section first introduces a security model for IoT network applications and then designs a post-quantum identity-based signature (PQ-IDS) scheme with the lattice assumption.

4.1. Security Model for IoT Network Application

As the former described in Section 1, the security issue is an essential part of information communication and data exchange processes in every IoT application, such as IoMT, Home IoT, IoV, and IIoT. This section divides the IoT network into three layers: perceptual, network, and application (as shown in Figure 2), and gives detailed descriptions of these three layers, security demands, and solutions.

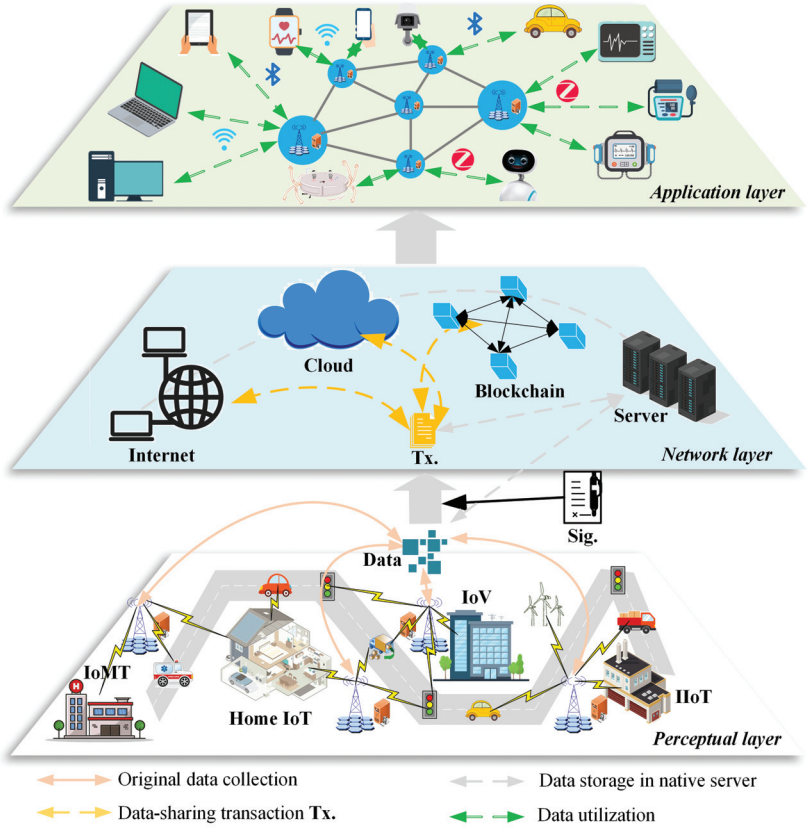


Figure 2. IoT network security model.

- *Perceptual layer*: This layer is a data collection process. From different smart IoT devices, many types of data are selected. These data mainly appear in text, figure, and video forms. As in the IoMT environment, the medical data generally contain health data collected from wearable smart devices, diagnosis data created by patients and doctors, testing data generated by medical devices, and other related data about health insurance, banking, etc. In the Home IoT environment, the data are collected from smart home devices, such as door locks, cameras, refrigerators, washing machines, sweepers, smart speakers, air purifiers, etc. As in the IoV environment, the data are collected from smart vehicles, road detection, cameras, traffic lights, etc. In the IIoT environment, the data are collected from logistics trucks, wind-generating sets, and other related industrial devices. These collected data are signed with the signatures of

IoT devices or device owners. The identities inserted in these signatures guarantee the data verifiability and traceability by utilizing the PQ-IDS scheme.

- *Network layer*: This layer is a data transmission process. The selected data are transmitted through the public network, such as the Internet, cloud, or blockchain. Under the help of communication protocol, these data are transmitted into the transaction Tx, and they also contain the signatures of IoT devices or device owners. Through those public networks, these transactions go to the node that needs them. Meanwhile, it needs a server with enough storage space for data storage and management. The Internet and cloud networks are generally public, which cannot provide a strong security guarantee for cross-domain data sharing. Blockchain technology changes the traditional centralized form to a distributed form, which improves data-sharing transparency and cross-institution data-sharing ability. As in blockchain-based IoT networks, the system transaction verification also can be executed by the PQ-IDS to achieve network-wide consistency. Some security issues of data tampering, malicious theft, and other network attacks generally occur in this layer.
- *Application layer*: This is a data-utilizing process. A number of IoT devices are included in the IoT network, which continuously performs data exchange and processing. The collection, storage, calculation, and analysis of massive data support different IoT application purposes for people's daily lives. For example, IoV data can solve some complex problems in traffic. On the one hand, the data can help understand the traffic flow situation, and improve the driving route for the vehicle owner; On the other hand, the data can also provide decision making for the traffic department in the aspects of road planning, real-time scheduling of signal lights, and vehicle diversion. More importantly, the PQ-IDS scheme plays an essential role in guaranteeing data credibility. This is because the operations throughout the life cycle of the data are recorded, and the identity information of all operators can be traced through signature verification.

In order to trace the data source and find the IoT fault point, it is imperative to label the identity of the operator in the processes of data generation, transmission, processing, and use. Identity-based signatures can guarantee security, integrity, and non-repudiation, which can help to achieve IoT data secure transmission and utilization. Meanwhile, current IoT network-based information systems are equated with traditional cryptographic algorithms that cannot provide security in the quantum computer age. This paper designs a post-quantum identity-based signature (PQ-IDS) scheme with lattice assumption, which can cover the security demands for data-sharing among IoT devices in different layers of the IoT network.

4.2. Details of PQ-IDS

To improve the privacy security of data-sharing transactions in BIoMT, an identity-based signature scheme with lattice assumption has been given. Next, this scheme mainly contains four algorithms.

Setup (1^n): Given the security parameter n, a prime number q with $q = q(n) \geq 3$, a positive integer m with $m \geq 5nlogq$, $L = O(\sqrt{nlogq})$, and $\sigma = L \cdot \omega(\sqrt{logn})$, the KGC first derives the system parameters and master private key. Details of this algorithm are presented in Algorithm 1.

Algorithm 1 Setup

Input: Security parameter κ, a prime number q, a positive integer m
Output: Public parameter pp and master key msk

1: KGC derives an approximate random distribution matrix $mpk = A \in Z_q^{n \times m}$ as the master public key, and a basis $T \in Z_q^{m \times m}$ from $\Lambda^{\perp}(A)$, which satisfy $||\tilde{T}|| \leq L$;
2: Selects two hash functions $H_1 : \{0,1\}^* \to Z_q^n, H_2 : \{0,1\}^* \to Z_q$;
3: Serves $msk = T$;
4: Outputs $pp = \{A, H_1, H_2\}$, and keeps $msk = T$ secretly.

KeyExt (ID_i, msk, pp): Given the system user's identity ID_i, KGC generates the key pairs (pk, sk) in relation to ID_i. Details of this algorithm are presented in Algorithm 2.

Algorithm 2 KeyExt

Input: pp, msk, and system user's identity ID_i
Output: Key pairs (pk, sk)
1: KGC computes $a_{ID_i} = H_1(ID_i) \in Z_q^n$;
2: Computes $s_{ID_i} \leftarrow Samplepre(A, T, a_{ID_i}, \sigma) \in Z_q^m$, where $\sigma \geq ||\tilde{T}||\omega(\sqrt{logm})$, $a_{ID_i} modq = A \cdot s_{ID_i}$, and $||s_{ID_i}|| \leq \sigma\sqrt{m}$;
3: Outputs the public key $pk = a_{ID_i}$ and secret key $sk = s_{ID_i}$ for system user with ID_i.

Sign (pp, pk, sk, μ): Given the message μ, the signer utilizes his secret key sk to sign on the data-sharing transaction. Details of this algorithm are presented in Algorithm 3.

Algorithm 3 Sign

Input: Message μ, system public key A, signer's private key s_{ID_i}
Output: Signature e
1: The user ID_i randomly selects $x \in D_\sigma^m$;
2: Computes $c = H_2(Ax, \mu)$;
3: Computes $e = s_{ID_i} c + x$;
4: Output the signature $< e, c >$ with probability $min(\frac{D_\sigma^m(e)}{M_2 D_{s_{\mu,\sigma}}^m(z)}, 1)$; otherwise, restart.

Verify$(pp, \mu, a_{ID_i}, < e, c >)$: The verifier utilizes the system parameter pp and $pk = a_{ID_i}$ to verify the legality of $< e, c >$ in relation to message μ. Details of this algorithm are presented in Algorithm 4.

Algorithm 4 Verify

Input: μ, a_{ID_i}, and $< e, c >$
Output: Reject or accept
1: **if** $||e|| > L$ **then**
2: Reject it
3: **end if**
4: **if** $||e||_\infty > q/4$ **then**
5: Reject it
6: **end if**
7: If $c = H_2(Ae - a_{ID_i} c \ mod \ q, \mu)$, accept; otherwise, reject.

This PQ-IDS scheme establishes data verification, identity authentication, and data-traceable mechanisms for IoT data sharing. Meanwhile, the lattice assumption guarantees the anti-quantum attack property for IoT network applications.

4.3. Example Application

This subsection presents an example application analysis of the former proposed security model and PQ-IDS scheme into the blockchain-based IoMT system. This signature scheme acts as two roles in the blockchain transactions in the IoMT system: the transaction signature and system transaction verification. Under the proposed security model, the data-sharing transaction through the blockchain-based IoMT system can be divided into the following six steps, also shown in the following Figure 3.

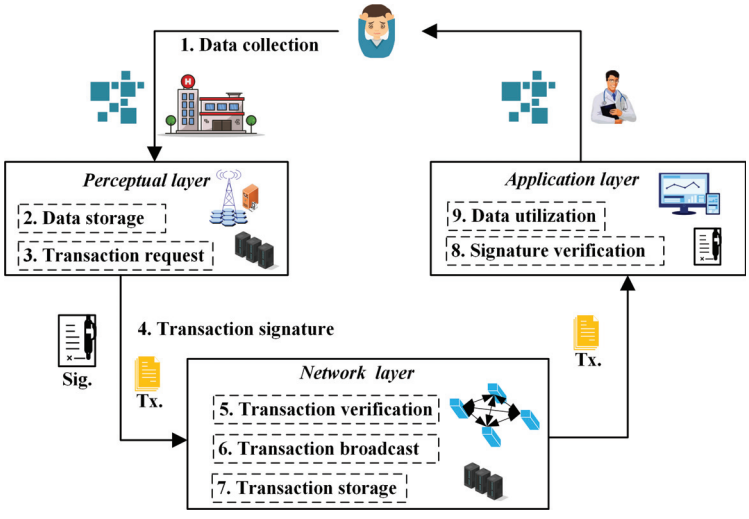


Figure 3. Data-sharing transaction through the blockchain-based IoMT.

- *Data collection*: Through smart medical devices, the daily health data and hospital test data are collected and uploaded to the IoMT system. To establish a traceability mechanism, the medical data are labeled with the signatures of related IoMT devices or operators.
- *Data storage*: In the blockchain-based IoMT system, the real data are stored in the native server. The storage address, operation records, and other related lightweight information are uploaded into the blockchain ledger.
- *Transaction request*: When one patient wants to authorize the doctor to view their health data, they execute a transaction request. This situation is a process of seeing a doctor or scientific research as one user attempts to share data with the other system user. This transaction only sends authorization for these data and the target user must pass this authentication before they can access control on the real medical data.
- *Transaction signature*: Based on the former proposed PQ-IDS scheme, this patient signs the transaction with their private key sk. This signature is a proof that other users confirm the data's validity by opening it. Meanwhile, this patient cannot deny this signature when some medical disputes occur.
- *Transaction verification*: This is the process of verifying the consistency of blockchain transactions across the network. The packaged transactions are sent to the verification nodes who check the legality of the transactions and sign them. The verification process is according to the consensus protocol proof of work (PoW) or proof of stake (PoS).
- *Transaction Broadcast*: The verified transactions will be broadcast to all the nodes in the IoMT network. This process mainly synchronizes the ledger across the network, but it also needs to obtain the access control rights of the real data before the target user can view the original data.
- *Transaction storage*: These transaction data belong to lightweight information, which is recorded in the blockchain ledger. This blockchain ledger is public and immutable, guaranteeing the transaction data's security and traceability.
- *Signature verification*: The target user (doctor) first verifies the signature of the transaction originator.
- *Data utilization*: These medical data can be used for disease diagnosis, drug discovery, medical research, and medical device development.

In general, the proposed security model with a PQ-IDS scheme provides a fundamental framework for data-sharing in the blockchain-based IoMT system. It is also suitable for privacy protection in the data-sharing processes in other IoT network applications.

5. Security Analysis

This security analysis of the PQ-IDS scheme is given in this section. The correctness analysis is given first, and then the unforgeability, non-repudiation, and non-transferability are proved in the random oracle model under the principle defined in the former section.

5.1. Correctness

As shown in Algorithm 4, when the signature satisfies $||e|| > L$ or $||e||_\infty > q/4$, the signature $< e, c >$ is valid iff the following equation Equation (4) holds.

$$H_2(Ae - a_{ID_i}c \; modq, \mu) = H_2(Ax, \mu) \tag{4}$$

Detailed processes of the equation $Ae - a_{ID_i}c = Ax \; modq$ are shown in Equation (5) as follows:

$$\begin{aligned} Ae - a_{ID_i}c &= A(s_{ID_i}c + x) - a_{ID_i}c \\ &= As_{ID_i}c + Ax - a_{ID_i}c \\ &= a_{ID_i}c + Ax - a_{ID_i}c \\ &= Ax \; modq \end{aligned} \tag{5}$$

5.2. Unforgeability

Theorem 1. *This PQ-IDS is secure against the forgery attack as this signature cannot be forged under the hardness of SIS problem* $\mathbb{Z} - SIS^{\kappa}_{q,n,m,\beta}$.

Proof. A query–response game is established first in which an adversary A acts as the query member, and a challenger C acts as the response member. Then, these two members execute this game to prove that this PQ-IDS can capture the security property of unforgeability. In this game, A attempts to obtain enough signature information (without the target message μ^* and user ID_i^*) and has the ability to create a valid signature. C responds to A's queries according to the signature scheme construction, and attempts to solve a $\mathbb{Z} - SIS^{\kappa}_{q,n,m,\beta}$ instance with the knowledge of the forged signature in polynomial time. Next are the detailed steps of the query–response game. □

- *Initialize*: C performs the **Setup** algorithm to derive (n, m, q, k, σ).
- *Query*: A queries the Hash, secret key, and signature algorithms with enough times in polynomial time, respectively.
  - H_1 *query*: A queries the hash function H_1 about the user's identity ID_i. Next, C keeps a list L_{H_1} to store these query results. When they obtain the queries from A, C first checks the list L_{H_1} whether the result pair (a_{ID_i}, ID_i) exists or not. If yes, they send a_{ID_i} to A as the response of H_1's query about ID_i. If no, they compute $a_{ID_i} = H_1(ID_i)$, send a_{ID_i} to A, and record (a_{ID_i}, ID_i) in the list L_{H_1}. Here, A can execute this query with enough times q_{H_1} about different identities $\{ID_1, ID_2, \ldots, ID_{q_{H_1}}\}$.
  - H_2 *query*: A queries the hash function H_2 about the message μ_i (is not the target message μ^*). Next, C keeps a list L_{H_2} to store these query results. When they obtain the queries from A, C first checks the list L_{H_2} whether the result pair (c_i, μ_i) exists or not. If yes, they send c_i to A as the response of H_2's query about $I\mu_i$. If no, they randomly choose $x \in D^m_\sigma$, compute $c = H_2(Ax, \mu_i)$, send c_i to A, and record (c_i, μ_i) in the list L_{H_2}. Here, A can execute this query with enough times q_{H_2} about different messages $\{\mu_1, \mu_2, \ldots, \mu_{q_{H_2}}\}$.
  - *Private key query*: A queries the private key of the user ID_i (is not the target user ID_i^*). Next, C keeps a list L_P to store these query results. When they obtain the queries from A, C first checks the list L_P whether the result pair (s_{ID_i}, ID_i) exists or not. If yes, they send s_{ID_i} to A as the response of a private key query about

ID_i. If no, they compute $s_{ID_i} \leftarrow Samplepre(A, T, a_{ID_i}, \sigma) \in Z_q^m$, sends s_{ID_i} to A, and records (s_{ID_i}, ID_i) in the list L_P. Here, A can execute this query with enough times q_P about different identities $\{ID_1, ID_2, \ldots, ID_{q_P}\}$.

– *Signature query*: A queries the signature of the user ID_i about the message μ_i (is not the target message μ^* and user ID_i^*). Next, C keeps a list L_S to store these query results. When they obtain the queries from A, C first checks the list L_S whether the result pair (e_i, c_i, ID_i, μ_i) exists or not. If yes, they send s_{ID_i} to A as the response of a private key query about ID_i. If no, they execute the Hash algorithm H_2 to obtain (c_i, μ_i), compute $e_i = s_{ID_i}c + x$, send $< e_i, c_i >$ to A, and record (e_i, c_i, ID_i, μ_i) in the list L_S. Here, A can execute this query with enough times q_S about different identities $\{ID_1, ID_2, \ldots, ID_{q_S}\}$ and different messages $\{\mu_1, \mu_2, \ldots, \mu_{q_S}\}$.

- *Forge*: A utilizes the former received information to generate a valid signature (e_i^*, c_i^*) about the target user ID_i^* and message μ^*. Then, they send this forged signature (e_i^*, c_i^*) to C.
- *Challenge*: C utilizes the forking lemma to generate the other valid signature (e_i^{**}, c_i^*) about the target user ID_i^* and message μ^*. Here, C utilizes the same selected $x \in D_\sigma^m$. Then, it has

$$c_i^* = H_2(Ae_i^* - a_{ID_i^*}c_i^*, \mu^*) = H_2(Ae_i^{**} - a_{ID_i^*}c_i^*, \mu^*) \tag{6}$$

With the same Hash function H_2 and message μ^*, it has

$$Ae_i^* - a_{ID_i^*}c_i^* = Ae_i^{**} - a_{ID_i^*}c_i^* \; modq \tag{7}$$

It also has

$$A(e_i^* - e_i^{**}) = a_{ID_i^*}(c_i^* - c_i^*) = 0 \; modq \tag{8}$$

Because of $||e_i^*||_\infty$, $||e_i^{**}||_\infty \leq \; q/4$, it has $||e_i^* - e_i^{**}||_\infty \leq \; q/2$ with overwhelming probability. Hence, it can derive a solution $v = e_i^* - e_i^{**}$ ($||v||_\infty \leq \beta$) for $\mathbb{Z} - SIS_{q,n,m,\beta}^{\kappa}$ instance as shown in the following Equation (9).

$$A(e_i^* - e_i^{**}) \; modq = 0 \tag{9}$$

- *Analyze*: In the common sense, the lattice assumption is not solved with the current highest computing power. This implies that the hypothesis of the former adversary A is invalid. Meanwhile, from the former query processes, A can create a valid signature (e_i^*, c_i^*) with the probability $\frac{\xi}{q_{H_1}+q_{H_2}+q_P+q_S}$. With the query times q_{H_1}, q_{H_2}, q_P, and q_S increasing, this probability will decrease to 0.

Here, it completes this theme proof, and the proposed PQ-IDS can resist the forgery attack from the malicious adversary.

5.3. Non-Repudiation and Traceability

Digital signature plays a role in information protection in most current IoT network applications, and this PQ-IDS scheme utilizes the identity mechanism to achieve secure data exchange among different IoT devices. The proposed security model and PQ-IDS scheme co-guarantee the non-repudiation and traceability of the data-sharing processes in IoT networks.

- *Non-repudiation 1*: The data-sharing transactions are all signed by related workers from the collection to processing and storage. These signatures are signed with the workers' private keys s_{ID_i}, and they can be verified by anyone with their public keys a_{ID_i}. So, the workers cannot deny when the signatures pass the verification.
- *Non-repudiation 2*: The private key is generated by $s_{ID_i} \leftarrow Samplepre(A, T, a_{ID_i}, \sigma) \in Z_q^m$, and the user's public key is generated by $a_{ID_i} = H_1(ID_i)$ where the ID_i is the personal identity, such as identity number, email address, phone number, etc. The personal information guarantees that the signature cannot be denied.

- *Traceability 1*: This identity signature mechanism guarantees the traceability of IoT data exchange processes as the related workers will be traced by opening the corresponding signature when some security incidents occur.
- *Traceability 2*: The blockchain technology provides an immutable public ledger for the records of IoT data storage and operational processes, which guarantees traceability for data sharing in different IoT networks. Once some disputes occur, it can find all the operation records and related operators by opening the signatures.

5.4. Post-Quantum Security

As in most current IoT network applications, the cryptographic algorithms inserted in different information systems are generally based on the classical mathematical difficult problems of discrete logarithms and large integer decomposition. These algorithms cannot provide security guarantees for IoT data sharing facing quantum attacks. However, the proposed PQ-IDS scheme in this paper can solve this problem, which is constructed with the lattice assumption as it has strong security against quantum attacks. Meanwhile, to guarantee the traceability of the IoT data-sharing process, this PQ-IDS scheme utilizes the identity mechanism to ensure that every data-processing record is signed with the corresponding operator's signature. Based on this PQ-IDS scheme, the IoT network applications will have strong security in the future quantum computer age.

6. Comparison and Performance

IoT network applications need more efficient algorithms with high computing power and low power consumption. The performance index is the ultimate goal of algorithm design. The efficiency comparison and performance evaluation are given in the following two subsections.

6.1. Efficiency Comparison

Key size is the essential item to show the efficiency and practicability of the proposed lattice-based IDS, which also influences the transaction execution in blockchain-based IoT network applications. This section selects five main items of mpk, msk, pk, sk, and sig. for comparison, and Table 3 gives the result. For the mpk of the IoT network, the key sizes with the vector form in Refs. [34,35] are smaller than those with matrix form in Ref. [33] and the proposed IDS. The key sizes of mpk, msk, pk, sk, and sig. in the proposed IDS scheme are equal to or less than those in the other three schemes. These comparison results show that this lattice-based IDS scheme has obvious advantages of efficiency and practicability for data sharing in IoT network applications. Smaller key size leads to more efficient data and transaction verification processes, which is suitable for data sharing in different IoT network applications. Along with the developments of edge computing ability and quantum technology, post-quantum algorithms for IoT network applications will be increasingly in demand.

Table 3. Key size comparison.

Schemes	mpk	msk	pk	sk	sig.
Ref. [33]	$mnlogq$	m^2logq	$mnlogq$	m^2logq	$2mlogq$
Ref. [34]	$nlogq$	$4n^2logq$	$nlogq$	$2mlogq$	$2mlogq$
Ref. [35]	$nlogq$	$4n^2logq$	$nlogq$	$2mlogq$	$mlogq$
This PQ-IDS	$mnlogq$	m^2logq	$nlogq$	$mlogq$	$mlogq$

6.2. Performance Evaluation

The performance evaluations are executed on a Windows 11 desktop with Intel Core i7-9700 CPU 3.2 GHz and 16 GB RAM.

(1) Key size evaluation

Firstly, the system parameters have been unified. The system parameters are set as $n = 256$, $q = 2^{23}$, $m = 3545$, which are equal to the 80-bit security level. The key size comparisons of mpk, msk, pk, and sk with 80-bit security are shown in Figure 4, and the results show that this PQ-IDS scheme is less than or equal to those in the other three schemes. Meanwhile, the signature size comparisons with 80-bit and 192-bit security levels are given as shown in Figure 5. Here, the system parameters are set as $n = 256$, $q = 2^{27}$, $m = 7807$, which are equal to the 192-bit security level. The results show that the signature size of the proposed PQ-IDS scheme is no more than that in the other three schemes, and it also shows that the proposed PQ-IDS scheme has good applicability and efficiency for data sharing in different IoT network applications. These results also prove the former theory analyses in Section 5.1. Although the key size of mpk is not small, it can provide a strong security level for IoT network applications facing quantum attacks, and it also can be pre-generated to alleviate the consumption of key generation.

(2) Transaction performance for IoT data sharing

Under the former stated performance environment, the performance for IoT data-sharing transactions has been given. Here, the throughput and latency are selected for performance, which are the main items for blockchain transactions. Meanwhile, these two items will influence the transaction efficiency. As shown in the following Figures 6 and 7, the performance results of these two items in relation to the "CreateAccount", "Query", and "Transaction" are presented. Note that the performance results are the average level of 10 times cross-chain transaction performance, which can eliminate errors from different simulations of the system. From the results, the user registrations have low latency and high throughput, and the establishment transaction amounts remain stable with the number increasing. However, the query times of cross-chain transaction originations have high latency and low throughput compared with the other two items because the IoT data-sharing transactions need high frequency and large quantity.

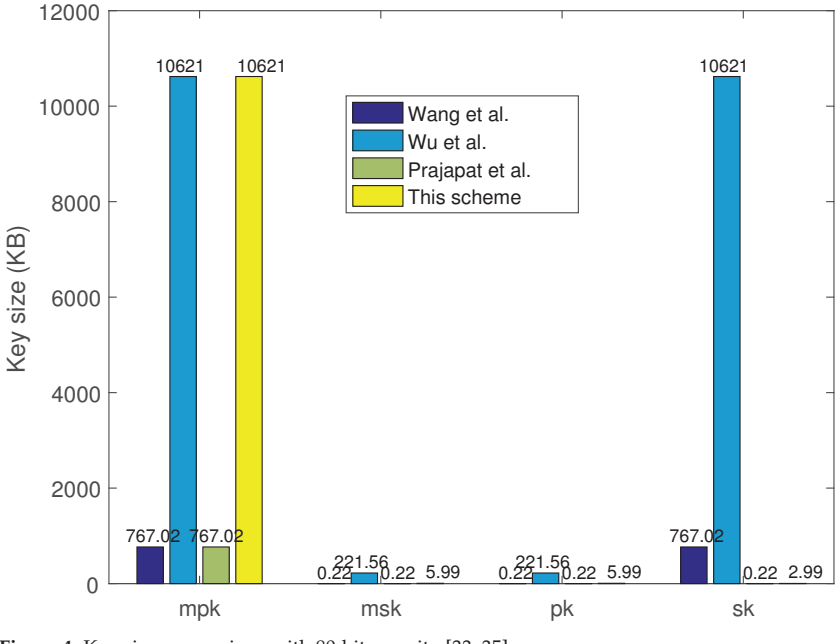


Figure 4. Key size comparison with 80-bit security [33–35].

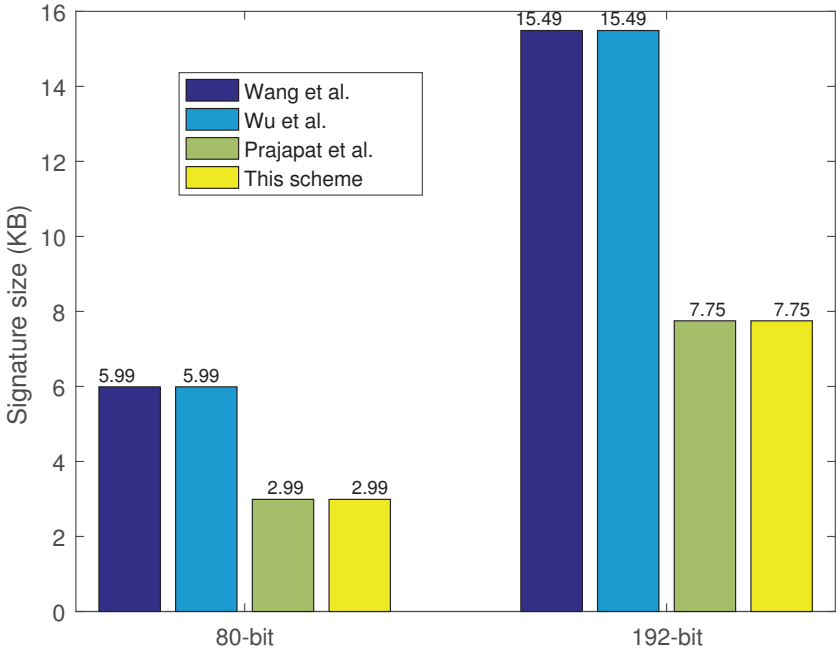


Figure 5. Signature size comparison [33–35].

Above performance evaluations show that the proposed security model is more efficient and practical for data-sharing in IoT network applications, and the proposed PQ-IDS scheme can well support the protection of system data and user privacy.

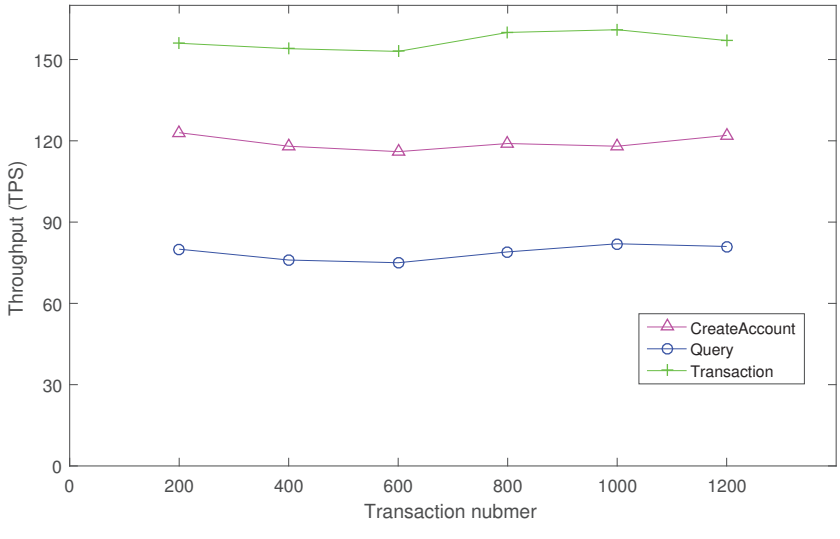


Figure 6. Transaction throughput for IoT data sharing.

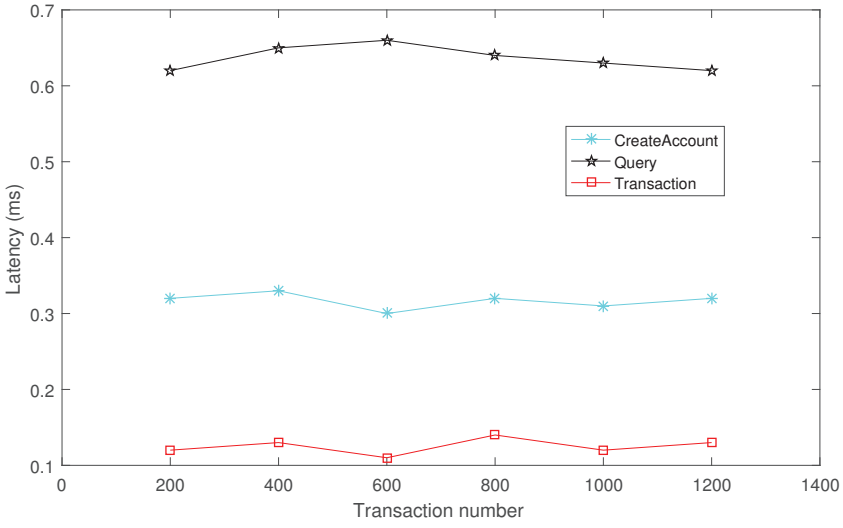


Figure 7. Transaction latency for IoT data sharing.

7. Conclusions

Facing security issues in IoT network applications, this paper introduces a PQ-IDS scheme to guarantee data-sharing security among IoT devices in different layers of the IoT network. An IoT network security model is designed first, which divides the IoT network applications into three layers and states the security issues. Meanwhile, a PQ-IDS scheme with lattice assumption has been proposed, which guarantees IoT data-sharing security in the current or future quantum computer age. The identity-based signature mechanism can also provide a traceability mechanism for IoT node failure. Then, the security analysis shows that this PQ-IDS can obtain the security properties of unforgeability, non-repudiation, traceability, and post-quantum security. Moreover, the efficiency comparisons show that the proposed PQ-IDS scheme has smaller key sizes than other schemes, and the transaction performances show that the proposed IoT network security model has good properties. These evaluation results show that the proposed PQ-IDS is efficient and practical for IoT network applications.

The key size of mpk is bigger than that in NTRU-based lattice cryptography, so it is one research purpose to reduce the key size to satisfy the demands of high throughput and low computing power of IoT devices. Meanwhile, as the number of IoT devices increases and the scale of the network increases, the refined control of the access rights and capabilities of different device subjects is an important guarantee for the privacy protection of IoT data and system users. Therefore, access control in a more fine-grained way will be considered more for secure data sharing among different IoT devices or IoT networks in future work.

Author Contributions: Conceptualization, methodology, and validation, Y.Z. and C.L.; writing—original draft preparation and formal analysis, Y.T. and C.L.; writing—review and editing, supervision, and funding acquisition, C.L., H.Z. and H.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under Grant Numbers 62272090, 72293583, 72293580, the Foundation and Cutting-Edge Technologies Research Program of Henan Province (CN) under Grant Numbers 242102211073, 242102110363, the Key Research and Development Project of Henan Province under Grant 31111113200, the Foundation of State Key Laboratory of Public Big Data under Grant PBD2023-25, and the Doctor Scientific Research Fund of Zhengzhou University of Light Industry under Grant 2021BSJJ033.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Laghari, A.A.; Wu, K.; Laghari, R.A.; Ali, M.; Khan, A.A. A review and state of art of Internet of Things (IoT). *Arch. Comput. Methods Eng.* **2021**, *29*, 1395–1413. [CrossRef]
2. Omolara, A.E.; Alabdulatif, A.; Abiodun, O.I.; Alawida, M.; Alabdulatif, A.; Arshad, H. The Internet of things security: A survey encompassing unexplored areas and new insights. *Comput. Secur.* **2022**, *112*, 102494. [CrossRef]
3. Ghubaish, A.; Salman, T.; Zolanvari, M.; Unal, D.; Al-Ali, A.; Jain, R. Recent advances in the internet-of-medical-things (IoMT) systems security. *IEEE Internet Things J.* **2020**, *8*, 8707–8718. [CrossRef]
4. Philip, N.Y.; Rodrigues, J.J.; Wang, H.; Fong, S.J.; Chen, J. Internet of Things for in-home health monitoring systems: Current advances, challenges and future directions. *IEEE J. Sel. Areas Commun.* **2022**, *39*, 300–310. [CrossRef]
5. Qureshi, K.N.; Din, S.; Jeon, G.; Piccialli, F. Internet of vehicles: Key technologies, network model, solutions and challenges with future aspects. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 1777–1786. [CrossRef]
6. Serror, M.; Hack, S.; Henze, M.; Schuba, M.; Wehrle, K. Challenges and opportunities in securing the industrial Internet of things. *IEEE Trans. Ind. Inform.* **2020**, *17*, 2985–2996. [CrossRef]
7. Dai, H.N.; Zheng, Z.; Zhang, Y. Blockchain for Internet of Things: A survey. *IEEE Internet Things J.* **2019**, *6*, 8076–8094. [CrossRef]
8. Gervais, A.; Karame, G.O.; Wüst, K.; Glykantzis, V.; Ritzdorf, H.; Capkun, S. On the security and performance of proof of work blockchains. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24 October 2016; pp. 3–16.
9. Saleh, F. Blockchain without waste: Proof-of-stake. *Rev. Financ. Stud.* **2021**, *34*, 1156–1190. [CrossRef]
10. Nechvatal, J.; Barker, E.; Bassham, L.; Burr, W.; Dworkin, M.; Foti, J.; Roback, E. Report on the development of the Advanced Encryption Standard (AES). *J. Res. Natl. Inst. Stand. Technol.* **2001**, *106*, 511. [CrossRef]
11. Hankerson, D.; Menezes, A. Elliptic curve cryptography. In *Encyclopedia of Cryptography, Security and Privacy*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 1–2.
12. Hellman, M.E. An overview of public key cryptography. *IEEE Commun. Mag.* **2002**, *40*, 42–49. [CrossRef]
13. Bernstein, D.J.; Lange, T. Post-quantum cryptography. *Nature* **2017**, *549*, 188–194. [CrossRef]
14. Micciancio, D.; Regev, O. Lattice-based cryptography. In *Post-Quantum Cryptography*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 147–191.
15. Peikert, C. Public-key cryptosystems from the worst-case shortest vector problem. In Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing, Washington, DC, USA, 31 May–2 June 2009; pp. 333–342.
16. Nguyen, P.Q.; Regev, O. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. *J. Cryptol.* **2009**, *22*, 139–160. [CrossRef]
17. Brakerski, Z.; Langlois, A.; Peikert, C.; Regev, O.; Stehlé, D. Classical hardness of learning with errors. In Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing, Palo Alto, CA, USA, 1–4 June 2013; pp. 575–584.
18. Rehman, Z.; Gondal, I.; Ge, M.; Dong, H.; Gregory, M.; Tari, Z. Proactive defense mechanism: Enhancing IoT security through diversity-based moving target defense and cyber deception. *Comput. Secur.* **2024**, *139*, 103685. [CrossRef]
19. Xu, G.; Zhou, Z.; Dong, J.; Zhang, L.; Song, X. A blockchain-based federated learning scheme for data sharing in industrial internet of things. *IEEE Internet Things J.* **2023**, *10*, 21467–21478. [CrossRef]
20. Singh, R.; Dwivedi, A.D.; Srivastava, G.; Chatterjee, P.; Lin, J.C.W. A privacy preserving Internet of things smart healthcare financial system. *IEEE Internet Things J.* **2023**, *10*, 18452–18460. [CrossRef]
21. Zhang, X.; Hao, L.; Gui, G.; Wang, Y.; Adebisi, B.; Sari, H. An automatic and efficient malware traffic classification method for secure Internet of Things. *IEEE Internet Things J.* **2023**, *11*, 8448–8458. [CrossRef]
22. Li, T.; Zhang, J.; Shen, Y.; Ma, J. Hierarchical and multi-group data sharing for cloud-assisted industrial internet of things. *IEEE Trans. Serv. Comput.* **2023**, *16*, 3425–3438. [CrossRef]
23. Malik, M.; Dutta, M. Feature engineering and machine learning framework for DDoS attack detection in the standardized Internet of things. *IEEE Internet Things J.* **2023**, *10*, 8658–8669.
24. Khanday, S.A.; Fatima, H.; Rakesh, N. Implementation of intrusion detection model for DDoS attacks in Lightweight IoT Networks. *Expert Syst. Appl.* **2023**, *215*, 119330. [CrossRef]
25. Mehmood, M.; Amin, R.; Muslam, M.M.; Xie, J.; Aldabbas, H. Privilege escalation attack detection and mitigation in cloud using machine learning. *IEEE Access* **2023**, *11*, 46561–46576. [CrossRef]
26. Liu, H.; Han, D.; Cui, M.; Li, K.C.; Souri, A.; Shojafar, M. IdenMultiSig: Identity-based decentralized multi-signature in Internet of things. *IEEE Trans. Comput. Soc. Syst.* **2023**, *10*, 1711–1721. [CrossRef]
27. Jia, X.; He, D.; Liu, Q.; Choo, K.K.R. An efficient provably-secure certificateless signature scheme for Internet-of-Things deployment. *Ad Hoc Networks* **2018**, *71*, 78–87. [CrossRef]
28. Du, H.; Wen, Q.; Zhang, S.; Gao, M. A new provably secure certificateless signature scheme for Internet of Things. *Ad Hoc Networks* **2020**, *100*, 102074. [CrossRef]

29. Li, T.; Wang, H.; He, D.; Yu, J. Permissioned blockchain-based anonymous and traceable aggregate signature scheme for industrial internet of things. *IEEE Internet Things J.* **2020**, *8*, 8387–8398. [CrossRef]
30. Bao, Z.; He, D.; Khan, M.K.; Luo, M.; Xie, Q. PBidm: Privacy-Preserving Blockchain-Based Identity Management System for Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2022**, *19*, 1524–1534. [CrossRef]
31. Cheng, C.; Lu, R.; Petzoldt, A.; Takagi, T. Securing the Internet of things in a quantum world. *IEEE Commun. Mag.* **2017**, *55*, 116–120. [CrossRef]
32. Srivastava, V.; Debnath, S.K.; Bera, B.; Das, A.K.; Park, Y.; Lorenz, P. Blockchain-envisioned provably secure multivariate identity-based multi-signature scheme for Internet of Vehicles environment. *IEEE Trans. Veh. Technol.* **2022**, *71*, 9853–9867. [CrossRef]
33. Wang, L.; Huang, C.; Cheng, H. Novel proxy signature from lattice for the post-quantum Internet of Things. *J. Ambient. Intell. Humaniz. Comput.* **2023**, *14*, 9939–9946. [CrossRef]
34. Wu, F.; Zhou, B.; Zhang, X. Identity-based proxy signature with message recovery over NTRU lattice. *Entropy* **2023**, *25*, 454. [CrossRef]
35. Prajapat, S.; Gautam, D.; Kumar, P.; Jangirala, S.; Das, A.K.; Park, Y.; Lorenz, P. Secure lattice-based aggregate signature scheme for vehicular Ad Hoc networks. *IEEE Trans. Veh. Technol.* **2024**. Available online: https://ieeexplore.ieee.org/abstract/document/10487896 (accessed on 25 June 2024).
36. Sun, L.; Wang, Y.; Ren, Y.; Xia, F. Path signature-based xai-enabled network time series classification. *Sci. China Inf. Sci.* **2024**, 1–15.
37. Ducas, L.; Durmus, A.; Lepoint, T.; Lyubashevsky, V. Lattice signatures and bimodal Gaussians. In *Advances in Cryptology, Proceedings of the CRYPTO 2013: Annual Cryptology Conference, Santa Barbara, CA, USA, 18–22 August 2013*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 1–41, 40–56.

sensors

MDPI

Article

Tides of Blockchain in IoT Cybersecurity

Love Allen Chijioke Ahakonye [1], Cosmas Ifeanyi Nwakanma [1] and Dong-Seong Kim [2,*]

[1] ICT-Convergence Research Center, Kumoh National Institute of Technology, Gumi 39177, Republic of Korea; loveahakonye@kumoh.ac.kr (L.A.C.A.); cosmas.ifeanyi@kumoh.ac.kr (C.I.N.)
[2] IT Convergence Engineering, Kumoh National Institute of Technology, Gumi 39177, Republic of Korea
* Correspondence: dskim@kumoh.ac.kr

Abstract: This paper surveys the implementation of blockchain technology in cybersecurity in Internet of Things (IoT) networks, presenting a comprehensive framework that integrates blockchain technology with intrusion detection systems (IDS) to enhance IDS performance. This paper reviews articles from various domains, including AI, blockchain, IDS, IoT, and Industrial IoT (IIoT), to identify emerging trends and challenges in this field. An analysis of various approaches incorporating AI and blockchain demonstrates the potentiality of integrating AI and blockchain to transform IDS. This paper's structure establishes the foundation for further investigation and provides a blueprint for the development of IDS that is accessible, scalable, transparent, immutable, and decentralized. A demonstration from case studies integrating AI and blockchain shows the viability of combining the duo to enhance performance. Despite the challenges posed by resource constraints and privacy concerns, it is notable that blockchain is the key to securing IoT networks and that continued innovation in this area is necessary. Further research into lightweight cryptography, efficient consensus mechanisms, and privacy-preserving techniques is needed to realize all of the potential of blockchain-powered cybersecurity in IoT.

Keywords: BaaS; blockchain; IoT; intrusion detection; smart contracts; security

Citation: Ahakonye, L.A.C.; Nwakanma, C.I.; Kim, D.-S. Tides of Blockchain in IoT Cybersecurity. *Sensors* **2024**, *24*, 3111. https://doi.org/10.3390/s24103111

Academic Editor: Jian Li

Received: 27 March 2024
Revised: 3 May 2024
Accepted: 13 May 2024
Published: 14 May 2024

1. Introduction

The Internet of Things (IoT) has revolutionized computing and sensing across various domains, offering ubiquitous connectivity and data exchange [1,2]. With industrial networks embracing IoT on a massive scale, the interconnectedness of billions of devices presents both opportunities and challenges [3]. This proliferation amplifies the potential for vulnerabilities and intrusions, significantly expanding the attack surface. Consequently, continuous monitoring becomes imperative to detect and mitigate security threats swiftly. As IoT deployments continue to burgeon, ensuring the resilience of networks against cyber attacks demands vigilant surveillance and proactive measures to safeguard sensitive assets and critical infrastructure [4].

The IoT system's security and privacy could be improved with the help of blockchain technology [2,5], with open, unchangeable, and safe transactions [5], thereby enhancing security and privacy by utilizing blockchain technology [6]. Integrating IoT ecosystems with blockchain technology is a promising strategy to fortify cybersecurity measures in the evolving digital landscape. This introduction elucidates the pivotal role of blockchain in enhancing IoT cybersecurity, emphasizing its transformative capabilities and urgent relevance in contemporary digital environments. As IoT systems proliferate and interconnect, leveraging blockchain presents a compelling solution to address the escalating cybersecurity challenges. By elucidating the symbiotic relationship between IoT and Blockchain, this discourse underscores the imperative for integrating these technologies to safeguard digital assets and mitigate emerging threats effectively.

Artificial intelligence (AI) algorithms are essential for detecting intrusions and attacks in IoT, providing advanced capabilities to secure IoT ecosystems [7]. These algorithms

apply machine learning (ML) techniques to analyze real-time data from IoT devices and identify anomalies and prospective security threats. By constantly observing network activity, sensor data, and device behavior patterns, AI algorithms can identify unusual activities indicative of malicious intent [8,9]. One widely used approach is anomaly detection, where AI models learn normal device behavioral patterns and flag deviations as potential threats [3]. Supervised learning algorithms are frequently employed and trained on labeled datasets to distinguish benign from anomalous conduct. Convolutional and recurrent neural networks (RNNs) are two examples of very effective deep learning techniques. CRNNs are skilled at processing sequential data, which makes them helpful in identifying intricate attack patterns, whereas CNNs examine structured data, such as network traffic [9,10]. In addition, reinforcement learning algorithms can adapt and improve over time by continuously interacting with the IoT environment and learning from feedback to enhance intrusion detection capabilities. AI algorithms offer powerful tools for detecting intrusions and attacks in IoT, enabling proactive security measures to mitigate risks and safeguard IoT networks from cyber threats [11].

Blockchain technology has recently been used by researchers in intrusion detection systems (IDSs) for improved monitoring and detection, prevention of malicious activities or attacks, and tamper-proof transactions and storage in IoT devices and networks [2,12]. Blockchain immutability promotes security and efficient data storage for resource-constrained systems. Despite offering decentralization, scalability, transparency, and immutability, this technique has challenges compared to the existing IDS approaches [5,13]. Over the past decade, the focus on IDS for securing IoT/IIoT has intensified, driven by the need for stringent controls, ensuring real-time operation, data integrity, and compatibility with limited telecommunication protocols [1,3,14]. While existing literature offers insights into risks to security and critical surveillance, this paper provides a comprehensive unconventional survey, explicitly exploring the fusion of blockchain and AI for fortified IoT/IIoT IDS approaches.

Blockchain's potential for IoT security lies in its decentralization, immutability, transparency, and smart contracts. Integrating it with ML, encryption, and identity management systems strengthens IoT security frameworks [15,16]. Industry interest and advancements in blockchain align with current research priorities, but gaps exist in understanding its full potential and scalability [15]. Analyzing blockchain's current state and future trends guides informed decisions and advancements, making it a crucial aspect of IoT security evolution [15]. This study explores the intersection of blockchain and IDS for IoT cybersecurity, as blockchain's features position it to enhance IoT security.

This article outlines blockchain research opportunities for IoT network cybersecurity, trends, future direction, and challenges. First, we conduct a background study on IoT and an overview of blockchain technology to provide context and clarity, followed by its unique features and challenges, motivating a general-purpose IDS. We propose an IDS and define its relationship to blockchain technology, forming a comprehensive blockchain-based IDS framework. The presented results demonstrate the role of blockchain in improving intrusion detection performance. Finally, we discuss research opportunities framed by our blockchain-integrated IDS. This article provides a framework within which researchers can better engage and collaborate in studying and exploring next-generation IDS for IoT networks with a vision toward ubiquitous, scalable, transparent, immutable, and decentralized IoT networks. Notably, this review makes the following significant contributions:

1. Utilizing the preferred reporting items for systematic reviews and meta-analyses (PRISMA) article collection approach, this study systematically gathers articles on AI, blockchain, IDS, IoT, and IIoT, shedding light on challenges, trends, and emerging review areas in IoT IDS designs for security.
2. Focusing on articles published between 2019 and 2024, this review captures recent advancements in AI and blockchain-based IDS designs, ensuring relevance and currency of insights.

3. Evaluation of various AI blockchain integration techniques prioritizes factors like fidelity, transparency, immutability, robustness, and compactness, providing a nuanced understanding of their performance.
4. This study underscores blockchain's pivotal role in fortifying IoT/IIoT security measures by showcasing its efficacy in enhancing intrusion detection performance.

2. Background Study

This section provides an essential framework for a clearer understanding of the discussion by elucidating the trends, limitations, and opportunities in IoT. It also provides concepts pertinent to the current investigation of blockchain applications.

2.1. Exploring Evolving Opportunities, Trends, and Striking Demands in the Internet of Things

The IoT represents a significant breakthrough in connectivity by connecting billions of internet-enabled devices, fostering intelligent interactions, and integrating physical infrastructure with digital systems. IoT has expanded to encompass various industries, including smart factories, healthcare, smart cities, and transportation [17]. Recent forecasts indicated that IoT-connected devices exceeded 10 billion in 2021, and are expected to reach 41 billion by 2027, with smart home and factory devices driving the market [18]. Sensors and actuators are vital to the Internet of Things as they gather data and manipulate actual environments. However, IoT faces challenges like security vulnerabilities resulting from resource constraints in sensor nodes and interoperability problems caused by diverse communication protocols. Despite these obstacles, IoT's convergence with data analytics and AI enables real-time decision-making and predictive maintenance, leading to significant process improvements. To ascertain the potentialities of IoT, it is necessary to address privacy, security, data heterogeneity, and interoperability concerns.

The proliferation of IoT devices, the diverse nature of these devices, and the evolution of communication protocols have led to a surge in enabling technologies from an engineering standpoint [19]. AI and ML techniques have further enhanced the potential of IoT by extracting insights from heterogeneous sensor data, thereby reshaping business operations [18,20]. The modular design of IoT systems, which abstracts these systems into separate components, enhances their adaptability and clarifies their architecture [20]. As in Figure 1, the layered structure of IoT comprises the perception, transport, processing, application, and analytics layers. The perception layer encompasses physical devices that sense the environment and communicate data to higher layers, while the transport layer facilitates communication between devices and cloud-based services. The processing layer, typically hosted on edge or cloud platforms, provides storage and computation capabilities, enabling scalability and interoperability. The application layer governs system operations, interacts with users, and manages logical processes. Finally, the analytical layer offers users actionable insights, enhancing decision-making. However, security challenges persist, particularly in resource-constrained IoT devices and cloud-based middleware, highlighting the need for robust security measures [20].

The IoT conceptual framework links billions of devices with internet access, allowing data to interact with each other and their environment. The evolution of IoT has led to ubiquitous data access, enabling real-time connectivity and interaction between physical and digital systems across various domains. From its origins supporting radio frequency identification (RFID) technology, IoT has expanded into diverse healthcare, transportation, and smart factory/city applications [21]. Recent statistics indicate a significant rise in connected IoT devices, with projections reaching 41 billion by 2027, translating to over 152,000 new connections per minute by 2025 [22,23]. This growth reflects a booming market, with the global IoT market reaching USD 157.9 billion in 2021, primarily driven by industrial applications and intelligent devices [22,23].

IoT presents opportunities to enhance productivity through real-time asset monitoring and control. Industries make informed decisions by leveraging data from IoT devices, such as sensors and actuators, improving operational efficiency. Additionally, IoT facilitates

the development of smart applications in various sectors like factories, homes, cities, and agriculture, leading to increased convenience and efficiency in daily operations.

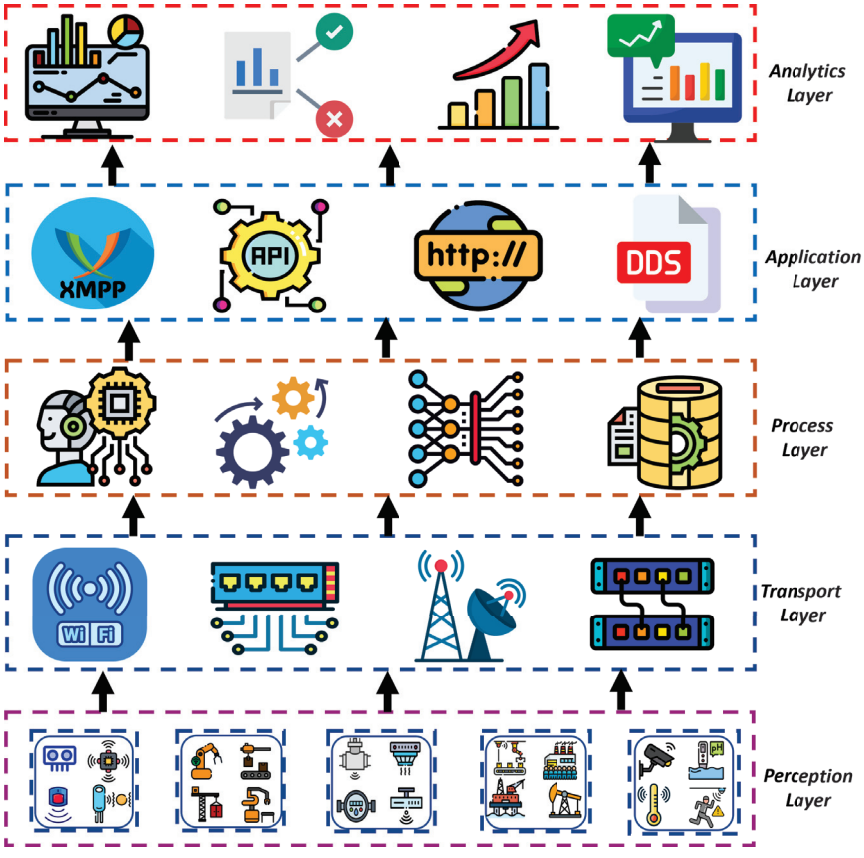


Figure 1. IoT architectural layers.

2.2. IoT/IIoT Vulnerabilities and Attacks

Security remains a paramount concern in IoT and IIoT, given the potential safety and risks associated with compromised systems. Vulnerabilities persist due to protocol limitations, insufficient mitigation strategies, and challenges in real-time monitoring, as highlighted by various studies [24–27]. Attacks target various substrata of the IoT framework, often aligning with the Open Systems Interconnection (OSI) model [25,26].

A comprehensive examination of SCADA attacks reveals their significance within the IIoT [24]. SCADA is a pivotal component of the IIoT infrastructure responsible for monitoring industrial processes. SCADA systems integrate data acquisition, transmission, and human–machine interface (HMI) techniques. HMIs serve as interfaces connecting individuals to devices, facilitating data visualization and real-time monitoring of production processes, as well as machine input and output. SCADA architecture typically comprises the master terminal unit/station (MTU/MSU), which functions as the control center, sub-MSUs/sub-MTUs as secondary control centers, remote terminal units/remote station units (RTUs/RSUs) acting as programmable logic controllers (PLCs) and intelligent end devices (IEDs), used for sensor and actuator data monitoring. The authors looked at common IoT/IIoT attacks, their methodologies, and frequently employed tools in IoT/IIoT [24,28] as follows:

1. Distributed denial-of-service and denial-of-service (DDoS/DoS) attacks deprive authorized users of network resources, primarily targeting availability requirements [4]. In such scenarios, a compromised remote terminal unit (RTU) inundates the master terminal unit (MTU) with arbitrary packets, causing network capacity depletion and hindering the accessibility of resources for legitimate users. The RTU and MTU's communication ability is interfered with, impeding supervision and process tracking. Low Orbit Ion Cannon (LOIC), Slowloris, Raksmart, Hulk, and Tor's Hammer are typically used attack tools [24,29,30].
2. The man-in-the-middle (MiTM) attack intercepts network traffic by infiltrating device communication paths. It is achieved by observing the network, inserting irregularities into transmissions, and relaying them to the intended recipient. Successful execution of this attack hinges on maintaining the session connection while keeping the attacker's presence concealed, using spoofed IPs to evade detection [28,31]. SSLStrip, Evilgrade, and Ettercap are standard tools that enable MiTM attacks [24,32,33].
3. MTUs and sub-MTUs can access the wired or wireless network through passive or active eavesdropping, allowing attackers to introduce spyware and exploit vulnerabilities [28,34].
4. Masquerade attacks involve impersonating legitimate network users through fake identities and IP spoofing, facilitating the theft of sensitive information. Attackers may employ brute force tactics to exploit stolen passwords for unauthorized access [28,35].
5. Viruses, Trojan horses, and worms are deployed by attackers post-MitM or masquerade attacks to infect MTUs. These malicious codes grant unauthorized access to the infected system, allowing attackers to launch further assaults or propagate throughout the network, potentially causing system instability or collapse [28].
6. Fragmentation attacks exploit weaknesses in packet reassembly processes, causing MSU/MTU failure when transmitting oversized data, leading to system collapse [28].
7. Doorknob rattling involves preparatory actions, such as limited system access attempts, to test security measures' readiness and responsiveness before an attack [28].
8. Attacks known as reconnaissance aim to learn more about a network and its hardware characteristics. Guarding sensor readings from the operational procedure is, therefore, essential. Attacks such as response injection introduce deceptive inputs into a control system, prompting control algorithms to make incorrect decisions. In a command injection assault, fictitious control commands infiltrated the control system. Human intervention may cause an improper control action, or bogus commands may be injected and cause RTU and field device register values to be overwritten [24,28,36].

2.3. Overview of Blockchain Technology

A blockchain, originally designed to support cryptocurrencies, has evolved into a disruptive force across various sectors. It serves as a decentralized ledger, ensuring secure and transparent recording of transactions. Its distributed architecture and cryptographic principles guarantee data integrity and resilience against tampering [37]. Security issues with IoT devices, blockchain, and connecting IoT devices to blockchain networks are cybersecurity concerns for blockchain-based IoT systems [2,38].

A blockchain operates as an open digital ledger on a peer-to-peer network, recording timestamped transactions in immutable blocks. Each block is connected to the following and encrypted, maintaining transparency and integrity without centralized control. It encompasses public and private variants, with public blockchains offering universal access and private ones restricting entry to authorized entities [38]. Transactions undergo digital signing, grouping, and storage in a distributed electronic database, ensuring consensus and verification to prevent tampering. This decentralized approach ensures data consistency across all ledger copies. Figure 2 shows the blockchain data transmission approach within the IoT networks.

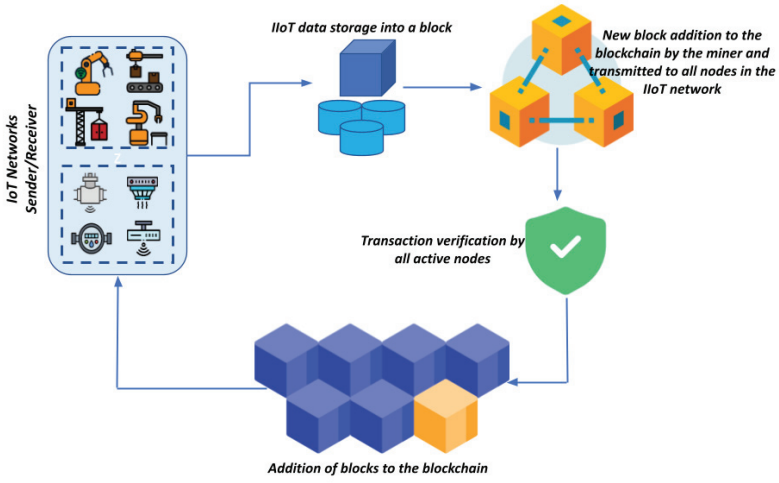


Figure 2. Blockchain transaction process.

2.3.1. Major Blockchain Security Features

1. Data immutability and integrity: Blockchain's immutability guarantees that recorded data remain unchangeable without network consensus, making it ideal for securing critical IoT data like sensor readings, supply chain details, and device logs. This feature is crucial for maintaining data integrity, a top priority in IoT systems requiring accurate and unaltered data throughout storage and transmission [2,39].
2. Decentralization and transparency: Acting as decentralized and distributed ledgers, transactions are recorded across numerous nodes, ensuring no single entity controls the network. The decentralized architecture in IoT devices lessens the dependence on central authorities and promotes transparent and tamper-resistant transactions. It eliminates single points of failure and bolsters system resilience against cyber threats [2,40].
3. Smart contracts: These self-executing agreements coded on the blockchain automatically execute actions based on conditions, reducing reliance on intermediaries in IoT transactions [41,42]. By automating predefined tasks, such as maintenance alerts or data validation, smart contracts improve efficiency and minimize the need for intermediaries and potential vulnerabilities in IoT transactions [40].
4. Consensus mechanisms: Consensus mechanisms are sets of rules and protocols used in blockchain networks to achieve agreement among network participants regarding the validity of transactions and the state of the distributed ledger [43]. This ensures that all nodes in the network reach a consensus or joint decision about the current state of the blockchain. Various consensus mechanisms facilitate agreement and trust in decentralized networks by establishing rules for adding new transactions to the blockchain and resolving conflicts among participants [44]. Some of these mechanisms are as follows:
    a. Proof of Work (PoW): Requires solving complex puzzles for transaction validation and block creation; ideal for highly secure IoT systems like industrial control systems [42–44].
    b. Proof of stake (PoS): Selects validators based on staked coins; offers energy efficiency suitable for resource-constrained IoT devices like smart home systems [42,43].
    c. Delegated proof of stake (DPoS): Uses elected nodes for transaction validation, ensuring high speed and scalability for real-time IoT applications like smart cities.

d. Proof of authority (PoA): Validators verify identity; benefits enterprise IoT deployments like supply chain management and ensures accountability [42,43].
e. Practical Byzantine fault tolerance (PBFT): Focuses on low latency and high throughput, making it suitable for financial IoT systems or autonomous vehicles requiring rapid consensus. These mechanisms collectively ensure data integrity, security, and trust in IoT, tailored to specific IoT application needs and constraints [42–44].

5. Identity management and authentication: Blockchain-based identity solutions enable secure and verifiable identity management in IoT to establish trust among themselves, ensuring that only authorized devices participate in the network [45].
6. Encryption: Transactions and data stored on the blockchain are encrypted using advanced cryptographic algorithms, ensuring that data remain private and secure, and protecting sensitive IoT data against vulnerability [45].
7. Privacy and confidentiality: Private blockchains provide controlled access to data, guaranteeing confidentiality and making them suitable for scenarios where sensitive information needs secure sharing. IoT leverages private blockchains for securely exchanging critical data, such as patient health records or industrial process data [2,45].

Blockchain's features contribute significantly to IoT cybersecurity by providing trust, transparency, and robustness [2,5]. However, scalability remains a hurdle for expansive IoT implementations due to energy-intensive consensus mechanisms. In addition, interoperability is needed across various blockchain and IoT protocols for seamless device integration [5,38]. Researchers continue to explore innovative solutions to address challenges and enhance the synergy between blockchain and IoT. Figure 3 elucidates blockchain integration with AI for intrusion detection.

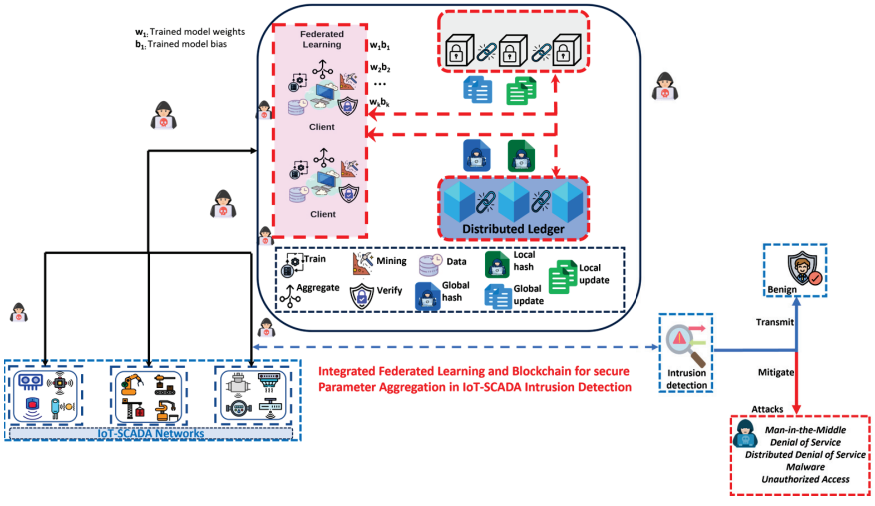


Figure 3. A diagram of an example of the combination of blockchain and IDS, visually demonstrating the benefits of intrusion detection [46].

2.3.2. Opportunities and Challenges in Blockchain–IoT Convergence

The convergence of blockchain and IoT presents a compelling synergy, combining blockchain's decentralized, transparent ledger with the interconnected network of IoT devices [47,48]. Blockchain, known for its secure transaction data storage, has expanded beyond cryptocurrencies like Bitcoin to include immutable data chains [49]. IoT, conversely, encompasses a vast array of interconnected devices exchanging data, facilitating innovative interactions, and bridging physical and digital domains [47]. Integrating IoT with blockchain networks presents opportunities to enhance data integrity, trust, and

decentralization in IoT systems [2,50]. Through blockchain, IoT devices can securely transmit data, ensuring immutable records of transactions [51]. This approach fosters trust between devices, reducing dependence on central authorities and intermediaries, thus enhancing security and resilience [51,52]. The intersection of IoT and blockchain faces challenges in scalability, resource constraints, and privacy [2,21,50,52–57]. As IoT expands, ensuring blockchain scalability while considering the limitations of IoT devices becomes crucial [56]. Optimizing blockchain solutions for IoT devices' computational and storage constraints poses a challenge [58,59]. Balancing transparency and data privacy is also complex, requiring innovative approaches to maintain the benefits of a transparent ledger while safeguarding sensitive information.

2.3.3. Trends and Innovations in Blockchain and IoT Convergence

Several significant trends mark the convergence of blockchain and IoT [48]. Edge computing is rising, boosting real-time processing for IoT devices and reducing latency, while AI integration enables advanced decision-making and predictive analytics [54]. The deployment of high-speed 5G networks further empowers IoT applications [60], complemented by developing communication frameworks like blockchain-enabled architectures to ensure secure data exchange among IoT nodes [21]. However, navigating this landscape requires addressing challenges and seizing opportunities to shape the future of interconnected intelligent systems [21,54].

Blockchain technology has garnered attention for its secure transaction methods and potential to address internet security issues. Recent innovations like federated blockchain offer enhanced scalability and transaction privacy, exemplified by platforms like R3 and Corda [21,61,62]. Blockchain-as-a-service (BaaS) simplifies blockchain app development and maintenance, while Ricardian contracts provide legally valid electronic documents linked to agreements [21,61,63]. Blockchain interoperability facilitates seamless data exchange and cross-chain transactions, while integration into social networking incentivizes content creation through token rewards. Hybrid blockchains offer enhanced security and flexibility by combining public and private blockchain elements [21,61–63].

2.3.4. Potential Use and Applications

The IoT industry is witnessing a surge in interest, focusing on integrating real-time data analytics directly into IoT standards. Previously seen as a passive data monitoring tool, IoT now empowers autonomous applications with real-time decision-making capabilities, becoming a fundamental requirement in deployments [5,64]. For instance, integrating real-time analytics with equipment monitoring systems in manufacturing facilities has transformed predictive maintenance practices [2,21]. By combining sensor data with advanced analytics, manufacturers can predict equipment failures proactively, minimizing downtime and optimizing production efficiency, especially in critical industries like automotive manufacturing or semiconductor fabrication [2,64].

Meanwhile, the modern internet landscape is evolving to prioritize the availability and security of connected resources. With its decentralized ledger system, blockchain technology is reshaping transactions across sectors like finance, healthcare, and supply chain management [2,21,51,65,66]. Blockchain enables smart devices to offer users greater control and insights, while integrated sensors enhance real-time monitoring in supply chains [65]. Smart contracts streamline agreement execution, reducing reliance on intermediaries [2]. Furthermore, blockchain applications extend to healthcare, electronic voting systems, digital identity verification, and property registration, addressing governance and asset management challenges [67]. In addressing global challenges such as the COVID-19 pandemic, blockchain, and IoT integration offer transparent and efficient solutions for collective action [2].

The blockchain's decentralized and impenetrable structure offers several benefits when integrated into IoT systems. Some critical use cases and applications are as follows:

1. Data integrity and immutability: Ensures data integrity from IoT devices, offering an immutable ledger where data records are cryptographically linked, time-stamped, and unalterable. Each transaction is securely stored, guaranteeing the authenticity and reliability of IoT data [2,56,68].
2. Secure device identity and authentication: Verifies IoT device identities and prevents unauthorized access. Blockchain-based digital certificates uniquely identify devices, with smart contracts enforcing access control. Only authorized devices, validated through cryptographic measures, can interact within the network [21,56,66].
3. Decentralized access control: Reduces reliance on central authorities and eliminates single points of failure. Blockchain enables decentralized access control through smart contracts, ensuring distributed permissions management. No single entity controls the entire IoT network, enhancing resilience and security [2,21,56].
4. Supply chain transparency and traceability: Tracks the journey and origin of data and records all transactions transparently and tamper-proof, providing an immutable audit trail. This fosters trust and reduces the risk of anomaly [2,52,56].
5. Secure firmware updates: Ensures secure over-the-air (OTA) updates for IoT devices and verifies the authenticity of firmware updates, allowing devices to validate software integrity before installation. This safeguards against malicious updates and ensures device security [2,21,56].
6. Distributed threat intelligence sharing: Collaborates on threat intelligence across IoT networks and facilitates secure sharing of threat data among devices and organizations. Malware signatures, attack patterns, and other threat intelligence can be exchanged, enhancing collective defense mechanisms [2,21,56].
7. Privacy-preserving data sharing: Enables selective data sharing while protecting privacy, employing privacy-preserving techniques like ZKPs to enable selective data disclosure. Users can share specific data without revealing sensitive information, ensuring privacy while promoting collaboration [2,21,56].
8. Smart contracts for automated security policies: Automates security policies and responses. Smart contracts execute predefined security rules autonomously. For instance, compromised devices can be automatically isolated from the network, preventing further threats and maintaining network integrity [2,51,56].

2.4. Examining Blockchain's Progression in the Quantum Age

Notwithstanding the industrial transformation by blockchain's decentralized and secure characteristics, the rise of quantum computing presents significant challenges to the conventional cryptographic methods underpinning blockchain systems. Quantum computers exploit quantum bits (qubits) to perform computations exponentially faster than classical computers, potentially breaking widely used encryption algorithms like Rivest–Shamir–Adleman (RSA) and elliptic curve cryptography (ECC) [69,70]. Shor's algorithm exemplifies this threat by efficiently factoring large numbers and exposing current encryption schemes [71]. Researchers are actively developing post-quantum cryptographic (PQC) algorithms that resist attacks from classical and quantum computers to counter quantum threats [71–73]. These PQC algorithms ensure long-term security for blockchain systems, including lattice-based cryptography, code-based cryptography, hash-based signatures, and multivariate polynomial cryptography [69,70].

Blockchain platforms are exploring the integration of post-quantum cryptography to address these challenges [70]. This integration entails upgrading cryptographic primitives such as digital signatures, key exchange protocols, and hash functions to PQC standards while maintaining backward compatibility and transaction efficiency [72]. Adopting proactive measures and fostering collaboration among blockchain developers, cryptographers, and quantum computing experts are crucial for securely navigating this transition [70,71,73]. Implementing robust post-quantum cryptographic standards is paramount to upholding the integrity and security of blockchain systems amidst quantum advancements [69,70,72].

2.5. Analysis of Survey on Blockchain for Security Concerns of IoT/IIoT

Ferrag et al. [55] conducted a summary of existing studies on IoT network security with blockchain and investigated blockchain-based security and privacy systems across various types of IoT applications. The work compared consensus algorithms based on nine properties and classified security analysis techniques into four categories. Finally, they highlighted steps for constructing and assessing security systems built on blockchain. Exploring blockchain's potential beyond cryptocurrency, Padma et al. [53] focused on its applications in IoT monitoring and supply chain management, proposing blockchain integration as a viable solution and highlighting research challenges and opportunities in leveraging blockchain for IoT advancement. Abubakar et al. [57] conducted a thorough survey on integrating blockchain with IoT, outlining limitations, benefits, and architectural insights across various IoT domains, addressing challenges, exploring solutions, and proposing future research directions.

Ankit et al. [38] systematically surveyed recent IoT security technologies, emphasizing session keys, blockchain integration, AI-based authentication, and authentication, stressing the need for ongoing improvements and collaboration in security mechanisms to address evolving threats and inspire future research in IoT security. Blockchain's decentralized nature offers promising solutions for enhancing IoT security, as explored by Banda et al. [74], who reviewed security enhancements facilitated by blockchain for IoT, including identity management, authentication, and data privacy, and examined its role in enabling end-to-end food traceability. Alam et al. [21] investigated blockchain integration with IoT architecture, evaluated academic research, discussed issues of interoperability and stability, and explored trends and advantages of combining blockchain with IoT.

Mathew et al. [12] analyzed blockchain's capabilities in IIoT, identifying vulnerabilities and proposing blockchain integration with collaborative IDS to enhance trust in IIoT networks. Shamar et al. [58] concluded that integrating blockchain with IoT presents significant security challenges and emphasized the need for pre-validation of data, security measures in public spaces, and tailored lightweight blockchain-powered solutions for IoT requirements. Alzoubi et al. [56] evaluated the state of blockchain-integrated IoT, examining challenges, proposed solutions, future research directions, and emerging trends, aiding practitioners and researchers in navigating integration complexities. A new paradigm investigated the possibility of handling security and privacy concerns, particularly in IDS for IIoT networks employing a combination of federated learning (FL) and blockchain [59]. Exploring the potential of blockchain in FL for enhancing IDS in monitoring IIoT network traffic offered recommendations for effective implementation. It also discussed concerns and potential avenues for future study in the duo for cybersecurity and intrusion detection for IIoT.

2.6. An Overview of Related Works and Areas for Research

The review mentioned above in blockchain and IoT integration shows that the applicability of blockchain to IoT cybersecurity has not yet been thoroughly studied. This evaluation effort boldly depicts the connection between explainable AI (XAI) and IoT security. Although there are surveys on the integration of blockchain and IoT for cybersecurity, as far as we know, research has yet to consider demonstrating the role of blockchain in improving intrusion detection performance. However, despite the multiple surveys regarding the security of IoT and blockchain, a limited survey is available on integrating both technologies for intrusion detection in IoT. Thus, this is the first attempt to comprehensively review incorporating blockchain and IoT for secure intrusion detection in IoT networks. Table 1 summarizes the research on blockchain technology's role in IoT cybersecurity, focusing on its limitations and the advancements proposed in this research. Existing studies offer valuable insights into how blockchain can enhance IoT security, particularly in IDS. This research guides future investigations, suggesting avenues for further integration and exploration in this field.

Table 1. Summary of the review of the literature highlighting the key findings of each paper (Yes: √, No: χ).

Author	Year	Systematic Review Methodology	IoT	Blockchain Integration	IDS	Use Case Demonstration
[75]	2018	χ	√	χ	χ	χ
[53]	2019	χ	√	χ	χ	χ
[74]	2019	χ	√	√	χ	χ
[68]	2019	√	√	χ	χ	χ
[54]	2019	χ	√	χ	χ	χ
[76]	2020	χ	χ	χ	χ	χ
[77]	2020	χ	√	√	√	χ
[78]	2020	χ	√	√	χ	χ
[55]	2021	χ	√	√	χ	χ
[58]	2021	χ	√	χ	χ	χ
[12]	2022	√	√	√	√	χ
[38]	2022	χ	√	χ	√	χ
[57]	2022	χ	√	χ	χ	χ
[56]	2022	χ	√	χ	χ	χ
[52]	2022	χ	√	√	χ	χ
[65]	2022	√	√	√	χ	χ
[79]	2022	χ	√	√	χ	χ
[21]	2023	χ	√	√	χ	χ
[80]	2023	χ	√	√	√	χ
[59]	2024	√	√	√	√	χ
This Study	2024	√	√	√	√	√

3. Review Methodology

This section presents a methodical illustration utilized for the in-depth evaluation. The 'mentefacto conceptual design' [81] and the meta-analysis (PRISMA) [82,83] served as inspiration for the creative reviewing methodologies used in this work. Articles released between 2019 and 2024 were given precedence during the selection process. However, the year of publication becomes inconsequential in the event of a historical context in the evaluation. Ref. [84] states that the following databases are the best places to find research about computer science and engineering: IEEE Xplore, ScienceDirect, Springer, a few social media sites like Academia and Google Scholar, ResearchGate, and Sage. Furthermore, we only considered papers prepared in English for our final evaluation. Table 2 summarizes the lists of documents based on the database source using the critical search terms AI', 'CYBERSECURITY', 'BLOCKCHAIN', 'IDS', 'IoT', and 'IIoT'. Similarly, Figure 4 displays the flow diagram for PRISMA used for the comprehensive analysis and the criteria for choosing the last set of papers. For IDS, only AI-BLOCKCHAIN integrated documents were used for quantitative analysis. As a result, a thorough narrative analysis enabled a systematic summary and described the findings of the screened literature. The following are the requirements for inclusion of papers in the survey:

1. The articles must be original works released as conference proceedings, journals, or arXiv.
2. The final discussion does not consider background and history; only papers published between 2019 and 2024 are included.
3. Only articles that discuss the problems and challenges of integrating AI-BLOCKCHAIN for cybersecurity are considered for the qualitative study.

4. To be eligible for comparison, this review paper must address blockchain and AI integration for IDS and security compared to other recent review works.
5. English must be used to write all of the papers.
6. Finally, publications with access restrictions are excluded because the writers could not access the databases.

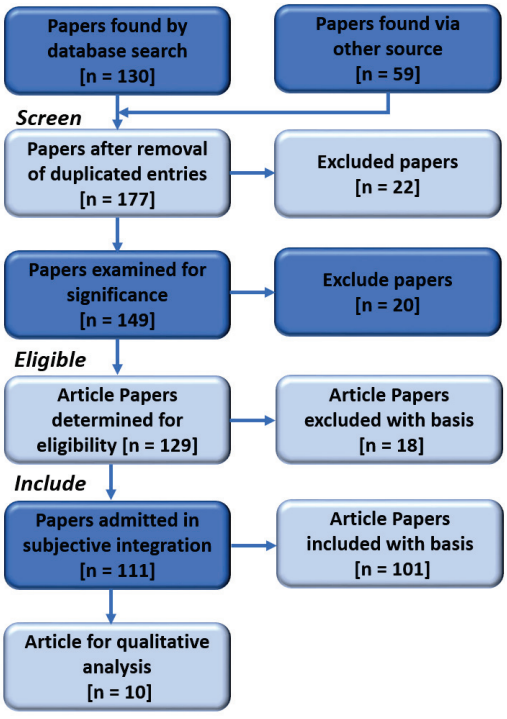


Figure 4. PRISMA flow illustrates how the final 111 papers at the reference were chosen and the 10 publications that specifically addressed combining blockchain with AI for IDS in IoT.

Table 2. Publications employed in this study.

Database Source	No. of Documents	% Freq
IEEE Xplore (Journals)	30	27.03
IEEE Xplore (Conferences)	10	9.01
MDPI	18	16.22
Springer	18	16.22
ACM	3	2.70
arXiv Pre-print	2	1.80
Google Scholar	5	4.50
Hindawi	1	0.9
Frontiers	2	1.80
Taylor & Francis	1	1.09
ScienceDirect (Elsevier)	12	10.81
Other Sources (Blogs, Reports, and Websites)	10	9.01
Total	111	100.00

Table 2 and Figure 4 contain the usage summary and document searches, respectively. A total of 189 (130 + 59) publications were found throughout the search. Screening out 22 papers due to duplication left 167. Following a relevancy screening and removing papers with open abstracts but restricted access to the full text, 20 were eliminated. With the above inclusion criteria, 20 of the 129 left documents were eliminated. For the survey, 111 publications in all were used. Of these, 101 were used for qualitative analysis, and the remaining 10 (see Table 3) were only articles on AI and BLOCKCHAIN integration implementation for cybersecurity. Thus, they were used for the particular review.

Table 3. Articles on blockchain-based IDS techniques for IoT/IIoT.

Study	Technique	Focus	Achievement	Year
[85]	Proposed a combination of blockchain and CNN for Software-defined network (SDN)-based IIoT architectures	To detect and prevent security threats in the application and network security layers of the SDN-based IIoT architectures.	Minimized the impact of attacks on SDN-based IIoT architecture layers.	2023
[86]	Creation of an IDS powered by ML algorithms and blockchain to improve the privacy and security of IoT devices.	Aims to encrypt interactions between IoT devices.	Simulation results could improve privacy and security by providing a tamper-proof decentralized communication system.	2023
[87]	Deep learning with blockchain orchestration for safe data transfer in IoT-enabled healthcare systems.	The approach ensures secure data transmission and integrity by exploiting the zero-knowledge proof (ZKP) scheme	Using the Ethereum smart contract to handle data security concerns with the interplanetary file system (IPFS) for off-chain storage to alleviate the problem of data storage costs.	2023
[88]	A hybrid decision tree method	To integrate ML with blockchain for anomaly detection	Predict attack within the shortest time with high detection accuracy.	2023
[89]	A lightweight blockchain security model driven by AI.	To guarantee the security and privacy of cloud-based IIoT systems.	Improved performance in anomaly detection when compared with other models.	2023
[90]	A secure aggregation mechanism for FL based on blockchain	By ensuring secure aggregation, local device data masking stops hostile servers from compromising and reconstructing training data.	The technique minimizes resource waste and quickens the global model's convergence rate by synchronizing clients with an antiquated model.	2023
[91]	A blockchain network is used in the proposed system for a safe FL model aggregation.	To safely carry out the FL-based aggregation and produce a global model.	According to experimental results, the framework's processing time was nearly identical to that of the original FL model.	2023
[92]	Multi-signature authentication is used to confirm the integrity of the global ML model and TEE is used to safeguard each client's local model training.	To give a verifiable ML model and guarantee the participant's local model training security.	The training on the secure enclave resulted in a slight drop in accuracy, according to the experimental findings. Additionally, multi-signature execution time has no discernible impact on blockchain network speed.	2023
[93]	A blockchain-driven edge intelligence methodology	Incorporates blockchain based on a reputation for decentralized transaction recording and verification, guaranteeing privacy and data protection.	The simulation findings validate the approach's efficiency and robustness over state-of-the-art cyberattack detection methods.	2022

Table 3. *Cont.*

Study	Technique	Focus	Achievement	Year
[94]	A security architecture that combines SDN and blockchain technology.	To defend industrial control processes from counterfeit commands and stop misrouting attacks on OpenFlow rules in industrial IoT systems with SDN enabled.	The assessment's findings confirm the suggested security measures' effectiveness and efficiency.	2019

The rapid IoT expansion, connecting billions of devices, faces several critical challenges, including counterfeit hardware, communication security, system management complexities, and data privacy issues. Although in its infancy, studies have shown that the emergence of blockchain technology, known for its decentralization, transparency, and security, aims to enhance resilience against single points of failure. Immutable records enable transparent data sharing and auditing with cryptographic mechanisms to improve data integrity and authentication while automating processes and enforcing rules within the IoT network. Table 3 highlights some attempts at blockchain-based intrusion detection systems for IoT. The next section spotlights this study's findings on blockchain applications for IoT intrusion detection.

4. Findings and Discussion

4.1. Role of Blockchain in Enhancing IDS Security

Blockchain enhances IDS security by ensuring the integrity and reliability of security data through its immutable and decentralized architecture [85–87,95]. Timestamping and cryptographically linking data blocks enable secure and tamper-proof logging of IDS events, mitigating the risk of data manipulation or deletion [91,92]. Additionally, blockchain's distributed consensus mechanism eliminates single points of failure in IDS networks. It provides a transparent audit trail of security events, enhancing reliability and fault tolerance against cyber threats [87,90], and strengthening IDS security. The immutability of blockchain prevents unauthorized alterations once data are recorded, while distributed nodes fortify resilience and resistance against attacks [88,89,94]. Encryption further safeguards identity data, making any attempts to alter data detectable due to the blockchain's transparent nature [87,93]. Existing implementations and case studies validate blockchain-based decentralized identity management systems, with key players like IBM spearheading advancements in various sectors using blockchain technology [96]. The decentralized identity management systems empower consumers with control over their data, ensuring enhanced privacy, security, and interoperability while fostering a user-centric approach [96]. Overcoming challenges and effectively adopting blockchain can lead to developing safer and more respectful digital identity management systems [96].

Integrating blockchain technology with ML holds immense potential for strengthening security in IoT networks. It offers tamper-resistant and transparent data integrity and transaction verification, which enhances intrusion detection capabilities [16]. This synergy enables real-time monitoring, pattern identification, and anomaly detection, facilitating efficient resource management and proactive maintenance [97]. Combining optimization techniques yields a robust and scalable solution for intrusion detection, boosting efficiency and security in interconnected environments [98]. Improves process efficiency, reducing spatiotemporal scenarios and enabling smart manufacturing processes while facilitating extensive data analysis in industrial IoT networks [12]. Combining blockchain's immutable nature with ML's resilience enhances IDS accuracy, trustworthiness, and transparency, ensuring data integrity, eliminating single points of failure, and providing auditable and tamper-proof logs and audit trails for better monitoring and accountability in security incidents.

Some studies have explored the integration of blockchain and ML for intrusion detection, as in Table 3. These attempts showed significance in the various applications. Figure 5 demonstrates the combination of blockchain and IDS in a federated learning approach.

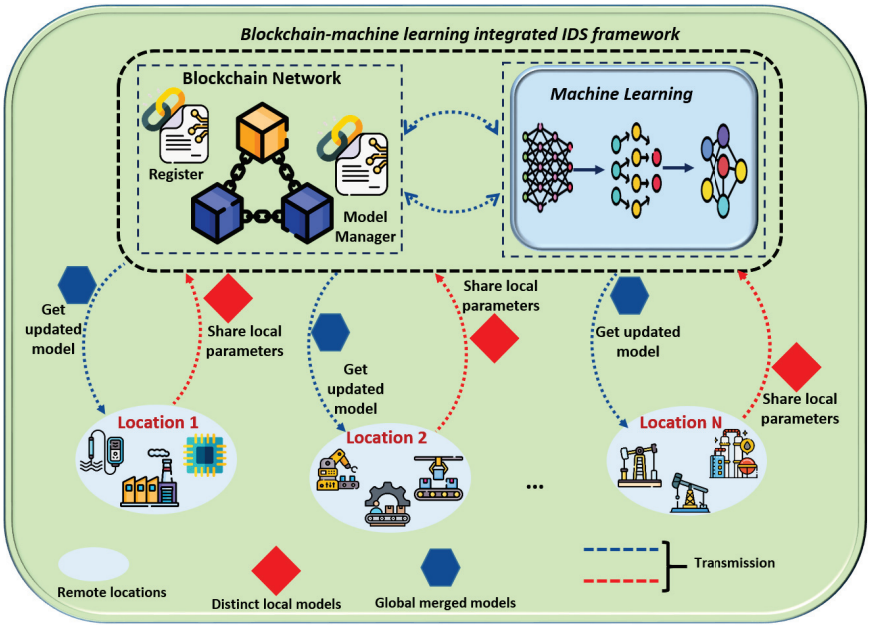


Figure 5. Illustrating the integration of blockchain and machine learning-based intrusion detection.

4.2. Security of IoT Devices

IoT systems encounter security risks like data breaches and malware due to the sensitive nature of stored information, leading to identity theft and fraud. Exploring IoT and blockchain security delves into blockchain's potential applications in enhancing IoT security. IoT devices are fundamental components of IoT systems, yet they face significant security challenges due to resource constraints and diverse deployment environments. Authentication and authorization methods for IoT devices in blockchain-based systems must be lightweight and scalable to accommodate limited resources while ensuring security and privacy. Additionally, ensuring firmware integrity and facilitating secure updates are critical but complex tasks in IoT device security, which blockchain smart contracts and consensus mechanisms can address. Moreover, securing communication channels and data privacy and addressing physical security concerns are essential to protecting blockchain-based IoT systems from various threats [2,5,52].

4.3. IoT Network Security

In blockchain-based IoT systems, network security is paramount for ensuring system reliability. Challenges include mitigating DDoS attacks, where the distributed network's nature amplifies risks, and addressing Sybil attacks through identity verification mechanisms. Detecting and mitigating rogue devices is crucial, facilitated by blockchain's tracking capabilities and anomaly detection algorithms. Additionally, interoperability challenges between IoT devices and blockchain networks necessitate standardized communication protocols and APIs [68]. Furthermore, as IoT device interconnections grow, adopting blockchain can streamline data flow by enabling direct device interactions without centralized servers, enhancing system efficiency and security [5,21,75].

4.4. Blockchain Security in IoT

Blockchain technology, fundamental to blockchain-based IoT systems, introduces unique cybersecurity challenges. Consensus mechanisms, crucial for transaction validation, pose challenges due to IoT device constraints, requiring lightweight and energy-efficient solutions. Scalability and performance issues arise from the high transaction volume, demanding scalable blockchain solutions tailored to IoT requirements. Privacy concerns necessitate privacy-preserving techniques like ZKPs, while smart contracts require rigorous security measures to mitigate vulnerabilities. Achieving consensus and governance among diverse stakeholders and navigating regulatory complexities further underscore the multifaceted security challenges in blockchain-based IoT systems [2,51].

IoT devices face security risks like data breaches and malware due to design flaws, interoperability, and remote deployment. Blockchain integration offers encryption, authentication, access control, and vulnerability management to bolster IoT security [2,66]. Although XAI enhances accountability, debugging, adaptation, and deterrence against attacks in blockchain-based IoT systems [83,99,100], additional methodologies such as blockchain, feature attribution, model summarization, counterfactual justifications, and causal modeling enhance system accountability and transparency [100,101]. Despite limitations such as standardization, data privacy, and computational complexity, the potential benefits of XAI in blockchain systems merit further exploration [83,100]. Table 3 summarizes the solutions and challenges across various integration approaches of blockchain for IoT network cybersecurity, highlighting the importance of leveraging blockchain security features to fortify IoT systems against threats and safeguard their operations and data integrity.

4.5. Blockchain Application in IoT

The implementation of blockchain has a lot of possibilities to improve the functionality and security of IoT, particularly in smart factories/cities, by ensuring increased trust, transparency, and efficiency [2]. Critical blockchain applications in IoT systems include data integrity and security, achieved through tamper-resistant ledgers distributed across multiple nodes. These are crucial for managing vast data generated in intelligent systems [2,76,80]. Blockchain-based identity management ensures secure access to devices and services by assigning unique cryptographic identities stored on the blockchain, establishing a trustworthy framework for managing IoT ecosystem entities [76,102,103]. Smart contracts automate transactions and processes in smart cities, enabling efficiency and transparency in areas like energy trading and supply chain management [2,55,59]. Blockchain-based IoT networks with decentralized infrastructure and interconnectedness have fewer intermediaries and single points of failure, increasing security and reliability and boosting resilience for industrial applications [55,104,105]. Moreover, blockchain enables micropayments and value exchange between IoT devices, simplifying transactions in scenarios like automated processes in smart factories [77,80]. While this integration promises to revolutionize industrial processes, scalability, interoperability, and governance must be addressed for large-scale IoT deployments [2]. Data transparency and the need for security, availability, and trustworthiness have motivated blockchain technology in IoT. The performance, methodology, and year of publication of a few recent research on blockchain-based IDS for IoT are summarized in Table 3.

The surge in cybersecurity attacks presents a significant challenge for protecting IoT networks due to their inherent vulnerabilities and resource constraints. Integrating IoT with AI has gained traction to bolster security by leveraging AI's analytical capabilities to detect attack patterns across network traffic [65,78]. However, centralized AI-based approaches face trust and scalability issues, making them incompatible with the decentralized nature of IoT. By facilitating safe data flow between untrusted nodes and offering decentralized defense strategies, blockchain promises to improve IoT security. Despite its potential, blockchain solutions encounter challenges such as limited insight into IoT networks and scalability issues [65,106]. More efficient and intelligent decentralized defense solutions are

needed to overcome these hurdles, with AI and blockchain emerging as promising allies, combining AI's analytical prowess with blockchain's decentralized architecture.

A blockchain-based training scheme, a secure support vector machine, was presented by Shen et al. [107] to protect IoT data privacy in smart city applications. By leveraging blockchain, the model enables secure data sharing among IoT data providers without depending upon a trusted agent, utilizing a Paillier cryptosystem and encryption methods to guarantee data ownership, integrity, and privacy during training. IoT devices transmit encrypted data to data providers through a unified blockchain ledger, which is stored securely. The proposed approach, evaluated with real-world datasets like the heart disease and breast cancer Wisconsin datasets, maintains SVM classifier accuracy while preserving IoT data privacy.

BlockIoTIntelligence is a framework developed to improve big data analytics by combining blockchain and AI [67]. It consists of four layers: cloud intelligence, fog intelligence, edge intelligence, and device intelligence. Each layer uses blockchain and AI to process and analyze data. The suggested approach proved highly accurate in tackling IoT security, confidentiality, storage capacity, and data flow concerns via qualitative and quantitative analysis, offering a more effective solution than conventional IoT schemes.

4.6. Case Study of AI-Blockchain Integration and Result Evaluation

To solve the security and privacy issues with IIoT systems, the works of [89] combine blockchain with AI. In addition, the framework achieves improved classification and detection accuracy with reduced execution time. The anomaly detection performance is improved when auto-encoder-based transformation and blockchain authentication are combined, as seen by the suggested model's ability to combine high security and computational efficiency compared to alternative methods. However, the major obstacle is the computational complexity of the consensus mechanism.

In [90], researchers utilized blockchain in an FL framework to protect end device data from malicious servers. They introduced a callback mechanism to streamline communication between FL servers and devices, addressing issues like stragglers and dropouts while ensuring secure aggregation of masked models to minimize complexity and resource usage, especially for IoT devices. Despite the computational complexities inherent in blockchain, this method enhanced computational efficiency and ensured secure communication within the federated network. However, using an aggregator/server limits the system to a single point of failure, which is quite concerning.

Blockchain was introduced by Kalapaaking et al. [91] to provide safe FL localized model aggregation in IoT networks. Every blockchain node performed secure aggregation activities using a trusted execution environment (TEE) based on Intel Software Guard Extensions (SGX). Experiments showed comparable processing times to the original FL model and a slight 2% decrease in accuracy. Further practical improvements are needed, such as support for diverse jobs in blockchain-based FL with TEE-based secure aggregation, even though a hash-based consensus mechanism assures model fidelity.

Similarly, a subsequent study by [92] presented a secure and verifiable FL framework for IoT systems, utilizing a TEE and multi-signature scheme to protect the training process and ensure model integrity. Participants trained local models within the TEE, which were verified by the blockchain aggregation manager before aggregation by blockchain nodes. The resulting global model was stored in tamper-proof storage, verified through multi-signature, and distributed to FL participants for subsequent rounds, ensuring the integrity of the FL process. Despite the achievement, the approach still requires enhancing the security and speed of the training process within the secure enclave.

Recently, a study integrated blockchain to secure data privacy during aggregation and transmission, ensuring model update integrity and transparency through smart contracts [46]. This combination minimized global model loss with improved detection accuracy against adversarial perturbation, with average latency for training and aggregation, respectively, enhancing security, scalability, and participant trust.

4.7. Practical Implementation and Evaluation Results

Monitoring and optimizing the performance of the IoT systems is essential for intelligent industrial operations, with accurate attack prediction key for effective security management. IoT devices serve as crucial sensors and data collectors in monitoring industrial processes, with real-time data used to train ML models for cybersecurity threat detection, bolstering critical infrastructure security against attacks. However, sharing sensor data centrally raises concerns about privacy and security. FL shows promise in addressing these issues by conducting ML model training on the clients but faces a vulnerability in model parameter aggregation, leading to sensitive information leakage through aggregated model updates.

This study includes exploratory results of a blockchain implementation to secure model aggregation to tackle this challenge. The proposed system integrates blockchain with FL, utilizing a private Ethereum platform to establish a blockchain network. FL, executed through a Solidity-based smart contract, incorporates data encryption and aggregation facilitated by the web3.py Python library for blockchain communication. Training involves a convolutional neural network model with enhanced security via adversarial training, fortifying the model against attacks. This methodology combines blockchain platforms, smart contracts, and ML algorithms aided by the application binary interface (ABI), as demonstrated in Figures 6–9. This results in a robust FL system with strong privacy guarantees and resistance to adversarial attacks.

```
#Compiling Solidity Smart Contracts in Python
install_solc("0.6.0")
compiled_sol = compile_standard(
    {
        "language": "Solidity",
        "sources": {"smart_contract.sol": {"content": simple_storage_file}},
        "settings": {
            "outputSelection": {
                "*": {
                    "*": ["abi", "metadata", "evm.bytecode", "evm.bytecode.sourceMap"]
                }
            }
        },
    },
    solc_version="0.6.0"
)
with open("compiled_code.json", "w") as file:
    json.dump(compiled_sol, file)
```

Figure 6. Screenshot showing the compilation of the smart contract.

```
# get abi and contract addresss
abi = json.loads(
    compiled_sol["contracts"]["smart_contract.sol"]["FuCBCFL"]["metadata"])["output"]["abi"]
```

Figure 7. Screenshot showing the connection of the smart contract ABI.

```
#set up the connection between Python and a local blockchain network (Ganache) using the Web3 library.
w3 = Web3(Web3.HTTPProvider("HTTP://192.168.0.145:7545",request_kwargs={'timeout': 50}))
```

Figure 8. Screenshot showing the connection between Python and Ganache.

```
def register(contract, public_key, connection):
    greeting_transaction = contract.functions.register().transact({ "from": public_key,})
    tx = connection.w3.eth.wait_for_transaction_receipt(greeting_transaction)
    send_data_Transactions(connection,tx,greeting_transaction)
    send_data_blocks(connection,greeting_transaction)
    print(f"Client registred...")
```

Figure 9. Screenshot showing the blockchain interaction function.

The smart contract, built on the Ethereum blockchain platform, automates contract terms enforcement and facilitates the transmission of FL-trained models to the blockchain. Development involved using Remix IDE (https://github.com/ethereum/remix-project, accessed on 1 February 2024) and Solidity linters for accuracy and robustness checks, ensuring adherence to best practices and security standards. Complementing the blockchain, the InterPlanetary File System (IPFS) [108] offers distributed file storage and sharing, enhancing decentralization and data availability. The Web3.py (https://github.com/ethereum/web3.py, accessed on 1 February 2024) Python library enables interaction with the Ethereum blockchain, simplifying smart contract handling and providing tools for development and testing. Challenges in storing model updates on the blockchain due to high gas fees and scalability constraints are addressed by utilizing IPFS for secure storage, minimizing the blockchain's burden. Interaction functions, such as *register*(), facilitate client participation in FL by initiating registration transactions, and *send_data_transactions*() and *send_data_blocks*() functions ensure transparency by visualizing transaction details and blockchain interaction blocks in the FL process. Table 4 illustrates the performance evaluation of blockchain–FL integration for securing model aggregation against adversarial attacks [46,109].

Table 4. Comparison of the blockchain–FL integration performance between Edge-IIoT and IoT Network intrusion datasets.

	Edge-IIoT Dataset						**IoT Network Intrusion Dataset**					
	Local Model			**Global Model**			**Local Model**			**Global Model**		
Clients	**Accuracy (%)**	**Loss**	**Train Time (s)**	**Accuracy (%)**	**Loss**	**Aggregation Time (s)**	**Accuracy (%)**	**Loss**	**Train Time (s)**	**Accuracy (%)**	**Loss**	**Aggregation Time (s)**
1	81.72	0.7359	580	84.80	0.5895	501	94.11	0.2205	486	95.74	0.1551	401
2	82.59	0.7538	572	85.09	0.5892	555	93.88	0.1685	475	95.68	0.1579	445
3	80.69	0.7475	563	84.77	0.5873	564	93.82	0.2270	464	95.47	0.1638	456
4	79.60	0.7619	420	84.97	0.5871	599	94.03	0.2215	388	95.57	0.1603	548
5	82.79	0.7326	389	84.80	0.5990	530	93.84	0.2311	400	95.94	0.1568	509

Simulation results show the possibility of integrating blockchain and FL to enhance secure model aggregation, resulting in higher accuracy and reduced loss values. It supports the advocated strategy of blockchain integration to safeguard model aggregation in FL against adversarial attacks. Efficient data transfer and aggregation are crucial, especially in FL frameworks with blockchain, considering network conditions and model complexity. Network latency becomes a critical factor in scenarios involving dispersed IoT devices.

This study validated its concepts using the Edge-IIoT set (https://ieee-dataport.org/documents/edge-iiotset-new-comprehensive-realistic-cyber-security-dataset-iot-and-iiot-applications, accessed on 2 November 2023) and IoT network intrusion detection (https://ieee-dataport.org/open-access/iot-network-intrusion-dataset, accessed on 2 November 2023) datasets, featuring 62 predictors, 15 classifications, and 157,800 observations in the Edge-IIoT set. The IoT environment covered by these datasets includes a variety of network attack types, including DoS, unauthorized commands, MiTM, reconnaissance, command injection and backdoor attacks, metadata, event data, device identifiers, communication protocols, and regular traffic. Particular sensor data, selected for their applicability in representing communication in a highly vulnerable IIoT network, are included. The sensor data include attributes like IP addresses, ports, protocols, packet length, flow time, and statistics. On an Intel(R) Core(TM) i5-8500 CPU @ 3.00 GHz PC with 8 GB RAM running Windows 11, the experimentation environment consisted of Visual Studio Code, Ganache v2.7.1, Solidity v0.8.22, and Python 3.6.13.

4.8. Blockchain-as-a-Service (BaaS): IoT Cybersecurity Perspective

Blockchain-as-a-service (*BaaS*) in IoT has gained significant attention due to its potential to address cybersecurity concerns [2,50,54,79]. By integrating blockchain technology into IoT networks, BaaS offers several advantages in enhancing cybersecurity:

1. Data integrity and immutability: Providing a tamper-resistant and immutable ledger ensures data integrity stored within IoT networks. Due to the cryptographic links between every transaction on the blockchain and earlier transactions, it is nearly complicated to change past data without the network's participants' consent [90].
2. Secure data exchange: Enables safe, direct, peer-to-peer data transfer between IoT devices [46,91,92]. To reduce the danger of data modification or illegal access, smart contracts and programmable self-executing agreements on the blockchain enable automatic and safe data exchanges based on established conditions [46].
3. Decentralization and resilience: Its decentralized architecture eliminates single points of failure, enhancing the resilience of IoT networks against cyberattacks. With no central authority controlling the network, blockchain ensures that data remain accessible even in node failures or malicious attacks [46,91–93].
4. Identity and access management: It makes it possible for IoT devices to have strong identification and access control systems, mitigating the risk of unauthorized access and identity spoofing. By confirming the identity of network participants, *BaaS* improves the security of IoT devices using decentralized authentication procedures [67,107].
5. Audibility and transparency: Real-time auditing of transactions within IoT networks is made possible by the transparent nature of blockchain technology. Data exchange and operations recorded on the blockchain are traceable to their origin, enabling forensic analysis and accountability in case of security breaches [51,89].

While promising solutions are available to improve cybersecurity in the IoT by offering data integrity, decentralized control, and transparency, resolving identified issues is necessary to utilize *BaaS* in IoT network security fully.

4.9. Open Issues and Future Direction

In IoT deployments, devices' limited computational power, memory, and energy pose challenges for integrating blockchain solutions [110]. Running full blockchain nodes on resource-constrained devices is impractical, and *BaaS* introduces additional latency and strain on such devices due to network communication and consensus requirements [110]. Moreover, the transparency of public blockchains raises privacy concerns, necessitating privacy-preserving techniques for sensitive IoT data [79]. Integrating *BaaS* with IoT systems requires careful planning to address interoperability and data synchronization complexities [79]. As researchers explore lightweight cryptographic schemes and efficient integration methods, it is crucial to balance the security benefits of blockchain with the computational demands imposed on IoT devices.

While *BaaS* offers benefits for IoT, it faces challenges in scalability, consensus mechanisms, and latency, given the volume of interconnected devices. The overhead costs of blockchain operations, including transaction fees and infrastructure, may limit its feasibility in resource-constrained IoT environments. Additionally, the immutable nature of blockchain transactions raises privacy concerns, necessitating privacy-preserving techniques like ZKPs or private blockchains to protect sensitive IoT data [2,54,79].

Layer-2 protocols, such as lightning networks, boost blockchain scalability by handling transactions off-chain and settling periodically on the main chain. Bridging diverse blockchains via standards like Polkadot and Cosmos fosters seamless communication among IoT networks [54,65]. PoS offers energy efficiency compared to PoW, which is particularly beneficial for power-intensive IoT mining activities [54,111]. Directed acyclic graphs provide scalability and energy efficiency, aligning well with IoT needs. Hybrid models merging blockchain with traditional databases ensure security and efficiency by storing metadata on-chain while keeping raw data off-chain [54]. Blockchain-driven edge

computing enables edge devices to manage a lightweight blockchain locally, reducing latency and enhancing privacy.

The ongoing advancement of quantum computing underscores the importance of fortifying blockchain against quantum attacks through research on quantum-safe cryptographic algorithms [69–71]. Addressing blockchain's energy consumption requires focusing on energy-efficient consensus protocols like PoS or PoA [54]. Scalability solutions such as sharding, sidechains, or off-chain protocols are critical for large-scale IoT deployments. Future exploration areas include interoperability standards, privacy-preserving mechanisms like ZKPs, and integrating AI/ML with blockchain for robust cybersecurity [54,65]. Real-world testing, regulatory compliance frameworks, and collaborative industry efforts are pivotal for validating and implementing blockchain-based IoT security solutions [2,54].

This study reveals that to uphold blockchain integrity and security against quantum threats, it is crucial for blockchain platforms to seamlessly integrate algorithms like PQC while upgrading cryptographic primitives like digital signatures, key exchange protocols, and hash functions to PQC standards [71–73]. There is a need to maintain reverse collaboration to enhance transaction efficiency. Collaboration among blockchain developers, cryptographers, and quantum computing specialists is necessary to navigate this quantum transition safely [71–73].

5. Conclusions

This paper comprehensively reviews blockchain applications for cybersecurity in IoT networks, presenting a robust framework for integrating blockchain into IDS within IoT networks and addressing crucial research avenues, current trends, and notable challenges. The study illuminates emerging areas in IoT security through a systematic analysis of articles spanning AI, blockchain, IDS, IoT, and IIoT. By evaluating recent advancements and diverse AI blockchain integration methods, this research underscores the pivotal role of blockchain in bolstering intrusion detection performance. This framework offers a roadmap for collaborative exploration, aiming to advance IDS for universally accessible, scalable, transparent, immutable, and decentralized IoT networks, driving innovation in IoT cybersecurity. Simulation results from highlighted case studies demonstrate that—despite resource constraints and privacy issues—blockchain's intense presence in IoT networks ensures ongoing progress toward a more secure and resilient IoT landscape, necessitating further research into lightweight cryptography, efficient consensus mechanisms, and privacy-preserving techniques to overcome existing barriers.

Author Contributions: Conceptualization, L.A.C.A. and C.I.N.; methodology, All authors contributed equally; software, L.A.C.A. and C.I.N.; validation, L.A.C.A., C.I.N. and D.-S.K.; formal analysis, L.A.C.A.; investigation, All authors contributed equally; resources, L.A.C.A. and D.-S.K.; data curation, L.A.C.A.; writing—original draft preparation, all authors contributed; writing—review and editing, L.A.C.A. and C.I.N.; visualization, L.A.C.A. and C.I.N.; supervision, D.-S.K.; project administration, C.I.N. and D.-S.K.; funding acquisition, L.A.C.A. and D.-S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by Innovative Human Resource Development for Local Intellectualization program through the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (IITP-2024-2020-0-01612, 50%) and Priority Research Centers Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2018R1A6A1A03024003, 50%)

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: This work was partly supported by Innovative Human Resource Development for Local Intellectualization program through the Institute of Information & Communications Tech-

nology Planning & Evaluation (IITP), the Ministry of Science and ICT (MSIT) and Priority Research Centers Program through the National Research Foundation of Korea (NRF).

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

AI	artificial intelligence
ABI	application binary interface
API	application programming interface
ANN	artificial neural network
BaaS	blockchain-as-a-service
CNN	convolutional neural network
CRNN	convolutional and recurrent neural network
DDoS	distributed denial-of-service
DL	deep learning
DoS	denial-of-service
DPoS	delegated proof of stake
ECC	elliptic curve cryptography
FL	federated learning
HMI	human–machine interface
ICS	industrial control system
IDE	integrated development environment
IED	intelligent end device
IDS	intrusion detection system
IoT	Internet of Things
IIoT	Industrial Internet of Things
IPFS	interplanetary file system
LOIC	Low Orbit Ion Cannon
MiTM	man-in-the-middle
ML	machine learning
MTU	master terminal unit
MSU	master station unit
OTA	over-the-air
OSI	open systems interconnection
PBFT	practical Byzantine fault tolerance
PLC	programmable logic controller
PoA	proof of authority
PoS	proof of stake
PoW	proof of work
PQC	post-quantum cryptographic
PRISMA	preferred reporting items for systematic reviews and meta-analyses
RFID	radio frequency identification
RNN	recurrent neural network
RSA	Rivest–Shamir–Adleman
RSU	remote station unit
RTU	remote terminal unit
SCADA	supervisory control and data acquisition
SDN	software-defined network
SGX	Software Guard Extensions
SVM	support vector machine
TEE	trusted execution environment
XAI	explainable AI
ZKP	zero-knowledge proof

References

1. Ahakonye, L.A.C.; Nwakanma, C.I.; Lee, J.M.; Kim, D.S. Trees Bootstrap Aggregation for Detection and Characterization of IoT-SCADA Network Traffic. *IEEE Trans. Ind. Inform.* **2023**, *20*, 5217–5228. [CrossRef]
2. Alajlan, R.; Alhumam, N.; Frikha, M. Cybersecurity for Blockchain-Based IoT Systems: A Review. *Appl. Sci.* **2023**, *13*, 7432. [CrossRef]
3. Ahakonye, L.A.C.; Nwakanma, C.I.; Lee, J.M.; Kim, D.S. Agnostic CH-DT Technique for SCADA Network High-Dimensional Data-Aware Intrusion Detection System. *IEEE Internet Things J.* **2023**, *10*, 10344–10356. [CrossRef]
4. Ahakonye, L.A.C.; Nwakanma, C.I.; Lee, J.M.; Kim, D.S. SCADA Intrusion Detection Scheme Exploiting the Fusion of Modified Decision Tree and Chi-Square Feature Selection. *Internet Things* **2023**, *21*, 100676. [CrossRef]
5. Panarello, A.; Tapas, N.; Merlino, G.; Longo, F.; Puliafito, A. Blockchain and IoT Integration: A Systematic Survey. *Sensors* **2018**, *18*, 2575. [CrossRef] [PubMed]
6. Mahmood, S.; Chadhar, M.; Firmin, S. Cybersecurity Challenges in Blockchain Technology: A Scoping Review. *Hum. Behav. Emerg. Technol.* **2022**, *2022*, 7384000. [CrossRef]
7. Elrawy, M.F.; Awad, A.I.; Hamed, H.F. Intrusion Detection Systems for IoT-Based Smart Environments: A Survey. *J. Cloud Comput.* **2018**, *7*, 21. [CrossRef]
8. Abdullahi, M.; Baashar, Y.; Alhussian, H.; Alwadain, A.; Aziz, N.; Capretz, L.F.; Abdulkadir, S.J. Detecting Cybersecurity Attacks in Internet of Things Using Artificial Intelligence Methods: A Systematic Literature Review. *Electronics* **2022**, *11*, 198. [CrossRef]
9. Ahakonye, L.A.C.; Amaizu, G.C.; Nwakanma, C.I.; Lee, J.M.; Kim, D.S. Classification and Characterization of Encoded Traffic in SCADA Network using Hybrid Deep Learning Scheme. *J. Commun. Netw.* **2024**, *26*, 65–79. [CrossRef]
10. Shareena, J.; Ramdas, A.; Haripriya, A.P. Intrusion Detection System for IoT Botnet Attacks using Deep Learning. *SN Comput. Sci.* **2021**, *2*, 205. [CrossRef]
11. Albulayhi, K.; Smadi, A.A.; Sheldon, F.T.; Abercrombie, R.K. IoT Intrusion Detection Taxonomy, Reference Architecture, and Analyses. *Sensors* **2021**, *21*, 6432. [CrossRef] [PubMed]
12. Mathew, S.S.; Hayawi, K.; Dawit, N.A.; Taleb, I.; Trabelsi, Z. Integration of Blockchain and Collaborative Intrusion Detection for Secure Data Transactions in Industrial IoT: A Survey. *Clust. Comput.* **2022**, *25*, 4129–4149. [CrossRef]
13. Khonde, S.; Ulagamuthalvi, V. Hybrid Intrusion Detection System using Blockchain Framework. *EURASIP J. Wirel. Commun. Netw.* **2022**, *2022*, 58. [CrossRef]
14. Ahakonye, L.A.C.; Nwakanma, C.I.; Lee, J.M.; Kim, D.S. Efficient Classification of Enciphered SCADA Network Traffic in Smart Factory Using Decision Tree Algorithm. *IEEE Access* **2021**, *9*, 154892–154901. [CrossRef]
15. Alfandi, O.; Khanji, S.; Ahmad, L.; Khattak, A. A Survey on Boosting IoT Security and Privacy Through Blockchain: Exploration, Requirements, and Open Issues. *Clust. Comput.* **2021**, *24*, 37–55. [CrossRef]
16. Hemashree, P.; Kavitha, V.; Mahalakshmi, S.; Praveena, K.; Tarunika, R. Machine Learning Approaches in Blockchain Technology-Based IoT Security: An Investigation on Current Developments and Open Challenges. In *Blockchain Transformations: Navigating the Decentralized Protocols Era*; Springer: Cham, Switzerland, 2024; pp. 107–130. [CrossRef]
17. Scully, P. Top 10 IoT Applications in 2020. IoT Analytics: Market Insights for the Internet of Things. 2020. Available online: https://iot-analytics.com/top-10-iot-applications-in-2020 (accessed on 1 February 2024).
18. Elgazzar, K.; Khalil, H.; Alghamdi, T.; Badr, A.; Abdelkader, G.; Elewah, A.; Buyya, R. Revisiting the Internet of Things: New Trends, Opportunities and Grand Challenges. *Front. Internet Things* **2022**, *1*, 1073780. [CrossRef]
19. Bouguettaya, A.; Sheng, Q.Z.; Benatallah, B.; Neiat, A.G.; Mistry, S.; Ghose, A.; Nepal, S.; Yao, L. An Internet of Things Service Roadmap. *Commun. ACM* **2021**, *64*, 86–95. [CrossRef]
20. Lin, J.; Yu, W.; Zhang, N.; Yang, X.; Zhang, H.; Zhao, W. A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. *IEEE Internet Things J.* **2017**, *4*, 1125–1142. [CrossRef]
21. Alam, T. Blockchain-Based Internet of Things: Review, Current Trends, Applications, and Future Challenges. *Computers* **2023**, *12*, 6. [CrossRef]
22. Newman, P. THE INTERNET OF THINGS 2020: Here's What over 400 IoT Decision-Makers Say about the Future of Enterprise Connectivity and How IoT Companies Can Use it to Grow Revenue. Business Insider. 2020. Available online: https://www.businessinsider.com/internet-of-things-report (accessed on 1 February 2024).
23. Wegner, P. Global IoT Market Size to Grow 19% in 2023—IoT Shows Resilience Despite Economic Downturn. IoT Analytics. 2020. Available online: https://iot-analytics.com/iot-market-size/ (accessed on 1 February 2024).
24. Ghosh, S.; Sampalli, S. A Survey of Security in SCADA Networks: Current Issues and Future Challenges. *IEEE Access* **2019**, *7*, 135812–135831. [CrossRef]
25. Hindy, H.; Brosset, D.; Bayne, E.; Seeam, A.K.; Tachtatzis, C.; Atkinson, R.; Bellekens, X. A Taxonomy of Network Threats and the Effect of Current Datasets on Intrusion Detection Systems. *IEEE Access* **2020**, *8*, 104650–104675. [CrossRef]
26. Zhu, B.; Joseph, A.; Sastry, S. A Taxonomy of Cyber Attacks on SCADA Systems. In Proceedings of the 2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing, Dalian, China, 19–22 October 2011; pp. 380–388. [CrossRef]
27. Hilal, H.; Nangim, A. Network Security Analysis SCADA System Automation on Industrial Process. In Proceedings of the 2017 International Conference on Broadband Communication, Wireless Sensors and Powering (BCWSP), Jakarta, Indonesia , 21–23 November 2017; pp. 1–6. [CrossRef]

28. Anand, P.; Singh, Y.; Selwal, A.; Singh, P.K.; Felseghi, R.A.; Raboaca, M.S. IoVT: Internet of Vulnerable Things? Threat Architecture, Attack Surfaces, and Vulnerabilities in Internet of Things and Its Applications towards Smart Grids. *Energies* **2020**, *13*, 4813. [CrossRef]
29. Kalluri, R.; Mahendra, L.; Kumar, R.S.; Prasad, G.G. Simulation and Impact Analysis of Denial-of-Service Attacks on Power SCADA. In Proceedings of the 2016 National Power Systems Conference (NPSC), Bhubaneswar, India, 19–21 December 2016; pp. 1–5. [CrossRef]
30. Amaizu, G.; Nwakanma, C.; Bhardwaj, S.; Lee, J.; Kim, D. Composite and Efficient DDoS Attack Detection Framework for B5G Networks. *Comput. Netw.* **2021**, *188*, 107871. [CrossRef]
31. Esfahani, A.; Mantas, G.; Ribeiro, J.; Bastos, J.; Mumtaz, S.; Violas, M.A.; Duarte, A.M.D.O.; Rodriguez, J. An Efficient Web Authentication Mechanism Preventing Man-in-the-Middle Attacks in Industry 4.0 Supply Chain. *IEEE Access* **2019**, *7*, 58981–58989. [CrossRef]
32. Melnick, J. Top 10 Most Common Types of Cyber Attacks. *Netwrix Blog* **2018**. Available online: https://blog.netwrix.com/2018/05/15/top-10-most-common-types-of-cyber-attacks/ (accessed on 1 February 2024).
33. Zhang, Y.; Xiang, Y.; Wang, L. Reliability Analysis of Power Grids with Cyber Vulnerability in SCADA System. In Proceedings of the 2014 IEEE PES General Meeting | Conference & Exposition, National Harbor, MD, USA, 27–31 July 2014; pp. 1–5. [CrossRef]
34. El Shafie, A.; Chihaoui, H.; Hamila, R.; Al-Dhahir, N.; Gastli, A.; Ben-Brahim, L. Impact of Passive and Active Security Attacks on MIMO Smart Grid Communications. *IEEE Syst. J.* **2018**, *13*, 2873–2876. [CrossRef]
35. Ustun, T.S.; Farooq, S.M.; Hussain, S.S. A Novel Approach for Mitigation of Replay and Masquerade Attacks in Smartgrids using IEC 61850 Standard. *IEEE Access* **2019**, *7*, 156044–156053. [CrossRef]
36. Rakas, S.V.B.; Stojanović, M.D.; Marković-Petrović, J.D. A Review of Research Work on Network-Based SCADA Intrusion Detection Systems. *IEEE Access* **2020**, *8*, 93083–93108. [CrossRef]
37. Prinz, W.; Rose, T.; Urbach, N. Blockchain Technology and International Data Spaces. In *Designing Data Spaces*; Springer: Cham, Switzerland, 2022. [CrossRef]
38. Attkan, A.; Ranga, V. Cyber-Physical Security for IoT Networks: A Comprehensive Review on Traditional, Blockchain and Artificial Intelligence Based Key-Security. *Complex Intell. Syst.* **2022**, *8*, 3559–3591. [CrossRef]
39. Li, W.; He, M.; Haiquan, S. An Overview of Blockchain Technology: Applications, Challenges and Future Trends. In Proceedings of the 2021 IEEE 11th International Conference on Electronics Information and Emergency Communication (ICEIEC), Beijing, China, 18–20 June 2021; pp. 31–39. [CrossRef]
40. Christidis, K.; Devetsikiotis, M. Blockchains and Smart Contracts for the Internet of Things. *IEEE Access* **2016**, *4*, 2292–2303. [CrossRef]
41. Wang, S.; Ouyang, L.; Yuan, Y.; Ni, X.; Han, X.; Wang, F.Y. Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends. *IEEE Trans. Syst. Man, Cybern. Syst.* **2019**, *49*, 2266–2277. [CrossRef]
42. Zheng, Z.; Xie, S.; Dai, H.; Chen, X.; Wang, H. An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. In Proceedings of the 2017 IEEE International Congress on Big Data (BigData Congress), Honolulu, HI, USA, 25–30 June 2017; pp. 557–564. [CrossRef]
43. Solanki, M.S. Overview of Blockchain Technology: Consensus, Architecture, and Its Future Trends. *Int. J. Innov. Res. Comput. Sci. Technol.* **2021**, *9*, 47–51. [CrossRef] [PubMed]
44. Dixit, P.; Bansal, A.; Rathore, P.S.; Payal, M. An Overview of Blockchain Technology: Architecture, Consensus Algorithm, and Its Challenges. In *Blockchain Technology and the Internet of Things*; Apple Academic Press: Palm Bay, FL, USA, 2020; pp. 21–46.
45. Dorri, A.; Kanhere, S.S.; Jurdak, R.; Gauravaram, P. Blockchain for IoT Security and Privacy: The Case Study of A Smart Home. In Proceedings of the 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom workshops), Kona, HI, USA, 13–17 March 2017; pp. 618–623. [CrossRef]
46. Ahakonye, L.A.C. Integrating Blockchain and Federated Learning for Secure Model Aggregation in IoT-SCADA Networks. Ph.D. Thesis, Kumoh National Institute of Technology, Gumi, Republic of Korea, 2023.
47. Jahid, A.; Alsharif, M.H.; Hall, T.J. The Convergence of Blockchain, IoT and 6G: Potential, Opportunities, Challenges and Research Roadmap. *J. Netw. Comput. Appl.* **2023**, *217*, 103677. [CrossRef]
48. Sandner, P.; Gross, J.; Richter, R. Convergence of Blockchain, IoT, and AI. *Front. Blockchain* **2020**, *3*, 522600. [CrossRef]
49. Komalavalli, C.; Saxena, D.; Laroiya, C. Overview of Blockchain Technology Concepts. In *Handbook of Research on Blockchain Technology*; Elsevier: Amsterdam, The Netherlands, 2020; pp. 349–371. [CrossRef]
50. Garces, E.; Li, S.; Daim, T.U. Cybersecurity and Technology Convergence: Analysis of AI, Blockchain, and IoT Using SNA. In *Cybersecurity: A Technology Landscape Analysis*; Springer: Cham, Switzerland, 2023; pp. 39–70. [CrossRef]
51. Awan, S.M.; Azad, M.A.; Arshad, J.; Waheed, U.; Sharif, T. A Blockchain-Inspired Attribute-Based Zero-Trust Access Control Model for IoT. *Information* **2023**, *14*, 129. [CrossRef]
52. Raju, M.C.; Paul, K.S. A Comprehensive Review of Cyber Security in Blockchain-Based IoT. *Math. Stat. Eng. Appl.* **2022**, *71*, 10646–10659. [CrossRef]
53. Padma, M.; KasiViswanath, N.; Swathi, T. Blockchain for IoT Application: Challenges and Issues. *Int. J. Recent Technol. Eng.* **2019**, *7*, 34–37.
54. Cui, P.; Guin, U.; Skjellum, A.; Umphress, D. Blockchain in IoT: Current Trends, Challenges, and Future Roadmap. *J. Hardw. Syst. Secur.* **2019**, *3*, 338–364. [CrossRef]

55. Ferrag, M.A.; Shu, L. The Performance Evaluation of Blockchain-Based Security and Privacy Systems for the Internet of Things: A Tutorial. *IEEE Internet Things J.* **2021**, *8*, 17236–17260. [CrossRef]
56. Alzoubi, Y.I.; Al-Ahmad, A.; Kahtan, H.; Jaradat, A. Internet of Things and Blockchain Integration: Security, Privacy, Technical, and Design Challenges. *Future Internet* **2022**, *14*, 216. [CrossRef]
57. Abubakar, M.; Jarocheh, Z.; Al-Dubai, A.; Liu, X. A Survey on The Integration of Blockchain and IoT: Challenges and Opportunities. In *Big Data Privacy and Security in Smart Cities*; Springer: Cham, Switzerland, 2022; pp. 197–221. [CrossRef]
58. Shammar, E.A.; Zahary, A.T.; Al-Shargabi, A.A. A survey of IoT and Blockchain Integration: Security Perspective. *IEEE Access* **2021**, *9*, 156114–156150. [CrossRef]
59. Ali, S.; Li, Q.; Yousafzai, A. Blockchain and Federated Learning-Based Intrusion Detection Approaches for Edge-Enabled Industrial IoT Networks: A Survey. *Ad Hoc Netw.* **2024**, *152*, 103320. [CrossRef]
60. Moudoud, H.; Cherkaoui, S.; Khoukhi, L. An Overview of Blockchain and 5G Networks. In *Computational Intelligence in Recent Communication Networks*; Springer: Cham, Switzerland, 2021; pp. 1–20. [CrossRef]
61. Su, X.; Ullah, I.; Wang, M.; Choi, C. Blockchain-based system and methods for sensitive data transactions. *IEEE Consum. Electron. Mag.* **2021**, *13*, 87–96. [CrossRef]
62. Abrol, A. Blockchain Interoperability–Understanding Cross-Chain Technology. *Blockchain Council*, 1 March 2022, pp. 1–5.
63. Faridi, A.R.; Hafeez, A.; Masood, F. Federated Learning with Blockchain: A Study of the Latest Decentralized Couple. In Proceedings of the 2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 4–5 November 2022; pp. 91–96. [CrossRef]
64. Salah, K.; Rehman, M.H.U.; Nizamuddin, N.; Al-Fuqaha, A. Blockchain for AI: Review and Open Research Challenges. *IEEE Access* **2019**, *7*, 10127–10149. [CrossRef]
65. Alharbi, S.; Attiah, A.; Alghazzawi, D. Integrating Blockchain with Artificial Intelligence to Secure IoT Networks: Future Trends. *Sustainability* **2022**, *14*, 16002. [CrossRef]
66. Algarni, S.; Eassa, F.; Almarhabi, K.; Almalaise, A.; Albassam, E.; Alsubhi, K.; Yamin, M. Blockchain-Based Secured Access Control in An IoT System. *Appl. Sci.* **2021**, *11*, 1772. [CrossRef]
67. Singh, S.K.; Rathore, S.; Park, J.H. Blockiotintelligence: A Blockchain-Enabled Intelligent IoT Architecture with Artificial Intelligence. *Future Gener. Comput. Syst.* **2020**, *110*, 721–743. [CrossRef]
68. Casino, F.; Dasaklis, T.K.; Patsakis, C. A Systematic Literature Review of Blockchain-Based Applications: Current Status, Classification and Open Issues. *Telemat. Inform.* **2019**, *36*, 55–81. [CrossRef]
69. Ciulei, A.T.; Creţu, M.C.; Simion, E. Preparation for Post-Quantum Era: A Survey About Blockchain Schemes From A Post-Quantum Perspective. Cryptology ePrint Archive, Paper 2022/026, 2022. Available online: https://eprint.iacr.org/2022/026 (accessed on 1 February 2024).
70. Alahmari, M. Barriers of Adopting Quantum Technology in Blockchain: A Prioritization-Based Framework. *Soft Comput.* **2023**, 1–15. [CrossRef]
71. Fernández-Caramès, T.M.; Fraga-Lamas, P. Towards Post-Quantum Blockchain: A Review on Blockchain Cryptography Resistant to Quantum Computing Attacks. *IEEE Access* **2020**, *8*, 21091–21116. [CrossRef]
72. Gurung, D.; Pokhrel, S.R.; Li, G. Performance Analysis and Evaluation of Post Quantum Secure Blockchained Federated Learning. *arXiv* **2023**, arXiv:2306.14772. [CrossRef].
73. Alghamdi, S.; Almuhammadi, S. The Future of Cryptocurrency Blockchains in the Quantum Era. In Proceedings of the 2021 IEEE International Conference on Blockchain (Blockchain), Melbourne, Australia, 6–8 December 2021; pp. 544–551. [CrossRef]
74. Alotaibi, B. Utilizing Blockchain to Overcome Cyber Security Concerns in the Internet of Things: A Review. *IEEE Sens. J.* **2019**, *19*, 10953–10971. [CrossRef]
75. Kumar, N.M.; Mallick, P.K. Blockchain Technology for Security Issues and Challenges in IoT. *Procedia Comput. Sci.* **2018**, *132*, 1815–1823. [CrossRef]
76. Hakak, S.; Khan, W.Z.; Gilkar, G.A.; Imran, M.; Guizani, N. Securing Smart Cities Through Blockchain Technology: Architecture, Requirements, and Challenges. *IEEE Netw.* **2020**, *34*, 8–14. [CrossRef]
77. Sengupta, J.; Ruj, S.; Bit, S.D. A Comprehensive Survey on Attacks, Security Issues and Blockchain Solutions for IoT and IIoT. *J. Netw. Comput. Appl.* **2020**, *149*, 102481. [CrossRef]
78. Liu, Y.; Yu, F.R.; Li, X.; Ji, H.; Leung, V.C. Blockchain and Machine Learning for Communications and Networking Systems. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1392–1431. [CrossRef]
79. Zafar, S.; Bhatti, K.; Shabbir, M.; Hashmat, F.; Akbar, A.H. Integration of Blockchain and Internet of Things: Challenges and Solutions. *Ann. Telecommun.* **2022**, *77*, 13–32. [CrossRef]
80. Issa, W.; Moustafa, N.; Turnbull, B.; Sohrabi, N.; Tari, Z. Blockchain-Based Federated Learning for Securing Internet of Things: A Comprehensive Survey. *ACM Comput. Surv.* **2023**, *55*, 1–43. [CrossRef]
81. Torres-Carrión, P.V.; González-González, C.S.; Aciar, S.; Rodríguez-Morales, G. Methodology for Systematic Literature Review Applied to Engineering and Education. In Proceedings of the 2018 IEEE Global Engineering Education Conference (EDUCON), Santa Cruz de Tenerife, Spain, 17–20 April 2018; pp. 1364–1373. [CrossRef]
82. Moher, D.; Liberati, A.; Tetzlaff, J.; Altman, D.G.; Prisma Group. Preferred Reporting Items for Systematic Reviews and Meta-analyses: The PRISMA Statement. *Ann. Intern. Med.* **2009**, *151*, 264–269. [CrossRef] [PubMed]

83. Nwakanma, C.I.; Ahakonye, L.A.C.; Njoku, J.N.; Odirichukwu, J.C.; Okolie, S.A.; Uzondu, C.; Ndubuisi Nweke, C.C.; Kim, D.S. Explainable Artificial Intelligence (XAI) for Intrusion Detection and Mitigation in Intelligent Connected Vehicles: A Review. *Appl. Sci.* **2023**, *13*, 1252. [CrossRef]
84. Misra, S. A Step by Step Guide for Choosing Project Topics and Writing Research Papers in ICT Related Disciplines. In Proceedings of the International Conference on Information and Communication Technology and Applications, Minna, Nigeria, 24–27 November 2020; pp. 727–744. [CrossRef]
85. Poorazad, S.K.; Benzaıd, C.; Taleb, T. Blockchain and Deep Learning-Based IDS for Securing SDN-Enabled Industrial IoT Environments. *arXiv* **2023**, arXiv.2401.00468. [CrossRef].
86. Alsharif, N.A.; Mishra, S.; Alshehri, M. IDS in IoT using Machine Learning and Blockchain. *Eng. Technol. Appl. Sci. Res.* **2023**, *13*, 11197–11203. [CrossRef]
87. Kumar, P.; Kumar, R.; Gupta, G.P.; Tripathi, R.; Jolfaei, A.; Islam, A.N. A Blockchain-Orchestrated Deep Learning Approach for Secure Data Transmission in IoT-Enabled Healthcare System. *J. Parallel Distrib. Comput.* **2023**, *172*, 69–83. [CrossRef]
88. Mishra, S. Blockchain and Machine Learning-Based Hybrid IDS to Protect Smart Networks and Preserve Privacy. *Electronics* **2023**, *12*, 3524. [CrossRef]
89. Selvarajan, S.; Srivastava, G.; Khadidos, A.O.; Khadidos, A.O.; Baza, M.; Alshehri, A.; Lin, J.C.W. An Artificial Intelligence Lightweight Blockchain Security Model for Security and Privacy in IIoT Systems. *J. Cloud Comput.* **2023**, *12*, 38. [CrossRef]
90. Mbonu, W.E.; Maple, C.; Epiphaniou, G. An End-Process Blockchain-Based Secure Aggregation Mechanism Using Federated Machine Learning. *Electronics* **2023**, *12*, 4543. [CrossRef]
91. Kalapaaking, A.P.; Khalil, I.; Rahman, M.S.; Atiquzzaman, M.; Yi, X.; Almashor, M. Blockchain-Based Federated Learning With Secure Aggregation in Trusted Execution Environment for Internet-of-Things. *IEEE Trans. Ind. Inform.* **2023**, *19*, 1703–1714. [CrossRef]
92. Kalapaaking, A.P.; Khalil, I.; Atiquzzaman, M. Blockchain-Enabled and Multisignature-Powered Verifiable Model for Securing Federated Learning Systems. *IEEE Internet Things J.* **2023**, *10*, 21410–21420. [CrossRef]
93. Abdel-Basset, M.; Moustafa, N.; Hawash, H. Privacy-Preserved Cyberattack Detection in Industrial Edge of Things (IEoT): A Blockchain-Orchestrated Federated Learning Approach. *IEEE Trans. Ind. Inform.* **2022**, *18*, 7920–7934. [CrossRef]
94. Derhab, A.; Guerroumi, M.; Gumaei, A.; Maglaras, L.; Ferrag, M.A.; Mukherjee, M.; Khan, F.A. Blockchain and Random Subspace Learning-Based IDS for SDN-Enabled Industrial IoT Security. *Sensors* **2019**, *19*, 3119. [CrossRef]
95. Babu, E.S.; SrinivasaRao, B.; Nayak, S.R.; Verma, A.; Alqahtani, F.; Tolba, A.; Mukherjee, A. Blockchain-Based Intrusion Detection System of IoT urban data with Device Authentication Against DDoS Attacks. *Comput. Electr. Eng.* **2022**, *103*, 108287. [CrossRef]
96. Buttar, A.M.; Shahid, M.A.; Arshad, M.N.; Akbar, M.A. Decentralized Identity Management Using Blockchain Technology: Challenges and Solutions. In *Blockchain Transformations: Navigating the Decentralized Protocols Era*; Springer: Cham, Switzerland, 2024; pp. 131–166. [CrossRef]
97. Shinde, N.K.; Seth, A.; Kadam, P. Exploring the Synergies: A Comprehensive Survey of Blockchain Integration with Artificial Intelligence, Machine Learning, and IoT for Diverse Applications. In *Machine Learning and Optimization for Engineering Design*; Springer: Singapore, 2023; pp. 85–119. [CrossRef]
98. Sunanda, N.; Shailaja, K.; Kandukuri, P.; Krishnamoorthy; Rao, V.S.; Godla, S.R. Enhancing IoT Network Security: ML and Blockchain for Intrusion Detection. *Int. J. Adv. Comput. Sci. Appl.* **2024**, *15*, 947–958. [CrossRef]
99. Jagatheesaperumal, S.K.; Pham, Q.V.; Ruby, R.; Yang, Z.; Xu, C.; Zhang, Z. Explainable AI Over the Internet of Things (IoT): Overview, State-of-the-Art and Future Directions. *IEEE Open J. Commun. Soc.* **2022**, *3*, 2106–2136. [CrossRef]
100. Ahakonye, L.A.C.; Nwakanma, C.I.; Lee, J.M.; Kim, D.S. Machine Learning Explainability for Intrusion Detection in the Industrial Internet of Things. *IEEE Internet Things Mag.* **2024**, *7*, 68–74. [CrossRef]
101. Bosri, R.; Rahman, M.S.; Bhuiyan, M.Z.A.; Al Omar, A. Integrating blockchain with artificial intelligence for privacy-preserving recommender systems. *IEEE Trans. Netw. Sci. Eng.* **2020**, *8*, 1009–1018. [CrossRef]
102. Košt'ál, K.; Helebrandt, P.; Belluš, M.; Ries, M.; Kotuliak, I. Management and Monitoring of IoT Devices using Blockchain. *Sensors* **2019**, *19*, 856. [CrossRef] [PubMed]
103. Venkatraman, S.; Parvin, S. Developing an IoT Identity Management System using Blockchain. *Systems* **2022**, *10*, 39. [CrossRef]
104. Khalid, U.; Asim, M.; Baker, T.; Hung, P.C.; Tariq, M.A.; Rafferty, L. A Decentralized Lightweight Blockchain-Based Authentication Mechanism for IoT Systems. *Clust. Comput.* **2020**, *23*, 2067–2087. [CrossRef]
105. Abbassi, Y.; Benlahmer, H. IoT and Blockchain Combined: For Decentralized Security. *Procedia Comput. Sci.* **2021**, *191*, 337–342. [CrossRef]
106. Ozyilmaz, K.R.; Yurdakul, A. Designing a Blockchain-Based IoT with Ethereum, Swarm, and LoRa: The Software Solution to Create High Availability with Minimal Security Risks. *IEEE Consum. Electron. Mag.* **2019**, *8*, 28–34. [CrossRef]
107. Shen, M.; Tang, X.; Zhu, L.; Du, X.; Guizani, M. Privacy-Preserving Support Vector Machine Training Over Blockchain-Based Encrypted IoT Data in Smart Cities. *IEEE Internet Things J.* **2019**, *6*, 7702–7712. [CrossRef]
108. Trautwein, D.; Raman, A.; Tyson, G.; Castro, I.; Scott, W.; Schubotz, M.; Gipp, B.; Psaras, Y. Design and evaluation of IPFS: A storage layer for the decentralized web. In Proceedings of the ACM SIGCOMM 2022 Conference, New York, NY, USA, 22–26 August 2022; pp. 739–752. [CrossRef]
109. Nair, A.K.; Raj, E.D.; Sahoo, J. A Robust Analysis of Adversarial Attacks on Federated Learning Environments. *Comput. Stand. Interfaces* **2023**, *86*, 103723. [CrossRef]

110. Romashkova, I.; Komarov, M.; Ometov, A. Demystifying Blockchain Technology for Resource-Constrained IoT Devices: Parameters, Challenges and Future Perspective. *IEEE Access* **2021**, *9*, 129264–129277. [CrossRef]
111. Zubaydi, H.D.; Varga, P.; Molnár, S. Leveraging Blockchain Technology for Ensuring Security and Privacy Aspects in Internet of Things: A Systematic Literature Review. *Sensors* **2023**, *23*, 788. [CrossRef] [PubMed]

sensors

MDPI

Article

End-to-End Encrypted Message Distribution System for the Internet of Things Based on Conditional Proxy Re-Encryption

Shi Lin [1], Li Cui [2] and Niu Ke [1,*]

[1] School of Cryptographic Engineering, Engineering University of PAP, Xi'an 710000, China; slshilin@126.com
[2] School of Information and Communication, National University of Defense Technology, Wuhan 430000, China; lc_licui17@nudt.edu.cn
* Correspondence: niuke@163.com

Abstract: In light of the existing security vulnerabilities within IoT publish–subscribe systems, our study introduces an improved end-to-end encryption approach using conditional proxy re-encryption. This method not only overcomes limitations associated with the reliance on a trusted authority and the challenge of reliably revoking users in previous proxy re-encryption frameworks, but also strengthens data privacy against potential collusion between the broker and subscribers. Through our innovative encryption protocol, unauthorized re-encryption by brokers is effectively prevented, enhancing secure communication between publisher and subscriber. Implemented on HiveMQ, an open-source MQTT platform, our prototype system demonstrates significant enhancements. Comparison to the state-of-the-art end-to-end encryption work, encryption overhead of our scheme is comparable to it, and the decryption cost is approximately half of it. Moreover, our solution significantly improves overall security without compromising the asynchronous communication and decentralized authorization foundational to the publish–subscribe model.

Keywords: Internet of Things; end-to-end encryption; conditional proxy re-encryption; message broker; HiveMQ

Citation: Lin, S.; Cui, L.; Ke, N. End-to-End Encrypted Message Distribution System for the Internet of Things Based on Conditional Proxy Re-Encryption. *Sensors* **2024**, *24*, 438. https://doi.org/10.3390/s24020438

Academic Editor: Jian Li

Received: 29 November 2023
Revised: 1 January 2024
Accepted: 2 January 2024
Published: 10 January 2024

1. Introduction

To realize data communication among a large number of entities, large-scale Internet of Things (IoT) system generally uses the publish–subscribe (pub/sub) paradigm for data distribution. The most commonly used protocols which work in the pub/sub paradigm are Message Queuing Telemetry Transport (MQTT) [1], Advanced Message Queuing Protocol (AMQP) [2], etc. The subscribers can subscribe to a "message topic", and publisher can publish messages to the "message topic". All subscribers will receive the publisher's message through the routing of the message broker between the publisher and subscribers. The pub/sub paradigm decouples the senders and receivers in time and space. They do not need to be directly connected or online simultaneously. It is more flexible, efficient, and scalable than the point-to-point data exchange mode. A typical pub/sub-based IoT system includes three types of components: IoT devices, a message broker, and a user management application. The device and the management application serve as the publisher and subscriber, respectively. The device publishes the data collected by its sensor to a specific topic, and the authorized user of the device subscribes to that topic through the application. The message broker in the middle routes the data to all the authorized users.

At present, Transport Layer Security (TLS) [3,4] is widely used in the industry to protect the data between the client (publisher or subscriber) and the message broker. The broker decrypts the ciphertext from the publisher to obtain the plaintext, encrypts the plaintext again with the key negotiated with each subscriber, and then forwards the corresponding ciphertext to the subscriber. However, the broker can obtain all the data generated by the client and has complete control over the user's data [5]. Furthermore, the message broker maintained by IoT manufacturers is not completely trusted. Recent research

shows that no-security policies are implemented by a large number of accessible message brokers, which allows anyone to receive data or inject messages [6]. Some researchers investigated a specific traffic monitoring system and discovered that the MQTT message broker was disclosing the traffic flow conditions in a specific area of Mexico City [7]. Even if the message broker deploys a security policy, users need to totally trust the broker. Currently, the message broker in pub/sub-based IoT system are deployed either by IoT device manufacturers based on existing commercial message servers (such as EMQX [8], HiveMQ [9] or Solace [10]), or based on the IoT cloud platform provided by third-party cloud computing manufacturers (such as Alibaba cloud [11], Amazon AWS cloud [12]). The message broker is established and maintained by the IoT device manufacturer or the IoT cloud platform [13]. These manufacturers are not entirely reliable. If the administrator operates incorrectly or is bribed by a spy, or if the manufacturer is for profit purposes, user data are likely to be abused or shared with unauthorized entities, which poses a threat to the security of user data.

In addition, at present, most pub/sub-based IoT systems are constructed based on MQTT protocol, which is not designed for hostile environments. Jia Yan et al. [14] found that the MQTT protocol has serious defects. The platform using this protocol can enable adversaries to steal users' private information and forge users' device status.

In order to prevent the threats brought by malicious message brokers, PICADOR [15] uses proxy re-encryption (PRE [16]) technology to provide end-to-end encryption from publishers to subscribers. In PRE, given a re-encryption key to the semi-trusted proxy, the proxy can convert the ciphertext encrypted with the public key of user A into the ciphertext encrypted with the public key of user B without obtaining any plaintext information from the ciphertext. The proxy re-encryption process can be described as: $E(pk_A, m) \xrightarrow{rk_{A\to B}} E(pk_B, m)$]; here, $rk_{A\to B}$ is the re-encryption key from A to B.

In PICADOR, the publisher encrypts its message with its public key. The broker re-encrypts the published message using the re-encryption key of each subscriber and then sends the corresponding ciphertext to each subscriber. The subscriber can decrypt the ciphertext with their private key. PICADOR needs a trusted authority to generate the re-encryption key for each subscriber according to the publisher's private key and each subscriber's public key. When revoking the authorization of a subscriber, the simplest way is that the broker no longer re-encrypts messages for the revoked subscribers. However, the broker is not entirely trusted. If it is compromised and still re-encrypts for the revoked user, then the revoked user can still receive the latest message. Therefore, depending on the broker for user revocation is not completely reliable. If we do not rely on the broker to revoke users, then we can only change the public–private key pair of the publisher once revocation is required and regenerate the re-encryption key for all the remaining authorized users, which would result in frequent changes to the public–private key pair of the publisher. The long-term public–private key of the user is usually used for authentication either, both between users and between users and brokers. If the user's public–private key pair changes frequently, then there will be inconveniences encountered during authentication.

The reason of the problem in PICADOR is that traditional proxy re-encryption allows the proxy to convert all ciphertexts without restriction [17]. As long as the proxy possesses the re-encryption key from A to B ($rk_{A\to B}$), the proxy can convert all ciphertexts encrypted with the public key of A into ciphertexts which could be decrypted using the private key of B. This all-or-nothing feature is not suitable for applications that need fine-grained authorization of decryption capability. Based on this, Weng et al. proposed the concept of Conditional Proxy Re-encryption (CPRE) [18], which allows for the conditional conversion of ciphertext. In CPRE, when generating ciphertext using user A's public key, a condition value is introduced at the same time, and the re-encryption key from A to B are also related to a condition value. Only when the condition value used for generating the ciphertext is equal to the condition value related to the re-encryption key, the proxy can convert the ciphertext encrypted with the public key of A into the ciphertext encrypted with the public key of B. A can prevent the proxy from performing unauthorized re-encryption

by controlling the change of condition value [17]. The process of conditional proxy re-encryption can be described as: $E(pk_A, m, \omega) \overset{rk}{\underset{A^{\omega} \to B^{\omega'}}{\rightarrow}} E(pk_B, m)\left(\omega = \omega'\right)$.

Because CPRE has significant advantages over PRE in fine-grained authorization, this paper introduces CPRE for the first time to realize end-to-end encryption in pub/sub-based IoT system to prevent broker from performing unauthorized re-encryption. We investigate a large number of existing conditional proxy re-encryption schemes. According to the principles of low computing and communication overhead and high security, the conditional proxy re-encryption algorithm proposed by Weng et al. in 2009 [19] is selected in our system.

In our system, the publisher uses its private key, conditional value, and subscriber's public key to generate the conditional re-encryption key for each subscriber, and sends the conditional re-encryption key to the broker. When publishing a message, the publisher encrypts the message with its public key and condition value and sends the ciphertext to the broker. The broker uses the re-encryption key of each subscriber to re-encrypt the message and then sends the re-encrypted ciphertext to the corresponding subscriber. Finally, the subscriber can obtain the plaintext by decrypting the message with their private key. When a subscriber is revoked, the publisher updates the condition value and generates new condition re-encryption keys for each remaining subscriber. Specifically, the main contributions of our system are as follows:

(1) The conditional proxy re-encryption (CPRE) algorithm is introduced to solve the end-to-end encryption problem in the pub/sub-based IoT system. the re-encryption key is associated with a condition value. By changing the condition value, the publisher can ensure that the proxy can not perform unauthorized re-encryption, thereby achieving reliable revocation of subscribers.

(2) By using an open-source MQTT message server, HiveMQ, we implement a prototype end-to-end encryption system for a pub/sub-based IoT system based on CPRE, and further enhance the system's performance through hybrid encryption and hash chain. Moreover, the performance of the system is tested, which shows that our system is not only easy to implement on existing commercial message servers, but also has high performance.

2. Related Works

2.1. End-to-End Encryption in IoT

At present, a large number of scholars have studied the end-to-end security in pub/sub-based systems [20,21]. This section survey the current research status of end-to-end encryption schemes according to the technology used.

The scheme based on a trusted message broker: Jia Yan [14] proposed MOUCON to solve access control issues in MQTT, the MQTT broker in MOUCON is responsible for verifying each client's access to the message. Clients must fully trust the broker and cannot resolve the security threat to user data caused by an untrusted broker.

The scheme based on the trusted key server: Markus et al. [5] propose an end-to-end security scheme for Cyber–Physical Systems (CPS). The scheme relies on trusted key servers to distribute topic keys for publishers and subscribers. The key server stores the global authorization information of the system and the encryption key. If the key server is compromised by the adversary, all the account information, authorization information, and the encryption keys of the system will be exposed.

The scheme based on identity-based encryption (IBE): JEDI [20] realizes the end-to-end encryption between devices and users in IoT, facilitates asynchronous communication, and supports decentralized authorization for the key. The scheme does not require any modifications to the message broker, which is convenient for deployment. However, this method uses the identity-based encryption algorithm with wildcards (WKD-IBE) [21], the algorithm has a high computational complexity, making it unsuitable for resource-constrained IoT devices.

The scheme based on secret sharing: Sana Belguith et al. [22] proposed an efficient and revocable secure publish–subscribe system. The system divided the broker into three parts to handle the functions of topic matching, routing, and message sending separately. As long as the adversary does not simultaneously break all three brokers, the solution remains secure. However, this solution requires the customization of a special message broker, which is inconvenient to deploy.

The scheme based on special hardware: Segarra et al. [23] restrict the broker to run only in a trusted execution environment (TEE) [24], thus ensuring that the broker functions as intended by the deployer. However, the installation and deployment of TEE requires professional management, as well as its maintenance, which results in high costs. In addition, there are security attacks against the current mainstream TEE [25], so TEE still has some security risks.

The scheme based on proxy re-encryption: PICADOR [15] implements end-to-end encryption between publishers and subscribers using proxy re-encryption. As can be inferred from the above analysis, this scheme depends on a trusted authorization center to generate re-encryption keys and relies on the broker to revoke users. However, this approach has the drawback of unreliable revocation.

2.2. Conditional Proxy Re-Encryption Schemes

Mambo and Okamoto [26] first introduced the concept of decryption capability authorization, which has higher performance than the method of decrypting and then encrypting the ciphertext. In 1998, Blaze, Bleuner, and Strauss formally introduced the concept of proxy re-encryption (PRE) [16], and since then, lots of research has been carried out around PRE. PRE allows a semi-trusted agent to transform the decryption capability of a ciphertext without obtaining any valid information about the ciphertext, and is widely used in encrypted email transmission, secure distributed file systems, and encrypted spam filtering, etc. PRE can be categorized according to different criteria. Based on the direction of re-encryption, PRE can be categorized into one-way PRE and two-way PRE. Additionally, based on the number of re-encryptions allowed, it can be categorized into single-hop PRE and multi-hop PRE.

Traditional proxy re-encryption is unable to provide fine-grained authorization of decryption capabilities. In response, Weng et al. introduced the concept of conditional proxy re-encryption (CPRE) [18] and developed the first CPRE scheme. The re-encryption key of the scheme consists of two parts: the re-encryption key and the conditional key. However, the scheme only considered the security of the second ciphertext layer and not the first ciphertext layer. Weng [27] pointed out that the scheme described in the literature [18] is vulnerable to Chosen-Ciphertext Attacks (CCAs). Weng [27] redefined a more stringent security model for CPRE and proposed a new efficient CPRE scheme. Both Shao [28] and Liang [29] proposed CCA-secure identity-based CPRE under the DBDH (Decisional Bilinear Diffie–Hellman Problem) assumption. However, the literature [30] points out that the scheme given by Liang [29] is insecure.

Fang et al. [31] proposed an anonymous CPRE scheme that enables keyword search. Subsequently, Jae Woo Seo et al. [32] proposed a type-based privacy requirement engineering (Type-based PRE) scheme, where "type" refers to a keyword that is equivalent to "condition" in CPRE. Thus, this scheme is essentially similar to the CPRE scheme, which achieves fine-grained authorization of user decryption capabilities. Son et al. [33] proposed a CPRE for big data sharing on cloud platforms by outsourcing the re-encryption key generation and decryption to the servers. Qiu et al. [19] and Liang et al. [34] proposed CCA-secured CPREs, respectively. Ge et al. [35] proposed an identity-based CPRE scheme. The authors proposed an identity-based CPRE scheme that enables contingent gate computation on conditionals. Hu Xiong et al. [36] introduced a unidirectional multi-hop identity-based CPRE scheme that facilitates flexible and efficient data authorization in cloud computing environments and demonstrated its security in the standard model. Arinjita et al. [37] presented a conditional proxy re-encryption scheme that does not require pairwise operations.

The scheme is not reliant on the bilinear pair operation for construction and has lower computational overhead. However, in this scheme, if the receiver conspires with the agent, the agent can compute the sender's private key as long as it possesses two conditional re-encryption keys. Therefore, the scheme proposed by Arinjita et al. [37] is not able to resist the conspiracy attack between the agent and the receiver.

The following compares various CPRE (Conditional Proxy Re-encryption) schemes, and the results are presented in Tables 1 and 2. The schemes proposed in the literature [18,29,37] exhibit security issues and are therefore not included in the comparison. Let $|G|$ and $|GT|$ denote the bit lengths of elements in groups G and GT, respectively. $|Z_p|$ represents the bit length of elements in the prime field $|Z_p|$. $|m|$ stands for the bit length of the plaintext. t_p and t_e represent the time required for a single bilinear pairing operation and a single exponentiation operation, respectively. $|\sigma|$ represents the bit length of the signature output by a strongly unforgeable one-time signature algorithm. svk is the length of the verification key for the strongly unforgeable one-time signature algorithm. t_v is the time taken to verify a strongly unforgeable one-time signature. t represents the size of the access tree, and w denotes the size of attributes.

Table 1. Comparison of the computation overhead of each CPRE algorithm.

Scheme	Re-Encryption Key Generation	Encrypt	Re-Encrypt	Decrypt
[27]	$4t_e$	$3t_e + t_p$	$3t_p$	$2t_e + t_p$
[28]	$5t_e + t_p$	$8t_e + t_p$	$6t_p$	$4t_e + 8t_p$
[31]	$2t_e$	t_e+$5t_p + t_s$	$2t_e + t_p + t_v$	$t_e + t_p$
[32]	t_e	$5t_e + t_p$	$5t_e$+$2t_p$	$2t_e + 4t_p$
[33]	$4t_e$	$3t_e$+$2t_p$	t_e+$2t_p$	$3t_e$+$2t_p$
[19]	$3t_e$	$4t_e + t_p$	$4t_p$	$2t_e + t_p$
[34]	$9t_e + t_p$	$5t_e + t_p$	$6t_e$+$7t_p$	$5t_e$+$12t_p$
[35]	$8t_e + t_p + t_s$	$6t_e + t_p + t_s$	$(9 + 2w + t)t_p + t_v$	$t_e + 8t_p + t_v$
[36]	$6t_e$	$6t_e + 2t_p$	$t_e + 4t_p + t_v$	$t_e + 3t_p + t_v$

Table 2. Comparison of the communication overhead of each CPRE algorithm.

Scheme	Initial Ciphertext	Re-Encryption Key	Ciphertext after Re-Encryption
[27]	$2\|G\| + \|G_T\| + \|m\|$	$2\|G\|$	$2\|G\| + \|G_T\| + \|m\|$
[28]	$4\|G\| + \|G_T\| + \|Z_p\|$	$2\|G\| + 2\|G_T\|$ +2 $\|Z_p\|$	$4\|G\| + \|G_T\| + \|m\|$
[31]	$\|G\| + 3\|G_T\| + svk + \|\sigma\|$	$2\|G\| + 2\|Z_p\|$	$2\|G_T\| + svk + \|m\|$
[32]	$2\|G\| + \|G_T\| + svk + \|\sigma\|$	$\|G\| + \|Z_p\|$	$4\|G\| + \|G_T\| + svk + \|\sigma\|$
[33]	$\|G\| + 2\|G_T\|$	$2\|G\|$	$2\|G\| + 2\|G_T\|$
[19]	$2\|G\| + \|m\|$	$2\|G\|$	$\|G_T\| + \|m\|$
[34]	$4\|G\| + \|G_T\| + svk + \|\sigma\|$	$6\|G\| + \|G_T\| + svk + \|\sigma\|$	$3\|G\| + \|G_T\| + svk + \|\sigma\| + \|m\|$
[35]	$(w + 4)\|G\| + \|G_T\| + w + \|\sigma\| + \|m\|$	$6\|G\| + \|G_T\| + \|\sigma\| + t$	$(w + 6)\|G\| + 2\|G_T\| + 2\|\sigma\| + 2\|m\|$
[36]	$3\|G\| + 2\|G_T\| + svk + \|\sigma\|$	$2\|G\| + \|Z_p\|$	$3\|G\| + 2\|G_T\| + svk + \|\sigma\|$

In the practical implementation of the CPRE algorithm, the re-encryption algorithm is executed by a semi-trusted agent, which is generally deployed on servers or clouds with more abundant resources. In contrast, the encryption and decryption operations are typically performed by IoT devices or personal handheld electronic devices with significantly smaller computational resources than the agent. Therefore, the overall principle in selecting CPRE algorithms is to choose the scheme with lower computational and communication overhead. When computational and communication overheads are comparable, the scheme with lower encryption and decryption overheads is chosen. In terms of security, the schemes in the table are all provably CCA secure in either the standard model or the stochastic prediction model. Although the schemes proven to be secure under the standard model are theoretically more reliable, in this paper, we choose the schemes proven to be secure under the stochastic prediction model. This is because the security of such security schemes depends only on the hash function itself, and no practical attack has yet occurred

that can compromise a practical cryptographic algorithm proven to be secure under the stochastic prediction model (excluding some carefully constructed counterexamples by humans [38]). In addition, these security schemes are computationally more efficient and have a wider range of applications.

Based on the aforementioned principles, this paper utilizes the conditional proxy re-encryption algorithm proposed by Weng et al. [27] in 2009 to achieve the encryption of the publish–subscribe system of this paper from the publisher side to the subscriber side.

3. Preliminaries

The conditional proxy re-encryption CPRE scheme includes the following algorithms, and its workflow is given in Figure 1:

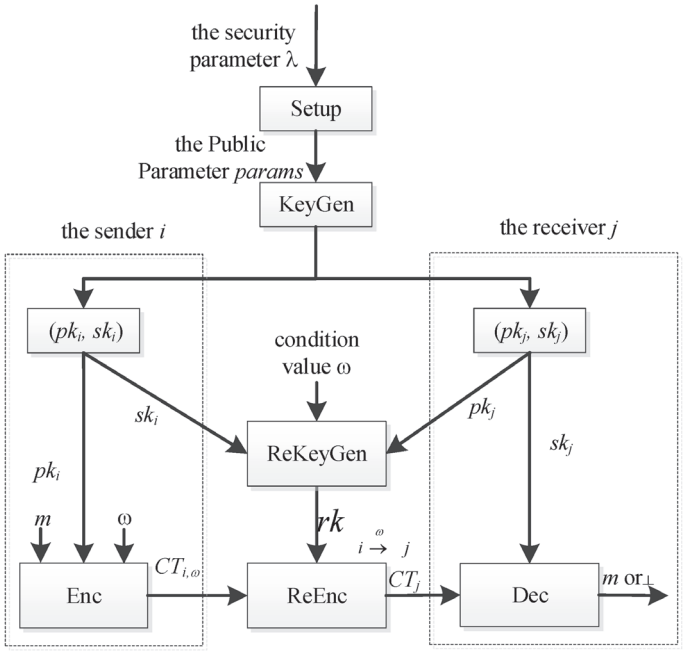


Figure 1. Workflow of CPRE.

$Setup\left(\lambda^k\right)$: Input the security parameter λ^k, and the algorithm outputs the public parameter *params*.

$KeyGen\left(\lambda^k\right)$: Each entity uses this randomized key generation algorithm to generate a public–private key pair (pk_i, sk_i).

$ReKeyGen(sk_i, \omega, pk_j)$: Input the private key sk_i of the sender, the condition value ω and the public key pk_j of the receiver, the re-encryption key generation algorithm outputs the re-encryption key $rk_{i\xrightarrow{\omega}j}$ from sender i to receiver j.

$Enc_1(pk_i, m)$: Input the public key pk_i of the sender and plaintext m, the first layer encryption algorithm outputs the first layer ciphertext CT_i. The ciphertext cannot be re-encrypted.

$Enc_2(pk_i, m, \omega)$: Input the public key pk_i of the sender, the plaintext m and the condition value ω, the second-layer encryption algorithm will output the second-layer ciphertext $CT_{i,\omega}$. This ciphertext can be re-encrypted using an appropriate re-encryption key into a first-layer ciphertext for different recipients.

$ReEnc\left(CT_{i,\omega}, rk_{i\xrightarrow{\omega}j}\right)$: Input the second-layer ciphertext $CT_{i,\omega}$, the re-encryption key $rk_{i\xrightarrow{\omega}j}$, the proxy runs the re-encryption algorithm to output the first-layer ciphertext CT_j .

$Dec_1(CT_j, sk_j)$: Input the first-layer ciphertext CT_j and private key sk_j, the first-layer decryption algorithm outputs plaintext m or error symbol $\perp$.

$Dec_2(CT_{i,\omega}, sk_i)$: Input the second-layer ciphertext $CT_{i,\omega}$ and private key sk_i, the second-layer decryption algorithm outputs the plaintext message m or error symbol $\perp$.

By introducing a conditional value into the generation of the re-encryption key and the second layer encryption algorithm, the conditional proxy re-encryption algorithm ensures that the proxy cannot perform unauthorized re-encryption.

4. End-to-End Encryption System Based on CPRE

4.1. System Framework

Our CPRE-based end-to-end encryption system consists of three types of entities: IoT devices (referred to as the sender), message broker, and multiple authorized users (referred to as the receiver).

Our system utilizes the conditional proxy re-encryption algorithm proposed by Weng et al. [19] to achieve end-to-end encryption from the publisher to the subscribers in pub/sub-based IoT system. The specific algorithm design can be found in the literature [19]. In our system, the device owner generates a re-encryption key for each authorized user and send the re-encryption key to the Broker. The device acts as the sender and encrypts the message with its public key and a condition value. The broker re-encrypts for each subscriber using the corresponding re-encryption key. All authorized users can decrypt the ciphertext using their private key. The framework of our system is shown in Figure 2.

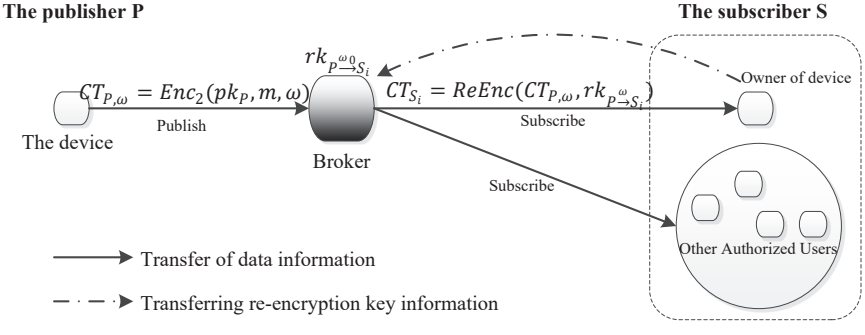


Figure 2. The framework of IoT end-to-end encryption system.

CPRE differs from PRE in that its re-encryption key $rk_{P\xrightarrow{\omega}S_i}$ and the second-layer ciphertext $CT_{P,\omega}$ are both associated with a condition value ω. If the authorization for certain users need to be revoked, the device owner just need to generate a new condition value ω' and send to the device. Then, the device owner generates a new re-encryption key $rk_{P\xrightarrow{\omega'}S_i}$ for each remaining recipient with the device's private key sk_P, the new condition value ω', and the public keys of the remaining authorized users pk_{S_i}. The owner no longer generates re-encryption keys for the revoked users. The device uses the new condition value ω' to generate the second layer ciphertext $CT_{P,\omega'}$. The broker re-encrypts with the new re-encryption key $CT_{P,\omega'}$ for the remaining authorized users, so that the remaining legitimate users can decrypt the message correctly. Since the re-encryption key of the revoked users is not updated, it is still associated with the previous condition value ω, and the new ciphertext corresponds to the updated condition value ω'. Even if the broker is compromised and still re-encrypts the message for the revoked users—as the condition value in the re-encryption key and second layer ciphertext are not equal—the revoked user cannot decrypt the re-encrypted ciphertext correctly with their private key.

4.2. System Workflow

Specifically, the workflow of our CPRE-based end-to-end encryption scheme includes the following steps.

4.2.1. User Registration

When a user wants to utilize our system for device management or monitoring, he must complete the user registration process through the client application. The algorithm $KeyGen\left(1^k\right)$ of CPRE is integrated into the application, allowing the user to generate their public and private key pairs $\left(pk_{S_i}, sk_{S_i}\right)$. The user keeps their private key sk_{S_i} secret.

4.2.2. Device Registration

When a user purchases a new IoT device, the user is the owner of the device and is responsible for device registration and authorization control. Typically, a newly purchased IoT device starts its life cycle with "device discovery" [3]. In this stage, the device owner requests to add a device through the client APP, and the APP establishes a local connection with the device to complete the device registration and the binding of the device and its owner. We assumes that during the "device registration" phase, the device interacts with the client APP of the device owner through a local connection, exchanging basic information and performing mutual authentication. During the device registration process, the device owner utilizes the built-in key generation algorithm $KeyGen(1^k)$ of the APP to generate public and private key pairs (pk_P, sk_P) for the device. Additionally, the owner generates a random initial condition value ω_0, and transfers the initial condition value ω_0 and device's public–private key pair (pk_P, sk_P) to the device through the "Local Connection" established in registration phase. The device owner also keeps the private key sk_P of the device secretly, which is used to generate the conditional re-encryption key for other authorized users.

4.2.3. Authorization Phase

When the device owner wants to authorize the access rights of the device to other users, the device owner uses the private key sk_p of the device, the condition value ω_0, and the public key pk_{S_i} of each authorized user to generate a conditional re-encryption key $rk_{P \xrightarrow{\omega_0} S_i}$ for each authorized user, which will be sent to the broker of the publish–subscribe system.

4.2.4. Message Transmission Stage

The device runs the CPRE encryption algorithm, encrypts the collected information with its public key pk_P and condition value ω_0 and obtain the ciphertext $CT_{P,\omega_0} = Enc_2\left(pk_p, m, \omega_0\right)$, which will be sent to the broker. Then, the broker re-encrypts the ciphertext according to the conditional re-encryption key $rk_{P \xrightarrow{\omega_0} S_i}$ of each authorized user and sends the re-encrypted ciphertext $CT_{S_i} = ReEnc\left(CT_{P,\omega_0}, rk_{P \xrightarrow{\omega_0} S_i}\right)$ to the corresponding authorized user. Finally, each authorized user uses their private key sk_{S_i} to decrypt the ciphertext CT_{S_i} and obtains the plaintext $m = Dec_1\left(CT_{S_i}, sk_{S_i}\right)$.

4.2.5. Revocation Phase

When the device owner needs to revoke the access permission of one user, first, the device owner randomly selects a new condition value ω_1; then, the owner generates a new conditional re-encryption key for each remaining authorized user with the user's public key pk_{S_j}, the device's private key of sk_P, and the new condition value ω_1. Furthermore, the updated re-encryption keys are sent to the broker. Finally, the device owner encrypts the new condition value with the public key of the device and sends the ciphertext to the device.

The device decrypts with its private key sk_P to obtain the new condition value ω_1, it updates the condition value to the new one. Similarly, when the broker receives the new conditional re-encryption key $rk_{P \xrightarrow{\omega_1} S_j}$ distributed by the device owner, the conditional

re-encryption key will also be updated from $rk_{P \xrightarrow{\omega_0} S_i}$ to $rk_{P \xrightarrow{\omega_1} S_j}$. Then, the device encrypts the message with the new condition value, and the broker re-encrypts with the new conditional re-encryption key.

4.3. System Optimization

4.3.1. Hybrid Encryption

Most IoT devices are low-power devices with limited resources, so we use hybrid encryption to further reduce device-side overhead. Before encrypting the collected messages, the device first selects a random symmetric key k, encrypts the key with CPRE and sends to the broker. Each authorized subscriber can decrypt the re-encrypted ciphertext with their private key, thereby obtaining the same symmetric key k. Since then, subsequent communications between the device and each subscriber can be encrypted using the symmetric key k.

4.3.2. Hash Chain

Whenever the set of authorized users changes (such as new users join or old users are revoked), if CPRE is used to generate new conditional re-encryption keys for all remaining authorized users, when the authorized user set changes frequently, generating and transmitting conditional re-encryption keys imposes significant overhead on the device owner. Therefore, when a new user joins, a symmetric key is distributed to the new user by means of the hash chain [39]. The CPRE algorithm is used only when the user is revoked.

Specifically, the process of distributing a symmetric key to a new user by using a hash chain is as follows: Assume that the symmetric key shared between the device and each subscribing user is k before the new user joins. Before a new user joins, the device owner uses k as the input of the one-way trapdoor function to obtain a new session key $k_1 = Hash(k)$, and sends the new session key to the new joined user. At the same time, the device owner broadcasts the key update command, so that the device and other legitimate users can also update their shared key k to k_1 through the one-way trapdoor function. Then, the consistency of the shared session key between the device and all its authorized users can be ensured.

4.4. System Analysis

Below, we analyze our CPRE-based end-to-end encryption system in IoT.

4.4.1. Satisfy Confidentiality

When a new user is authorized to join, the session key is updated through the hash chain. Based on the unidirectionality of the hash function, the new user cannot deduce the session key k before joining by using the session key k_1.

When the user is revoked, the system uses CPRE to update the session key. The device owner randomly selects a new condition value, and generates new conditional re-encryption keys for the remaining users, but no longer generates new conditional re-encryption keys for revoked users. Therefore, when the device uses CPRE encryption to transmit a randomly selected new session key, the new condition value is used, and the broker also uses the new conditional re-encryption key for re-encryption, so that the remaining legitimate users can use their own private key to decrypt and obtain the new session key. The conditional re-encryption key of the revoked user is also related to the old condition value, even if the broker colludes with the revoked user, still re-encrypts with the old conditional re-encryption key. As the condition value in the cipertext of the device is not equal to the condition value in the revoked user's re-encryption key, so the revoked user cannot decrypt or obtain the new session key. Based on the above analysis, the revoked user cannot obtain a new session key even if he colludes with the broker. In addition, our scheme is constructed based on the CPRE algorithm, and the broker can only use the re-encryption key to convert the ciphertext, and cannot obtain any plaintext message of the ciphertext based on the security proof of CPRE in [19].

4.4.2. Support Asynchronous Communication

When a new user joins, the session key is updated through the hash chain, and the offline user does not affect the session key update process between the online user and the device. When the user is revoked, the update of the conditional re-encryption key involves the device owner. As long as the device owner is online, the broker can obtain the latest conditional re-encryption key, and the device can obtain the latest conditional value, offline users do not affect the process. In short, offline users do not affect the key update process of online users, and our scheme supports asynchronous communication.

4.4.3. Support Decentralized Authorization

In our scheme, the authorization and revocation of device access rights are only controlled by its owner and do not rely on trusted third parties. If a device owner is compromised by an adversary, only the security of the owner and its managed devices will be affected, and the security of other devices and users will not be affected.

4.4.4. Support the Decoupling of Publishers and Subscribers

Our scheme is designed based on the CPRE scheme. The device uses its public key to encrypt the data. The device does not need to encrypt for each authorized user, and does not need to care about which users subscribed to its data. The broker completes the conversion and distribution of the ciphertext sent by the device. Therefore, the publisher (ie, device) and subscriber (ie, user) in our scheme are decoupled, and their relationship is controlled and managed by the device owner.

5. Prototype Implementation and Performance Analysis

This section implements the above-mentioned IoT end-to-end encryption system and tests the system's performance.

5.1. Implementation of the Prototype System

We implement each module based on the Java language. Our prototype system includes three types of entities: the publisher, multiple subscribers, and the message broker which can perform re-encryption operations. The implementation of the CPRE algorithm [19] is based on the JPBC (the Java Pairing-Based Cryptography Library) [40]. The establishment of the message broker is based on the open-source MQTT server HiveMQ [9], which supports customized function extensions. The development of the publisher and the subscriber is based on the Eclipse Paho Java Client library [41]. The prototype system runs on a laptop configured with an Intel Core i7-5600U 2.6GHZ CPU and 8G RAM (Intel, Santa Clara, CA, USA). The computer operating system is Windows 7. The development environment of the publisher, subscriber, and Broker is IntelliJ IDEA 2018.1.6. The functions of the publisher and subscriber are simulated using a Java console program.

5.1.1. CPRE Implementation

The CPRE scheme used in our system is constructed based on the symmetric bilinear group. Therefore, we use the configuration file "a.properties" provided by the JPBC library to generate a type A symmetric bilinear group based on a prime-order elliptic curve $y^2 = x^3 + xmodp(p \equiv 3mod4)$, while the base field size is 512 bits, and the embedding degree is 2.

5.1.2. Implementation of Message Broker

The construction of Broker is based on the open-source MQTT server HiveMQ Community Edition [42], which provides the SDK (HiveMQ Extension SDK) [43] that supports extension development. Through this, users can develop custom business logic to extend the functions of HiveMQ with the SDK, such as intercepting or controlling MQTT messages, integrating other services, statistics, and adding fine-grained security solutions. We use the core part of HiveMQ to implement message routing and forwarding, implements message

re-encryption in a customized extension, and re-encrypts the message payload for each subscriber based on the re-encryption key of each subscriber.

HiveMQ includes multiple types of Interceptors, which provide convenience for intercepting and modifying MQTT messages in extensions. Our customized extension is implemented through HiveMQ Interceptors. The goal of our customized extension is to re-encrypt the payload of the message according to the subscriber's re-encryption key after the message is routed, but before the message is forwarded to the subscriber. Therefore, our customized extension is implemented using the **Publish Outbound Interceptor** in HiveMQ, which allows the extension to intercept the PUBLISH message after the broker is routed, and allows different modifications to the payload corresponding to each subscriber.

5.1.3. Implementation of the Client

The implementation of the publisher and the subscriber is relatively easy, which is based on the Eclipse Paho Java Client library and the JPBC library is added to support the encryption and decryption of the published and received messages based on CPRE.

5.2. Performance Analysis

This section tests the performance of each module of our system based on the prototype system built in the previous section.

5.2.1. Overhead of Distributing Session Keys Using CPRE

When the device receives a new condition value, it will randomly generate a new session key. The key will be encrypted with the device's public key and the new condition value, and then ciphertext will be sent to the Broker. The broker re-encrypts the ciphertext for each subscriber and sends the re-encrypted ciphertext to each subscriber. Each subscriber decrypts the re-encrypted ciphertext with its private key, and obtains the session key. When the device uses CPRE to transmit an AES (Advanced Encryption Standard) [44] key (128 bits), the computational overhead of each module (device, broker, and subscriber) in our system is shown in Figure 3.

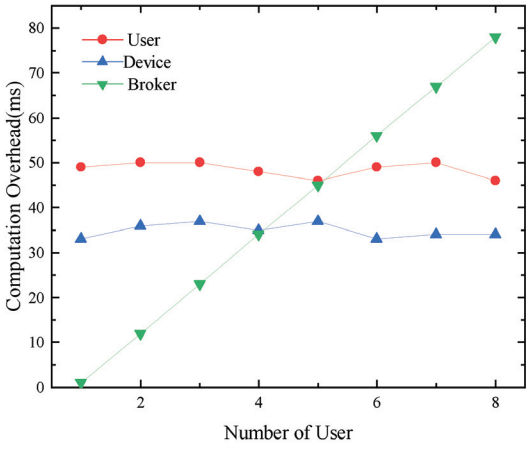


Figure 3. Computation overhead of each module to distribute session key based on CPRE.

The computation overhead of the device is approximately 46 ms. The overhead of the subscriber to is approximately 35 ms. The processing time required by the broker increases linearly with the number of users. As shown in Figure 3, for each additional user, the processing time of the Broker increases by approximately 9 ms. The Broker is generally deployed on a server or cloud platform with rich resources, which can easily cope with

this overhead increase. The computation overhead of the device and users does not change with the number of users, making it suitable for IoT devices with limited resources.

5.2.2. Overhead of Secure Communication

After the device and the subscriber have established a shared symmetric key, the communication between the device and the subscribers are encrypted by the symmetric encryption algorithm AES. We compare the computation overhead of the publisher and subscriber when they use AES for message transmission than when they transmit in plaintext.

For multiple message sizes (128 B, 512 B, 1 KB, 2 KB), Figure 4 shows that encrypted transmission between the publisher and the subscriber incurs an increased overhead of approximately 0.2–0.3 ms per message. Therefore, the use of symmetric keys for secure transmission brings only a slightly increase in computation cost.

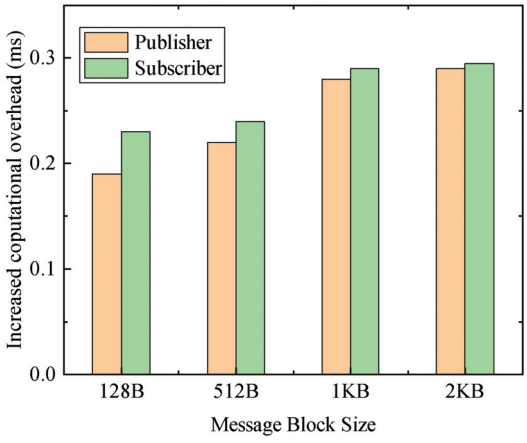


Figure 4. Increased computation overhead for encrypted transfers.

5.2.3. Comparison with Related Schemes

Table 3 compares the current end-to-end encryption schemes in IoT in terms of security, whether it supports decentralized authorization, whether it is convenient to deploy, and performance.

Table 3. Comparison of existing end-to-end encryption scheme.

Scheme	Confidentiality	Decentralization Authorization	Easy Deploy	Performance
[14]	No	No	Yes	-
[5]	Yes	No	Yes	Fast
[22]	Yes	No	No	-
[23]	Yes	No	No	-
[15]	Yes	No	Yes	Comparable to [20]
[20]	Yes	Yes	Yes	Slow (encrypt ≈ 42 ms, decrypt ≈ 62 ms)
Our scheme	Yes	Yes	Yes	Faster than [20] (encrypt ≈ 46 ms, decrypt ≈ 35 ms)

Note: - represents the performance of these schemes was not evaluated on our prototype.

- Security: The message brokers in most IoT systems are not completely trusted. In a scheme that completely relies on the message broker, the broker can obtain all the information of the user, which does not meet the confidentiality requirements.
- Support decentralized authorization: If relying on a third-party trusted key server, the authorization and revocation of device access rights must be completed through the key server, and decentralized authorization is not supported. PICADOR relies on a trusted authority to generate re-encryption keys for all users of the system and does not support decentralized authorization too.
- Easy to deploy: Reference [22] divides the functions of a broker into multiple ones, and new brokers need to be customized to meet the corresponding functions. Reference [39] needs to install special hardware, which is difficult to deploy.
- Performance: The schemes that rely on trusted brokers do not meet the confidentiality requirements, and the literature [22,23] is difficult to deploy. Therefore, these schemes are not re-implemented on our experimental platform. Ref. [5] relies on a trusted key server, the scheme uses symmetric key to establish session key, so [5] distributes symmetric keys faster than our scheme. However, in the secure communication stage, the overhead of using symmetric keys to encrypt and transmit messages is equal to our scheme. Ref. [20] implements the scheme of PICADOR, whose performance is comparable to [20]. In order to have a fair comparison with [20], we re-implement the WKD-IBE algorithm in [20] using our crypto library. In [20], the encryption algorithm takes almost 42 ms to encrypt 128 bits of data, and the decryption algorithm takes about 62 ms to decrypt and obtain the plaintext (the decryption time contains the time to generate a decryption key for the encrypted pattern and time to decrypt the ciphertext. When testing the computation overhead, we use a pattern of 20 attributes representing the URI and the last six attributes representing the time.). The encryption overhead of our scheme is comparable to that presented in [20], while the decryption cost is approximately half that presented in [20].

Above all, some existing schemes rely on the trustworthiness of the broker, and their security assumptions are strong, which cannot meet the confidentiality requirements; some schemes rely on a third-party trusted server for authorization and revocation, and do not support decentralized authorization; some solutions need a customized broker or rely on special hardware, which is inconvenient to deploy. Ref. [20] is a solution with good security and deployability. However, it relies on the more complex WKD-IBE algorithm, and its performance is lower compared to our proposed solution.

6. Conclusions

In the publish–subscribe-based IoT system, the communication between devices and users is not one-to-one direct communication, but one-to-many asynchronous communication and forwarded by the broker located in the middle. Currently, TLS is commonly employed to safeguard the security of data transmission between the device and the broker. However, the broker has the ability to access the plaintext of all messages, rendering this method ineffective in preventing the security and privacy risks posed by untrustworthy brokers to device data. We present and implement a new end-to-end encryption system based on conditional proxy re-encryption. Theoretical analysis and experimental results demonstrate that our proposed scheme is not only theoretically safe, but also highly practical, feasible, and efficient in practice. However, our system cannot efficiently revoke users, which will form the focus of our future research.

Author Contributions: Conceptualization, S.L. and L.C.; methodology, S.L. and N.K.; software, S.L.; validation, S.L., L.C. and N.K.; writing—original draft preparation, L.C.; writing—review and editing, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are included in this article.

Acknowledgments: We thank anonymous reviewers for their useful comments.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

IoT	Internet of Things
CPRE	conditional proxy re-encryption
MQTT	Message Queuing Telemetry Transport
AMQP	Advanced Message Queuing Protocol
TLS	Transport Layer Security
PRE	proxy re-encryption
CPS	Cyber–Physical Systems
WKD-IBE	identity-based encryption with wildcards
JPBC	The Java Pairing Based Cryptography Library
SDK	HiveMQ Extension SDK
AES	Advanced Encryption Standard

References

1. Banks, A.; Briggs, E.; Borgendale, K.; Gupta, R. *MQTT*, Version 5.0; Technical Report, OASIS Standard; OASIS: Burlington, MA, USA, 2019.
2. Godfrey, R.; Ingham, D.S.R. *Advanced Message Queuing Protocol (AMQP)*, Version 1.0; Technical Report, OASIS Standard; OASIS: Burlington, MA, USA, 2012.
3. Zhou, W.; Jia, Y.; Yao, Y.; Zhu, L.; Guan, L.; Mao, Y.; Liu, P.; Zhang, Y. Discovering and Understanding the Security Hazards in the Interactions between IoT Devices, Mobile Apps, and Clouds on Smart Home Platforms. In Proceedings of the 28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, 14–16 August 2019; Heninger, N., Traynor, P., Eds.; USENIX Association: Berkeley, CA, USA, 2019; pp. 1133–1150.
4. Wilson, J.; Wahby, R.S.; Corrigan-Gibbs, H.; Boneh, D.; Levis, P.A.; Winstein, K. Trust but Verify: Auditing the Secure Internet of Things. In Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys'17, Niagara Falls, NY, USA, 19–23 June 2017; Choudhury, T., Ko, S.Y., Campbell, A., Ganesan, D., Eds.; ACM: New York, NY, USA, 2017; pp. 464–474. [CrossRef]
5. Dahlmanns, M.; Pennekamp, J.; Fink, I.B.; Schoolmann, B.; Wehrle, K.; Henze, M. Transparent End-to-End Security for Publish/Subscribe Communication in Cyber-Physical Systems. In Proceedings of the SAT-CPS@CODASPY 2021, Proceedings of the 2021 ACM Workshop on Secure and Trustworthy Cyber-Physical Systems, Virtual Event, 28 April 2021; Gupta, M., Abdelsalam, M., Mittal, S., Eds.; ACM: New York, NY, USA, 2021; pp. 78–87. [CrossRef]
6. Maggi, F.; Vosseler, R.; Quarta, D. *The Fragility of Industrial IoT's Data Backbone: Security and Privacy Issues in MQTT and CoAP Protocols*; Technical Report, Trend Micro Research; Trend Micro Inc.: Tokyo, Japan, 2018.
7. Huq, N.; Vosseler, R.; Swimmer, M. *Cyberattacks against Intelligent Transportation Systems*; Technical Report, Trend Micro Research; 2017.
8. EMQ. *EMQX Broker Docs*; v4.3; Technical report; EMQ: Hong Kong, China, 2021.
9. HiveMQ. *HiveMQ Documentation*; v4.7; Technical Report; HiveMQ: Landshut, Germany, 2021.
10. Solace. *Solace Cloud*; Technical Report; Solace: Ottawa, ON, Canada, 2021.
11. Alibaba. *Alibaba Cloud Iot Platform*; Technical report; Alibaba: Hangzhou, China, 2021.
12. Amazon. *Aws Iot Core*; Technical Report; Amazon: Bellevue, WA, USA, 2021.
13. Henze, M.; Matzutt, R.; Hiller, J.; Mühmer, E.; Ziegeldorf, J.H.; van der Giet, J.; Wehrle, K. Complying With Data Handling Requirements in Cloud Storage Systems. *IEEE Trans. Cloud Comput.* **2022**, *10*, 1661–1674. [CrossRef]
14. Jia, Y.; Xing, L.; Mao, Y.; Zhao, D.; Wang, X.; Zhao, S.; Zhang, Y. Burglars' IoT Paradise: Understanding and Mitigating Security Risks of General Messaging Protocols on IoT Clouds. In Proceedings of the 2020 IEEE Symposium on Security and Privacy, SP 2020, San Francisco, CA, USA, 18–21 May 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 465–481. [CrossRef]
15. Borcea, C.; Gupta, A.D.; Polyakov, Y.; Rohloff, K.; Ryan, G.W. PICADOR: End-to-end encrypted Publish-Subscribe information distribution with proxy re-encryption. *Future Gener. Comput. Syst.* **2017**, *71*, 177–191. [CrossRef]

16. Blaze, M.; Bleumer, G.; Strauss, M. Divertible Protocols and Atomic Proxy Cryptography. In Proceedings of the Advances in Cryptology-EUROCRYPT'98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, 31 May–4 June 1998; Nyberg, K., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1998; Volume 1403, pp. 127–144. [CrossRef]
17. Lee, E. Improved Security Notions for Proxy Re-Encryption to Enforce Access Control. In Proceedings of the Progress in Cryptology-LATINCRYPT 2017-5th International Conference on Cryptology and Information Security in Latin America, Havana, Cuba, 20–22 September 2017; Revised Selected Papers; Lange, T., Dunkelman, O., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2017; Volume 11368, pp. 66–85. [CrossRef]
18. Weng, J.; Deng, R.H.; Ding, X.; Chu, C.; Lai, J. Conditional proxy re-encryption secure against chosen-ciphertext attack. In Proceedings of the 2009 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2009, Sydney, Australia, 10–12 March 2009; Li, W., Susilo, W., Tupakula, U.K., Safavi-Naini, R., Varadharajan, V., Eds.; ACM: New York, NY, USA, 2009; pp. 322–332. [CrossRef]
19. Qiu, J.; Hwang, G.; Lee, H. Efficient Conditional Proxy Re-encryption with Chosen-Ciphertext Security. In Proceedings of the Ninth Asia Joint Conference on Information Security, AsiaJCIS 2014, Wuhan, China, 3–5 September 2014; Computer Society; IEEE: Piscataway, NJ, USA, 2014; pp. 104–110. [CrossRef]
20. Kumar, S.; Hu, Y.; Andersen, M.P.; Popa, R.A.; Culler, D.E. JEDI: Many-to-Many End-to-End Encryption and Key Delegation for IoT. In Proceedings of the 28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, 14–16 August 2019; Heninger, N., Traynor, P., Eds.; USENIX Association: Berkeley, CA, USA, 2019; pp. 1519–1536.
21. Abdalla, M.; Catalano, D.; Dent, A.W.; Malone-Lee, J.; Neven, G.; Smart, N.P. Identity-Based Encryption Gone Wild. In Proceedings of the Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, 10–14 July 2006; Bugliesi, M., Preneel, B., Sassone, V., Wegener, I., Eds.; Proceedings, Part II; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4052, pp. 300–311. [CrossRef]
22. Belguith, S.; Cui, S.; Asghar, M.R.; Russello, G. Secure publish and subscribe systems with efficient revocation. In Proceedings of the 33rd Annual ACM Symposium on Applied Computing, SAC 2018, Pau, France, 9–13 April 2018; Haddad, H.M., Wainwright, R.L., Chbeir, R., Eds.; ACM: New York, NY, USA, 2018; pp. 388–394. [CrossRef]
23. Segarra, C.; Delgado-Gonzalo, R.; Schiavoni, V. MQT-TZ: Secure MQTT Broker for Biomedical Signal Processing on the Edge. In Proceedings of the Digital Personalized Health and Medicine-Proceedings of MIE 2020, Medical Informatics Europe, Geneva, Switzerland, 28 April–1 May 2020; Pape-Haugaard, L.B., Lovis, C., Madsen, I.C., Weber, P., Nielsen, P.H., Scott, P., Eds.; Studies in Health Technology and Informatics; IOS Press: Amsterdam, The Netherlands, 2020; Volume 270, pp. 332–336. [CrossRef]
24. Maene, P.; Götzfried, J.; de Clercq, R.; Müller, T.; Freiling, F.C.; Verbauwhede, I. Hardware-Based Trusted Computing Architectures for Isolation and Attestation. *IEEE Trans. Comput.* **2018**, *67*, 361–374. [CrossRef]
25. Sabt, M.; Achemlal, M.; Bouabdallah, A. Trusted Execution Environment: What It is, and What It is Not. In Proceedings of the 2015 IEEE TrustCom/BigDataSE/ISPA, Helsinki, Finland, 20–22 August 2015; IEEE: Piscataway, NJ, USA, 2015; Volume 1, pp. 57–64. [CrossRef]
26. Mambo, M.; Okamoto, E. Proxy Cryptosystems: Delegation of the Power to Decrypt Ciphertexts (Special Section on Cryptography and Information Security). *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **1997**, *80*, 54–63.
27. Weng, J.; Yang, Y.; Tang, Q.; Deng, R.H.; Bao, F. Efficient Conditional Proxy Re-encryption with Chosen-Ciphertext Security. In Proceedings of the Information Security Conference, Pafos, Cyprus, 18–20 May 2009.
28. Shao, J.; Wei, G.; Ling, Y.; Xie, M. Identity-Based Conditional Proxy Re-Encryption. In Proceedings of the 2011 IEEE International Conference on Communications (ICC), Kyoto, Japan, 5–9 June 2011; pp. 1–5.
29. Liang, K.; Liu, Z.; Tan, X.; Wong, D.S.; Tang, C. A CCA-Secure Identity-Based Conditional Proxy Re-Encryption without Random Oracles. In Proceedings of the International Conference on Information Security and Cryptology, Seoul, Republic of Korea, 28–30 November 2012.
30. He, K.; Weng, J.; Deng, R.H.; Liu, J.K. On the security of two identity-based conditional proxy re-encryption schemes. *Theor. Comput. Sci.* **2016**, *652*, 18–27. [CrossRef]
31. Fang, L.; Susilo, W.; Ge, C.; Wang, J. Chosen-ciphertext secure anonymous conditional proxy re-encryption with keyword search. *Theor. Comput. Sci.* **2012**, *462*, 39–58. [CrossRef]
32. Seo, J.W.; Yum, D.H.; Lee, P.J. Proxy-invisible CCA-secure type-based proxy re-encryption without random oracles. *Theor. Comput. Sci.* **2013**, *491*, 83–93. [CrossRef]
33. Son, J.; Kim, D.; Hussain, R.; Oh, H. Conditional proxy re-encryption for secure big data group sharing in cloud environment. In Proceedings of the 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Toronto, ON, Canada, 27 April–2 May 2014; pp. 541–546.
34. Liang, K.; Susilo, W.; Liu, J.K.; Wong, D.S. Efficient and Fully CCA Secure Conditional Proxy Re-Encryption from Hierarchical Identity-Based Encryption. *Comput. J.* **2015**, *58*, 2778–2792. [CrossRef]
35. Ge, C.; Susilo, W.; Wang, J.; Fang, L. Identity-based conditional proxy re-encryption with fine grain policy. *Comput. Stand. Interfaces* **2017**, *52*, 1–9. [CrossRef]
36. Xiong, H.; Wang, Y.; Li, W.; Chen, C. Flexible, Efficient, and Secure Access Delegation in Cloud Computing. *ACM Trans. Manag. Inf. Syst. (TMIS)* **2019**, *10*, 2. [CrossRef]

37. Paul, A.; Selvi, S.S.D.; Rangan, C.P. A Provably Secure Conditional Proxy Re-Encryption Scheme without Pairing. *J. Internet Serv. Inf. Secur.* **2019**, *11*, 1–21.
38. Katz, J.; Lindell, Y. *Introduction to Modern Cryptography*, 2nd ed.; Computer Science, Mathematics, 2014. Available online: https://api.semanticscholar.org/CorpusID:9506320 (accessed on 28 November 2023).
39. Canetti, R.; Malkin, T.; Nissim, K. Efficient Communication-Storage Tradeoffs for Multicast Encryption. In Proceedings of the Advances in Cryptology-EUROCRYPT'99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, 2–6 May 1999; Stern, J., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1999; Volume 1592, pp. 459–474. [CrossRef]
40. Caro, A.D.; Iovino, V. jPBC: Java pairing based cryptography. In Proceedings of the 16th IEEE Symposium on Computers and Communications, ISCC 2011, Kerkyra, Corfu, Greece, 28 June–1 July 2011; IEEE Computer Society: Piscataway, NJ, USA, 2011; pp. 850–855. [CrossRef]
41. e Foundation. *Eclipse Paho Java Client*; Technical Report; e Foundation: Paris, France, 2021.
42. HiveMQ. *Hivemq-Community-Edition*; Technical Report; HiveMQ: Landshut, Germany, 2021.
43. HiveMQSDK. *HiveMQ Extension SDK 4.7.1 API*; Technical Report; HiveMQSDK: Hongkong, China, 2021.
44. Dirk, F. *AES*; Datenschutz und Datensicherheit, Advanced Encryption Standard (AES). 1999. Available online: https://api.semanticscholar.org/CorpusID:31476420 (accessed on 28 November 2023).

sensors

MDPI

Article

Cryptanalysis and Improvement of Several Identity-Based Authenticated and Pairing-Free Key Agreement Protocols for IoT Applications

Haiyan Sun, Chaoyang Li *, Jianwei Zhang, Shujun Liang and Wanwei Huang

College of Software Engineering, Zhengzhou University of Light Industry, Zhengzhou 450001, China; sunhaiyan@zzuli.edu.cn (H.S.)
* Correspondence: lichaoyang2013@163.com

Abstract: Internet of Things (IoT) applications have been increasingly developed. Authenticated key agreement (AKA) plays an essential role in secure communication in IoT applications. Without the PKI certificate and high time-complexity bilinear pairing operations, identity-based AKA (ID-AKA) protocols without pairings are more suitable for protecting the keys in IoT applications. In recent years, many pairing-free ID-AKA protocols have been proposed. Moreover, these protocols have some security flaws or relatively extensive computation and communication efficiency. Focusing on these problems, the security analyses of some recently proposed protocols have been provided first. We then proposed a family of eCK secure ID-AKA protocols without pairings to solve these security problems, which can be applied in IoT applications to guarantee communication security. Meanwhile, the security proofs of these proposed ID-AKA protocols are provided, which show they can hold provable eCK security. Some more efficient instantiations have been provided, which show the efficient performance of these proposed ID-AKA protocols. Moreover, comparisons with similar schemes have shown that these protocols have the least computation and communication efficiency at the same time.

Keywords: AKA; identity-based cryptography; eCK security model; attacks

Citation: Sun, H.; Li, C.; Zhang, J.; Liang, S.; Huang, W. Cryptanalysis and Improvement of Several Identity-Based Authenticated and Pairing-Free Key Agreement Protocols for IoT Applications. *Sensors* **2024**, *24*, 61. https://doi.org/10.3390/s24010061

Academic Editor: Alessandra Rizzardi

Received: 22 November 2023
Revised: 19 December 2023
Accepted: 19 December 2023
Published: 22 December 2023

1. Introduction

Internet of Things (IoT) applications have been increasingly exploited as information technology, operational technology, communication technology, and electronic technology develop. Examples include Smart Grid, Internet of Vehicles, Industrial IoT, and Agriculture IoT, which have enhanced living conditions greatly. With the development of 5G, secure and efficient communication demands have become huge as massive IoT devices are participating in these IoT applications. In general, IoT end devices and IoT servers play important roles in IoT applications. IoT end devices, mounted with sensors, collect useful information and transmit data to the IoT servers over the open network. When these IoT data are transmitted among different IoT devices, the sensitive information inserted in these IoT data needs more secure cryptographic algorithms to protect IoT user privacy and data security.

To protect the security of data transmission, cryptographic means that require secret keys are presented. For IoT user identity authentication, the shared key, public key certificate, and zero-knowledge proof are some common cryptographic methods [1]. For secure IoT data transmission, the digital signature can help confirm data ownership [2], the key agreement can help establish a secure session key [3], and the public key encryption can guarantee IoT data security with ciphertext in the public Internet environment [4]. For IoT data with fine-grained access control, attribute-based encryption can help establish a secure access control strategy with the attributes of IoT users and data [5], and security policy protocol can solve Internet control message protocol (ICMP) attacks [6]. Many

cryptographic algorithms guarantee secure and efficient communication among different IoT applications—there are too many to mention one by one. However, these cryptographic algorithms are public; therefore, the keys are essential for secure data transmission. Moreover, blockchain technology has been applied to IoT applications to solve the centralized management problem [7].

Key agreement (KA) is an important method to protect keys, which supports building a common session key between no less than two different users for subsequent communications. Authenticated key agreement (AKA) not only generates a common session key, but also prevents active attacks and implicitly authenticates the participants simultaneously. In the public-key infrastructure (PKI) setting, the user's long-term public key is matched with the corresponding identity in a certificate, which is derived by a trusted certificate authority. However, managing and transmitting these certificates results in heavy computation and storage costs. Considering IoT end-devices are usually resource-constrained and have limited memory, computation power, storage, and battery life, PKI-based AKA protocols (e.g., [3,8]) are not suitable for IoT applications. Therefore, to avoid the PKI certificate problem, AKA (ID-AKA) protocols with identity are proposed [9]. In ID-AKA protocols, every user owns a unique identity, the long-term public key is constructed by the user's personal identity, and the private key is composed of the identity and a master key, which is created by the trusted key generation center (KGC). Since the introduction of the first ID-AKA protocol in [10], many ID-AKA protocols have been introduced based on bilinear pairings [11–18], which is a high time-complexity operation. Because of the resource restriction of IoT end devices, ID-AKA protocols without pairings are more suitable for IoT applications.

Currently, the modified Bellare and Rogaway (mBR) model [12], the Canetti–Krawczyk (CK) model [19], and the extended Canetti–Krawczyk (eCK) model [20] are some famous security models for AKA protocols. In particular, the eCK model can achieve the most security properties [13]. Meanwhile, to satisfy the communication demands of the high speed, low latency, and large connections in IoT applications, a more secure and efficient AKA protocol is needed. Thus, designing efficient ID-AKA protocols without pairings while maintaining eCK security would be more suitable for IoT applications.

Motivation and Our Contribution

Aiming at considering the aforementioned problems, this paper presents some efficient ID-AKA protocols while holding the eCK security for IoT applications.

- We provide the preliminaries of some Diffie–Hellman assumptions and the forking lemma, provide a detailed description of the eCK-security model for two-party ID-AKA protocols, and draw a figure to show the network model for these protocols.
- We analyze three recently proposed ID-AKA protocols without pairings [21–23], and point out the security flaws against some known attacks.
- We propose a family of pairing-free ID-AKA protocols in the eCK security model. The security proof also considers the case where the public key materials related to the long-term private key can be altered by an active adversary. Furthermore, events in our security proof are complementary.
- We provide some more efficient protocols that need four elliptic curve point multiplication operations. Protocol comparison shows that our efficient protocols have the advantage over similar protocols with the items of security, computation, and communication efficiency.

The paper is organized as follows: Section 2 provides some related work, Section 3 presents the cryptanalysis of several ID-AKA protocols, Section 4 proposes a family of pairing-free ID-AKA protocols, Section 5 shows some more efficient instantiations, Section 6 presents the performance and comparison, and Section 7 gives the conclusion.

2. Related Work

Some related works about the cryptographic methods for IoT applications and developments of KA protocols are provided in the following two subsections.

2.1. Cryptographic Methods for IoT Applications

In IoT applications, the IoT data generally contain plenty of sensitive information about the system users, such as their personal identity, location, and production data. When these IoT data are transmitted among different IoT devices and applications, cryptographic technologies play an essential role in the processes of user identity authentication, IoT data transmission, IoT data fine-grained access control, and so on. Jayabalasamy et al. [2] proposed an aggregate signature scheme to solve the nonrepudiation problem in blockchain ecosystems, which could also confirm the data ownership in the untrusted IoT environment. Li et al. [3] designed a KA protocol based on the SM2 algorithm, which could help establish a secure session key between different system users in smart grid communications. Pu et al. [4] introduced a public key authentication encryption scheme, and added the function of keyword search into this scheme to guarantee IoT data security in industrial IoT applications. Rasori et al. [5] provided a survey of the attributed-based encryption protocols in recent years, and figured out the problems and challenges of IoT data fine-grained access control in most current IoT applications. Onyema et al. [6] presented a security policy protocol for the detection and prevention of Internet control message protocol (ICMP) attacks in software-defined networks. For these different cryptographic algorithms, the key is an essential part of secure IoT communication. Therefore, establishing a secure session key is the first issue that should be taken into consideration for secure communication in IoT applications.

2.2. Developments of KA Protocols

Numerous interesting pairing-free ID-AKA protocols have been introduced in recent years. In 2010, Cao et al. [24] provided a one-round ID-AKA protocol without pairing and presented the security proof in the mBR model. Then, two pairing-free ID-AKA protocols were proposed in [25,26], which utilized the CK model to prove its security. In 2016, Ni et al. [27] showed that the security proof of Sun et al.'s protocol neglected the case where the public key materials related to the long-term private key could be altered by an active adversary, and proposed two eCK-secure ID-AKA protocols without pairings. However, events in the security proof of their protocols were not complementary, which resulted in a mismatch in the freshness definition. Moreover, KGC needed to generate one extra long-term private key, which increased the storage, computation, and communication costs with the increase in the number of users. Bala et al. [21] presented an ID-AKA scheme without pairing for wireless sensor networks and claimed it was secure in the eCK security model. But Dang et al. [28] showed that its security proof had the same problem. Furthermore, the eCK security model was not secure as it suffered from an ephemeral key reveal attack.

In 2018, Dang et al. [28] provided a pairing-free ID-AKA protocol that could achieve eCK security in vehicular ad hoc networks. However, in 2021, Deng et al. [29] showed that it was not eCK-secure and put forward a new scheme that required only four scale multiplication operations. However, we found that the security proof had some flaws, for example, it was inappropriate that the challenger grasped the private key of KGC in the proof Cases CA2, CA3, CA4, and CA6. Mohammadali et al. [22] proposed the NIKE protocol, an ID-AKA protocol without pairing. However, there were still some drawbacks to it. Firstly, we found this scheme had a design flaw, i.e., if an individual was to learn about the long-term private keys of two parties (say $A(Meter)$ and $B(AHE)$), they could easily obtain KGC's master key. Secondly, it could not resist a KCI attack. Thirdly, although the NIKE protocol only needed, at most, three elliptic curve point multiplication operations, it needed three hash-to-point operations, which was a more time-consuming operation than the point multiplication operation. In 2019, Zhang et al. [23] gave two pairing-free and unbalanced ID-AKA protocols for disaster scenarios. Their protocols were actually

unbalanced versions of the protocol in [24]. Their protocols reduced one elliptic curve point multiplication operation for the limited party. However, Zhang et al.'s protocols did not have ephemeral key reveal resistance. In 2020, Daniel et al. [30] pointed out that Bala et al.'s protocol [21] could not resist an ephemeral key reveal attack. They also provided an ID-AKA protocol and presented its security proof in the eCK model [27]. However, its computational cost was still higher, which needed five-point multiplication operations in the elliptic curve. Furthermore, they pointed out that protocol [27] suffered from key offset attacks; however, key offset attacks could be simply avoided by adding a message authentication code using the same method in [30].

In 2021, Kumar and Chand [31] presented an ID-AKA protocol with cloud for the wireless body area network for anonymous health data authentication. However, Rakeei and Moazami [32] pointed out that this protocol could not resist a man-in-the-middle attack and achieve perfect forward secrecy. In 2022, Pu et al. [33] provided a mutual authentication and KA protocol for data privacy preserving in unmanned aerial vehicles (UAVs). Zhang et al. [34] designed a group key agreement (GKA) protocol for user privacy protection and data resource secure sharing in an intelligent IoT system. In 2023, Zhou et al. [35] presented an AGKA protocol for an AI-based automation system, which utilized a semi-trusted authority to perform precomputation operations. Pan et al. [36] focused on the communication security of UAVs to introduce a heterogeneous AKA protocol. Zhang et al. [37] provided a symmetric-key AKA protocol for edge-cloud IIoT, which could achieve perfect forward secrecy based on both authentication and derivation master keys. Abdussami et al. [38] proposed an AKA protocol for secure patient health-related data sharing in IoMT.

The security comparisons of these ID-AKA protocols are shown in Table 1, and the security items of the KGC's master key (MSS), weak perfect forward secrecy (wPFS), key compromise impersonation resilience (KCIR), ephemeral secrets reveal resistance (ESRR), and assumption (AS) were compared. Facing these problems and secure IoT communication demands, a secure ID-AKA protocol is needed to strengthen the security of session keys between different IoT users. The next sections will present the cryptanalysis of several ID-AKA protocols [21–23] first, and then provide the proposed ID-AKA protocols.

Table 1. Security comparisons.

Protocols	Security Model	wPFS	KCIR	ESRR	MSS	AS
CKD [24]	mBR*	Yes	Yes	No	Yes	GDH
FG-I [25]	CK	Yes	Yes	No	Yes	GDH
XW [26]	CK	Yes	Yes	No	Yes	GDH
PF-ID-2PAKA [21]	eCK* (flawed)	Yes	Yes	No	Yes	GDH
DXCLCZF-18 [28]	eCK (flawed)	Yes	No	No	Yes	GDH
ZHWY-19 [23]	mBR* (flawed)	No	No	No	Yes	GDH
NIKE [22]	– (flawed)	Yes	No	Yes	No	–
NCL-16-II [27]	eCK@	Yes	Yes	Yes	Yes	GDH
DRS-20 [30]	eCK	Yes	Yes	Yes	Yes	GDH
DSH-21 [29]	eCK*	Yes	Yes	Yes	Yes	GDH
PWCAS-22 [33]	eCK*	Yes	Yes	Yes	Yes	PUF
ZHVLH-23 [37]	eCK*	Yes	Yes	Yes	Yes	PRF

eCK* and mBR* are without the case where an active adversary may alter all public key materials not only the temporary public key. "–" denotes that there is no formal security proof for the protocol. eCK@ denotes that events in the proof are not complementary.

3. Preliminaries

Some basic concepts including complexity assumptions and the eCK security model for two-party ID-AKA protocols are reviewed in this section.

3.1. Complexity Assumptions

Let $\mathbb{G}$ be an elliptic curve additive group with a large prime order q, and P is a generator of $\mathbb{G}$. Some Diffie–Hellman assumptions over $\mathbb{G}$ are recalled as follows.

- **Computational Diffie–Hellman (CDH) Assumption**: Given three points, P, aP, and bP, where $a, b \in \mathbb{Z}_q^*$, the advantage $Adv_{\mathcal{M}}^{CDH}$ to compute abP is negligible for any probabilistic polynomial time (PPT) adversary $\mathcal{M}$.
- **Decision Diffie–Hellman (DDH) Assumption**: Given four points P, aP, bP, and cP, where $a, b, c \in \mathbb{Z}_q^*$, the advantage $Adv_{\mathcal{M}}^{DDH}$ to decide whether $c \equiv ab \bmod q$ is negligible for any PPT adversary $\mathcal{M}$.
- **Gap Diffie–Hellman (GDH) Assumption**: Given four points P, aP, bP, and cP, where $a, b, c \in \mathbb{Z}_q^*$, the advantage $Adv_{\mathcal{M}}^{GDH}$ to compute abP by accessing a DDH oracle is negligible for any PPT adversary $\mathcal{M}$.

3.2. The Forking Lemma

The forking lemma is applied in the security proof of our proposed protocols. Here, we recall it described in [27].

Let Θ_S be a generic digital signature scheme. Given an input message m, Θ_S produces a triple (K, h, v), where K is a randomly selected value in a large set, h is the hash value of (m, K), and v is only dependent on K, m, and h. Assume that a PPT algorithm $\mathcal{M}$ can produce a valid signature (K, h, v) on the message m with non-negligible probability. Then, with non-negligible probability, a replay of this algorithm can output two valid signatures (K, h, v) and $(K, \overline{h}, \overline{v})$ on the same message m, such that $h \neq \overline{h}$.

3.3. eCK-Security Model for Two-Party ID-AKA Protocols

We now recall the eCK-security model for two-party ID-AKA protocols in [13,27].

- **Participants.** Let $\mathcal{U} = \{ID_1, \cdots, ID_L\}$ be a finite set of L honest parties. Each participant $ID_i \in \mathcal{U}$ is modeled as a PPT Turing machine. Any two parties can be involved in a protocol execution. Each party may execute multiple instances (sessions) in parallel. Let $\prod_{i,j}^m$ denote the mth protocol session, which runs at party ID_i (the owner) with intended partner party ID_j. Every session $\prod_{i,j}^m$ has internal state variables $State_{i,j}^m$ and $tran_{i,j}^m$ to record the state of $\prod_{i,j}^m$, and the transcript of messages sent and received by $\prod_{i,j}^m$, respectively. If $\prod_{i,j}^m$ can compute a session key $SK_{i,j}^m$, $State_{i,j}^m = completed$. The messages in $tran_{i,j}^m$ are ordered according to the protocol specification.
- **Adversary Model.** The adversary $\mathcal{M}$ is modeled as a PPT Turing machine and has full control of the communication network. Active attacks are formulated by allowing the adversary $\mathcal{M}$ to perform the following queries:
  - EphemeralKeyReveal ($\prod_{i,j}^m$). The adversary $\mathcal{M}$ is provided the ephemeral private key of $\prod_{i,j}^m$.
  - SessionKeyReveal ($\prod_{i,j}^m$). The session key held by a completed session $\prod_{i,j}^m$ is returned to $\mathcal{M}$.
  - Corrupt (ID_i). The long-term private key of ID_i is returned to the adversary $\mathcal{M}$.
  - KGCStaticKeyReveal. The adversary $\mathcal{M}$ obtains the master key of KGC. This query is used to model master key forward secrecy.
  - RegCT (ID_i). Via this query, the adversary $\mathcal{M}$ is able to register a dishonest party with identity ID_i. Meanwhile, $\mathcal{M}$ obtains ID_i's long-term private key and totally controls ID_i.
  - Send ($\prod_{i,j}^m, M$). Via this query, the adversary $\mathcal{M}$ can send any message M to party ID_i in session $\prod_{i,j}^m$ on behalf of party ID_j. The adversary $\mathcal{M}$ is responded to according to the protocol specification. $\prod_{i,j}^m$ can be initiated by ID_i when $M = \lambda$. In general, for simplicity $ID_i \neq ID_j$ is required, i.e., two identical participants will not run a session. Internal states of $\prod_{i,j}^m$ should be maintained accordingly.

  - **Test ($\prod_{i,j}^{m}$)**. The input session $\prod_{i,j}^{m}$ must be fresh. In response to this query, $\prod_{i,j}^{m}$ flips a fair coin $b \in \{0,1\}$, and returns the real session key if $b = 0$, or a random sample from the distribution of the session key if $b = 1$.
- **Security Experiment.** The security experiment between the adversary $\mathcal{M}$ and the challenger $\mathcal{CH}$ consists of the following phases.
  - **Setup.** The challenger $\mathcal{CH}$ generates the system parameters along with the master private key and valid long-term secret keys for each party. The adversary $\mathcal{M}$ is then provided all public data, including the identities of all the honest parties.
  - **The first phase of the game.** Adversary $\mathcal{M}$ is allowed to issue a polynomial number of EphemeralKeyReveal, SessionKeyReveal, Corrupt, KGCStaticKeyReveal, RegCT, and Send queries in any order.
  - **The second phase of the game.** At some point, adversary $\mathcal{M}$ chooses a fresh session $\prod_{i,j}^{m}$ (see Definition 2) and issues a Test($\prod_{i,j}^{m}$) query at most once. After this, adversary $\mathcal{M}$ can keep asking other queries under the condition that the test session must remain fresh.
  - **The end of the game.** $\mathcal{M}$ makes a guess b' for b.
- **Advantage.** $\mathcal{M}$ wins the above security experiment if the test session $\prod_{i,j}^{m}$ is still fresh and $b' = b$. The advantage of $\mathcal{M}$ in winning the above security experiment is defined as $Adv^{AKE}(\mathcal{M}) = |2Pr[\mathcal{M}\ wins] - 1|$, where $\mathcal{M}$ wins refers to $\mathcal{M}$ can distinguish the tested session key from a random string.

In the following, we introduce definitions for Matching Session, Freshness, and eCK Security.

Definition 1 (Matching Session)**.** *If two completed sessions $\prod_{i,j}^{m}$ and $\prod_{j,i}^{n}$ have the same message transcript, they are said to be matching.*

Definition 2 (Freshness)**.** *Let $\prod_{i,j}^{m}$ be a completed session between honest party ID_i and ID_j, $\prod_{i,j}^{m}$ is said to be fresh if none of the following three conditions hold:*

(1) The adversary $\mathcal{M}$ knows the session key of $\prod_{i,j}^{m}$ or its matching session $\prod_{j,i}^{n}$ (if $\prod_{j,i}^{n}$ exists);

(2) $\prod_{i,j}^{m}$ has a matching session $\prod_{j,i}^{n}$, and the adversary $\mathcal{M}$ knows both the long-term private key of participant ID_i and the ephemeral private key of $\prod_{i,j}^{m}$, or both the long-term private key of participant ID_j and the ephemeral private key of $\prod_{j,i}^{n}$.

(3) $\prod_{i,j}^{m}$ has no matching session, and the adversary $\mathcal{M}$ knows both the long-term private key of participant ID_i and the ephemeral private key of $\prod_{i,j}^{m}$, or the long-term private key of participant ID_j.

Remark 1. *The first condition in Definition 2 is to exclude the trivial attack that $\mathcal{M}$ obtains the session key directly. The second and third conditions in Definition 2 are to exclude the trivial attack that $\mathcal{M}$ obtains the long-term private key and the ephemeral private key of one party simultaneously. If $\prod_{i,j}^{m}$ has no matching session, it means that $\mathcal{M}$ has obtained the ephemeral private key of participant ID_j; therefore, $\mathcal{M}$ cannot obtain the long-term private key of ID_j.*

Definition 3 (eCK Security)**.** *We say that an ID-AKA protocol is secure in the eCK model if the following conditions hold:*

(1) If two honest parties successfully complete matching sessions, they both compute the same session key.

(2) For any PPT adversary, $\mathcal{M}$, $Adv^{AKE}(\mathcal{M})$ is negligible in security parameter k.

Remark 2. *If a protocol is secure under Definition 3, then it achieves implicit mutual key authentication and the basic security properties, including weak perfect forward secrecy (wPFS), key compromise impersonation resilience (KCIR), ephemeral secrets reveal resistance (ESRR), known key*

security, no key control, resistance to basic impersonation attack, replay attack resilience, resistance to man-in-the-middle attack, and unknown key share resilience.

4. Cryptanalysis of Several ID-AKA Protocols

Figure 1 shows the network model for ID-AKA protocols considered in our paper. Here, user A, user B, and trusted authority (acts as KGC) are the three main parties of an ID-AKA protocol. A and B obtain their long-term private keys in the extract phase and use them to reach AKA each other. Next, this section provides the cryptanalysis of several ID-AKA protocols.

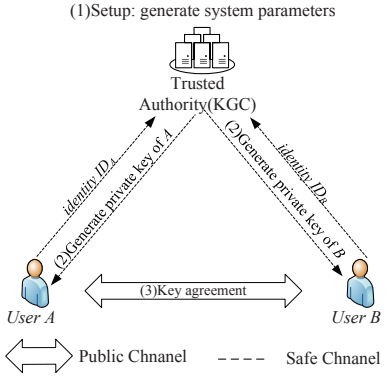


Figure 1. Network model for ID-AKA protocols.

4.1. The PF-ID-2PAKA Protocol

PF-ID-2PAKA [21] is also composed of three stages, i.e., setup, extract, and key agreement. The former two stages are the same as those of the DXCLCZF-18 protocol [28]. Here, we only describe the key agreement stage in Figure 2.

$A(ID_A, R_A, s_A)$		$B(ID_B, R_B, s_B)$
	$M_1 = \{ID_A, R_A, T_A\}$ →	
$e_A \in_R \mathbb{Z}_q^*$		$e_B \in_R \mathbb{Z}_q^*$
$T_A = e_A P$	← $M_2 = \{ID_B, R_B, T_B\}$	$T_B = e_B P$
$K_{AB}^1 = (e_A + s_A)(T_B + H_1(ID_B \Vert R_B)T_B)$		$K_{BA}^1 = (e_B + s_B)(T_A + H_1(ID_A \Vert R_A)T_A)$
$K_{AB}^2 = e_A T_B$		$K_{BA}^2 = e_B T_A$
$sk = H_2(ID_A \Vert ID_B \Vert T_A \Vert T_B \Vert K_{AB}^1 \Vert K_{AB}^2)$		$sk = H_2(ID_A \Vert ID_B \Vert T_A \Vert T_B \Vert K_{BA}^1 \Vert K_{BA}^2)$

Figure 2. The key agreement phrase of the PF-ID-2PAKA protocol [21].

4.1.1. Ephemeral Key Reveal Attack

Suppose that A's ephemeral key e_A is compromised by an adversary. Next, we show that the PF-ID-2PAKA protocol suffers from an ephemeral key reveal attack.

(1) The adversary $\mathcal{M}$ initializes a session $\prod_{A,B}^{\ell}$ through the query Send($\prod_{A,B}^{\ell}, \perp$), and then obtains the message $M_1 = \{ID_A, R_A, T_A\}$, where $T_A = e_A P$ with a random element $e_A \in \mathbb{Z}_q^*$.

(2) Upon receiving the message, M_1, $\mathcal{M}$ randomly picks an ephemeral private key $e_B^{\mathcal{M}} \in \mathbb{Z}_q^*$, calculates $T_B^{\mathcal{M}} = e_B^{\mathcal{M}} P - R_B - H_1(ID_B, R_B)P_{pub}$, and returns $M_2 = \{ID_B, R_B, T_B^{\mathcal{M}}\}$ to $\prod_{A,B}^{\ell}$ impersonating B via the query Send($\prod_{A,B}^{\ell}, M_2$). Note that B's identity ID_B

and correct public key material R_B can be obtained from the response of the query Send($\prod_{B,U}^{t}, \perp$).

(3) Upon the receipt of M_2, A calculates the shared session key and completes this session. Specifically, A calculates $sk = H_2(ID_A||\ ID_B||T_A||T_B^{\mathcal{M}}||K_{AB}^1||K_{AB}^2)$, where $K_{AB}^1 = (s_A + e_A)(R_B + H_1(ID_B, R_B)P_{pub} + T_B^{\mathcal{M}})$ and $K_{AB}^2 = e_A T_B^{\mathcal{M}}$.

(4) Now, $\mathcal{M}$ performs the query EphemeralKeyReveal($\prod_{A,B}^{\ell}$) to reveal an ephemeral private key e_A. With the knowledge of $e_B^{\mathcal{M}}$ and e_A, $\mathcal{M}$ computes $K_{BA}^{1\mathcal{M}} = e_B^{\mathcal{M}}(R_A + H_1(ID_A, R_A)P_{pub} + T_A)$, $K_{BA}^{2\mathcal{M}} = e_A T_B^{\mathcal{M}}$, and $sk = H_2(ID_A||\ ID_B||T_A||T_B^{\mathcal{M}}||K_{BA}^{1\mathcal{M}}||K_{BA}^{2\mathcal{M}})$.

Correctness. The following provides the correctness of the attack process.
As a result of $T_B^{\mathcal{M}} = e_B^{\mathcal{M}} P - R_B - H_1(ID_B, R_B)P_{pub}$, we can obtain

$$\begin{aligned} K_{AB}^1 &= (s_A + e_A)(R_B + H_1(ID_B, R_B)P_{pub} + T_B^{\mathcal{M}}) \\ &= (s_A + e_A)e_B^{\mathcal{M}} P \\ &= e_B^{\mathcal{M}}(s_A P + e_A P) \\ &= e_B^{\mathcal{M}}(R_A + H_1(ID_A, R_A)P_{pub} + T_A) \\ &= K_{BA}^{1\mathcal{M}}. \end{aligned}$$

Thus, A and $\mathcal{M}$ receive the same session key, which means that the PF-ID-2PAKA protocol suffers from an ephemeral key reveal attack. Note that our construction cannot suffer from the above attack as both of the two shared secrets not only depend on the ephemeral private key, but also depend on the long-term private key, and they are linearly independent. Therefore, it is impossible to remove the long-term private key from the two shared secrets simultaneously.

4.1.2. Flaws in the Security Proof

The PF-ID-2PAKA protocol can not be proved secure under the hardness of the GDH problem in Case 3 (i.e., $\mathcal{M}$ can neither obtain the long-term private key of ID_A nor that of ID_B). Meanwhile, the ephemeral key t_B of ID_B may be created by $\mathcal{M}$, then $\mathcal{CH}$ cannot know t_B. In ignorance of t_B, $\mathcal{CH}$ does not calculate CDH$(U, V) = K_1 - t_A(T_B + V) - t_B U$. Therefore, the challenger $\mathcal{CH}$ cannot solve the GDH instance.

4.2. The ZHWY-19 Protocol

Here, we only describe the key agreement stage of the ZHWY-19-I protocol [23] in Figure 3. For more details, one can refer to [23]. Note that Cheng et al. [39] pointed out that the ZHWY-19-I protocol [23] cannot achieve forward security and resist the key compromise impersonation attack. Here, we point out that this protocol is weak against the ephemeral key reveal attack and flaws in the security proof.

4.2.1. Ephemeral Key Reveal Attack

If e_A and e_B in a past session have been compromised by the adversary $\mathcal{M}$, $\mathcal{M}$ can compute the session key sk.

(1) $\mathcal{M}$ accesses to $M_1 = \{R_A, V_A, u_A\}$ and $M_2 = \{R_B, V_B, T_B, T_A, mac_B\}$ of the session.

(2) Given $\{ID_A, e_A, R_A\}$ and $\{ID_B, e_B, R_B\}$, $\mathcal{M}$ calculates the keys of $PK_A = R_A + H_1(ID_A||R_A)P_{pub}$, $PK_B = R_B + H_1(ID_B||R_B)P_{pub}$, $K_1 = e_A PK_B + e_B PK_A$, $K_2 = e_A e_B P$, and $sk = H_2(ID_A||ID_B||T_A||T_B||K_1||K_2)$ as the shared session key.

Thus, the ZHWY-19-I protocol is weak against the ephemeral secret key leakage. Note that they did not claim their protocol holds ephemeral key reveal resistance.

4.2.2. Flaws in the Security Proof

In the ZHWY-19-I protocol, the answer to oracle Corrupt(ID_i) is improper. Actually, Corrupt(ID_i) should return the long-term private key (s_i, R_i, v_i, V_i) rather than s_i to the adversary.

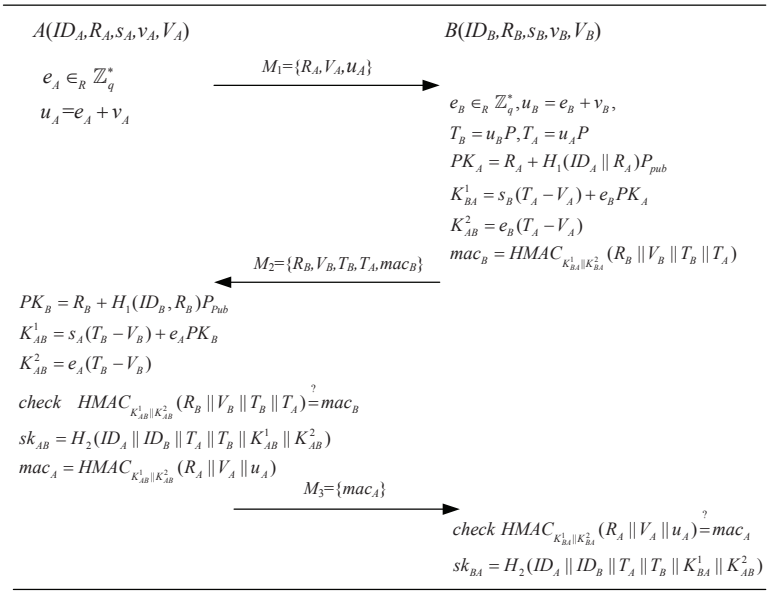


Figure 3. The key agreement phrase of the ZHWY-19-I protocol [23].

4.3. Mohammadali et al.'s Protocols

Mohammadali et al. [22] proposed two protocols, the NIKE protocol and the NIKE$^+$ protocol. These two protocols contain three stages, namely setup, extract, and key agreement. The NIKE protocol is briefly shown in Figure 4. Note that, here, we did not analyze the flaws in the security proof as there was no security proof in [22].

4.3.1. The Insecurity of the KGC's Master Key

If the user $A(Meter)$ and the user $B(AHE)$ launch collusion attacks, they can know the KGC's master key s. As $y_A = H_2(ID_B, Y_B)s$, they can obtain $s = H_2(ID_B, Y_B)^{-1} y_A$ with the knowledge of y_A and Y_B.

4.3.2. Key Compromise Impersonation (KCI) Attack

The NIKE protocol suffers from a key compromise impersonation (KCI) attack, i.e., if the user $B(AHE)$'s long-term private key Y_B is compromised by $\mathcal{M}$, $\mathcal{M}$ can impersonate any user (say $A(Meter)$) with $B(AHE)$'s long-term private key Y_B to communicate with $B(AHE)$. The details are as follows. Note that they did not claim their protocol held KCI resistance.

(1) $\mathcal{M}$ obtains ID_A, R_A by eavesdropping on a connection between $A(Meter)$ and any user. Then, $\mathcal{M}$ picks $e_A^{\mathcal{M}} \in \mathbb{Z}_q^*$ at random, calculates $T_A^{\mathcal{M}} = e_A^{\mathcal{M}} P - H_2(ID_B, Y_B) P_{pub} - R_A$, and sends $\{ID_A, T_A^{\mathcal{M}}, R_A\}$ to $B(AHE)$.
(2) Upon the receipt of $\{ID_A, T_A^{\mathcal{M}}, R_A\}$, $B(AHE)$ generates t_B, T_B, K_{BA}, m_B according to the protocol.
(3) Upon receiving $\{ID_B, T_B, m_B\}$, $\mathcal{M}$ calculates $K_{AB}^{\mathcal{M}} = e_A^{\mathcal{M}} T_B$, $m_A^{\mathcal{M}} = H_1(1, K_{AB}^{\mathcal{M}})$ and $SK = H_1(ID_A || ID_B || K_{AB}^{\mathcal{M}})$, and finally sends $m_A^{\mathcal{M}}$ to $B(AHE)$.

(4) Upon receiving $m_A^{\mathcal{M}}$, $B(AHE)$ verifies $m_A^{\mathcal{M}}$ is correct and computes the session key according to the protocol.

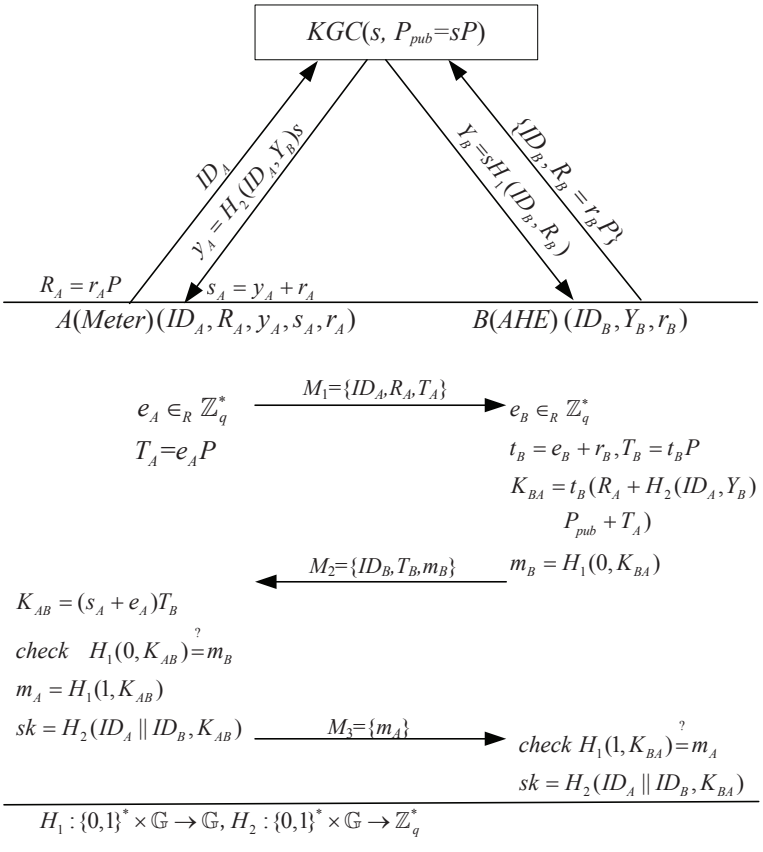


Figure 4. The NIKE protocol [22].

Correctness. The following provides the correctness of the attack process. As $T_A^{\mathcal{M}} = e_A^{\mathcal{M}} P - H_2(ID_B, y_B) P_{pub} - R_A$ and $T_B = t_B P$, we can obtain

$$\begin{aligned} K_{BA} &= t_B(R_A + H_2(ID_A, Y_B)P_{pub} + T_A^{\mathcal{M}}) \\ &= t_B(R_A + H_2(ID_A, Y_B)P_{pub} + e_A^{\mathcal{M}} P \\ &\quad - H_2(ID_B, Y_B)P_{pub} - R_A) \\ &= t_B e_A^{\mathcal{M}} P \\ &= e_A^{\mathcal{M}} T_B \\ &= K_{AB}^{\mathcal{M}}. \end{aligned}$$

Thus, A and $\mathcal{M}$ obtain the same session key, which means that the NIKE protocol suffers from a KCI attack. The KCI attack on the NIKE$^+$ protocol is the same as above.

5. Our General Construction

This section firstly provides the construction $\sum_{C_1,C_2,C_3,C_4}$, secondly show the construction $\sum_{C_1,C_2,C_3,C_4}$ is correct, and thirdly provides the security proof.

5.1. Construction Description

The $\sum_{C_1,C_2,C_3,C_4}$ is composed of three stages, i.e., setup, extract, and key agreement.

- **Setup:** Select security parameter k, KGC performs as follows:
  (1) Pick an elliptic curve $E/\mathbb{F}_p$, where $\mathbb{F}_p$ is a finite filed, and p is a prime number with k bits.
  (2) Create a cyclic additive group $\mathbb{G}$ with the order q, which is generated by a base point P over $E/\mathbb{F}_p$.
  (3) Choose $s \in \mathbb{Z}_q^*$ randomly, and then set the master private key s and the system public key $P_{pub} = sP$.
  (4) Pick $H_1 : \{0,1\}^* \to \mathbb{Z}_q^*$ and $H_2 : \{0,1\}^* \to \{0,1\}^k$.
  (5) Expose $\langle E/\mathbb{F}_p, \mathbb{G}, q, P, P_{pub}, H_1, H_2 \rangle$, and meanwhile retain s unrevealed.
- **Extract:** KGC derives the long-term private key for user $ID_i \in \{0,1\}^*$ as below.
  (1) KGC randomly selects $r_i \in \mathbb{Z}_q^*$, and calculates $R_i = r_i P$ and $h_i = H_1(ID_i||R_i)$.
  (2) KGC computes $s_i = r_i + h_i s \bmod q$ and derives (s_i, R_i) as the user's long-term private key.
  (3) KGC sends (s_i, R_i) to the user securely.

  Upon receiving (s_i, R_i), the user can verify $s_i P \stackrel{?}{=} R_i + H_1(ID_i||R_i)P_{pub}$. If this verification succeeds, the key pair (s_i, R_i) is correct and valid. $s_i P$ serves as the real public key in relation to ID_i.
- **Key Agreement:** Assume that user A with identity ID_A hopes to compute a key with user B with identity ID_B.
  (1) A randomly picks an ephemeral secret key $e_A \in \mathbb{Z}_q^*$, calculates $T_A = e_A P$, and returns $M_1 = \{ID_A, R_A, T_A\}$ to B. The agreement process in Figure 5.
  (2) When $\{ID_A, R_A, T_A\}$ is received, B picks an ephemeral secret key $e_B \in \mathbb{Z}_q^*$, calculates $T_B = e_B P$, and returns $\{ID_B, R_B, T_B\}$ to A. Next, B calculates $sk_{BA} = H_2(ID_A||ID_B||R_A||R_B||T_A||T_B||K_{BA}^1||K_{BA}^2)$ as the shared session key, where $K_{BA}^1 = (s_B + C_2 e_B)(PK_A + C_1 T_A)$, $K_{BA}^2 = (s_B + C_4 e_B)(PK_A + C_3 T_A)$, $PK_A = R_A + H_1(ID_A||R_A)P_{pub}$, $C_i (i = 1,2,3,4) \in \mathbb{Z}_q^*$ and $C_1 \neq C_3, C_2 \neq C_4$. Finally, B sends $M_2 = \{ID_B, R_B, T_B\}$ to A.
  (3) When $M_2 = \{ID_B, R_B, T_B\}$ is received, A calculates $PK_B = R_B + H_1(ID_B||R_B)P_{pub}$, and two shared secrets $K_{AB}^1 = (s_A + C_1 e_A)(PK_B + C_2 T_B)$ and $K_{AB}^2 = (s_A + C_3 e_A)(PK_B + C_4 T_B)$, where $C_i (i = 1,2,3,4) \in \mathbb{Z}_q^*$ and $C_1 \neq C_3, C_2 \neq C_4$. Finally, A calculates the shared session key $sk_{AB} = H_2(ID_A||ID_B||R_A||R_B||T_A||T_B||K_{AB}^1||K_{AB}^2)$.

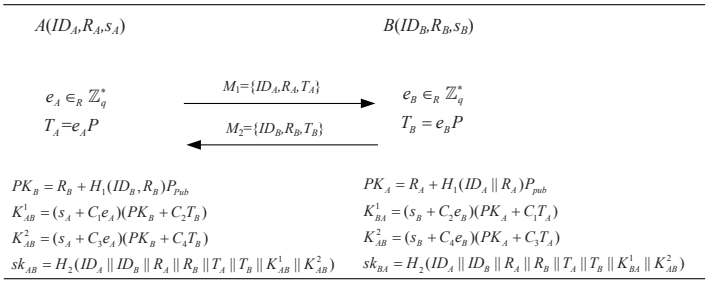


Figure 5. The key agreement phrase of our proposed construction $\sum_{C_1,C_2,C_3,C_4}$.

Note that our construction provides a method to construct eCK secure ID-AKA protocols; however, parameters C_1, C_2, C_3, and C_4 should be fixed in the real execution environment. One can choose a concrete and efficient protocol derived from our construction to

execute in the real environment, e.g., protocol $\sum_{1,1,-1,-1}$, protocol $\sum_{1,-1,-1,1}$ and protocol $\sum_{1,1,2,2}$ described in Section 6.

5.2. Construction Correctness

The following provides the correctness of our construction. As $PK_B = R_B + H_1(ID_B||R_B)$ $P_{pub} = s_BP$, $PK_A = R_A + H_1(ID_A||R_A)P_{pub} = s_AP$, $T_A = e_AP$ and $T_B = e_BP$, we can obtain:

$$\begin{aligned} K_{AB}^1 &= (s_A + C_1e_A)(PK_B + C_2T_B) \\ &= (s_A + C_1e_A)(s_BP + C_2e_BP) \\ &= (s_A + C_1e_A)(s_B + C_2e_B)P \\ &= (s_B + C_2e_B)(s_A + C_1e_A)P \\ &= (s_B + C_2e_B)(PK_A + C_1T_A) = K_{BA}^1 = K_1; \end{aligned}$$

$$\begin{aligned} K_{AB}^2 &= (s_A + C_3e_A)(PK_B + C_4T_B) \\ &= (s_A + C_3e_A)(s_BP + C_4e_BP) \\ &= (s_A + C_3e_A)(s_B + C_4e_B)P \\ &= (s_B + C_4e_B)(s_A + C_3e_A)P \\ &= (s_B + C_4e_B)(PK_A + C_3T_A) = K_{BA}^2 = K_2. \end{aligned}$$

Thus both A and B compute $sk_{AB} = sk_{BA} = sk = H_2(ID_A||ID_B||R_A||R_B||T_A||T_B||K_1||K_2)$ as their session key. Hence the correctness holds.

5.3. Security Proof

Here, the events in our security proof are complementary, while they are not complementary in [27], and the security proof can be reduced to the following theorems.

Theorem 1. *Provide two random oracles, H_1 and H_2, the proposed ID-AKA protocol is secure in the eCK model based on the GDH assumption over the elliptic curve group.*

Proof. This theorem is under the condition that the two conditions shown in Definition 3 hold. The correctness analysis shows that the first condition stands. The second condition would be proven by contradiction, i.e., there is an adversary who can execute a PPT algorithm to win the game with non-negligible probability, we can use $\mathcal{M}$ to create a GDH solver $\mathcal{CH}$ who can find a solution for the GDH instance. □

Assume $\mathcal{M}$ is a polynomially (in security parameter k) bounded adversary whose advantage is $Adv_{\mathcal{M}}(k)$. Suppose that $\mathcal{M}$ activates no more than $n_p(k)$ different honest parties, and each party can take part in no more than $n_s(k)$ sessions. Suppose that $\mathcal{M}$ chooses $\prod_{a,b}^{n}$, the nth protocol session which executes between party ID_a (the owner) and the target party ID_b (the peer) as the test session. Assume that $\mathcal{M}$ performs, at most, $n_h(k)$ H_2 queries.

According to $Adv^{AKE}(\mathcal{M}) = |2Pr[\mathcal{M}\ wins] - 1|$, we can derive that $Pr[\mathcal{M}\ wins]$ is non-negligible as $Adv^{AKE}(\mathcal{M})$ is non-negligible. As H_2 is modeled as a random oracle, $\mathcal{M}$ can make a clear distinction between a random string and the tested session key in the following three ways:

A1. Guessing attack: $\mathcal{M}$ directly guesses the correct session key.
A2. Key replication attack: $\mathcal{M}$ successfully creates a session that cannot match the test session while holding the same session key. Here, $\mathcal{M}$ can obtain the test session key by querying the non-matching session key.
A3. Forging attack: Sometimes, $\mathcal{M}$ makes H_2 queries on $(ID_a, ID_b, R_a, R_b, T_a, T_b, K_1, K_2)$ in the test session. Here, $\mathcal{M}$ calculates K_1 and K_2 itself.

The guessing of H_2's output is with the negligible probability $\mathcal{O}(1/2^k)$. If two sessions are different, H_2 has the same input by probability $\mathcal{O}(n_s(k)^2/2^k)$, which is also negligible.

Then, $\mathcal{M}$ can provide the difference between a random string and the tested session key only by *forging attack*.

Next, a reduction approach is applied to analyze the *forging attack*. This approach reduces the protocol security to the hardness of mathematical problems in the GDH assumption. By making assumptions about the adversary, a challenger can solve a GDH instance with the queried data and forged session key derived by a query-respond game between them. As the GDH instance cannot be solved with the current computation ability in polynomial time, the assumptions about the adversary are invalid, and the proposed ID-AKA protocols are secure.

Now, the detailed descriptions of the reduction proofs are as follows.

If $\mathcal{M}$ can successfully execute *forging attack* with non-negligible probability $Adv^{F}_{\mathcal{M}}(k)$, we will use $\mathcal{M}$ to create a GDH solver $\mathcal{CH}$ to find a solution for the GDH instance with $Adv^{GDH}_{\mathcal{S}}(k)$. Here, GDH instance is $(U = uP, V = vP)$, and $u, v \in \mathbb{Z}^{*}_{q}$, $P \in \mathbb{G}$, $\mathcal{CH}$ plans to calculate $\mathrm{GDH}(U, V) = uvP$ performing the DDH oracle. $\mathcal{CH}$ acts as a challenger that performs the eCK game with $\mathcal{M}$ and makes response for $\mathcal{M}$'s queries.

Before the game starts, $\mathcal{CH}$ guesses the test session that $\mathcal{M}$'s choices is $\prod^{n}_{a,b}$ with a correct probability at least $1/n_p(k)^2 n_s(k)$. Next, $\mathcal{CH}$ needs to guess the strategy that $\mathcal{M}$ adopts. Then, according to Definition 2, test session $\prod^{n}_{a,b}$ has the matching session $\prod^{l}_{b,a}$, then $\mathcal{M}$ can only passively forward messages between participant ID_a and participant ID_b, i.e., messages including public key materials and ephemeral keys of $\prod^{n}_{a,b}$ and $\prod^{l}_{b,a}$ are selected by $\mathcal{CH}$. Test session $\prod^{n}_{a,b}$ has no matching session, then $\mathcal{M}$ alters some messages at its own will, i.e., messages including the public key material and the ephemeral key of $\prod^{n}_{a,b}$ are chosen by $\mathcal{CH}$, and another one of ID_b is chosen by $\mathcal{M}$, and, thus, for ID_b, $\mathcal{CH}$ can only consider the long-term private key of ID_b. With the former analysis and the freeness definition, $\mathcal{CH}$ guesses the operation that $\mathcal{M}$ selects one of the following six complementary choices. Note that, strictly speaking, the ephemeral private key of ID_a refers to $\prod^{n}_{a,b}$'s ephemeral private key, and the ephemeral private key of ID_b refers to the matching session $\prod^{l}_{b,a}$'s ephemeral private key.

S1: $\prod^{n}_{a,b}$ has $\prod^{l}_{b,a}$, and $\mathcal{M}$ obtains neither the long-term private key of ID_a nor the ephemeral private key of ID_b.
S2: $\prod^{n}_{a,b}$ has $\prod^{l}_{b,a}$, and $\mathcal{M}$ cannot obtain any information about the ephemeral private keys of ID_a and ID_b.
S3: $\prod^{n}_{a,b}$ has $\prod^{l}_{b,a}$, and $\mathcal{M}$ knows neither the ephemeral private key of ID_a nor the long-term private key of ID_b.
S4: $\prod^{n}_{a,b}$ has $\prod^{l}_{b,a}$, and $\mathcal{M}$ cannot obtain any information about the long-term private keys of ID_a and ID_b.
S5: $\prod^{n}_{a,b}$ does not have a matching session, and $\mathcal{M}$ knows neither the ephemeral private key of ID_a nor the long-term private key of ID_b.
S6: $\prod^{n}_{a,b}$ does not have a matching session, and $\mathcal{M}$ does not know any information about the long-term private keys of ID_a and ID_b.

One of the former operation successes is if $\mathcal{M}$ succeeds in a forging attack with non-negligible probability. Therefore, the assumption about adversary $\mathcal{M}$ is invalid, and the proposed ID-AKA protocol is secure in the eCK model.

5.3.1. The Analysis of Strategy S1

In this subsection, we analyze strategy S1.

- **Setup:** $\mathcal{CH}$ initializes a list $Setup^{list}$ with entries of $(ID_i, (d_i, R_i), PK_i)$. $\mathcal{CH}$ creates the system parameters and long-term private keys of all parties as follows.
  - $\mathcal{CH}$ picks $P_{pub} \in \mathbb{G}$ at random, and exposes $\langle E/\mathbb{F}_p, \mathbb{G}, q, P, P_{pub}, H_1, H_2 \rangle$. Thus, $\mathcal{CH}$ cannot obtain any information about KGC's master key.
  - For ID_a, $\mathcal{CH}$ sets the long-term private key $(\perp, R_a)$, where $h_a \in_R \mathbb{Z}^{*}_{q}$, $R_a = U - h_a P_{pub}$. Thus, $PK_a = R_a + h_a P_{pub} = U$.

- For $ID_i (i \neq a)$, $\mathcal{CH}$ sets the long-term private key (s_i, R_i), where $h_i, s_i \in_R \mathbb{Z}_q^*$, $R_i = s_i P - h_i P_{pub}$. Thus, $PK_i = R_i + h_i P_{pub} = s_i P$.
- For every participant, $\mathcal{CH}$ transfers (ID_i, R_i) to $\mathcal{M}$, and stores the tuple $(ID_i, (d_i, R_i), PK_i)$ and (ID_i, R_i, h_i) in $Setup^{list}$ and ${H_1}^{list}$ (described later), respectively.

- **Queries:** $\mathcal{CH}$ maintains four lists, ${H_1}^{list}$, ${H_2}^{list}$, $Send^{list}$, and R^{list}, which are initially empty and used to record H_1, H_2, Send, and SessionKeyReveal oracles, respectively. $\mathcal{CH}$ starts $\mathcal{M}$ by answering $\mathcal{M}$'s queries, as follows.
  - $H_1(ID_i, R_i)$: If an entry (ID_i, R_i, h_i) is recorded in ${H_1}^{list}$, $\mathcal{CH}$ responds with h_i. Then, $\mathcal{CH}$ randomly selects $h_i \in \mathbb{Z}_q^*$, appends (ID_i, R_i, h_i) to ${H_1}^{list}$, and sends h_i back to $\mathcal{M}$.
  - $H_2(ID_i, ID_j, R_i, R_j, T_i, T_j, K_1, K_2)$: List ${H_2}^{list}$ is with $(ID_i, ID_j, R_i, R_j, T_i, T_j, K_1, K_2, h_2)$.
    - If a matching entry $(ID_i, ID_j, R_i, R_j, T_i, T_j, K_1, K_2, h_2)$ is stored in ${H_2}^{list}$, $\mathcal{CH}$ replies with h_2.
    - Else, $\mathcal{CH}$ seeks $(*, ID_i, ID_j, R_i, R_j, T_i, T_j, *)$ in R^{list}. Then, if such an entry exists, $\mathcal{CH}$ sees if K_1 and K_2 are produced correctly by validating $\text{DDH}(PK_i + C_1 T_i, PK_j + C_2 T_j, K_1) \stackrel{?}{=} 1$ and $\text{DDH}(PK_i + C_3 T_i, PK_j + C_4 T_j, K_2) \stackrel{?}{=} 1$, respectively, where $PK_i = R_i + H_1(ID_i, R_i) P_{pub}$ and $PK_j = R_j + H_1(ID_j, R_j) P_{pub}$. If both verifications pass, $\mathcal{CH}$ receives the corresponding $SK_{i,j}^m$ and sets $h_2 \leftarrow SK_{i,j}^m$. Otherwise (at least one verification fails or none), $\mathcal{CH}$ picks $h_2 \in \{0,1\}^k$ at random. Finally, $\mathcal{CH}$ inserts the tuple $(ID_i, ID_j, T_i, T_j, K_1, K_2, h_2)$ into ${H_2}^{list}$ and provides h_2 as the answer.
  - Corrupt(ID_i): If $ID_i = ID_a$, $\mathcal{CH}$ discontinues. Otherwise, $\mathcal{CH}$ responds with s_i.
  - KGCStaticKeyReveal: $\mathcal{CH}$ discontinues.
  - EphemeralKeyReveal($\prod_{i,j}^m$): If $\prod_{i,j}^m = \prod_{b,a}^l$, $\mathcal{CH}$ discontinues. Otherwise, $\mathcal{CH}$ provides the stored ephemeral key $r_{i,j}^m$ as the answer.
  - Send($\prod_{i,j}^m$,M): List $Send^{list}$ is with $(\prod_{i,j}^m, tran_{i,j}^m, r_{i,j}^m, State_{i,j}^m)$, where $tran_{i,j}^m$, $r_{i,j}^m$ and $State_{i,j}^m$ are the transcript by now, the ephemeral secret key, and the state by now, respectively.
    - If M is the second message on the transcript, $\mathcal{CH}$ sets $State_{i,j}^m = completed$ and updates $Send^{list}$.
    - Else $\mathcal{CH}$ executes as follows.
      - If $\prod_{i,j}^m = \prod_{b,a}^l$, $\mathcal{CH}$ sets $r_{i,j}^m = \perp$, gets R_b from $Setup^{list}$ and replies with $\{ID_b, R_b, V\}$.
      - Else $\mathcal{CH}$ randomly chooses $r_{i,j}^m \in \mathbb{Z}_q^*$, obtains R_i from $Setup^{list}$ and replies with $\{ID_i, R_i, r_{i,j}^m P\}$.
      - Finally, $\mathcal{CH}$ updates $Send^{list}$, and updates $State_{i,j}^m$ to $completed$ if the newly generated message is the second message on the transcript.
  - SessionKeyReveal($\prod_{i,j}^m$): List R^{list} is of the form $(\prod_{i,j}^m, ID_{ini}, ID_{resp}, R_{ini}, R_{resp}, T_{ini}, T_{resp}, SK_{i,j}^m)$, where $ini \in \{i,j\}$ and $resp \in \{i,j\}$ denote the initiator and the responder of $\prod_{i,j}^m$, respectively.
    - $\mathcal{CH}$ receives $State_{i,j}^m$ from $Send^{list}$. If $State_{i,j}^m \neq completed$, $\mathcal{CH}$ returns $\perp$.
    - Else if $\prod_{i,j}^m = \prod_{a,b}^n$ or $\prod_{i,j}^m = \prod_{b,a}^l$, $\mathcal{CH}$ aborts.
    - Else if the session key $SK_{i,j}^m$ already exists, $\mathcal{CH}$ responds with $SK_{i,j}^m$.
    - Else $\mathcal{CH}$ obtains $\{ID_{ini}, R_{ini}, T_{ini}\}$ and $\{ID_{resp}, R_{resp}, T_{resp}\}$ from $Send^{list}$, and looks up ${H_2}^{list}$ to see if there is a tuple $(*, ID_{ini}, ID_{resp}, R_{ini}, R_{resp}, T_{ini}, T_{resp}, *)$. Then, if it exists, $\mathcal{CH}$ sees if K_1 and K_2 are produced correctly by validating $\text{DDH}(PK_{ini} + C_1 T_{ini}, PK_{resp} + C_3 T_{resp}, K_1) \stackrel{?}{=} 1$ and $\text{DDH}(PK_{ini} + C_3 T_{ini}, PK_{resp}$

$+C_4T_{resp}, K_2) \stackrel{?}{=} 1$, respectively, where $PK_{ini} = R_{ini} + H_1(ID_{ini}, R_{ini})P_{pub}$ and $PK_{resp} = R_{resp} + H_1(ID_{resp}, R_{resp})P_{pub}$. If both verifications pass, $\mathcal{CH}$ receives the corresponding h_2 and sets $SK_{i,j}^m \leftarrow h_2$. Otherwise (at least one verification fails or no such a tuple exists), $\mathcal{CH}$ picks $SK_{i,j}^m \in \{0,1\}^k$ at random. Finally, $\mathcal{CH}$ inserts the tuple $(\prod_{i,j}^m, ID_{ini}, ID_{resp}, R_{ini}, R_{resp}, T_{ini}, T_{resp}, SK_{i,j}^m)$ into R^{list} and returns $SK_{i,j}^m$.

- Test($\prod_{i,j}^m$): If $\prod_{i,j}^m = \prod_{a,b}^n$, $\mathcal{CH}$ picks $\xi \in \{0,1\}^k$ at random and sends ξ back to $\mathcal{M}$. Otherwise, $\mathcal{CH}$ aborts.

- **Analysis:** If $\mathcal{M}$ can successfully execute a forging attack in Strategy S1 with non-negligible probability, the following conditions should be met.

  (1) $\mathcal{CH}$ continues following the above simulation. If $\mathcal{M}$ chooses Strategy S1, with $\prod_{a,b}^n$ and $\prod_{b,a}^l$ as the test session and its corresponding matching session, respectively, this condition can be met.

  (2) For the test session $\prod_{a,b}^n$, adversary $\mathcal{M}$ must have conducted H_2 queries on the values $\{ID_a, ID_b, R_a, R_b, T_a, V, K_1, K_2\}$, where R_a and R_b are the public key materials of ID_a and ID_b picked by the challenger $\mathcal{CH}$, respectively, T_a and V are the outgoing messages of ID_a and ID_b picked by the challenger $\mathcal{CH}$, respectively, and K_1 and K_2 are correctly formed.

     * If $\prod_{a,b}^n$ is an initiator, the correct input of H_2 should be $(ID_a, ID_b, R_a, R_b, T_a, V, K_1, K_2)$, where $K_1 = (DLOG(U) + C_1 r_{a,b}^n)(R_b + h_b P_{pub} + C_2 V)$, $K_2 = (DLOG(U) + C_3 r_{a,b}^n)(R_b + h_b P_{pub} + C_4 V)$, and $h_b = H_1(ID_b, R_b)$.
     * If $\prod_{a,b}^n$ is a responder, the correct input of H_2 should be $(ID_b, ID_a, R_b, R_a, V, T_a, K_1, K_2)$, where $K_1 = (DLOG(U) + C_2 r_{a,b}^n)(R_b + h_b P_{pub} + C_1 V)$, $K_2 = (DLOG(U) + C_4 r_{a,b}^n)(R_b + h_b P_{pub} + C_3 V)$, and $h_b = H_1(ID_b, R_b)$.

  Finally, $\mathcal{CH}$ receives the item in $H_2{}^{list}$ and outputs $\mathrm{GDH}(U, V) = (C_2 - C_4)^{-1}\Big(K_1 - K_2 + r_{a,b}^n(C_3 - C_1)(R_b + h_b P_{pub}) + r_{a,b}^n(C_3C_4 - C_1C_2)V\Big)$ if $\prod_{a,b}^n$ is an initiator or $\mathrm{GDH}(U, V) = (C_1 - C_3)^{-1}\Big(K_1 - K_2 + r_{a,b}^n(C_4 - C_2)(R_b + h_b P_{pub}) + r_{a,b}^n(C_3C_4 - C_1C_2)V\Big)$ if $\prod_{a,b}^n$ is a responder by the knowledge of $r_{a,b}^n$. Note that since $C_i (i = 1, 2, 3, 4) \in \mathbb{Z}_q^*$ and $C_1 \neq C_3, C_2 \neq C_4$, the solution of $\mathrm{GDH}(U, V)$ is correct. The $\mathcal{CH}$ success rate is at least

$$Adv_{\mathcal{S}}^{GDH}(k) >= \frac{Adv_{\mathcal{M}}^{F}(k)}{6n_h(k)n_p(k)^2n_s(k)^2}.$$

As $Adv_{\mathcal{M}}^{F}(k)$ is non-negligible, $Adv_{\mathcal{S}}^{GDH}(k)$ can also be seen as non-negligible. Now, it derives the contradiction of the GDH assumption.

5.3.2. The Analysis of Strategy S2

In this subsection, we analyze strategy S2.

- **Setup:** $Setup^{list}$ is an initially empty list with $(ID_i, (d_i, R_i), PK_i)$. $\mathcal{CH}$ creates the system parameters and all parties' long-term private keys as follows.
  - $\mathcal{CH}$ picks $s \in \mathbb{Z}_q^*$ at random, computes $P_{pub} = sP$, and exposes system parameters $\langle E/\mathbb{F}_p, \mathbb{G}, q, P, P_{pub}, H_1, H_2 \rangle$. Thus, $\mathcal{CH}$ cannot obtain any information about KGC's master key.
  - For ID_i, $\mathcal{CH}$ sets the long-term private key (s_i, R_i), where $h_i, s_i \in_R \mathbb{Z}_q^*$, $R_i = s_iP - h_iP_{pub}$. Thus, $PK_i = R_i + h_iP_{pub} = s_iP$.
  - For every participant, $\mathcal{CH}$ transfers (ID_i, R_i) to $\mathcal{M}$, and stores the tuple $(ID_i, (d_i, R_i), PK_i)$ and (ID_i, R_i, h_i) in $Setup^{list}$ and $H_1{}^{list}$ (described later), respectively.

- **Queries:** $\mathcal{CH}$ maintains four lists $H_1{}^{list}$, $H_2{}^{list}$, $Send^{list}$, and R^{list} to store H_1, H_2, Send, and SessionKeyReveal oracles, respectively. $\mathcal{CH}$ performs the queries game with $\mathcal{M}$ as follows:
  - $H_1(ID_i, R_i)$, SessionKeyReveal($\prod_{i,j}^m$), Test($\prod_{i,j}^m$), and $H_2(ID_i, ID_j, R_i, R_j, T_i, T_j, K_1, K_2)$: These four queries are described in the same as those in Strategy S1.
  - Corrupt(ID_i): $\mathcal{CH}$ responds with (s_i, R_i).
  - KGCStaticKeyReveal: $\mathcal{CH}$ responds with s to $\mathcal{M}$.
  - EphemeralKeyReveal($\prod_{i,j}^m$): If $\prod_{i,j}^m = \prod_{a,b}^n$ or $\prod_{i,j}^m = \prod_{b,a}^l$, $\mathcal{CH}$ discontinues. Otherwise, $\mathcal{CH}$ provides the stored ephemeral key $r_{i,j}^m$ as the answer.
  - Send($\prod_{i,j}^m$,M): List $Send^{list}$ has $(\prod_{i,j}^m, tran_{i,j}^m, r_{i,j}^m, State_{i,j}^m)$, where $tran_{i,j}^m$, $r_{i,j}^m$ and $State_{i,j}^m$ are the transcript by now, the ephemeral secret key, and the state by now, respectively.
    - If M is the second message on the transcript, $\mathcal{CH}$ sets $State_{i,j}^m = completed$ and updates $Send^{list}$.
    - Else $\mathcal{CH}$ performs the following steps.
      - If $\prod_{i,j}^m = \prod_{a,b}^n$, $\mathcal{CH}$ sets $r_{i,j}^m = \perp$, receives R_a from $Setup^{list}$, and replies with $\{ID_a, R_a, U\}$.
      - Else If $\prod_{i,j}^m = \prod_{b,a}^l$, $\mathcal{CH}$ sets $r_{i,j}^m = \perp$, receives R_b from $Setup^{list}$, and replies with $\{ID_b, R_b, V\}$.
      - Else $\mathcal{CH}$ randomly chooses $r_{i,j}^m \in \mathbb{Z}_q^*$, obtains R_i from $Setup^{list}$, and replies with $\{ID_i, R_i, r_{i,j}^m P\}$.
      - Finally, $\mathcal{CH}$ updates $Send^{list}$, and updates $State_{i,j}^m$ to *completed* if the newly generated message is the second message on the transcript.
- **Analysis:** Here, we assume that $\prod_{a,b}^n$ is an initiator here. If $\mathcal{M}$ indeed chooses Strategy S2, $\prod_{a,b}^n$ and $\prod_{b,a}^l$ as the test session and its matching session, respectively, then $\mathcal{CH}$ continues this simulation. If $\mathcal{M}$ successfully executes the forging attack, it must have queried oracle $H_2(ID_a, ID_b, R_a, R_b, U, V, K_1, K_2)$, where R_a, R_b, U, and V are all picked by the challenger $\mathcal{CH}$, $K_1 = (s_a + C_1 DLOG(U))(R_b + h_b P_{pub} + C_2 V)$, $K_2 = (s_a + C_3 DLOG(U))(R_b + h_b P_{pub} + C_4 V)$, and $h_b = H_1(ID_b, R_b)$.
  Finally, $\mathcal{CH}$ receives the item in $H_2{}^{list}$, and outputs $\mathrm{GDH}(U, V) = (C_1 C_3 (C_2 - C_4))^{-1}\Big(C_3 K_1 - C_1 K_2 + s_a(C_1 - C_3)(R_b + h_b P_{pub}) + s_a(C_1 C_4 - C_2 C_3)V\Big)$ by the knowledge of s_a. Note that as $C_i (i = 1, 2, 3, 4) \in \mathbb{Z}_q^*$ and $C_1 \neq C_3, C_2 \neq C_4$, the solution of $\mathrm{GDH}(U, V)$ is correct. $\mathcal{CH}$'s success probability is at least

$$Adv_{\mathcal{S}}^{GDH}(k) >= \frac{Adv_{\mathcal{M}}^{F}(k)}{6 n_h(k) n_p(k)^2 n_s(k)^2}.$$

  As $Adv_{\mathcal{M}}^{F}(k)$ is non-negligible, $Adv_{\mathcal{S}}^{GDH}(k)$ can also be seen as also non-negligible. Now, it derives the contradiction of the GDH assumption.

5.3.3. The Analysis of Strategy S3

Here, we omit the detailed analysis of Strategy S3 as the analysis is almost the same as that for Strategy S1.

5.3.4. The Analysis of Strategy S4

In this subsection, we analyze strategy S4.

- **Setup:** $\mathcal{CH}$ initializes a list $Setup^{list}$ with $(ID_i, (d_i, R_i), PK_i)$. $\mathcal{CH}$ creates the system parameters and all parties' long-term private keys.

- $\mathcal{CH}$ picks $P_{pub} \in \mathbb{G}$ at random, and exposes $\langle E/\mathbb{F}_p, \mathbb{G}, q, P, P_{pub}, H_1, H_2 \rangle$. Thus, $\mathcal{CH}$ does not obtain any information about KGC's master key.
- For ID_a, $\mathcal{CH}$ sets the long-term private key $(\perp, R_a)$, where $h_a \in_R \mathbb{Z}_q^*$, $R_a = U - h_a P_{pub}$. Thus, $PK_a = R_a + h_a P_{pub} = U$.
- For ID_b, $\mathcal{CH}$ sets the long-term private key $(\perp, R_b)$, where $h_b \in_R \mathbb{Z}_q^*$, $R_b = V - h_b P_{pub}$. Thus, $PK_b = R_b + h_b P_{pub} = V$.
- For $ID_i (i \neq a, i \neq b)$, $\mathcal{CH}$ sets the long-term private key (s_i, R_i), where $h_i, s_i \in_R \mathbb{Z}_q^*$, $R_i = s_i P - h_i P_{pub}$. Thus, $PK_i = R_i + h_i P_{pub} = s_i P$.
- For every participant, $\mathcal{CH}$ transfers (ID_i, R_i) to $\mathcal{M}$, and stores $(ID_i, (d_i, R_i), PK_i)$ and (ID_i, R_i, h_i) in $Setup^{list}$ and ${H_1}^{list}$ (described later), respectively.

- **Queries:** $\mathcal{CH}$ maintains four lists ${H_1}^{list}$, ${H_2}^{list}$, $Send^{list}$, and R^{list}, which are initially empty and used for recording H_1, H_2, Send, and SessionKeyReveal oracles, respectively. $\mathcal{CH}$ performs the queries game with $\mathcal{M}$ as follows:
  - $H_1(ID_i, R_i)$, SessionKeyReveal($\prod_{i,j}^m$), Test($\prod_{i,j}^m$), $H_2(ID_i, ID_j, R_i, R_j, T_i, T_j, K_1, K_2)$, and KGCStaticKeyReveal: These five queries are described in the same way as those in Strategy S1.
  - Corrupt(ID_i): If $ID_i = ID_a$ or $ID_i = ID_b$, $\mathcal{CH}$ discontinues. Otherwise, $\mathcal{CH}$ responds with (s_i, R_i).
  - EphemeralKeyReveal($\prod_{i,j}^m$): $\mathcal{CH}$ responses with $r_{i,j}^m$.
  - Send($\prod_{i,j}^m$,M): List $Send^{list}$ is of the form $(\prod_{i,j}^m, tran_{i,j}^m, r_{i,j}^m, State_{i,j}^m)$, where $tran_{i,j}^m$, $r_{i,j}^m$, and $State_{i,j}^m$ are the transcript by now, the ephemeral secret key, and the state by now, respectively.
    - If M is the second message on the transcript, $\mathcal{CH}$ sets $State_{i,j}^m = completed$ and updates $Send^{list}$.
    - Else $\mathcal{CH}$ randomly chooses $r_{i,j}^m \in \mathbb{Z}_q^*$, obtains R_i from $Setup^{list}$, and replies with $\{ID_i, R_i, r_{i,j}^m P\}$. Then, $\mathcal{CH}$ updates $Send^{list}$, and updates $State_{i,j}^m$ to *completed* if the newly generated message is the second message on the transcript.
- **Analysis:** Here, we assume that $\prod_{a,b}^n$ is an initiator. If $\mathcal{M}$ selects Strategy S4, $\prod_{a,b}^n$ and $\prod_{b,a}^l$ as the test session and its matching session, then $\mathcal{CH}$ does not abort in the simulation. If $\mathcal{M}$ successfully performs the forging attack, it must have queried oracle $H_2(ID_a, ID_b, U - h_a P_{pub}, V - h_b P_{pub}, T_a, T_b, K_1, K_2)$, where T_a, T_b, U, and V are all picked by the challenger $\mathcal{CH}$, $K_1 = (DLOG(U) + C_1 r_{a,b}^n)(V + C_2 T_b)$, $K_2 = (DLOG(U) + C_3 r_{a,b}^n)(V + C_4 T_b)$, $h_a = H_1(ID_a, R_a)$, and $h_b = H_1(ID_b, R_b)$.

  Finally, $\mathcal{CH}$ receives the item in ${H_2}^{list}$, and outputs $\mathrm{GDH}(U, V) = (C_4 - C_2)^{-1}\Big(C_4 K_1 - C_2 K_2 + r_{a,b}^n (C_2 C_3 - C_1 C4) V + r_{a,b}^n C_2 C_4 (C_3 - C_1) T_b\Big)$ by the knowledge of $r_{a,b}^n$. Note that as $C_i (i = 1, 2, 3, 4) \in \mathbb{Z}_q^*$ and $C_1 \neq C_3, C_2 \neq C_4$, the solution of $\mathrm{GDH}(U, V)$ is correct. $\mathcal{CH}$'s success probability is at least

$$Adv_{\mathcal{S}}^{GDH}(k) >= \frac{Adv_{\mathcal{M}}^{F}(k)}{6 n_h(k) n_p(k)^2 n_s(k)^2}.$$

  As $Adv_{\mathcal{M}}^{F}(k)$ is non-negligible, $Adv_{\mathcal{S}}^{GDH}(k)$ can also be seen as non-negligible. Now, it derives the contradiction of the GDH assumption.

5.3.5. The Analysis of Strategy S5

$\prod_{a,b}^n$ has no matching session in strategy S5, thus at least one of ID_b's public key material R_b and ID_b's ephemeral private key is chosen by $\mathcal{M}$. If the adversary selects R_b themselves, then the change in R_b means the change in the ID_b's long-term private key s_b. Hence, a GDH instance cannot be embedded in the long-term private key in strategy S5.

- **Setup:** $Setup^{list}$ is an initially empty list and $(ID_i, (d_i, R_i), PK_i)$ is needed in this phase. $\mathcal{CH}$ creates the system parameters and all parties' long-term private keys.
  - $\mathcal{CH}$ sets V as the system public key P_{pub} and exposes the system parameters $\langle E/\mathbb{F}_p, \mathbb{G}, q, P, P_{pub}, H_1, H_2 \rangle$. Thus $\mathcal{CH}$ cannot know KGC's master key.
  - For ID_i, $\mathcal{CH}$ sets the long-term private key (s_i, R_i), where $h_i, s_i \in_R \mathbb{Z}_q^*$, $R_i = s_i P - h_i P_{pub}$. Thus, $PK_i = R_i + h_i P_{pub} = s_i P$.
  - For every participant, $\mathcal{CH}$ transfers (ID_i, R_i) to $\mathcal{M}$, and stores $(ID_i, (d_i, R_i), PK_i)$ and (ID_i, R_i, h_i) in $Setup^{list}$ and ${H_1}^{list}$ (described later), respectively.
- **Queries:** $\mathcal{CH}$ maintains four lists, ${H_1}^{list}$, ${H_2}^{list}$, $Send^{list}$, and R^{list} to store H_1, H_2, Send, and SessionKeyReveal oracles, respectively. $\mathcal{CH}$ performs the queries game with $\mathcal{M}$ as follows:
  - $H_1(ID_i, R_i)$, KGCStaticKeyReveal, Test($\prod_{i,j}^m$), and $H_2(ID_i, ID_j, R_i, R_j, T_i, T_j, K_1, K_2)$ are described the same as those in Strategy S1.
  - Corrupt(ID_i): If $ID_i = ID_b$, $\mathcal{CH}$ discontinues. Otherwise, $\mathcal{CH}$ responds with (s_i, R_i).
  - EphemeralKeyReveal($\prod_{i,j}^m$): If $\prod_{i,j}^m = \prod_{a,b}^n$, $\mathcal{CH}$ discontinues. Otherwise, $\mathcal{CH}$ provides the stored ephemeral key $r_{i,j}^m$ as the answer.
  - Send($\prod_{i,j}^m$,M): List $Send^{list}$ is with $(\prod_{i,j}^m, tran_{i,j}^m, r_{i,j}^m, State_{i,j}^m)$, where $tran_{i,j}^m$, $r_{i,j}^m$, and $State_{i,j}^m$ are the transcript by now, the ephemeral secret key, and the state by now, respectively.
    - If M is the second message on the transcript, $\mathcal{CH}$ sets $State_{i,j}^m = completed$ and updates $Send^{list}$.
    - Else $\mathcal{CH}$ performs the following steps.
      - If $\prod_{i,j}^m = \prod_{a,b}^n$, $\mathcal{CH}$ sets $r_{i,j}^m = \perp$, receives R_a from $Setup^{list}$, and replies with $\{ID_a, R_a, U\}$.
      - Else $\mathcal{CH}$ randomly chooses $r_{i,j}^m \in \mathbb{Z}_q^*$, obtains R_i from $Setup^{list}$, and replies with $\{ID_i, R_i, r_{i,j}^m P\}$.
      - Finally, $\mathcal{CH}$ updates $Send^{list}$, and updates $State_{i,j}^m$ to *completed* if the newly generated message is the second message on the transcript.
  - SessionKeyReveal($\prod_{i,j}^m$): This query is the same as that in Strategy S1, except that "Else if $\prod_{i,j}^m = \prod_{a,b}^n$ or $\prod_{i,j}^m = \prod_{b,a}^l$" is modified to "Else if $\prod_{i,j}^m = \prod_{a,b}^n$". This is because the matching session $\prod_{b,a}^l$ certainly exists in Strategy S1, while $\prod_{b,a}^l$ does not exist in Strategy S5.
- **Analysis:** Here, we assume that $\prod_{a,b}^n$ is an initiator. If $\mathcal{M}$ selects Strategy S5 and $\prod_{a,b}^n$ as the test session, then $\mathcal{CH}$ continues using the above simulation. If $\mathcal{M}$ successfully performs the forging attack with non-negligible probability, it should execute the H_2 query on $(ID_a, ID_b, R_a, R_b, U, T_b, K_1, K_2)$, where $K_1 = (s_a + C_1 DLOG(U))(R_b + h_b V + C_2 T_b)$, $K_2 = (s_a + C_3 DLOG(U))(R_b + h_b V + C_4 T_b)$, and $h_b = H_1(ID_b, R_b)$. Note that ID_a's public key material R_a and outgoing message U are both picked by the challenger $\mathcal{CH}$, and at least one of ID_b's public key material R_b and outgoing message T_b is chosen by $\mathcal{M}$.

  By the forking lemma [27], $\mathcal{CH}$ replays $\mathcal{M}$ with the same input and tossing coins. Here, $\mathcal{CH}$ only changes the query results of $H_1(ID_b, R_b)$, i.e., $\mathcal{CH}$ sets $H_1(ID_b, R_b)$ to $\overline{h_b}$, where $\overline{h_b} \in \mathbb{Z}_q^*$ and $\overline{h_b} \neq h_b$. Then, if $\mathcal{M}$ succeeds, it should perform a query on H_2 with $(ID_a, ID_b, R_a, R_b, U, T_b, \overline{K_1}, \overline{K_2})$, where $\overline{K_1} = (s_a + C_1 DLOG(U))(R_b + \overline{h_b} V + C_2 T_b)$, $\overline{K_2} = (s_a + C_3 DLOG(U))\ (R_b + \overline{h_b} V + C_4 T_b)$.

  Finally, $\mathcal{CH}$ receives the item in ${H_2}^{list}$, and outputs $\mathrm{GDH}(U, V) = C_1^{-1}\Big((h_b - \overline{h_b})^{-1}(K_1 - \overline{K_1}) - s_a V\Big)$ using the knowledge of s_a. Note that as $C_1 \in \mathbb{Z}_q^*$, the solution of $\mathrm{GDH}(U, V)$

is correct. Let λ be a factor from the forking lemma for Strategy S5. $\mathcal{CH}$'s success probability is at least

$$Adv_{\mathcal{S}}^{GDH}(k) >= \frac{\lambda Adv_{\mathcal{M}}^{F}(k)}{6n_h(k)^2 n_p(k)^2 n_s(k)}.$$

As $Adv_{\mathcal{M}}^{F}(k)$ is non-negligible, $Adv_{\mathcal{S}}^{GDH}(k)$ can also be seen as non-negligible. Now, it derives the contradiction of the GDH assumption.

5.3.6. The Analysis of Strategy S6

In this subsection, we will analyze strategy S6. A GDH instance cannot be embedded in the long-term private key in Strategy S6.

- **Setup:** $Setup^{list}$ is an initially empty list, and $(ID_i, (d_i, R_i), PK_i)$ is needed in this phase. $\mathcal{CH}$ creates the system parameters and all parties' long-term private keys.
  - $\mathcal{CH}$ sets V as the system public key P_{pub} and exposes $\langle E/\mathbb{F}_p, \mathbb{G}, q, P, P_{pub}, H_1, H_2 \rangle$. Thus, $\mathcal{CH}$ does not obtain any information about KGC's master key.
  - For ID_a, $\mathcal{CH}$ sets the long-term private key $(\perp, R_a)$, where $h_a \in_R \mathbb{Z}_q^*$, $R_a = U - h_a P_{pub}$. Thus, $PK_a = R_a + h_a P_{pub} = U$.
  - For $ID_i (i \neq a)$, $\mathcal{CH}$ sets (s_i, R_i) as the long-term private key, where $h_i, s_i \in_R \mathbb{Z}_q^*$, $R_i = s_i P - h_i P_{pub}$. Thus, $PK_i = R_i + h_i P_{pub} = s_i P$.
  - For every participant, $\mathcal{CH}$ transfers (ID_i, R_i) to $\mathcal{M}$, and stores $(ID_i, (d_i, R_i), PK_i)$ and (ID_i, R_i, h_i) in $Setup^{list}$ and ${H_1}^{list}$ (described later), respectively.
- **Queries:** $\mathcal{CH}$ maintains four lists, ${H_1}^{list}$, ${H_2}^{list}$, $Send^{list}$, and R^{list}, which are initially empty and used for recording H_1, H_2, Send, and SessionKeyReveal oracles, respectively. $\mathcal{CH}$ starts $\mathcal{M}$ by answering $\mathcal{M}$'s queries as follows:
  - The five queries $H_1(ID_i, R_i)$, SessionKeyReveal($\prod_{i,j}^{m}$), Test($\prod_{i,j}^{m}$), KGCStaticKeyRevea, and $H_2(ID_i, ID_j, R_i, R_j, T_i, T_j, K_1, K_2)$ are described in the same way as those in Strategy S5.
  - Corrupt(ID_i), EphemeralKeyReveal($\prod_{i,j}^{m}$) and Send($\prod_{i,j}^{m}$,M): These three queries are described to be the same as those in Strategy S4.
- **Analysis:** Here, we assume that $\prod_{a,b}^{n}$ is an initiator. If $\mathcal{M}$ selects Strategy S6 and $\prod_{a,b}^{n}$ as the test session, then $\mathcal{CH}$ continues this simulation. If $\mathcal{M}$ successfully performs the forging attack, it should execute H_2 query on $(ID_a, ID_b, R_a, R_b, T_a, T_b, K_1, K_2)$, where $K_1 = (DLOG(U) + C_1 r_{a,b}^n)(R_b + h_b V + C_2 T_b)$,$K_2 = (DLOG(U) + C_3 r_{a,b}^n)(R_b + h_b V + C_4 T_b)$, and $h_b = H_1(ID_b, R_b)$. Note that ID_a's public key material R_a and outgoing message T_a are both picked by the challenger $\mathcal{CH}$, and at least one of ID_b's public key material R_b and outgoing message T_b is chosen by $\mathcal{M}$.
  By the forking lemma [27], $\mathcal{CH}$ replays $\mathcal{M}$ with the same input and tossing coins. Here, $\mathcal{CH}$ only changes the query results of $H_1(ID_b, R_b)$, i.e., $\mathcal{CH}$ sets $H_1(ID_b, R_b)$ to $\overline{h_b}$, where $\overline{h_b} \in \mathbb{Z}_q^*$ and $\overline{h_b} \neq h_b$. Then, if $\mathcal{M}$ succeeds, it should perform a query on H_2 with $(ID_a, ID_b, R_a, R_b, T_a, T_b, \overline{K_1}, \overline{K_2})$, where $\overline{K_1} = (DLOG(U) + C_1 r_{a,b}^n)(R_b + \overline{h_b} V + C_2 T_b)$, $\overline{K_2} = (DLOG(U) + C_3 r_{a,b}^n)\ (R_b + \overline{h_b} V + C_4 T_b)$.
  Here, $\mathcal{CH}$ receives the item in ${H_2}^{list}$, and outputs GDH$(U, V) = (h_b - \overline{h_b})^{-1}(K_1 - \overline{K_1}) - C_1 r_{a,b}^n V$ using the knowledge of $r_{a,b}^n$. Note that as $\overline{h_b} \neq h_b$, the solution of GDH(U, V) is correct. Let λ be a factor from the forking lemma in Strategy S6. $\mathcal{CH}$'s success probability is at least

$$Adv_{\mathcal{S}}^{GDH}(k) >= \frac{\lambda Adv_{\mathcal{M}}^{F}(k)}{6n_h(k)^2 n_p(k)^2 n_s(k)}.$$

As $Adv_{\mathcal{M}}^{F}(k)$ is non-negligible, $Adv_{\mathcal{S}}^{GDH}(k)$ can also be seen as non-negligible. Now, it derives the contradiction of the GDH assumption.

The former formal security proof has proven that the proposed ID-AKA protocol $\sum_{C_1,C_2,C_3,C_4}$ is secure against some comment attacks of guessing attacks, key replication attacks, and forging attacks. Its security can be reduced to the hardness of GDH assumption over the elliptic curve group in the eCK model.

6. More Efficient Instantiations

As $C_i (i = 1, 2, 3, 4) \in \mathbb{Z}_q^*$, our construction needs six scalar multiplications (here, we ignore less time-consuming point additions and general hash function outputs), which is a bit higher than the NCL-16-II protocol [27] at the same security level. However, the NCL-16-II protocol is only a special protocol, while our construction will result in different special protocols with different C_i values, for example, protocol $\sum_{1,1,-1,-1}$ and protocol $\sum_{1,1,H_1(R_A||R_B||T_A||T_B),H_1(R_B||R_A||T_B||T_A)}$. How should the values of C_1, C_2, C_3, and C_4 be chosen in the real execution environment? It would be better to select values that result in more efficient instantiation as different protocols have different computation costs. The following provides some efficient instantiations of our construction.

Protocol 1 ($\sum_{1,1,-1,-1}$). *In this protocol,* $C_1 = C_2 = 1, C_3 = C_4 = -1$. *A computes the shared secrets* $K_{AB}^1 = (s_A + e_A)(PK_B + T_B)$ *and* $K_{AB}^2 = (s_A - e_A)(PK_B - T_B)$. *B compute the shared secrets* $K_{BA}^1 = (s_B + e_B)(PK_A + T_A)$ *and* $K_{BA}^2 = (s_B - e_B)(PK_A - T_A)$. *This protocol reduces two scalar multiplications compared with the general construction.*

Protocol 2 ($\sum_{1,-1,-1,1}$). *In this protocol,* $C_1 = C_4 = 1, C_2 = C_3 = -1$. *A computes the shared secrets* $K_{AB}^1 = (s_A + e_A)(PK_B - T_B)$ *and* $K_{AB}^2 = (s_A - e_A)(PK_B + T_B)$. *B computes the shared secrets* $K_{BA}^1 = (s_B - e_B)(PK_A + T_A)$ *and* $K_{BA}^2 = (s_B + e_B)(PK_A - T_A)$. *This protocol has the same efficiency as* $\sum_{1,1,-1,-1}$.

Protocol 3 ($\sum_{1,1,2,2}$). *In this protocol,* $C_1 = C_2 = 1, C_3 = C_4 = 2$. *A computes the shared secrets* $K_{AB}^1 = (s_A + e_A)(PK_B + T_B)$ *and* $K_{AB}^2 = (s_A + 2e_A)(PK_B + 2T_B)$. *B computes the shared secrets* $K_{BA}^1 = (s_B + e_B)(PK_A + T_A)$ *and* $K_{BA}^2 = (s_B + 2e_B)(PK_A + 2T_A)$. *This protocol only adds a point addition operation compared with the protocol of* $\sum_{1,1,-1,-1}$.

7. Performance and Comparison

This section presents the efficiency and security comparison between our Protocols 1 and 2 with other competitive ID-AKA protocols. Note that only the HC protocols [13] were pairings-based ID-AKA protocols, the other ID-AKA protocols [21–28] and ours were all pairing-free.

7.1. Comparison of Computation Overheads

To evaluate the computational overhead, Table 2 lists the same execution time of different cryptographic operations, reported in [40]. The execution time was calculated using the MIRACL library on a Samsung Galaxy S5 smartphone, equipped with a 2.5 GHz ARM Krait processor with 2GB RAM memory running the Android 4.4.2 operating system.

Table 2. Execution time on a Samsung Galaxy S5.

Notation	Explanation (The Execution Time of)	Time (ms)
T_p	A bilinear pairing e: $\mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$	32.713
T_{psm}	A pairing-based scalar multiplication in $\mathbb{G}_1$	13.405
T_{ppa}	A pairing-based point addition in $\mathbb{G}_1$	0.056
T_{pexp}	An exponentiation operation in $\mathbb{G}_2$	2.249
T_{mtph}	A hash-to-point in $\mathbb{G}_1$	33.582
T_{esm}	An ECC-based scalar multiplication in $\mathbb{G}$	3.350
T_{epa}	An ECC-based point addition in $\mathbb{G}$	0.014
T_{emtph}	A hash-to-point in $\mathbb{G}$	8.250
T_h	A general hash function	0.006

Next, the total execution times of these two protocols and the competitive ID-AKA protocols [13,21–28] were computed, which are shown in Table 3. In our Protocols 1 and 2, to agree on a session key, each party needed to compute four ECC-based scalar multiplications, three ECC-based point additions, and two general hash function outputs. Therefore, the total computation time at each party was about $4T_{esm} + 3T_{epa} + 2T_h \approx 13.454$ ms. Similarly, the communication costs of protocols in [13,21,24–28] were computed. In protocols ZHWY-19 [23] and NIKE [22], two parties had unbalanced computation costs, i.e., one party had a lower computation cost than the other party. Here, we adopted the lower computation cost of one party. According to Table 2, our Protocols 1 and 2 were nearly 80% of protocols NCL-16-II [27] and CKD [24], 100% of protocols [25] and [21,23,28], 72% of protocol XW [26], and 8% of the HC protocol [13] with relation to the computation cost. That is to say, our Protocols 1 and 2 almost had the lowest computation cost. The comparison results are shown in Figure 6.

Table 3. Comparative computation overheads.

Protocols	Computations	Computation Cost (ms)	Energy Consumption (mj)
PF-ID-2PAKA [21]	$4T_{esm} + 2T_{epa} + 2T_h$	13.44	322.56
DXCLCZF-18 [28]	$4T_{esm} + 2T_{epa} + 2T_h$	13.44	322.56
NIKE [22]	$2T_{esm} + 3T_{emtph}$	31.45	754.8
ZHWY-19 [23]	$4T_{esm} + 3T_{epa} + 3T_h$	13.46	323.04
NCL-16-II [27]	$5T_{esm} + 4T_{epa} + 3T_h$	16.824	403.776
DRS-20 [30]	$5T_{esm} + 3T_{epa} + 5T_h$	16.822	403.728
DSH-21 [29]	$4T_{esm} + 3T_{epa} + 3T_h$	13.460	323.04
PWCAS-22 [33]	$4T_{esm} + 6T_{epa} + 5T_h$	13.514	324.336
ZHVLH-23 [37]	$5T_{esm} + T_{epa} + 16T_h$	16.860	404.64
Our protocols	$4T_{esm} + 3T_{epa} + 2T_h$	13.454	322.896

Energy consumption is one essential item for IoT communication, and the energy consumption of ID-AKA protocol decides the energy efficiency of IoT communication as it is executed by the IoT device. As shown in the former comparative computation overheads in Table 3, the computation costs were calculated. To compute the energy consumption of these key agreement algorithms, the IoT devices equipped with 3.0 V and 8.0 mA were selected. This parameter was set according to the power level of MICA 2 [41]. For the proposed ID-AKA protocol, the energy consumption was $3.0 * 8.0 * 13.454 = 322.896$ mj, and the comparison results with similar protocols are shown in Figure 7. Therefore, the low computation overheads led to low energy consumption, and the proposed ID-AKA protocol had more advantages than similar protocols in relation to the costs of computation and energy.

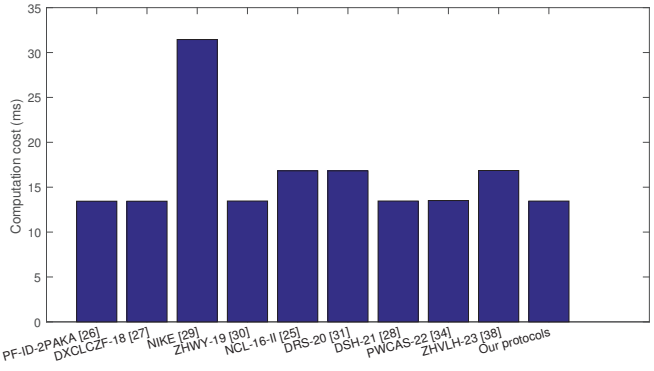


Figure 6. Comparison of computation overheads.

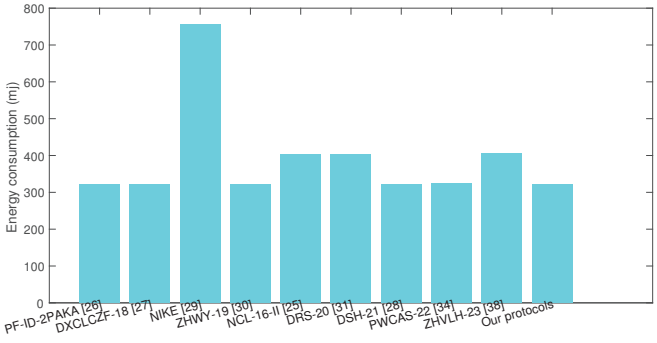


Figure 7. Comparison of energy consumption.

7.2. Comparison of Communication Overheads

Let $|\mathbb{G}_1|$, $|\mathbb{G}_2|$, $|\mathbb{G}|$, and $|\mathbb{Z}_q^*|$ represent elements sizes of $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}$, and $\mathbb{Z}_q^*$, respectively. Furthermore, assume $|ID|$ and $|H|$ represent the length of an identifier and a general hash output, respectively. Considering the Ate pairing and elliptic curves, $|\mathbb{G}_1|$, $|\mathbb{G}_2|$, $|\mathbb{G}|$, $|\mathbb{Z}_q^*|$, and $|H|$ are 1024, 1024, 320, 160, and 160 bits, respectively. We assumed $|ID|$ is 32 bits in length.

Table 4 demonstrates the communication cost comparison of the key agreement phase. Note that in our Protocol 1 (Protocol 2), party A sends $\{ID_A, R_A, T_A\}$ to party B, where $R_A, T_A \in \mathbb{G}$ and ID_A is the identity of A. Party B symmetrically sends $\{ID_B, R_B, T_B\}$ to party A, where $R_B, T_B \in \mathbb{G}$ and ID_B is the identity of B. Therefore, the communication cost of our protocol 1 (protocol 2) is $2 * (|ID|+2|\mathbb{G}|) = 2 * (32 + 2 * 320) = 1344$ bits. The results show that Protocols 1 and 2 have the lowest communication cost.

Table 4. Comparative communication overheads.

Protocols	Messages No.	Communication Cost	Cost (bits)
PF-ID-2PAKA [21]	2	$2\|ID\| + 4\|\mathbb{G}\|$	1344
DXCLCZF-18 [28]	2	$2\|ID\| + 6\|\mathbb{G}\|$	1984
ZHWY-19 [23]	3	$6\|\mathbb{G}\| + \|\mathbb{Z}_q^*\| + 2\|H\|$	2400
NIKE [22]	3	$2\|ID\| + 5\|\mathbb{G}\|$	1664
NCL-16-II [27]	2	$2\|ID\| + 6\|\mathbb{G}\|$	1984
DRS-20 [30]	2	$2\|ID\| + 4\|\mathbb{G}\| + \|H\|$	1504
DSH-21 [29]	2	$2\|ID\| + 4\|\mathbb{G}\|$	1344
PWCAS-22 [33]	2	$2\|ID\| + 4\|\mathbb{G}\| + 2\|H\|$	1664
ZHVLH-23 [37]	2	$3 * (2\|ID\| + \|\mathbb{G}\| + \|H\|)$	1632
Our protocols	2	$2\|ID\| + 4\|\mathbb{G}\|$	1344

7.3. Security Comparisons

As shown in Table 1, some related ID-AKA protocols can capture other security attributes, including known key security, no key control, resistance to basic impersonation attacks, replay attack resilience, resistance to man-in-the-middle attacks, and unknown key share resilience. But, for the proposed IA-AKA protocols, we did not consider explicit mutual authentication, as it can be easily achieved for all one-round protocols [13,21,24–28] by adding a key confirmation. Here, the protocol PWCAS-22 [33] is based on physical unclonable function (PUF), and ZHVLH-23 [37] is based on pseudo-random permutation (PRF). The HC protocol [13], the NCL-16-II protocol [27], the DRS-20 protocol [30], and our protocols are provably secure in the eCK security model. But events in the security proof of NCL-16-II [27] are not complementary, which mismatches the freshness definition. Table 3 shows that our Protocols 1 and 2 can reach the best computation efficiency.

Compared with similar ID-AKA protocols, the proposed Protocols 1 and 2 presented the lowest computation and communication overheads, which could improve IoT com-

munication efficiency in IoT applications. Meanwhile, with the increase in the number of devices, these ID-AKA protocols could also maintain high efficiency, as the key agreement process was executed between two different IoT users. The key agreement between the two parties was less affected by the number of devices in IoT applications and only affected by the hardware, software, and communication protocol in the public internet environment. Although the key agreement times will increase more, this can be ignored with the increasing IoT computation ability.

8. Conclusions

This paper first reviews several ID-AKA protocols without pairings in terms of security and efficiency. We carefully studied them and pointed out the security weaknesses against the ephemeral key reveal attack, key compromise impersonation attack, and launch collusion attack. We also proposed a family of ID-AKA protocols without pairings and proven the security in the eCK security model, a widely accepted security model for AKA protocols. Six strategy analyses were provided, and these ID-AKA protocols were proven to be secure in the eCK model based on the GDH assumption over the elliptic curve group. Then, the instantiations, performance and comparison were presented, and the results show that the proposed ID-AKA protocols were more efficient than other protocols in similar literature. In addition, these ID-AKA protocols no only improved communication security and efficiency in IoT applications, but also saved energy consumption for the communication process.

In the future, with the increasing amount of IoT devices, some security issues of identity authentication, data fine-grained access control, and user privacy protection should still be taken into consideration. Especially with the development of quantum computers and quantum computation, the anti-quantum attack security ID-AKA protocol will be a hot research direction. Meanwhile, many customized ID-AKA schemes should be designed to meet the special requirements of future IoT applications.

Author Contributions: Methodology, H.S. and C.L.; Validation, W.H.; Formal analysis, J.Z.; Investigation, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China (62072416), the Key Research and Development Special Project of Henan Province (221111210500), the Science and Technology Program of Henan Province (212102210107, 232102210125), the Doctor Scientific Research Fund of Zhengzhou University of Light Industry (2021BSJJ033) and the Foundation of State Key Laboratory of Public Big Data (No. PBD2023-25).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Khan, M.A.; Din, I.U.; Majali, T.E.; Kim, B.S. A survey of authentication in Internet of things-enabled healthcare systems. *Sensors* **2022**, *22*, 9089. [CrossRef]
2. Jayabalasamy, G.; Koppu, S. High-performance Edwards curve aggregate signature (HECAS) for nonrepudiation in IoT-based applications built on the blockchain ecosystem. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 9677–9687. [CrossRef]
3. Li, W.; Li, R.; Wu, K.; Cheng, R.; Su, L.; Cui, W. Design and implementation of an SM2-based security authentication scheme with the key agreement for smart grid communications. *IEEE Access* **2018**,*6*, 71194–71207. [CrossRef]
4. Pu, L.; Lin, C.; Chen, B.; He, D. User-friendly public-key authenticated encryption with keyword search for industrial Internet of things. *IEEE Internet Things J.* **2023**, *10* 13544–13555. [CrossRef]
5. Rasori, M.; La Manna, M.; Perazzo, P.; Dini, G. A survey on attribute-based encryption schemes suitable for the Internet of things. *IEEE Internet Things J.* **2022**, *9*, 8269–8290. [CrossRef]
6. Onyema, E.M.; Kumar, M.A.; Balasubaramanian, S.; Bharany, S.; Rehman, A.U.; Eldin, E.T.; Shafiq, M. A security policy protocol for detection and prevention of internet control message protocol attacks in software defined networks. *Sustainability* **2022**, *14*, 11950. [CrossRef]
7. Alam, S.; Shuaib, M.; Ahmad, S.; Jayakody, D.N.K.; Muthanna, A.; Bharany, S.; Elgendy, I.A. Blockchain-based solutions supporting reliable healthcare for fog computing and Internet of medical things (IoMT) integration. *Sustainability* **2022**, *14*, 15312. [CrossRef]

8. Sun, F.; He, S.; Zhang, X.; Zhang, J.; Li, Q.; He, Y. A fully authenticated Diffie-Hellman protocol and its application in WSNs. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 1986–1999. [CrossRef]
9. Shamir, A. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology: Proceedings of CRYPTO 84 4*; Springer: Berlin/Heidelberg, Germany, 1985; pp. 47–53.
10. Smart, N.P. Identity-based authenticated key agreement protocol based on Weil pairing. *Electron. Lett.* **2002**, *38*, 630–632. [CrossRef]
11. Wang, S.; Cao, Z.; Choo, K.K.R.; Wang, L. An improved identity-based key agreement protocol and its security proof. *Inf. Sci.* **2009**, *179*, 307–318. [CrossRef]
12. Chen, L.; Cheng, Z.; Smart, N.P. Identity-based key agreement protocols from pairings. *Int. J. Inf. Secur.* **2007**, *6*, 213–241. [CrossRef]
13. Huang, H.; Cao, Z. An ID-based authenticated key exchange protocol based on bilinear Diffie-Hellman problem. In Proceedings of the 4th International Symposium on Information, Computer, and Communications Security, Sydney, Australia, 10–12 March 2009; pp. 333–342.
14. Choo, K.K.R.; Nam, J.; Won, D. A mechanical approach to derive identity-based protocols from Diffie-Hellman-based protocols. *Inf. Sci.* **2014**, *281*, 182–200. [CrossRef]
15. Wu, L.; Wang, J.; Choo, K.K.R.; Li, Y.; He, D. An efficient provably-secure identity-based authentication scheme using bilinear pairings for Ad hoc network. *J. Inf. Secur. Appl.* **2017**, *37*, 112–121. [CrossRef]
16. Odelu, V.; Das, A.K.; Wazid, M.; Conti, M. Provably secure authenticated key agreement scheme for smart grid. *IEEE Trans. Smart Grid* **2016**, *9*, 1900–1910. [CrossRef]
17. Gupta, D.S.; Islam, S.H.; Obaidat, M.S.; Vijayakumar, P.; Kumar, N.; Park, Y. A provably secure and lightweight identity-based two-party authenticated key agreement protocol for IIoT environments. *IEEE Syst. J.* **2020**, *15*, 1732–1741. [CrossRef]
18. Lian, H.; Pan, T.; Wang, H.; Zhao, Y. Identity-Based Identity-Concealed Authenticated Key Exchange. In *Computer Security-ESORICS 2021: 26th European Symposium on Research in Computer Security, Darmstadt, Germany, 4–8 October 2021*; Proceedings, Part II 26; Springer International Publishing: Berlin/Heidelberg, Germany, 2021; pp. 651–675.
19. Canetti, R.; Krawczyk, H. Analysis of key-exchange protocols and their use for building secure channels. In Proceedings of the International conference on the theory and applications of cryptographic techniques, Innsbruck, Austria, 6–10 May 2001; Springer: Berlin/Heidelberg, Germany, 2001; pp. 453–474.
20. LaMacchia, B.; Lauter, K.; Mityagin, A. Stronger security of authenticated key exchange. In Proceedings of the International Conference on Provable Security, Wollongong, Australia, 1–2 November 2007; Springer: Berlin/Heidelberg, Germany, 2007; pp. 1–16.
21. Bala, S.; Sharma, G.; Verma, A.K. PF-ID-2PAKA: Pairing free identity-based two-party authenticated key agreement protocol for wireless sensor networks. *Wirel. Pers. Commun.* **2016**, *87*, 995–1012. [CrossRef]
22. Mohammadali, A.; Haghighi, M.S.; Tadayon, M.H.; Mohammadi-Nodooshan, A. A novel identity-based key establishment method for advanced metering infrastructure in smart grid. *IEEE Trans. Smart Grid* **2016**, *9*, 2834–2842. [CrossRef]
23. Zhang, J.; Huang, X.; Wang, W.; Yue, Y. Unbalancing pairing-free identity-based authenticated key exchange protocols for disaster scenarios. *IEEE Internet Things J.* **2018**, *6*, 878–890. [CrossRef]
24. Cao, X.; Kou, W.; Du, X. A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges. *Inf. Sci.* **2010**, *180*, 2895–2903. [CrossRef]
25. Fiore, D.; Gennaro, R. Making the Diffie-Hellman protocol identity-based. In *Topics in Cryptology-CT-RSA 2010: The Cryptographers' Track at the RSA Conference 2010, San Francisco, CA, USA, 1–5 March 2010*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 165–178.
26. Xie, M.; Wang, L. One-round identity-based key exchange with perfect forward security. *Inf. Process. Lett.* **2012**, *112*, 587–591. [CrossRef]
27. Ni, L.; Chen, G.; Li, J.; Hao, Y. Strongly secure identity-based authenticated key agreement protocols without bilinear pairings. *Inf. Sci.* **2016**, *367*, 176–193. [CrossRef]
28. Dang, L.; Xu, J.; Cao, X.; Li, H.; Chen, J.; Zhang, Y.; Fu, X. Efficient identity-based authenticated key agreement protocol with provable security for vehicular ad hoc networks. *Int. J. Distrib. Sens. Netw.* **2018**, *14*, 1550147718772545. [CrossRef]
29. Deng, L.; Shao, J.; Hu, Z. Identity based two-party authenticated key agreement scheme for vehicular ad hoc networks. *Peer-to-Peer Netw. Appl.* **2021**, *14*, 2236–2247. [CrossRef]
30. Daniel, R.M.; Rajsingh, E.B.; Silas, S. An efficient ECK secure identity based two party authenticated key agreement scheme with security against active adversaries. *Inf. Comput.* **2020**, *275*, 104630. [CrossRef]
31. Kumar, M.; Chand, S. A lightweight cloud-assisted identity-based anonymous authentication and key agreement protocol for secure wireless body area network. *IEEE Syst. J.* **2020**, *15*, 2779–2786. [CrossRef]
32. Rakeei, M.A.; Moazami, F. Cryptanalysis of an anonymous authentication and key agreement protocol for secure wireless body area network. *Cryptol. ePrint Arch.* **2020**, 1–4.
33. Pu, C.; Wall, A.; Choo, K.K.R.; Ahmed, I.; Lim, S. A lightweight and privacy-preserving mutual authentication and key agreement protocol for Internet of Drones environment. *IEEE Internet Things J.* **2022**, *9*, 9918–9933. [CrossRef]
34. Zhang, Q.; Zhu, L.; Li, Y.; Ma, Z.; Yuan, J.; Zheng, J.; Ai, S. A group key agreement protocol for intelligent internet of things system. *Int. J. Intell. Syst.* **2022**, *37*, 699–722. [CrossRef]

35. Zhou, T.; Wang, C.; Zheng, W.; Tan, H. Secure and efficient authenticated group key agreement protocol for AI-based automation systems. *ISA Trans.* **2023**, *141*, 1–9. [CrossRef]
36. Pan, X.; Jin, Y.; Li, F. An efficient heterogeneous authenticated key agreement scheme for unmanned aerial vehicles. *J. Syst. Archit.* **2023**, *136*, 102821. [CrossRef]
37. Zhang, Y.; He, D.; Vijayakumar, P.; Luo, M.; Huang, X. SAPFS: An Efficient Symmetric-Key Authentication Key Agreement Scheme with Perfect Forward Secrecy for Industrial Internet of Things. *IEEE Internet Things J.* **2023**, *10*, 9716–9726. [CrossRef]
38. Abdussami, M.; Amin, R.; Vollala, S. Provably secured lightweight authenticated key agreement protocol for modern health industry. *Ad Hoc Netw.* **2023**, *141*, 103094. [CrossRef]
39. Cheng, Q.; Li, Y.; Jiang, Q.; Li, X. Security Analysis of Two Unbalancing Pairing-free Identity-based Authenticated Key Exchange Protocols. *Int. J. Netw. Secur.* **2020**, *22*, 597–601.
40. He, D.; Wang, H.; Khan, M.K.; Wang, L. Lightweight anonymous key distribution scheme for smart grid using elliptic curve cryptography. *IET Commun.* **2016**, *10*, 1795–1802. [CrossRef]
41. Gura, N.; Patel, A.; Wander, A.; Eberle, H.; Shantz, S.C. Comparing elliptic curve cryptography and RSA on 8-bit CPUs. In *Cryptographic Hardware and Embedded Systems-CHES 2004: 6th International Workshop Cambridge, MA, USA, 11–13 August 2004*; Proceedings 6; Springer: Berlin/Heidelberg, Germany, 2004; pp. 119–132.

sensors

MDPI

Article

Provably Secure Lightweight Mutual Authentication and Key Agreement Scheme for Cloud-Based IoT Environments

Sieun Ju and Yohan Park *

Department of Computer Engineering, College of Engineering, Keimyung University, Daegu 42601, Republic of Korea; juse0204@gmail.com
* Correspondence: yhpark@kmu.ac.kr; Tel.: +82-53-580-5229

Abstract: A paradigm that combines cloud computing and the Internet of Things (IoT) allows for more impressive services to be provided to users while addressing storage and computational resource issues in the IoT environments. This cloud-based IoT environment has been used in various industries, including public services, for quite some time, and has been researched in academia. However, various security issues can arise during the communication between IoT devices and cloud servers, because communication between devices occurs in open channels. Moreover, issues such as theft of a user's IoT device or extraction of key parameters from the user's device in a remote location can arise. Researchers interested in these issues have proposed lightweight mutual authentication key agreement protocols that are safe and suitable for IoT environments. Recently, a lightweight authentication scheme between IoT devices and cloud servers has been presented. However, we found out their scheme had various security vulnerabilities, vulnerable to insider, impersonation, verification table leakage, and privileged insider attacks, and did not provide users with untraceability. To address these flaws, we propose a provably secure lightweight authentication scheme. The proposed scheme uses the user's biometric information and the cloud server's secret key to prevent the exposure of key parameters. Additionally, it ensures low computational costs for providing users with real-time and fast services using only exclusive OR operations and hash functions in the IoT environments. To analyze the safety of the proposed scheme, we use informal security analysis, Burrows–Abadi–Needham (BAN) logic and a Real-or-Random (RoR) model. The analysis results confirm that our scheme is secure against insider attacks, impersonation attacks, stolen verifier attacks, and so on; furthermore, it provides additional security elements. Simultaneously, it has been verified to possess enhanced communication costs, and total bit size has been shortened to 3776 bits, which is improved by almost 6% compared to Wu et al.'s scheme. Therefore, we demonstrate that the proposed scheme is suitable for cloud-based IoT environments.

Keywords: cloud computing; internet of things; authentication; cryptanalysis; real-or-random model; Burrow–Abadi–Needham logic

Citation: Ju, S.; Park, Y. Provably Secure Lightweight Mutual Authentication and Key Agreement Scheme for Cloud-Based IoT Environments. *Sensors* **2023**, *23*, 9766. https://doi.org/10.3390/s23249766

Academic Editor: Jian Li

Received: 10 November 2023
Revised: 1 December 2023
Accepted: 8 December 2023
Published: 11 December 2023

1. Introduction

The Internet of Things (IoT) is a network in which Internet-enabled objects interact with each other through the internet [1,2]. IoT objects collect data from their surroundings, provide web services to users, and communicate with each other. Therefore, IoT objects such as smart devices need significant resources to store data collected from sensors and perform real-time computations using limited hardware. Hence, addressing the limitations of storage and computing capacities is crucial for the formation of a network of IoT objects [3–5]. However, cloud computing technology refers to the practice of moving computational power and storage space from individual devices to larger shared data centers [6]. Cloud computing allows access to a shared pool of computing resources such as networks, servers, storage, and applications. By using cloud computing, it becomes possible to overcome the limitations inherent in IoT devices [7–10]. The development

and discussion of cloud-based IoT (CloudIoT) have been ongoing since before 2008 and continue to evolve. Figure 1 illustrates the structure of CloudIoT. This structure comprises three entities: user, cloud server, and control server. Users with IoT devices can access the resources provided by the cloud service provider's server anytime and anywhere through IoT objects. The cloud server collects user's requests and delivers the right service through IoT. The control server, acting as a trusted entity, generates the necessary parameters for communication between authenticated users and the cloud server through the registration process. Additionally, it monitors the key agreement phase to ensure that users and the cloud server establish the same session key for subsequent communications when needed.

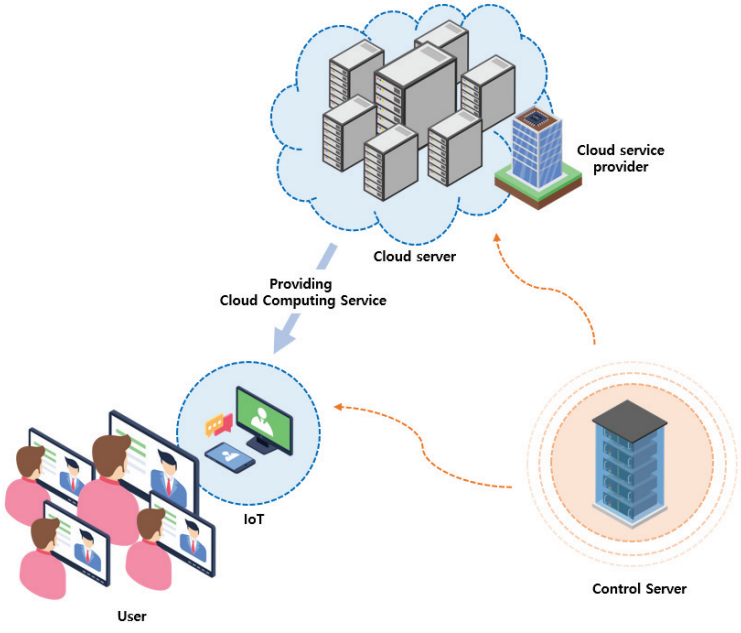


Figure 1. Cloud-based IoT enviroment.

In 2022, Wu et al. [11] proposed a lightweight authentication protocol for IoT-enabled cloud computing environments. The authors argued that their scheme could resist various attacks, such as man-in-the-middle, insider, DDoS, and masquerade attacks, and provides privacy, traceability, and integrity. However, we identified several vulnerabilities in Wu et al.'s scheme, including susceptibility to insider attacks, verification table attacks, user impersonation, and cloud server impersonation. Furthermore, the scheme lacks user untraceability, allowing an attacker to track the same user across different sessions through message eavesdropping alone. To address these vulnerabilities, we propose a provably secure lightweight mutual authentication and key agreement (MAKA) scheme. In our proposed scheme, we protect crucial parameters stored in the user's IoT smart card using the user's biometric information to prevent attacks like user impersonation and offline password-guessing. We also enhanced security by adding a secret key to the cloud server, preventing attackers from exploiting leaked database values. Additionally, we reduced the communication and computation overhead by employing only hash functions and exclusive-OR operations.

1.1. Research Contributions

We review and conduct a security analysis of Wu et al.'s authentication scheme. We demonstrate that Wu et al.'s scheme is vulnerable to insider attacks, verification table leakage attacks privileged insider attacks, user impersonation, and cloud server imperson-

ation. Additionally, we propose an MAKA for cloud-based IoT environments that leverages biometric information. The proposed scheme is tailored to the IoT environments, using only exclusive OR operations and hash functions to align with a lightweight architecture. Additionally, we use a Real-or-Random (RoR) model and Burrow–Abadi–Needham (BAN) logic to demonstrate formally the security and robustness of the proposed. Moreover, we substantiate the security of our scheme against different attacks, including insider attacks, impersonation attacks, reply and man-in-the-middle (MITM) attacks, privileged insider attacks, ephemeral security leakage, stolen verifier attacks, DoS attacks, and session key disclosure attacks. In addition, we confirmed that our scheme can provide user anonymity, user untraceability, perfect forward secrecy, and mutual authentication. Last, we evaluate the security features, communication costs, and computation costs of the proposed scheme with related schemes, including Wu et al.'s.

1.2. Organization

In Section 2, we introduce studies related to cloud-based IoT, IoT, and cloud computing. We present the system model and adversary model used in our proposed scheme in Section 3. Following that, we discuss Wu et al.'s scheme in Section 4. We then delve into the vulnerabilities we identified in Wu et al.'s scheme in Section 5. In Section 6, we introduce our proposed scheme, and in Section 7, we provide security analyses using tools such as BAN logic, and RoR model. Performance analyses, including security features, communication, and computation costs, are presented in Section 8. Finally, in Section 9, we conclude our paper and outline future plans.

2. Related Works

When providing services to users over the internet, application security is crucial in gaining user trust. To access various services, including storage services provided by cloud service providers, the environments should be well prepared to handle various attacks and security threats that may exist. Furthermore, in IoT environments, lightweight protocol computations are essential to provide users with a seamless real-time service anytime, anywhere. In the following sections, we will review the authentication protocols in the existing cloud-based IoT environments.

In 2019, Schouqi et al. [12] introduced an authentication protocol for IoT built on Nikooghadam et al.'s [13] protocol. The protocol of Nikooghadam et al. was developed as a responses to issues with the authentication protocol proposed by Kumari et al. [14]. However, Nikooghadam et al.'s scheme has already been analyzed by researchers in the field, including Limbasiya et al., Chandrakar-Om, and Sharma-Kalra [15–17]. These researchers raised concerns about its security, highlighting vulnerabilities to various attacks such as password-guessing, insiders, and modification attacks. They also indicated that the protocol lacked forward secrecy and did not provide session key verification and a biometric update phase. The author of the new scheme reviewed the security issues known in Nikooghadam et al.'s protocol and proposed enhancements based on these findings.

Prosanta and Biplab (Prosanta-Biplab) [18] proposed lightweight two-factor authentication scheme for IoT devices in 2019. They argued that two-factor authentication schemes that use a passwords and smartcards, often vulnerable to physical attacks. To overcome these security issues they suggested physically uncloneable functions(PUF) as an authentication factor for IoT devices. However, in 2020, Siddiqui et al. [19] demonstrated that the scheme is vulnerable to man-in-the-middle, impersonations, session- hijacking and conventional and differential template attacks.

In 2019, Zhou et al. [20] presented a lightweight two-factor authentication scheme for IoT devices available in the cloud environments. In the same year, Rafael et al. [21] indicated that Zhou et al.'s scheme has several security issues. Rafael et al. demonstrated that Zhou et al.'s scheme failed to provide mutual authentication, was unsuccessful in protecting the secret key, and was vulnerable to various attacks, including insider attacks and man-in-the-middle attacks.

In 2020, Alzahrani et al. [22] presented an authentication protocol for IoT environments based on self-certified public keys and elliptic curve cryptography (ECC). Alzahrani conducted research on protocols proposed by Islam-Biswas [23] and Mandal et al. [24], highlighting their failure to ensure user anonymity and vulnerability to impersonation attacks. Therefore, the author developed a protocol that guarantees anonymity among connected devices and addresses security vulnerabilities. However, this scheme does not guarantee security against physical attacks.

Chen et al. [25] proposed a lightweight user authentication and key-agreement scheme for IoT. Chen et al. utilized XOR operations, hash functions, and elliptical multiplication. Lee et al. [26] indicated that Chen et al.'s scheme did not provide a steal-resistant smartcard offline password, offline identity guessing and reply attack. Subsequently, in 2020, Ye et al. [27] proposed an authentication and key agreement scheme for IoT-based cloud computing environments by advancing the protocol developed by He et al. [28]. Ye et al. addressed various security issues in the scheme proposed by He et al, such as failure to resist insider attacks, offline password-guessing, user impersonation, and potential DoS attacks.

Table 1, summarizes cryptographic technologies and limitations of various authentication schemes related to IoT, cloud-based IoT, and cloud computing environments. Related papers propose various protocols to provide users with secure and fast services in the CloudIoT environment. However, there are still vulnerabilities and challenges in fully supporting security features, as some attacks persist. Additionally, methods using symmetric keys like ECC may incur higher computation costs in IoT environments. Therefore, our goal is to design a lightweight protocol tailored for IoT environments using XOR and to achieve higher security in our scheme.

Table 1. Authentication scheme overview.

Schemes	Cryptographic Technologies	Limitaion
Islam-Biswas [23]	- ECC - Self-certified public keys	- Cannot provide anonymity - Vulnerable to reply and clogging attacks
He et al. [28]	- Asymmetric cryptography	- Vulnerable to insider attacks, offline password-guessing, user impersonation attacks, and DoS attacks
Chen at al. [25]	- XOR operation - Hash function - Elliptic multiplication	- Vulnerable to stolen smartcard, offline password-guessing, offline identity guessing, and reply attacks
Prosanta-Biplab [18]	- PUF - Fuzzy extractor	-Vulnerable to man-in-the-middle attacks, impersonation attacks, session key hijaking, conventional and differential template attacks
Zhou et al. [20]	- XOR operation - Hash function	- Cannot provide mutual authentication - Vulnerable to insider attacks and man-in-the-middle attacks
Nikooghadam et al. [13]	- XOR operation - Hash function	- Vulnerable to reply attacks, privileged insider attacks, offline password-guessing, known key temporary imformation attacks, server spoofing attacks and impersonation attcks
Tsai-Lo [29]	- Single Sign On scheme	- Cannot provide sesssion key security, mutual authentication, and user anonymity - Vulnerable to impersonation attacks

Table 1. *Cont.*

Schemes	Cryptographic Technologies	Limitaion
Kumari et al. [30]	- Hash function - Diffie-Hellman	- Cannot provide user unlinkability and anonymity, data confidentiality - Vulnerable to known session-specific temporary information attacks, impersonation attacks, and desynchronization attacks
Bhuarya et al. [31]	- ECC	- Cannot provide mutual authentication - Vulnerable to impersonation attacks and man-in-the-middle attacks

3. Preliminaries

3.1. System Model

As shown in Figure 2, an IoT-enable cloud computing environment includes three entities: user, cloud server, and control server. Users can also use cloud computing provided by a cloud server, using IoT-enabled devices. Therefore, the user and cloud server should register and authenticate it through the control server. Finally, the user and cloud server share a session key for communication. The details are as follows

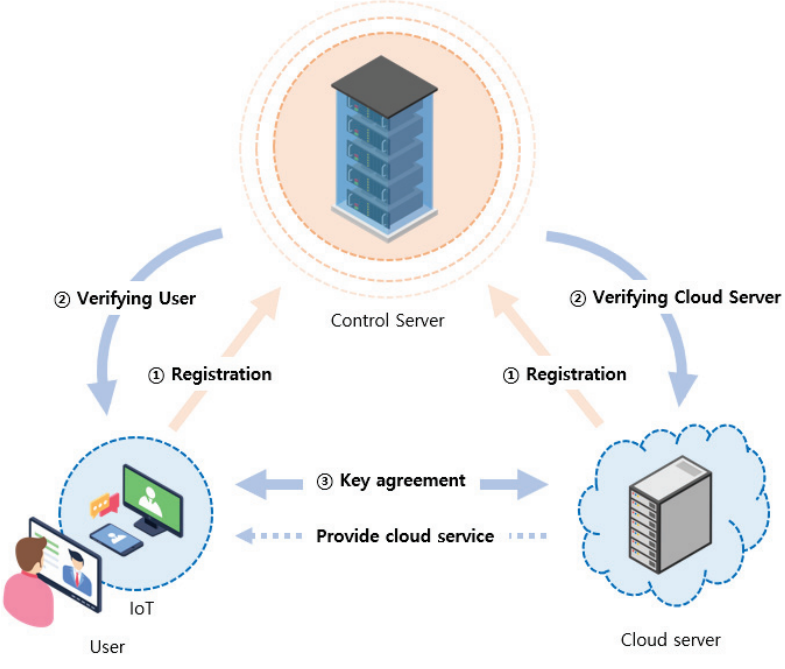


Figure 2. System model.

- User (U_i): User uses IoT devices with cloud services. Communicates with the cloud servers, then the user should register with the control server. The user can use smart cards and biometric technology to store sensitive information or the user's identity and password. We assumed that the user is an untrusted entity, implying that the user can execute unauthorized or malicious attacks.
- Cloud server (S_j): A cloud server provides cloud services to users using IoT devices. To achieve this, the cloud server should be registered with the control server. As a semi-trusted entity, the cloud server can misbehave; however, it cannot directly collude or participate.
- Control server (CS): This manages the registration of the user and cloud server, and helps generate the session key for authentication and subsequent communication.

As a semi-trusted entity, the control server can misbehave; however, it cannot directly collude or participate.

3.2. Adversary Model

We employ the widely used "Dolev–Yao (DY) model" [32] to define the capabilities of the adversary. The details are as follows:

- Within the DY model, entities in the IoT environments are considered trustworthy, and the communication channel is also considered insecure. Consequently, the adversary can engage in various actions through the insecure channel, including resending, eavesdropping, blocking, and deleting any messages transmitted.
- The adversary can extract sensitive information through power analysis attacks from stolen user smart cards. Additionally, because the control and cloud servers are semi-trusted entities, the adversary can also extract information from their databases.

Furthermore, the "Canetti–Krawczyk (CK) model" [33] assumes that stronger adversaries can also be adapted to our protocol. The adversaries in the CK model can obtain and use ephemeral values or long-term values, and using those, ephemeral leakage attacks can be performed.

4. Revisit of Wu et al.'s Scheme

4.1. Registration Phase

Before generating a session key for communication, the user and cloud server must go through the registration process via a secure channel. The detailed process is as follows.

4.1.1. User Registration Phase

Step 1: The U_i enters ID_i, PW_i and imprints B_i on the device. Then, calculates $Gen(B_i) = \sigma_i, \tau_i$, $HPW_i = h(PW_i||\sigma_i)$ and sends ID_i, HPW_i to CS as a registration request message through a secure channel.

Step 2: CS checks if U_i's identity is new, and generates a random number n_i to calculate $TID_i = h(ID_i)$, $A_1 = h(ID_{CS}||HPW_i) \oplus (n_i \oplus x)$. Then, stores $\{TID_i, HPW_i\}$ in its database, and stores $\{A_1, ID_{CS}\}$ to smart card SC. After that, sends SC to U_i through a secure channel.

Step 3: U_i computes $A_2 = h(ID_i||HPW_i)$ and store $\{A_1, A_2, ID_{CS}, Gen(\cdot), Rep(\cdot), \tau_i\}$ in SC.

4.1.2. Cloud Server Registration Phase

Step 1: S_j selects SID_j and random number n_j and sends $\{SID_j, n_j\}$ as a request message to CS through a secure channel.

Step 2: CS checks if S_j's identity is new, and chooses S_j's pseudo identity QID_j, computes $A_3 = h(SID_j||x \oplus n_j)$, then stores $\{QID_j, n_j\}$ in its database. Next, CS sends QID_j, n_j to S_j through secure channel.

Step 3: S_j computes $A_3^* = A_3 \oplus SID_j$, and stores $\{A_3^*, QID_j\}$.

4.2. Login and Authentication Phase

In this phase, the control server first verifies the identities of the user and the cloud server. If both are confirmed, a shared session key for subsequent communication is generated. The detailed process is as follows and illustrated in Figure 3.

User U_m	Cloud Server	Control Server D_n
Input ID_i, PW_i, imprint B_i Compute $Rep(B_i, \tau_i) = \sigma_i$ $HPW_i = h(PW_i \| \| \sigma_i)$ $A'_2 = h(ID_i \| \| HPW_i)$ Check $A'_2 \stackrel{?}{=} A_2$ Generate r_i, TS_1 Compute $(n_i \oplus x) = A_1 \oplus h(ID_{CS} \| \| HPW_i)$ $B_1 = r_i \oplus h(ID_{CS} \| \| HPW_i \oplus SID_j)$ $B_2 = SID_j \oplus h(ID)CS \| \| HPW_i)$ $B_3 = h(TID_i \| \| ID_{CS} \| \| n_i \oplus x) \oplus HPW_i$		
$\xrightarrow{M_1 = \{TID_i, A_1, B_1, B_2, B_3, TS_1\}}$		
	Check $\|TS_1 - TS_c\| \leqq \Delta T$ Generate r_j, TS_2 Compute $A_3 = SID_j \oplus A_3^*$ $B_4 = r_j \oplus h(A_3 \| \| SID_j)$ $B_5 = h(r_j \| \| A_3 \| \| SID_j)$	
	$\xrightarrow{M_2 = \{M_1, QID_j, B_4, B_5, TS_2\}}$	
		Check $\|TS_2 - TS_c\| \leqq \Delta T$ According TID_i to find HPW_i Compute $SID_j = B2 \oplus h(ID_{CS} \| \| HPW_i)$ $r_i = B_1 \oplus h(ID_{CS} \| \| HPW_i \oplus SID_j)$ $(n_i \oplus x) = A_1 \oplus h(ID_{CS} \| \| HPW_i)$ $B'_3 = h(TID_i \| \| ID_{CS} \| \| n_i \oplus x) \oplus HPW_i$ Check $B'_3 \stackrel{?}{=} B_3$ According QID_j to find n_j Compute $A_3 = h(SID_j \| \| x \oplus n_j)$ $r_j = B_4 \oplus h(A_3 \| \| SID_j)$ $B'_5 = h(r_j \| \| A_3 \| \| SID_j)$ Check $B'_5 \stackrel{?}{=} B_5$ Generate r_k, TS_3 Compute $SK = h(r_i \oplus HPW_i \| \| r_j \| \| r_k \| \| SID_j)$ $B_6 = (r_i \oplus HPW_i) \oplus A_3$ $B_7 = h(A_3 \oplus r_j \| \| SID_j) \oplus r_k$ $B_8 = h(r_j \| \| r_k \| \| SK \| \| TS_3)$ $B_9 = h(n_i \oplus x \| \| SID_j) \oplus r_j$ $B_10 = h(HPW_i \| \| r_i) \oplus r_k$ $B_11 = h(SK \| \| n_i \oplus x \| \| r_k \| \| r_j)$
	$\xleftarrow{M_3 = \{B_6, B_7, B_8, B_9, B_{10}, B_{11}, TS_3\}}$	
	Check $\|TS_3 - TS_c\| \leqq \Delta T$ Compute $(r_i \oplus HPW_i) = B_6 \oplus A_3$ $r_k = h(A_3 \| \| r_j \| \| SID_j) \oplus B_7$ $SK = h(r_i \oplus HPW_i \| \| r_j \| \| r_k \| \| SID_j)$ $B'\|8 = h(r_j \| \| r_k \| \| SK \| \| TS_3)$ Check $B'_8 \stackrel{?}{=} B_8$	
$\xleftarrow{M_4 = \{B_9, B_{10}, TS_4\}}$		
Checks $\|TS_4 - TS_c\| \leqq \Delta T$ $r_j = h(n_i \oplus x \| \| SID_j) \oplus B_9$ $r_k = h(HPW_i \| \| r_i) \oplus B_{10}$ $SK = h(r_i \oplus HPW_i \| \| r_j \| \| r_k \| \| SID_j)$ $B'_{11} = h(SK \| \| n_i \oplus x \| \| r_k \| \| r_j)$ Checks $B'_{11} \stackrel{?}{=} B_{11}$ Compute $B_{12} = h(SK \| \| r_j)$		
$\xrightarrow{M_5 = \{B_{12}\}}$		
	Compute $B'_{12} = h(SK \| \| r_j)$ Check $B'_{12} \stackrel{?}{=} B_{12}$	

Figure 3. AKA phase of Wu et al.'s scheme.

Step 1: U_i enters ID_i, PW_i and imprints B_i, and calculates $Rep(B_i, \tau_i) = \sigma_i$, $HPW_i = h(PW_i||\sigma_i)$, $A'2 = h(ID_i||HPW_i)$. Then, by confirming $A'_2 \stackrel{?}{=} A2$, U_i can be verified as a legitimate user. If this is valid, U_i selects a random number r_i and timestamp TS_1, then calculates $(n_i \oplus x) = A_1 \oplus h(ID_{CS}||HPW_i)$, $B_1 = r_i \oplus h(ID_{CS}||HPW_i \oplus SID_j)$, $B_2 = SID_j \oplus h(ID_{CS}||HPW_i)$, and $B_3 = h(TID_i||ID_{CS}||n_i \oplus x) \oplus HPW_i$. Finally, generates a message $M_1 = \{TID_i, A_1, B_1, B_2, B_3, TS_1\}$ and sends it to S_j via open channel.

Step 2: Upon receiving U_i's message, CS confirms timestamp $|TS_1 - TS_c| \leqq \Delta T$. If the timestamp is valid, S_j chooses a random value r_j and timestamp TS_2. S_j computes $A_3 = SID_j \oplus A_3^*$, $B_4 = r_j \oplus h(A_3||SID_j)$, and $B_5 = h(r_j||A_3||SID_j)$. Finally, message $M_2 = \{M_1, QID_j, B_4, B_5, TS_2\}$ is sent through an open channel.

Step 3: After receiving the M_2, S_j confirms timestamp $|TS_2 - TS_c| \leqq \Delta T$. If the timestamp is successfully verified, CS uses TID_i to find HPW_i and performs the following computations: $SID_j = B_2 \oplus h(ID_{CS}||HPW_i)$, $r_i = B_1 \oplus h(ID_{CS}||HPW_i \oplus SID_j)$, and $B'_3 = h(TID_i||ID_{CS}||n_i \oplus x) \oplus HPW_i$. And by checking $B'_3 \stackrel{?}{=} B_3$, the CS confirms whether U_i is the legitimate user. Next, CS utilizes the value of QID_j to find n_j and then performs the following computations: $A_3 = h(SID_j||x \oplus n_j)$, $r_j = B_4 \oplus h(A_3||SID_j)$, and $B'_5 = h(r_j||A_3||SID_j)$. After checking $B'_5 \stackrel{?}{=} B_5$ is valid, CS then selects r_k, TS_3, computes $SK = h(r_i \oplus HPW_i||r_j||r_k||SID_j)$, $B_6 = (r_i \oplus HPW_i) \oplus A_3$, $B_7 = h(A_3||r_j||SID_j) \oplus r_k$, $B_8 = h(r_j||r_k||SK||TS_3)$, $(n_i \oplus x) = A_1 \oplus h(ID_{CS}||HPW_i)$, $B_9 = h(n_i \oplus x||SID_j) \oplus r_j$, and $B_{10} = h(HPW_i||r_i) \oplus r_k$, $B_{11} = h(SK||n_i \oplus x||r_k||r_j)$. At last, CS generates message $M_3 = \{B_6, B_7, B_8, B_9, B_{10}, B_{11}, TS_3\}$ and sends to S_j through an open channel.

Step 4: Upon receiving M_3, S_j checks timestamp $|TS_3 - TS_c| \leqq \Delta T$. If the timestamp is valid, S_j calculates following computations: $(r_i \oplus HPW_i) = B_6 \oplus A_3$, $SK = h(r_i \oplus HPW_i||r_j||r_k||SID_j)$, and $B'_8 = h(r_j||r_k||SK||TS_3)$, and confirms $B'_8 \stackrel{?}{=} B_8$. If it confirms, S_j generates message $M_4 = \{B_9, B_{10}, TS_4\}$ to U_i via open channel.

Step 5: U_i verifies timestamp $|TS_4 - TS_c| \leqq \Delta T$. If the timestamp is valid, U_i calculates $r_j = h(n_i \oplus x||SID_j) \oplus B_9$, $r_k = h(HPW_i||r_i) \oplus B_{10}$, $SK = h(r_i \oplus HPW_i||r_j||r_k||SID_j)$, and $B'_{11} = h(SK||n_i \oplus x||r_k||r_j)$ and calculates $B'_{11} \stackrel{?}{=} B_{11}$. If it confirms, U_i computes $B_{12} = h(SK||r_j)$ and generates $M_5 = \{B_{12}\}$ and sends to S_j.

Step 6: S_j calculates the equation $B'_{12} = h(SK||r_j)$ and then checks $B'_{12} \stackrel{?}{=} B_{12}$. If they match, S_j stores SK for future communication.

5. Cryptanalysis of Wu et al.'s Scheme

Following the description of Section 3, adversary $\mathcal{A}$ can obtain important values from the user's smart card by using a power analysis attack. Furthermore, $\mathcal{A}$ can extract parameters from the cloud server and control server itself, because they are considered semi-trusted. With this information, various security attacks, including insider attack, verification table leakage attack, privileged insider attack, user impersonation, and cloud server impersonation, can be executed by $\mathcal{A}$. Details are described below.

5.1. Insider Attack

An adversary A, who has undergone the registration process as a legitimate user, can obtain session keys from another user U_i's sessions or impersonate U_i. The detailed process is as follows:

Step 1: After completing the registration process, A obtains B_6 of M_3 during their AKA process. Subsequently, A calculates A_3 of S_j using their own HPW_a and r_a.

Step 2: In another user U_i's session, A obtains message M_2 and uses B_4 and the previously acquired A_3 to deduce r_j.

Step 3: From B_6 of M_3, A calculates user U_i's r_i and HPW_i, and from B_7, A calculates r_k.

Step 4: Using the computed values, A can generate the session key $SK = h(r_i \oplus HPW_i||r_j||r_k||SID_j)$ for another user U_i and potentially disclose or exploit it.

Therefore, Wu et al.'s scheme cannot resist insider attacks.

5.2. Verification Table Leakage Attack

If $\mathcal{A}$ extracts verification table of cloud server, $\mathcal{A}$ can disclose session key. The following procedures are below:

Step 1: $\mathcal{A}$ extracts the verification table to take $\{A_3^*, QID_j\}$ from S_j. And also intercept message $M_2 = \{M_1, QID, B_4, B_5, TS_2\}$ transmitted in public channel.

Step 2: $\mathcal{A}$ calculates $A_3 = SID_j \oplus A_3^*$, and $r_j = B_4 \oplus h(A_3||SID_j)$to extract A_3 and r_j.

Step 3: $\mathcal{A}$ takes message $M_3 = \{B_6, B_7, B_8, B_9, B_{10}, B_{11}, TS_3\}$.

Step 4: $\mathcal{A}$ computes $(r_i \oplus HPW_i) = B6 \oplus A_3$, and $r_k = h(A_3||r_j||SID_j) \oplus B_7$. In addition by calculating $SK = h(r_i \oplus HPW_i||r_j||r_k||SID_j)$, A can generate a session key to disclose or exploit it.

Therefore, Wu et al.'s scheme cannot resist verification table leakage attacks.

5.3. Privileged Insider Attack

A privileged insider can take important information like $\{ID_j, HPW_j\}$ from the registration message and values stored in the user's smart card such as $\{A_1, A_2, ID_{cs}, Gen(), Rep(), \}$. Through support from this privileged insider, a malicious $\mathcal{A}$ can generate a session key through the following:

Step 1: $\mathcal{A}$ computes $SID_j = B_2 \oplus h(ID_{CS}||HPW_i)$, and $r_i = B_1 \oplus h(ID_{CS}||HPW_i \oplus SID_j)$, $(n_i \oplus x) = A_1 \oplus h(ID_{CS}||HPW_i)$; therefore, $\mathcal{A}$ can extract parameters SID_j, r_i, and $(n_i \oplus x)$.

Step 2: $\mathcal{A}$ intercepts message $M_4 = \{B_9, B_{10}, TS_4\}$.

Step 3: $\mathcal{A}$ calculates $r_j = h(n_i \oplus x||SID_j) \oplus B_9$, and $r_k = h(HPW_i||r_i) \oplus B_{10}$. Hence, $\mathcal{A}$ can compute session key $SK = h(r_i \oplus HPW_i||r_j||r_k||SID_j)$ and disclose it.

Thus, Wu et al.'s scheme is insecure against privileged insider attacks.

5.4. Impersonation

When $\mathcal{A}$ obtains the table information $\{k_n, SID_n\}$ of the control center, $\mathcal{A}$ can calculate $SK_{nm} = h(SID_m||SID_n||SID_c||A_8)$.

(1) User impersonation: If the privileged insider described in Section 5.3 generates random number r_i and time stamp TS_1, $\mathcal{A}$ can forge message $M_1 = \{TID_i, A_1, B_1, B_2, B_3, TS_1\}$. In addition, by $\mathcal{A}$ to take message $M_4 = \{B_9, B_{10}, TS_4\}$ from an unsecured public channel, $\mathcal{A}$ can generate session key and rj. Thus, $\mathcal{A}$ can send message $M_5 = \{B12\}$ impersonates user.

(2) Cloud server impersonation: According to the previous verification table attack in Section 5.2, $\mathcal{A}$ generates random number r_j and time stamp TS_2, and $\mathcal{A}$ can send $M_2 = \{M_1, QID_j, B_4, B_5, TS_2\}$. Second, $\mathcal{A}$ can generate $M_4 = \{B_9, B_{10}, TS_4\}$ after intercept message $M_3 = \{B_6, B_7, B_8, B_9, B_{10}, B_{11}, TS_3\}$. Hence $\mathcal{A}$ can impersonate cloud server.

Therefore, Wu et al.'s scheme cannot resist user and cloud impersonation attack.

5.5. Lack of Untraceability

If an attacker A continues to eavesdrop on $M_1 = \{TID_i, A_1, B_1, B_2, B_3, TS_1\}$ and compares the value of TID_i contained in M_1, A can track the user U_i. The reason is that the pseudo identity of U_i, TID_i, is a fixed value, and an attacker can easily obtain it through

eavesdropping on the message. Indeed, by verifying whether the value of TID_i matches the values from previous or subsequent communications, A can detect the user. In conclusion, Wu et al.'s scheme lacks anonymity and untraceability.

5.6. Impossibility of Offline Password Update

In the user registration phase in Wu et al.'s scheme, the value of HPW_i is created by concatenating the user's password PW_i with their biometric information σ_i. Additionally, this HPW_i is transmitted to the control server CS and undergoes the operation $A_1 = h(ID_{CS}||HPW_i) \oplus (n_i \oplus x)$, and stored in the CS's database as A_1. However, this design leads to a problem where users must communicate with the CS to update the A_1 value stored in the CS if they wish to change their password, because CS cannot create the HPW_i on its own. Consequently, Wu et al.'s scheme does not support offline password updates.

6. Proposed Protocol

6.1. Registration Phase

Before generating a session key for communication, the user and the cloud server must go through the registration process with the control server via a secure channel. In this phase, users register the information, such as identity, password, and biometrics, with the control server. The detailed process is as follows and illustrated in Figure 4.

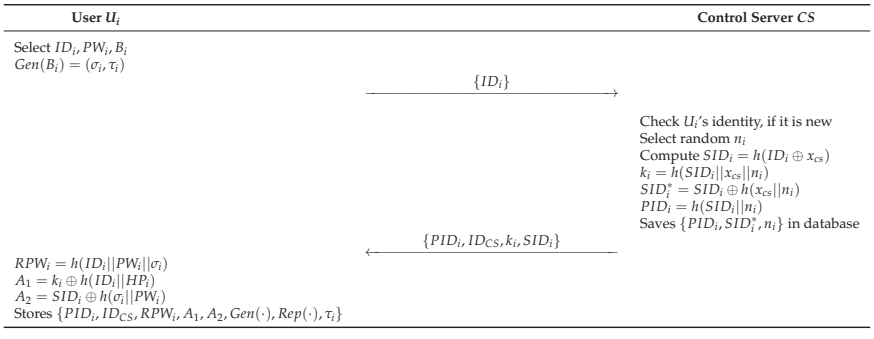


Figure 4. User registration phase of proposed scheme.

6.1.1. User Registration Phase

Step 1: The U_i enters ID_i, PW_i and imprints B_i on the device. Then, calculates $Gen(B_i) = \sigma_i$ and sends ID_i to CS as a registration request message through a secure channel.

Step 2: CS checks if U_i's identity is new, and generates a random number n_i to calculate $SID_i = h(ID_i \oplus x_{cs}, k_i = h(SID_i||x_{cs}||n_i)$. $SID_i^* = SID_i \oplus h(x_{cs}||n_i)$ and $PID_i = h(SID_i||n_i)$. Then, stores $\{PID_i, SID_i^*, n_i\}$ in its database, and sends $\{PID_i, ID_{cs}, k_i, SID_i\}$ to U_i through secure channel.

Step 3: U_i computes $RPW_i = h(ID_i||PW_i||\sigma_i)$, $A_1 = k_i \oplus h(ID_i||HP_i)$, and $A_2 = SID_i \oplus h(\sigma_i||PW_i)$. Then, store $\{PID_i, ID_{CS}, RPW_i, A_1, A_2, Gen(\cdot), Rep(\cdot), \tau_i\}$ in SC.

6.1.2. Cloud Server Registration Phase

In this phase, cloud servers register the information with the control server. The detailed process is as follows and illustrated in Figure 5.

Step 1: S_j selects SID_j and sends $\{ID_j\}$ as a request message to CS through a secure channel.

Step 2: CS checks if S_j's identity is new, and chooses random number n_j, computes $k_j = h(ID_j||n_j||x_{cs})$. Then, stores $\{ID_j, n_j\}$ in its database. Next, CS sends k_j to S_j through secure channel.

Step 3: S_j computes $A_3 = k_j \oplus x_j$, and stores $\{A_3\}$.

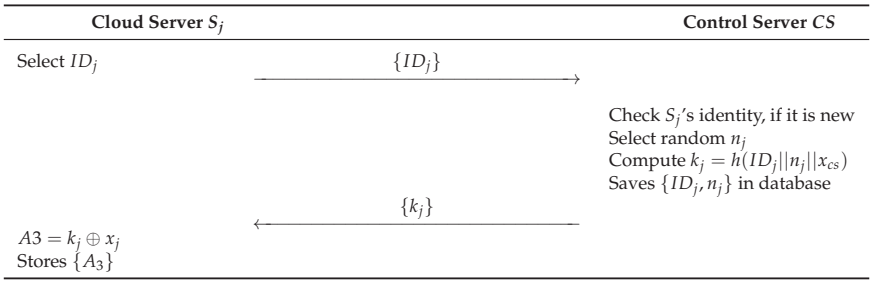


Figure 5. Cloud server registration phase of proposed scheme.

6.2. Login and Authentication Phase

In this phase, the control server first verifies the identities of the user and the cloud server. If both are confirmed, a shared session key for subsequent communication is generated. The detailed process is as follows and illustrated in Figure 6.

Step 1: U_i enters ID_i, PW_i, imprints B_i, and calculates $Rep(B_i, \tau_i) = \sigma_i$, $RPW_i' = h(ID - I||PW_i||\sigma_i)$, $k_i = A_1 \oplus (ID_i||PW_i)$ and $SID_i = A_q \oplus (\sigma_i||PW_i)$. Then, by confirming $RPW_i' \stackrel{?}{=} RPW_i$, U_i can be verified as a legitimate user. If this is valid, U_i selects a random number r_i and timestamp TS_1 then calculates $B_1 = r_i \oplus h(SID_i||ID_{CS}||k_i)$, $B_2 = ID_j \oplus h(ID_{CS}||SID_i||r_i)$ and $V_1 = h(ID_{CS}||TS_1||SID_i||k_i||r_i)$. Finally, it generates a message $M_1 = \{PID_i, B_1, B_2, V_1, TS_1\}$ and sends it to S_j via open channel.

Step 2: Upon receiving U_i's message, CS confirms timestamp $|TS_1 - TS_c| \leqq \Delta T$. If thetimestamp is valid, S_j chooses a random value r_j and timestamp TS_2. S_j computes $k_j = A_3 \oplus x_j$, $B_3 = r_j \oplus h(k_j||ID_j)$, and $V_2 = h(k_j||TS_2||ID_j||r_j)$. Finally, message $M_2 = \{M_1, B_3, V_2, TS_2\}$ is sent through an open channel.

Step 3: After receiving the M_2, S_j confirms timestamp $|TS_2 - TS_c| \leqq \Delta T$. If the timestamp is successfully verified, CS uses PID_i to find $\{SID_i^*, n_i\}$ and performs the following computations: $SID_i = SID_i^* \oplus (x_{CS}||n_i)$, $k_i = h(SID_i||x_{CS}||n_i)$, $r_i = B_1 \oplus h(SID_i||ID_{CS}||k_i)$ and $V_1' = h(ID_{CS}||TS_1||SID_i||k_i||r_i)$. And by checking $V_1' \stackrel{?}{=} V_1$, the CS confirms whether U_i is the legitimate user.

Step 4: Next, CS calculates $ID_j = B_2 \oplus h(ID_{CS}||SID_i||r_i)$ and utilizes the value of ID_j to find n_j. Then, it performs the following computations: $k_j = h(ID_j||n_j||x_{CS})$, $r_j = B_3 \oplus h(k_j||ID_j)$, and $V_2' = h(k_j||TS_2||ID_j||r_j)$. Subsequently, it checks if $V_2' \stackrel{?}{=} V_2$ is valid.

Step 5: CS then selects r_k, TS_3, computes $SID_j = ID_j \oplus h(k_j||r_k)$, $C_1 = h(SID_i \oplus SID_j \oplus ID_{CS})$, $PID_i^* = PID_i \oplus h(PID_i||r_k||r_i)$. Next, it updates the old PID_i to PID_i^*.

Step 6: $B_6 = (r_k||r_i) \oplus h(r_j||k_j)$, $B_7 = C_1 \oplus h(k_j||r_k)$, $B_8 = (r_j||r_k) \oplus h(SID_i||ID_{CS}||r_i)$, $B_9 = SID_j \oplus h(SID_i||r_k)$, $V_3 = h(r_j||r_k||C_1||ID_j||TS_3)$ and $V_4 = h(SID_i||r_j||r_k||C_1||TS_3)$. At last, CS generates message $M_3 = \{B_6, B_7, B_8, B_9, V_3, V_4, TS_3\}$ and sends to S_j through an open channel.

Step 7: Upon receiving M_3, S_j checks timestamp $|TS_3 - TS_c| \leqq \Delta T$. If the timestamp is valid, S_j calculates following computations: $(r_k||r_i) = B6oplush(r_j||k_j)$, $C_1 = B_7 \oplus h(k_j||r_k)$, $SK = h(C_1||r_i||r_j||r_k)$, and $V_3' = h(r_j||r_k||C_1||ID_j||TS_3)$. Subsequently, it confirms $V_3' \stackrel{?}{=} V_3$. If it confirms, S_j generates message $M_4 = \{B_8, B_9, V_4, TS_4\}$ to U_i via an open channel.

Step 8: U_i verifies timestamp $|TS_4 - TS_c| \leqq \Delta T$. If the timestamp is valid, U_i calculates $(r_j||r_k) = B_8 \oplus h(SID_i||ID_{CS}||r_i)$, $SID_j = B_9 \oplus h(SID_i||r_k)$, $C_1 = h(SID_i \oplus SID_j \oplus ID_{CS})$, $SK = h(C_1||r_i||r_j||r_k)$, and calculates $V_4' = h(SID_i||r_j||r_k||C_1||TS_3)$ to check $B_{11}' \stackrel{?}{=} B_{11}$. If it confirms, U_i computes $PID_i^* = PID_i \oplus h(PID_i||r_k||r_i)$ and update old PID_i to PID_i^*.

User U_i	Cloud Server S_j	Control Server CS
Inputs ID_i, PW_i, imprint B_i Compute $Rep(B_i, \tau_i) = \sigma_i$ $RPW_i' = h(ID_i \| \| PW_i \| \| \sigma_i)$ $k_i = A_1 \oplus (ID_i \| \| PW_i)$ $SID_i = A_2 \oplus (\sigma_i \| \| PW_i)$ Checks $RPW_i' \stackrel{?}{=} RPW_i$ Generate random r_i, timestamp TS_1 $B_1 = r_i \oplus h(SID_i \| \| ID_{CS} \| \| k_i)$ $B_2 = ID_j \oplus h(ID_{CS} \| \| SID_i \| \| r_i)$ $V_1 = h(ID_{CS} \| \| TS_1 \| \| SID_i \| \| k_i \| \| r_i)$		
$\xrightarrow{M_1 = \{PID_i, B_1, B_2, V_1, TS_1\}}$		
	Check $\|TS_1 - TS_c\| \leqq \Delta T$ Generate random r_j, timestamp TS_2 $k_j = A_3 \oplus x_j$ $B_3 = r_j \oplus h(k_j \| \| ID_j)$ $V_2 = h(k_j \| \| TS_2 \| \| ID_j \| \| r_j)$	
	$\xrightarrow{M_2 = \{M_1, B_3, V_2, TS_2\}}$	
		Check $\|TS_2 - TS_c\| \leqq \Delta T$ According PID_i to find $\{SID_i^*, n_i\}$ $SID_i = SID_i^* \oplus (x_{CS} \| \| n_i)$ $k_i = h(SID_i \| \| x_{CS} \| \| n_i)$ $r_i = B_1 \oplus h(SID_i \| \| ID_{CS} \| \| k_i)$ $V_1' = h(ID_{CS} \| \| TS_1 \| \| SID_i \| \| k_i \| \| r_i)$ Check $V_1' \stackrel{?}{=} V_1$ $ID_j = B_2 \oplus h(ID_{CS} \| \| SID_i \| \| r_i)$ According ID_j to find $\{n_j\}$ $k_j = h(ID_j \| \| n_j \| \| x_{CS})$ $r_j = B_3 \oplus h(k_j \| \| ID_j)$ $V_2' = h(k_j \| \| TS_2 \| \| ID_j \| \| r_j)$ Check $V_2' \stackrel{?}{=} V_2$ Generate random r_k, timestamp TS_3 $SID_j = ID_j \oplus h(k_j \| \| r_k)$ $C_1 = h(SID_i \oplus SID_j \oplus ID_{CS})$ $PID_i^* = PID_i \oplus h(PID_i \| \| r_k \| \| r_i)$ Update PID_i to PID_i^* $B_6 = (r_k \| \| r_i) \oplus h(r_j \| \| k_j)$ $B_7 = C_1 \oplus h(k_j \| \| r_k)$ $B_8 = (r_j \| \| r_k) \oplus h(SID_i \| \| ID_{CS} \| \| r_i)$ $B_9 = SID_j \oplus h(SID_i \| \| r_k)$ $V_3 = h(r_j \| \| r_k \| \| C_1 \| \| ID_j \| \| TS_3)$ $V_4 = h(SID_i \| \| r_j \| \| r_k \| \| C_1 \| \| TS_3)$
	$\xleftarrow{M_3 = \{B_6, B_7, B_8, B_9, V_3, V_4, TS_3\}}$	
	Check $\|TS_3 - TS_c\| \leqq \Delta T$ $(r_k \| \| r_i) = B_6 \oplus h(r_j \| \| k_j)$ $C_1 = B_7 \oplus h(k_j \| \| r_k)$ $SK = h(C_1 \| \| r_i \| \| r_j \| \| r_k)$ $V_3' = h(r_j \| \| r_k \| \| C_1 \| \| ID_j \| \| TS_3)$ Check $V_3' \stackrel{?}{=} V_3$	
$\xleftarrow{M_4 = \{B_8, B_9, V_4, TS_4\}}$		
Checks $\|TS_4 - TS_c\| \leqq \Delta T$ $(r_j \| \| r_k) = B_8 \oplus h(SID_i \| \| ID_{CS} \| \| r_i)$ $SID_j = B_9 \oplus h(SID_i \| \| r_k)$ $C_1 = h(SID_i \oplus SID_j \oplus ID_C S)$ $SK = h(C_1 \| \| r_i \| \| r_j \| \| r_k)$ $V_4' = h(SID_i \| \| r_j \| \| r_k \| \| C_1 \| \| TS_3)$ Checks $V_4' \stackrel{?}{=} V_4$ $PID_i^* = PID_i \oplus h(PID_i \| \| r_k \| \| r_i)$ Update PID_i to PID_i^*		

Figure 6. AKA phase of proposed scheme.

6.3. Offline Password and Biometric Template Update

In this phase, an authenticated user U can locally change their password and biometrics without a connection to CS. U must perform the login process on the IoT device before updating data offline. A logged-in user can update their password or biometric template. The detailed process is as follows and illustrated in Figure 7.

Step 1: U_i enters ID_i, PW_i and imprint B_i on the device. Compute $Rep(Bio_i, \tau_i) = \sigma_i$ and check $RPW_i' = h(ID_i||PW_i||\sigma_i)$ for login phase and confirm user.

Step 2: Then, ask U_i to change password and biometric data. U_i select new password PW_i^{new}, and compute $RPW_i^{new} = h(ID_i||PW_i^{new}||\sigma_i)$, $A_1^{new} = k_i \oplus h(ID_i||PW_i^{new})$ and $A_2^{new} = SID_i \oplus h(\oplus_i||PW_i^{new})$. Subsequently, update RPW_i, A_1, and A_2 with new data to change the password.

Step 3: Compute $Rep(Bi_i, \tau_i) = \sigma_i^{new}$, $RPW_i^{new} = h(ID||PW_i||\sigma_i^{new})$ and $A_2^{new} = SID_i \oplus h(\sigma_i^{new}||PW_i)$. Subsequently, update RPW_i^{new} and A_2^{new} with new data to change the biometric template.

User U_i		**Smart Card SC**
Inputs ID_i, PW_i and Imprint B_i to SC		
	$\{ID_i, PW_i, B_i\} \longrightarrow$	
		Compute $Rep(B_i, \tau_i) = \sigma_i$ $RPW_i' = h(ID_i\|\|PW_i\|\|\sigma_i)$ If $RPW_i' \stackrel{?}{=} RPW_i$, login is successful Ask U_i to change password and biometric data
Select PW_i^{new}		
	$\{PW_i^{new}\} \longrightarrow$	
		$RPW_i^{new} = h(ID\|\|PW_i^{new}\|\|\sigma_i)$ $A_1^{new} = k_i \oplus h(ID_i\|\|PW_i^{new})$ $A_2^{new} = SID_i \oplus h(\sigma_i\|\|PW_i^{new})$ Update $RPW_i \rightarrow RPW_i^{new}$ $A_1 \rightarrow A_1^{new}$ $A_2 \rightarrow A_2^{new}$
		$Rep(Bi_i, \tau_i) = \sigma_i^{new}$ $RPW_i^{new} = h(ID\|\|PW_i\|\|\sigma_i^{new})$ $A_2^{new} = SID_i \oplus h(\sigma_i^{new}\|\|PW_i)$ Update $RPW_i \rightarrow RPW_i^{new}$ $A_2 \rightarrow A_2^{new}$

Figure 7. Offline password and biometric template update of proposed scheme.

7. Security Analysis

7.1. ROR Model

In this section, we conduct an analysis of session key security using the ROR model [34]. To apply the proposed protocol to the ROR model, we first define participants, especially $U_{US}^{i_1}$, $U_{SJ}^{i_2}$, and $U_{CS}^{i_3}$ as user, cloud server, and control server, respectively. Note that i_k $(k = 1, 2, 3)$ is an instance for each participant. In ROR model, the adversary can eavesdrop, delete, intercept, and send messages through the public channel. Moreover, the adversary can extract secret parameters from the user $U_{US}^{i_1}$. These actions of the adversary can be defined as queries in the ROR model.

- $EX(U_{US}^{i_1}, U_{SJ}^{i_2}, U_{CS}^{i_3})$: This query is an eavesdropping attack that the adversary can obtain messages transmitted via a public channel. Thus, this query can be defined as a passive attack.

- $CoUD(U_{US}^{i_1})$: In this query, the adversary extracts secret parameters using the smart device of $U_{US}^{i_1}$. Therefore, we can define the query $CoUD$ is an active attack.
- $Sn(U_p^i)$: The adversary sends messages to legal participants through open channels. This query is an active attack.
- $Ts(U_p^i)$: In this query, the adversary flips an unbiased coin. When the result of the flipped coin is 0, the session key is not fresh. When the result of the flipped coin is 1, we can demonstrate that the session key is fresh. Otherwise, the result outputs $NULL$ ($\perp$).

Theorem 1. *We take a definition of P_{AD}, HA, q_{HA}, and q_{Sn} as the possibility of breaking session key, range space of hash function, number of hash functions, and number of send queries, respectively. Moreover, we define that s and C are the Zipf's parameters [35]. From that, the adversary tries to reveal the session key of the proposed protocol in polynomial time. Following [36–38], the ROR model analysis of the proposed protocol is composed of four games ($GAME_m$, $m = 0, 1, 2, 3$) and the winning possibility of the adversary is PW_{GAME_m} for each game $GAME_m$.*

$$P_{AD} \leq \frac{q_{HA}^2}{|HA|} + 2\{Cq_S^{Sn}\} \tag{1}$$

- *$GAME_0$: In this game, the adversary has no knowledge about the session key. Thus, the adversary picks a random bit B.*
$$P_{AD} = |2PW_{GAME_0} - 1| \tag{2}$$
- *$GAME_1$: The adversary conducts EX query to collect the messages transmitted via public channels. Thus, the adversary obtains $\{PID_i, B_1, B_2, V_1, TS_1\}$, $\{M_1, B_3, V_2\}$, $\{B_6, B_7, B_8, V_3, V_4, TS_3\}$, and $\{B_8, V_5, TS_4\}$. After that, the adversary flips an unbiased coin to execute the Ts query. However, the adversary has no knowledge of the session key $SK = h(C_1 \parallel r_i \parallel r_j \parallel r_k)$ because it is composed of random numbers r_i, r_j and r_k and masked in the hash functions. For these reasons, the adversary can obtain the following:*
$$PW_{GAME_1} = PW_{GAME_1} \tag{3}$$
- *$GAME_2$: The adversary conducts HA and Sn queries to reveal session key in this game. However, the session key is composed of fresh random numbers and a cryptographic hash function. Therefore, the adversary cannot make hash collisions to calculate the session key. Applying the birthday paradox [39], we obtain the following:*
$$|PW_{GAME_2} - PW_{GAME_1}| \leq \frac{q_{HA}^2}{|HA|} \tag{4}$$
- *$GAME_3$: In the last game, the adversary conducts CoUD query to obtain the secret parameters $\{PID_i, ID_{CS}, RPW_i, A_1, A_2, Gen(.), Rep(.), \tau_i\}$. However, the adversary cannot decrypt the secret parameters because these parameters are encrypted using the identity ID_i, password PW_i, and biometrics B_i. Since simultaneously guessing ID_i, PW_i, and B_i is a computationally infeasible task, the adversary has no advantage in this game. We obtain the following using Zipf's law [35].*
$$|PW_{GAME_3} - PW_{GAME_2}| \leq Cq_S^{Sn} \tag{5}$$
When all the games end, the adversary becomes a random bit B.
$$PW_{GAME_3} = \frac{1}{2} \tag{6}$$
We can calculate (7) utilizing (2) and (3).
$$\frac{1}{2}P_{AD} = |PW_{GAME_0} - \frac{1}{2}| = |PW_{GAME_3} - \frac{1}{2}| \tag{7}$$

Then, we use (6) and (7) to obtain (8).

$$\frac{1}{2}P_{AD} = |PW_{GAME_1} - PW_{GAME_3}| \tag{8}$$

We calculate (9) utilizing the triangular inequality.

$$\begin{aligned}\frac{1}{2}P_{AD} &= |PW_{GAME_1} - PW_{GAME_3}| \\ &\leq |PW_{GAME_1} - PW_{GAME_2}| \\ &\quad +|PW_{GAME_2} - PW_{GAME_3}| \\ &\leq \frac{q_{HA}^2}{2|HA|} + Cq_S^{Sn}\end{aligned} \tag{9}$$

We calculate (10) multiplying (9) by 2.

$$P_{AD} \leq \frac{q_{HA}^2}{|HA|} + 2\{Cq_S^{Sn}\} \tag{10}$$

We obtain the inEquation (10) which is the same as (1). It means that the adversary cannot distinguish random nonce and the session key using various security attacks, such as EX, $CoUD$, and Sn. Thus, we can prove the session key security of the proposed protocol.

7.2. BAN Logic

We analyze the mutual authentication of the proposed protocol using BAN logic [40]. Following [41–43], we define basic notations and descriptions of BAN logic in Table 2.

Table 2. Basic notations and decriptions.

Notation	Description
A_i, A_j	Principals
SK	Session key
T_1, T_2	Statements
$A_i \mid\equiv T_1$	A_i **believes** T_1
$A_1 \mid\sim T_1$	A_i once **said** T_1
$A_i \mapsto T_1$	A_i **controls** T_1
$A_i \lhd T_1$	A_i **receives** T_1
$\#T_1$	T_1 is **fresh**
$\{T_1\}_S$	T_1 is **encrypted** with S
$A_i \overset{SH}{\longleftrightarrow} A_j$	A_i and A_j have a shared key SH

7.2.1. Rules

In BAN logic, there are five rules, such as "Message meaning rule (MMR)", "Nonce verification rule (NVR)", "Jurisdiction rule (JR)", "Belief rule (BR)", and "Freshness rule (FR)".

1. Message meaning rule (MMR):

$$\frac{A_i \mid\equiv A_i \overset{SH}{\leftrightarrow} A_j, \quad A_i \lhd \{T_1\}_{SH}}{A_i \mid\equiv A_j \mid\sim T_1}$$

2. Nonce verification rule (NVR):

$$\frac{A_i \mid\equiv \#(T_1), \quad A_i \mid\equiv A_j \mid\sim T_1}{A_i \mid\equiv A_j \mid\equiv T_1}$$

3. Jurisdiction rule (JR):

$$\frac{A_i| \equiv A_j \mapsto T_1, \quad A_i| \equiv A_j| \equiv T_1}{A_i \Big| \equiv T_1}$$

4. Belief rule (BR):

$$\frac{A_i \Big| \equiv (T_1, T_2)}{A_i \Big| \equiv T_1}$$

5. Freshness rule (FR):

$$\frac{A_i \Big| \equiv \#(T_1)}{A_i \Big| \equiv \#(T_1, T_2)}$$

7.2.2. Goals

In our protocol, each participant authenticate the communication partner by establishing session key SK. Thus, goals of the proposed protocol can be shown as follows:

Goal 1: $UI| \equiv UI \overset{SK}{\longleftrightarrow} CS$

Goal 2: $UI| \equiv CS| \equiv UI \overset{SK}{\longleftrightarrow} CS$

Goal 3: $CS| \equiv UI \overset{SK}{\longleftrightarrow} CS$

Goal 4: $CS| \equiv UI| \equiv CS \overset{SK}{\longleftrightarrow} UI$

Goal 5: $CS| \equiv CS \overset{SK}{\longleftrightarrow} SJ$

Goal 6: $CS| \equiv SJ| \equiv CS \overset{SK}{\longleftrightarrow} SJ$

Goal 7: $SJ| \equiv CS \overset{SK}{\longleftrightarrow} SJ$

Goal 8: $SJ| \equiv CS| \equiv SJ \overset{SK}{\longleftrightarrow} CS$

7.2.3. Idealized Forms

In the proposed authentication phase, four messages are transmitted via open channels ($\{PID_i, B_1, B_2, V_1, TS_1\}$, $\{M_1, B_3, V_2\}$, $\{B_6, B_7, B_8, V_3, V_4, TS_3\}$, $\{B_8, V_5, TS_4\}$). To analyze these messages, we convert them into idealized forms.

$MSG_1 : UI \rightarrow SJ : \{r_i, ID_j, TS_1\}_{k_i}$

$MSG_2 : SJ \rightarrow CS : \{\{r_i, ID_j\}_{k_i}, \{r_j, TS_2\}_{k_j}\}$

$MSG_3 : CS \rightarrow SJ : \{\{r_k, r_i, C_1, TS_3\}_{k_j}, \{r_j, r_k, TS_3\}_{r_i}\}$

$MSG_4 : SJ \rightarrow UI : \{r_j, r_k, TS_4\}_{r_i}$

7.2.4. Assumptions

In the proposed protocol, participants agree on the freshness of the random number and secret parameters. Therefore, we show the assumptions to analyze the proposed authentication phase.

S_1**:** $CS| \equiv \#(TS_2)$

S_2**:** $SJ| \equiv \#(TS_3)$

S_3**:** $UI| \equiv \#(TS_4)$

S_4**:** $CS| \equiv \#(r_i)$

S_5**:** $CS| \equiv UI \overset{k_i}{\longleftrightarrow} CS$

S_6**:** $CS| \equiv SJ \overset{k_j}{\longleftrightarrow} CS$

S_7: $SJ| \equiv CS \overset{k_j}{\longleftrightarrow} SJ$

S_8: $UI| \equiv CS \overset{r_j}{\longleftrightarrow} UI$

7.2.5. BAN Logic Proof

Step 1: We obtain P_1 using MSG_2.

$$P_1\text{: } CS \lhd \{\{r_i, ID_j\}_{k_i}, \{r_j, TS_2\}_{k_j}\}$$

Step 2: We use S_5, S_6, and MMR to obtain P_2 and P_3 from P_1.

$$P_2\text{: } CS| \equiv UI| \sim (r_i, ID_j)$$
$$P_3\text{: } CS| \equiv SJ| \sim (r_j, TS_2)$$

Step 3: From P_2 and P_3, we use S_1, S_4, and FR to obtain P_4 and P_5.

$$P_4\text{: } CS| \equiv \#(r_i, ID_j)$$
$$P_5\text{: } CS| \equiv \#(r_j, TS_2)$$

Step 4: From P_2, P_3, P_4 and P_5, we use NVR to obtain P_6 and P_7.

$$P_6\text{: } CS| \equiv UI| \equiv (r_i, ID_j)$$
$$P_7\text{: } CS| \equiv SJ| \equiv (r_j, TS_2)$$

Step 5: We obtain P_8 using MSG_3.

$$P_8\text{: } SJ \lhd \{\{r_k, r_i, C_1, TS_3\}_{k_j}, \{r_j, r_k, TS_3\}_{r_i}\}$$

Step 6: We use S_7 and MMR to obtain P_9 from P_8.

$$P_9\text{: } SJ| \equiv CS| \sim (r_k, r_i, C_1, TS_3)$$

Step 7: From P_9, we use S_2 and FR to obtain P_{10}.

$$P_{10}\text{: } SJ| \equiv \#(r_k, r_i, C_1, TS_3)$$

Step 8: From P_9 and P_{10}, we use NVR to obtain P_{11}.

$$P_{11}\text{: } SJ| \equiv CS| \equiv (r_k, r_i, C_1, TS_3)$$

Step 9: Using P_7 and P_{11}, CS and SJ computes the session key $SK = h(C_1 \parallel r_i \parallel r_j \parallel r_k)$. Thus, we obtain the following:

$$P_{12}\text{: } SJ| \equiv CS| \equiv SJ \overset{SK}{\longleftrightarrow} CS \quad \textbf{(Goal 8)}$$
$$P_{13}\text{: } CS| \equiv SJ| \equiv CS \overset{SK}{\longleftrightarrow} SJ \quad \textbf{(Goal 6)}$$

Step 10: Using JR into P_{12} and P_{13}, We obtain the following goals:

$$P_{14}\text{: } SJ| \equiv SJ \overset{SK}{\longleftrightarrow} CS \quad \textbf{(Goal 7)}$$
$$P_{15}\text{: } CS| \equiv CS \overset{SK}{\longleftrightarrow} SJ \quad \textbf{(Goal 5)}$$

Step 11: We obtain P_{16} using MSG_4.

$$P_{16}\text{: } UI \lhd \{r_j, r_k, TS_4\}_{r_i}$$

Step 12: We use S_8 and MMR to obtain P_{17} from P_{16}.

$$P_{17}\text{: } UI| \equiv CS| \sim (r_j, r_k, TS_4)$$

Step 13: From P_{17}, we use S_3 and FR to obtain P_{18}.

$$P_{18}\text{: } UI\#(r_j, r_k, TS_4)$$

Step 14: From P_{17} and P_{18}, we use NVR to obtain P_{19}.

$$P_{19}\text{: } UI| \equiv CS| \equiv (r_j, r_k, TS_4)$$

Step 15: Using P_6 and P_{19}, UI and SJ agrees the session key $SK = h(C_1 \parallel r_i \parallel r_j \parallel r_k)$. Thus, we obtain the following:

$$P_{20}\text{: } UI| \equiv CS| \equiv UI \overset{SK}{\longleftrightarrow} CS \quad \textbf{(Goal 2)}$$
$$P_{21}\text{: } CS| \equiv UI| \equiv CS \overset{SK}{\longleftrightarrow} UI \quad \textbf{(Goal 4)}$$

Step 16: Using JR into P_{20} and P_{21}, We obtain the following goals:

$$P_{22}\text{: } UI| \equiv CS \overset{SK}{\longleftrightarrow} UI \quad \textbf{(Goal 1)}$$
$$P_{23}\text{: } CS| \equiv UI \overset{SK}{\longleftrightarrow} CS \quad \textbf{(Goal 3)}$$

7.3. Informal Security Analysis

7.3.1. Insider Attack

Malicious actor A, who has gone through the registration phase as a legitimate user, can attempt an insider attack using the acquired information. However, the attacker is unable to know the random values(ri, rj, rk) and k_j. As a result, the attacker cannot calculate C_1, rendering the attack impossible.

7.3.2. Impersonation Attack

(1) User impersonation: Adversary A needs to create a valid message $M_1 = \{PID_i, B_1, B_2, V_1, TS_1\}$ to impersonate the legitimate user U_i. While A might obtain PID_i from the user's device, it is impossible for A to access the necessary k_i and SID_i to calculate B_1, B_2, V_1, TS_1 needed to create the message. Therefore, A cannot generate the M_1 message on behalf of the user U_i and transmit it to the cloud server and control server. Thus, the proposed scheme is secure against user impersonation attacks.

(2) Cloud server impersonation: To execute this attack, A needs to send the message $M_2 = \{M_1, B_3, V_2, TS_2\}$ to the control server on behalf of the cloud server S_j. Even if A intercepts the transmission of M_1 over an open channel, they cannot generate the necessary B_3, V_2 for the message until they know k_j. Therefore, the proposed scheme is secure against cloud server impersonation attacks.

7.3.3. Reply and MITM Attacks

All users, the cloud server, and the control server attempt to validate the received messages through V_1', V_2', V_3', V_4'. Also, the sent messages are masked with different random values for each session, ensuring freshness. Therefore, the proposed scheme is secure against reply attacks and MITM attacks.

7.3.4. Privileged Insider Attack

In this attack scenario, external entity A is considered a privileged insider, implying that A possesses the user's registration request message ID_i and confidential values such as $\{A_1, A_2, ID_{CS}, Gen(\cdot), Rep(\cdot), \tau_i\}$ However, without the precise biometric information, ID, or PW values of the user, calculating $k_i = A_1 \oplus (ID_i||PW_i)$ or $SID_i = A_2 \oplus (\sigma_i||PW_i)$ is not possible. As a result, the proposed scheme is secure against privileged insider attacks.

7.3.5. Ephemeral Security Leakage Attack

To prevent adversary A from carrying out valid attacks, such as obtaining the session key through this attack scenario, it is essential to ensure that the session key is preserved even if the random values used in the session are exposed. Therefore, assuming A knows the values of r_i, r_j, r_k, it is postulated here that even with this knowledge, A cannot calculate SK without knowing SID_i, SID_j. Additionally, valid attacks like impersonating the user or cloud server using random values are not possible. Therefore, the proposed scheme is secure against ESL attacks.

7.3.6. Stolen Verifier Attack

We can assume that a malicious A, upon obtaining $\{A_3\}$ from the cloud server's database, attempts to calculate the session key $SK = h(C_1||r_i||r_j||r_k)$ or impersonate the cloud server. However, without the cloud server's secret key x_j, A cannot deduce the value of k_j from the stored A_3, nor can A determine the randomly generated values r_i, r_j, or r_k. Therefore, A is unable to compute the session key or impersonate the cloud server. Consequently, the proposed scheme is secure against verification table leakage attacks.

7.3.7. DoS Attack

The adversary A may intentionally attempt to send the message $M_1 = \{PID_i, B_1, B_2, V_1, TS_1\}$ repeatedly. However, to generate message M_1, A must go through the login process and pass the verification $RPW_i' \stackrel{?}{=} RPW_i$. However, to create a valid $RPW_i' = h(ID_i||PW_i||\sigma_i)$, A cannot have the required ID_i, PW_i, σ_i. Therefore, A cannot create and repeatedly send the message M_1, making the proposed scheme secure against DoS attacks.

7.3.8. User Anonymity and Untraceability

Due to the use of PID_i as a pseudo identity, the user's identity ID_i cannot be deduced by an adversary A. Additionally, the PID_i is updated as a new value with random elements for each session, making it impossible for A to compare PID_i values between previous and current sessions to compromise the user's untraceability. Therefore, the proposed scheme provides user anonymity and untraceability.

7.3.9. Session Key Disclosure Attack

To calculate the session key $SK = h(C_1||r_i||r_j||r_k)$, adversary A needs to have access to the values of SID_i, SID_j, r_i, r_j, and r_k. However, for A to discover SID_i, SID_j, they would need access to the secret key x_{cs} and the random values n_i and r_k. Additionally, random values like r_i, r_j, r_k are used temporarily and exist only within a single session. Therefore, the proposed scheme is secure against session key disclosure attacks.

7.3.10. Perfect Forward Secrecy

If the control server's secret key x_{cs} is compromised, adversary A may attempt to calculate the session key SK for a previous session. However, since $SK = h(C_1||r_i||r_j||r_k)$ does not contain x_{cs} and the values of r_i, r_j, r_k are random and cannot be deduced, A cannot perform the calculation. Furthermore, without n_i through x_{cs}, A cannot compute SID_i. Therefore, the proposed scheme ensures perfect forward secrecy.

7.3.11. Mutual Authentication

In the login and authentication phases, the messages $\{PID_i, B_1, B_2, V_1, TS_1\}$ and $\{M_1, B_3, V_2, TS_2\}$ included can be used by the control server to verify the legitimacy of the user and the cloud server through the transmitted V_1 and V_2. Additionally, messages $\{B_6, B_7, B_8, B_9, V_3, V_4, TS_3\}$ and $\{B_8, B_9, V_4, TS_4\}$ allow both the user and the cloud server to validate each other's identity using V_3 and V_4. Due to the unavailability of SID_i, kj values, and random values to adversaries through the open channel, the transparency of authentication is ensured. Therefore, the provided scheme offers mutual authentication.

8. Performance Analysis

8.1. Security Features Comparison

We visually compare the safety elements of the proposed scheme and related schemes [11,21,44–48] and record them in Table 3, which includes various types of safety elements such as "insider attack", "impersonation attack", "stolen verification attack", "ESL attack", "privileged attack", "perfect forward secrecy", "reply attack", "offline password-guessing attack", "session key disclosure", "mutual authentication", "DoS attack", "user anonymity", and "untraceability". Ultimately, the proposed scheme offers more security features compared to Wu et al.'s scheme, and it exhibits fewer features that are either unidentified or not provided, even when compared to the schemes of other related works.

Table 3. Security and functionality features(SFF) comparison.

SFF	[21]	[44]	[45]	[46]	[47]	[48]	[11]	Proposed
SP1	✓	✓	✓	✓	△	△	×	✓
SP2	×	✓	✓	×	✓	✓	×	✓
SP3	✓	✓	△	✓	△	△	×	✓
SP4	✓	✓	△	×	✓	✓	✓	✓
SP5	✓	✓	△	×	✓	✓	×	✓
SP6	✓	✓	△	✓	✓	△	✓	✓
SP7	×	✓	✓	✓	△	✓	✓	✓
SP8	✓	✓	×	✓	✓	△	✓	✓
SP9	×	✓	✓	×	△	△	×	✓
SP10	×	✓	✓	✓	△	✓	✓	✓
SP11	✓	✓	△	✓	△	△	✓	✓
SP12	×	✓	✓	✓	✓	✓	✓	✓
SP13	✓	✓	△	✓	✓	✓	×	✓

Note: SP1: insider attack; SP2: impersonation attack; SP3: stolen verification attack; SP4: ESL attack; SP5: privileged insider attack; SP6: perfect forward secrecy; SP7: reply attack; SP8 offline password-guessing attack SP9: session key disclosure; SP10: mutual authentication; SP11: DoS attack; SP12: user anonymity; SP13: untraceability; ✓: provides safety/functional features; ×: does not provides safety/functional features; △: not verified.

8.2. Communication Costs Comparison

We conducted a comparative analysis of communication costs between the related schemes [11,21,44–48] and the proposed scheme. Based on [11], we assume the bit lengths of hash function, timestamp, string, identity, random number, fuzzy extractor, and encryption operation to be 256, 32, 160, 160, 160, 8, and 256 bits, respectively. Therefore, during the MAKA phase of our proposed scheme, the exchanged message $M_1 = \{PID_i, B_1, B_2, V_1, TS_1\}$ requires (160 + 160 + 160 + 256 + 32 = 768 bits), message $M_2 = \{M_1, B_3, V_2, TS_2\}$ requires (768 + 160 + 256 + 32 = 1216 bits), message $M_3 = \{B_6, B_7, B_8, B_9, V_3, V_4, TS_3\}$ requires (160 + 160 + 160 + 160 + 256 + 256 + 32 = 1184 bits), and message $M_4 = \{B_8, V_5, TS_4\}$ requires (160 + 160 + 256 + 32 = 608 bits). Table 4 and Figure 8 present a summary of the communication costs for the associated schemes [11,21,44–48] and the proposed scheme.

Table 4. Comparison analysis of communication costs.

Scheme	Total Cost (bits)	Number of Messages
Martinez-Pelaez et al. [21]	4608 bits	6
Wu et al. (2020) [44]	4416 bits	5
Kang et al. [45]	4320 bits	4
Huang et al. [46]	3232 bits	4
Alam-Kumar [47]	2912 bits	4
Wu et al. (2023) [48]	3360 bits	4
Wu el al. [11]	4001 bits	5
Proposed	3776 bits	4

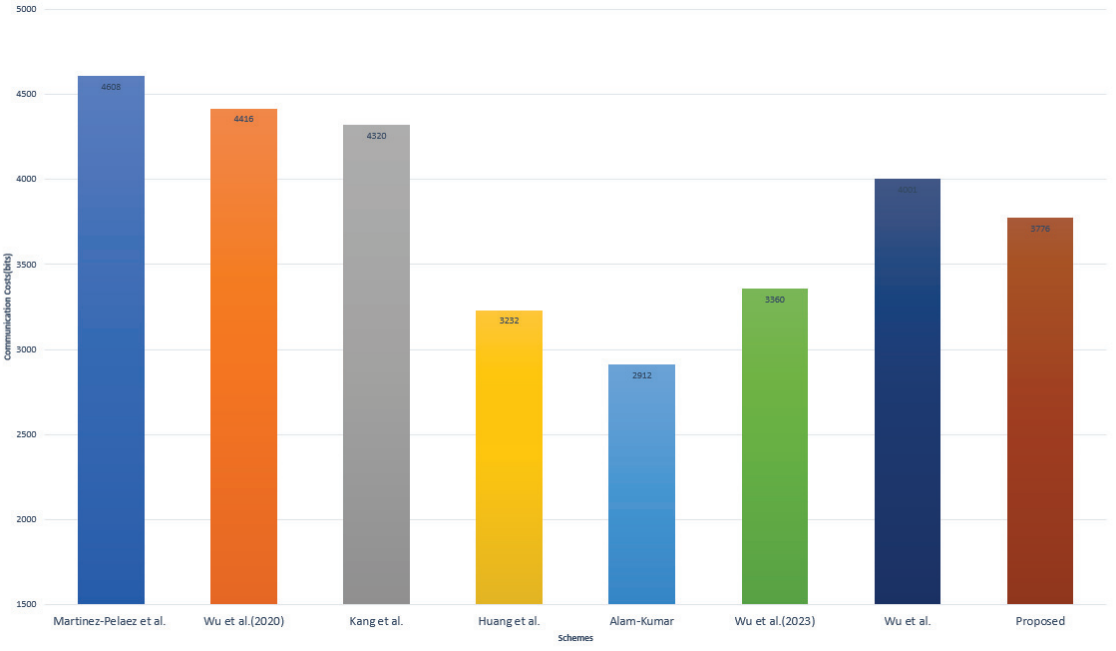


Figure 8. Communication costs comparison [7,11,17,21,40–48].

8.3. Computation Costs Comparison

We conducted a comparative analysis of computation costs for the AKA phase of the proposed scheme and related schemes [11,21,44–48]. Based on [49], we designed the environment for computing costs. The experimental environment and the performance of operation costs, including the minimum, maximum, and average values, are summarized in Table 5. We represent hash function as T_h and encryption/decryption operations of AES-256 as T_e. Using these values, we conducted a comparison of computation costs as shown in Table 6 and Figure 9.

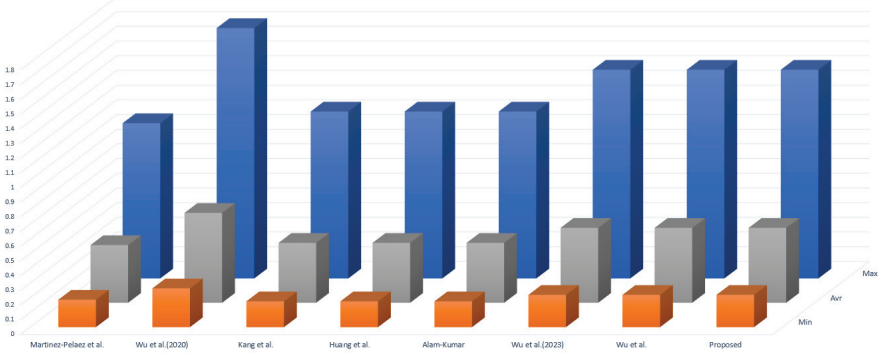


Figure 9. Computation costs comparison on user side devices [7,11,17,21,40–48].

Table 5. Hardware software enviroment and operation costs.

Hardware/Software	Operation	Max	Min	Average
Raspberry PI 4B with Linux Ubuntu 18.04.4 LTS with 64-bits, 8 GB, and MIRACL library	Hash function T_h	0.142 ms	0.022 ms	0.051 ms
	AES-256 T_e	0.021 ms	0.011 ms	0.012 ms

Table 6. Comparison analysis of user side computation costs.

Protocol	User Side	Max	Min	Average
[21]	$7T_h + 3T_e$	≈1.057 ms	≈0.187 ms	≈0.393 ms
[44]	$12T_h$	≈1.704 ms	≈0.264 ms	≈0.612 ms
[45]	$8T_h$	≈1.136 ms	≈0.176 ms	≈0.408 ms
[46]	$8T_h$	≈1.136 ms	≈0.176 ms	≈0.408 ms
[47]	$8T_h$	≈1.136 ms	≈0.176 ms	≈0.408 ms
[48]	$10T_h$	≈1.42 ms	≈0.22 ms	≈0.51 ms
[11]	$10T_h$	≈1.42 ms	≈0.22 ms	≈0.51 ms
Proposed	$10T_h$	≈1.42 ms	≈0.22 ms	≈0.51 ms

We can observe that the computational costs for users using the proposed scheme and users using Wu et al.'s scheme are the same. Next, we calculated the computational costs of the cloud server and control server for the proposed scheme and related schemes based on the environments provided in [49] as well. Table 7 represents the calculated computational costs for the proposed scheme and related schemes.

Table 7. Comparison analysis of cloud server side control server side computation costs.

Scheme	Cloud Server	Control Server	Total Average (ms)
[21]	$5T_h + 3T_e$	$21T_h + 2T_e$	≈1.386 ms
[44]	$8T_h$	$19T_h$	≈1.377 ms
[45]	$4T_h$	$11T_h$	≈0.765 ms
[46]	$4T_h$	$10T_h$	≈0.714 ms
[47]	$3T_h$	$6T_h$	≈0.459 ms
[48]	$5T_h$	$12T_h$	≈0.867 ms
[11]	$5T_h$	$13T_h$	≈0.918 ms
Proposed	$6T_h$	$16T_h$	≈1.122 ms

When comprehensively examining the results of the comparison with related schemes, we can elaborate as follows. Our proposed scheme offers more security elements compared to other schemes and is secure against various attacks such as insider attacks, impersonation attacks, stolen verification attacks, ESL attacks, privileged insider attacks, reply attacks, and offline password-guessing attacks. Simultaneously, it maintains reasonable user-side computation cost and communication cost suitable for the CloudIoT environment. However, it is noteworthy that to provide such robust security, additional computation operations on the server side have been introduced.

9. Conclusions

This study analyzed the key agreement protocol between cloud-enabled IoT devices and cloud servers as proposed by Wu et al. The scheme proposed by Wu et al. was found to be vulnerable to insider, privileged insiders, impersonation, and verification table leakage attacks and lacks user untraceability. In addition, it is inconvenient for users to update their passwords offline. To overcome these vulnerabilities and inconveniences, this study proposed a provably secure lightweight MAKA protocol for the cloud-based IoT environments.

The proposed protocol ensures safety against various attacks by preventing the exposure of critical parameters using user biometric information, and the cloud server's secret key. Furthermore, user untraceability was ensured by updating the user's pseudonym in every session and convenience was enhanced by adding an offline user password change and biometric template update phase. The safety of mutual authentication and the resulting session key was verified using the RoR model and BAN logic. Moreover, informal analysis was conducted to verify safety against attacks such as insider attacks, impersonation attacks, privileged attacks, ESL attacks, stolen verifier attacks and DoS attacks, while confirming security features such as user anonymity, untraceability, and perfect forward secrecy. The security features, communication costs, and computation costs of the proposed scheme were compared. This comparison demonstrated that the proposed scheme is rational in terms of communication and computation amounts in the cloud-based IoT environments, while being verified for safety.

In conclusion, the proposed scheme demonstrated robust safety and the ability to provide users with real-time services securely. Future research will focus on integrating the proposed scheme into real-world environments and various industrial settings where cloud-based IoT is applied.

Author Contributions: Conceptualization, S.J. and Y.P.; Methodology, S.J. and Y.P.; Formal analysis, Y.P.; Writing—original draft, S.J.; Writing—review editing, Y.P.; Supervision, Y.P.; Project administration, Y.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Bisa Research Grant of Keimyung University in 2022.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* **2013**, *29*, 1645–1660. [CrossRef]
2. Zhao, J.C.; Zhang, J.F.; Feng, Y.; Guo, J.X. The study and application of the IOT technology in agriculture. In Proceedings of the 2010 3rd International Conference on Computer Science and Information Technology, Chengdu, China, 9–11 July 2010; IEEE: Piscataway, NJ, USA, 2010; Volume 2, pp. 462–465.
3. Park, Y.; Park, Y. Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks. *Sensors* **2016**, *16*, 2123. [CrossRef] [PubMed]
4. Lee, S.; Kim, S.; Yu, S.; Jho, N.; Park, Y. Provably Secure PUF-Based Lightweight Mutual Authentication Scheme for Wireless Body Area Networks. *Electronics* **2022**, *11*, 3868. [CrossRef]
5. Park, Y.; Ryu, D.; Kwon, D.; Park, Y. Provably secure mutual authentication and key agreement scheme using PUF in internet of drones deployments. *Sensors* **2023**, *23*, 2034. [CrossRef] [PubMed]
6. Jadeja, Y.; Modi, K. Cloud computing-concepts, architecture and challenges. In Proceedings of the 2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET), Nagercoil, India, 21–22 March 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 877–880.
7. Dinh, T.; Kim, Y.; Lee, H. A location-based interactive model of internet of things and cloud (IoT-Cloud) for mobile cloud computing applications. *Sensors* **2017**, *17*, 489. [CrossRef] [PubMed]
8. Babu, S.M.; Lakshmi, A.J.; Rao, B.T. A study on cloud based Internet of Things: CloudIoT. In Proceedings of the 2015 Global Conference on Communication Technologies (GCCT), Thuckalay, India, 23–24 April 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 60–65.
9. Zargar, S.; Shahidinejad, A.; Ghobaei-Arani, M. A lightweight authentication protocol for IoT-based cloud environment. *Int. J. Commun. Syst.* **2021**, *34*, e4849. [CrossRef]
10. Kim, M.; Yu, S.; Lee, J.; Park, Y.; Park, Y. Design of secure protocol for cloud-assisted electronic health record system using blockchain. *Sensors* **2020**, *20*, 2913. [CrossRef]
11. Wu, T.Y.; Meng, Q.; Kumari, S.; Zhang, P. Rotating behind security: A lightweight authentication protocol based on iot-enabled cloud computing environments. *Sensors* **2022**, *22*, 3858. [CrossRef]
12. Shouqi, C.; Wanrong, L.; Liling, C.; Xin, H.; Zhiyong, J. An improved authentication protocol using smart cards for the Internet of Things. *IEEE Access* **2019**, *7*, 157284–157292. [CrossRef]

13. Nikooghadam, M.; Jahantigh, R.; Arshad, H. A lightweight authentication and key agreement protocol preserving user anonymity. *Multimed. Tools Appl.* **2017**, *76*, 13401–13423. [CrossRef]
14. Kumari, S.; Chaudhry, S.A.; Wu, F.; Li, X.; Farash, M.S.; Khan, M.K. An improved smart card based authentication scheme for session initiation protocol. *Peer Netw. Appl.* **2017**, *10*, 92–105. [CrossRef]
15. Limbasiya, T.; Soni, M.; Mishra, S.K. Advanced formal authentication protocol using smart cards for network applicants. *Comput. Electr. Eng.* **2018**, *66*, 50–63. [CrossRef]
16. Chandrakar, P.; Om, H. An extended ECC-based anonymity-preserving 3-factor remote authentication scheme usable in TMIS. *Int. J. Commun. Syst.* **2018**, *31*, e3540. [CrossRef]
17. Sharma, G.; Kalra, S. A lightweight multi-factor secure smart card based remote user authentication scheme for cloud-IoT applications. *J. Inf. Secur. Appl.* **2018**, *42*, 95–106. [CrossRef]
18. Gope, P.; Sikdar, B. Lightweight and privacy-preserving two-factor authentication scheme for IoT devices. *IEEE Internet Things J.* **2018**, *6*, 580–589. [CrossRef]
19. Siddiqui, Z.; Gao, J.; Khan, M.K. An improved lightweight PUF–PKI digital certificate authentication scheme for the Internet of Things. *IEEE Internet Things J.* **2022**, *9*, 19744–19756. [CrossRef]
20. Zhou, L.; Li, X.; Yeh, K.H.; Su, C.; Chiu, W. Lightweight IoT-based authentication scheme in cloud computing circumstance. *Future Gener. Comput. Syst.* **2019**, *91*, 244–251. [CrossRef]
21. Martínez-Peláez, R.; Toral-Cruz, H.; Parra-Michel, J.R.; García, V.; Mena, L.J.; Félix, V.G.; Ochoa-Brust, A. An enhanced lightweight IoT-based authentication scheme in cloud computing circumstances. *Sensors* **2019**, *19*, 2098. [CrossRef]
22. Alzahrani, B.A.; Chaudhry, S.A.; Barnawi, A.; Al-Barakati, A.; Shon, T. An anonymous device to device authentication protocol using ECC and self certified public keys usable in Internet of Things based autonomous devices. *Electronics* **2020**, *9*, 520. [CrossRef]
23. Islam, S.H.; Biswas, G. Design of two-party authenticated key agreement protocol based on ECC and self-certified public keys. *Wirel. Pers. Commun.* **2015**, *82*, 2727–2750. [CrossRef]
24. Mandal, S.; Mohanty, S.; Majhi, B. Cryptanalysis and enhancement of an anonymous self-certified key exchange protocol. *Wirel. Pers. Commun.* **2018**, *99*, 863–891. [CrossRef]
25. Chen, Y.; López, L.; Martínez, J.F.; Castillejo, P. A lightweight privacy protection user authentication and key agreement scheme tailored for the Internet of Things environment: LightPriAuth. *J. Sensors* **2018**, *2018*, 7574238. . [CrossRef]
26. Lee, J.; Yu, S.; Kim, M.; Park, Y.; Das, A.K. On the design of secure and efficient three-factor authentication protocol using honey list for wireless sensor networks. *IEEE Access* **2020**, *8*, 107046–107062. [CrossRef]
27. Yu, Y.; Hu, L.; Chu, J. A secure authentication and key agreement scheme for IoT-based cloud computing environment. *Symmetry* **2020**, *12*, 150. [CrossRef]
28. He, D.; Zeadally, S.; Kumar, N.; Wu, W. Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server architectures. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 2052–2064. [CrossRef]
29. Tsai, J.L.; Lo, N.W. A privacy-aware authentication scheme for distributed mobile cloud computing services. *IEEE Syst. J.* **2015**, *9*, 805–815. [CrossRef]
30. Kumari, A.; Kumar, V.; Abbasi, M.Y.; Kumari, S.; Chaudhary, P.; Chen, C.M. Csef: Cloud-based secure and efficient framework for smart medical system using ecc. *IEEE Access* **2020**, *8*, 107838–107852. [CrossRef]
31. Bhuarya, P.; Chandrakar, P.; Ali, R.; Sharaff, A. An enhanced authentication scheme for Internet of Things and cloud based on elliptic curve cryptography. *Int. J. Commun. Syst.* **2021**, *34*, e4834. [CrossRef]
32. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
33. Canetti, R.; Krawczyk, H. Universally composable notions of key exchange and secure channels. In Proceedings of the Advances in Cryptology—EUROCRYPT 2002: International Conference on the Theory and Applications of Cryptographic Techniques Amsterdam, The Netherlands, 28 April–2 May 2002; Springer: Berlin/Heidelberg, Germany, 2002; pp. 337–351.
34. Abdalla, M.; Fouque, P.A.; Pointcheval, D. Password-based authenticated key exchange in the three-party setting. In Proceedings of the Public Key Cryptography-PKC 2005: 8th International Workshop on Theory and Practice in Public Key Cryptography, Les Diablerets, Switzerland, 23–26 January 2005; Springer: Berlin/Heidelberg, Germany, 2005, pp. 65–84.
35. Wang, D.; Cheng, H.; Wang, P.; Huang, X.; Jian, G. Zipf's law in passwords. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2776–2791. [CrossRef]
36. Kwon, D.K.; Yu, S.J.; Lee, J.Y.; Son, S.H.; Park, Y.H. WSN-SLAP: Secure and lightweight mutual authentication protocol for wireless sensor networks. *Sensors* **2021**, *21*, 936. [CrossRef]
37. Yu, S.; Lee, J.; Sutrala, A.K.; Das, A.K.; Park, Y. LAKA-UAV: Lightweight authentication and key agreement scheme for cloud-assisted Unmanned Aerial Vehicle using blockchain in flying ad hoc networks. *Comput. Netw.* **2023**, *224*, 109612. [CrossRef]
38. Kim, M.; Lee, J.; Oh, J.; Kwon, D.; Park, K.; Park, Y.; Park, K.H. A Secure Batch Authentication Scheme for Multiaccess Edge Computing in 5G-Enabled Intelligent Transportation System. *IEEE Access* **2022**, *10*, 96224–96238. [CrossRef]
39. Boyko, V.; MacKenzie, P.; Patel, S. Provably secure password-authenticated key exchange using Diffie-Hellman. In Proceedings of the Advances in Cryptology—EUROCRYPT 2000: International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, 14–18 May 2000; Proceedings 19; Springer: Berlin/Heidelberg, Germany, 2000; pp. 156–171.
40. Burrows, M.; Abadi, M.; Needham, R. A logic of authentication. *ACM Trans. Comput. Syst. (TOCS)* **1990**, *8*, 18–36. [CrossRef]
41. Kwon, D.; Son, S.; Park, Y.; Kim, H.; Park, Y.; Lee, S.; Jeon, Y. Design of secure handover authentication scheme for urban air mobility environments. *IEEE Access* **2022**, *10*, 42529–42541. [CrossRef]

42. Son, S.; Kwon, D.; Lee, S.; Jeon, Y.; Das, A.K.; Park, Y. Design of Secure and Lightweight Authentication Scheme for UAV-Enabled Intelligent Transportation Systems using Blockchain and PUF. *IEEE Access* **2023**, *11*, 60240–60253. [CrossRef]
43. Cho, Y.; Oh, J.; Kwon, D.; Son, S.; Yu, S.; Park, Y.; Park, Y. A secure three-factor authentication protocol for e-governance system based on multiserver environments. *IEEE Access* **2022**, *10*, 74351–74365. [CrossRef]
44. Wu, H.L.; Chang, C.C.; Zheng, Y.Z.; Chen, L.S.; Chen, C.C. A secure IoT-based authentication system in cloud computing environment. *Sensors* **2020**, *20*, 5604. [CrossRef]
45. Kang, B.; Han, Y.; Qian, K.; Du, J. Analysis and improvement on an authentication protocol for IoT-enabled devices in distributed cloud computing environment. *Math. Probl. Eng.* **2020**, . [CrossRef]
46. Huang, H.; Lu, S.; Wu, Z.; Wei, Q. An efficient authentication and key agreement protocol for IoT-enabled devices in distributed cloud computing architecture. *EURASIP J. Wirel. Commun. Netw.* **2021**, *2021*, 150. [CrossRef]
47. Alam, I.; Kumar, M. A novel protocol for efficient authentication in cloud-based IoT devices. *Multimed. Tools Appl.* **2022**, *81*, 13823–13843. [CrossRef]
48. Wu, T.Y.; Kong, F.; Meng, Q.; Kumari, S.; Chen, C.M. Rotating behind security: An enhanced authentication protocol for IoT-enabled devices in distributed cloud computing architecture. *EURASIP J. Wirel. Commun. Netw.* **2023**, *2023*, 36. [CrossRef]
49. Park, K.; Park, Y. IAKA-CIOT: An improved authentication and key agreement scheme for cloud enabled internet of things using physical unclonable function. *Sensors* **2022**, *22*, 6264. [CrossRef] [PubMed]

sensors

MDPI

Article

Hierarchical Controlled Hybrid Quantum Communication Based on Six-Qubit Entangled States in IoT

Xiaoyu Hua, Dongfen Li *, You Fu, Yonghao Zhu, Yangyang Jiang, Jie Zhou, Xiaolong Yang and Yuqiao Tan

College of Computer Science and Cyber Security (Oxford Brookes College), Chengdu University of Technology, Chengdu 610059, China; huaxiaoyu@stu.cdut.edu.cn (X.H.); 2022050889@stu.cdut.edu.cn (Y.F.); 2022050893@stu.cdut.edu.cn (Y.Z.); 2022050894@stu.cdut.edu.cn (Y.J.); 2021020861@stu.cdut.edu.cn (J.Z.); 2021050831@stu.cdut.edu.cn (X.Y.); 2021020897@stu.cdut.edu.cn (Y.T.)
* Correspondence: lidongfen17@cdut.edu.cn

Abstract: The rapid development and extensive application of the Internet of Things (IoT) have brought new challenges and opportunities to the field of communication. By integrating quantum secure communication with the IoT, we can provide a higher level of security and privacy protection to counteract security threats in the IoT. In this paper, a hybrid quantum communication scheme using six-qubit entangled states as a channel is proposed for specific IoT application scenarios. This scheme achieves hierarchical control of communication protocols on a single quantum channel. In the proposed scheme, device A transmits data to device B through quantum teleportation, while device B issues control commands to device A through remote quantum state preparation technology. These two tasks are controlled by control nodes C and D, respectively. The transmission of information from device A to device B is a relatively less important task, which can be solely controlled by control node C. On the other hand, issuing control commands from device B to device A is a more crucial task requiring joint control from control nodes C and D. This paper describes the proposed scheme and conducts simulation experiments using IBM's Qiskit Aer quantum computing simulator. The results demonstrate that the fidelity of the quantum teleportation protocol (QTP) and the remote state preparation protocol (RSP) reach an impressive value of 0.999, fully validating the scheme's feasibility. Furthermore, the factors affecting the fidelity of the hybrid communication protocol in an IoT environment with specific quantum noise are analyzed. By combining the security of quantum communication with the application scenarios of the IoT, this paper presents a new possibility for IoT communication.

Keywords: IoT communication; hierarchical controlled hybrid quantum communication; six-qubit entangled state; quantum secure communication; quantum noise

Citation: Hua, X.; Li, D.; Fu, Y.; Zhu, Y.; Jiang, Y.; Zhou, J.; Yang, X.; Tan, Y. Hierarchical Controlled Hybrid Quantum Communication Based on Six-Qubit Entangled States in IoT. *Sensors* **2023**, *23*, 9111. https://doi.org/10.3390/s23229111

Academic Editor: Jian Li

Received: 2 October 2023
Revised: 2 November 2023
Accepted: 6 November 2023
Published: 10 November 2023

1. Introduction

The Internet of Things (IoT) enables communication and data exchange between various devices and systems by connecting them to the Internet. This seamless connectivity allows devices to perform real-time monitoring, remote control, and data sharing, bringing more convenient services and functions to users. However, IoT devices have limited computing power, which poses substantial security challenges for IoT deployment. In the era of quantum communication, these challenges become more severe because some attackers may have quantum computing capabilities, making IoT devices more vulnerable. Unlike traditional secure communication schemes, quantum communication does not rely on computational complexity to guarantee communication security. However, it uses the unique physical properties of quantum mechanics to transmit information, thus achieving secure communication. In recent years, quantum information technology has developed rapidly. In 2022, Xie et al. [1] proposed an asynchronous measurement-device-independent quantum key distribution protocol. In 2023, Zhou et al. [2] implemented an innovative measurement-device-independent quantum key distribution (MDI-QKD) scheme. In 2022,

Yin et al. [3] proposed an efficient quantum digital signature (QDS) protocol. In 2023, Liao et al. [4] proposed a continuous-variable quantum secret-sharing scheme based on multi-ring discrete modulation. In 2023, Zhou et al. [5] proposed a hybrid quantum-classical generative adversarial network (HQCGAN). In 2024, Gong et al. [6] designed a quantum convolutional neural network (QCNN) based on pure variational quantum circuits inspired by convolutional neural networks. Quantum communication technology provides new solutions for IoT security issues and application scenarios [7–10]. In 2021, Maha et al. [11] proposed a novel method that uses quantum key distribution (QKD) technology to encrypt data between IoT devices and servers, which is simple and effective. In the same year, Rajesh Kumar et al. [12] designed a quantum-communication-based IoT security architecture (QIoTSA) and analyzed its advantages and challenges. In 2022, Liu et al. [13] used quantum key distribution (QKD) technology to store quantum keys in IoT devices in advance and used them to encrypt and decrypt IoT-sensitive data.

Quantum teleportation (QTP) and remote state preparation (RSP) are quantum secure communication techniques that use quantum entanglement properties. QTP was first proposed by Bennett and Brassard [14] in 1993 and later experimentally verified by Anton Zeilinger and others in 1997. Since then, QTP has attracted the attention of many researchers and has produced many theoretical and experimental advances and variations in the past thirty years [15–18]. In 1998, Karlsson and Bourennane [19] pioneered the idea of controlled quantum teleportation (CQTP). In 2008, Chen et al. [20] designed a bidirectional CQTP scheme based on four-qubit entangled states. In 2015, Chen et al. [21] proposed a CQTP scheme based on three-particle partially entangled states. In 2018, Zhou et al. [22] proposed an efficient CQTP scheme based on two-qubit entangled states, which requires the control of a supervisor. In 2020, Li et al. [23] proposed a theoretical CQTP scheme based on seven-qubit entangled states. Remote quantum state preparation (RSP) is an essential branch of quantum communication technology. It differs from quantum teleportation (QTP) in that the sender knows the quantum state to be prepared, while the receiver does not. The sender can perform specific measurements on a certain qubit based on their information, thereby helping the remote receiver recover the original state. In 2000, Lo et al. [24] first proposed the concept of RSP. In recent years, some people have proposed a bidirectional hybrid controlled quantum communication (BHCQC) scheme that combines QTP and RSP. In 2017, Fang et al. [25] used a five-qubit Brown state as the channel to give a deterministic BHCQC protocol for any single-qubit state. In the same year, Sang et al. [26] also used a five-qubit cluster state as the channel to realize a deterministic BHCQC protocol for any single-qubit state. In 2018, Ma et al. [27] used a six-qubit entangled state to perform BHCQC. Hierarchical quantum communication is a novel means of multiparty quantum communication in recent years. It uses different levels of quantum entangled states to achieve quantum information transmission among multiple parties. In 2010, Wang et al. [28] proposed the first scheme of multiparty hierarchical controlled quantum teleportation (HCQTP) using four-particle entangled states. In 2013, Shukla and Pathak et al. [29] performed similar work using four-particle entangled states. In 2017, Shukla et al. [30] proposed a protocol for the hierarchical joint remote preparation of any single-qubit state using five-particle cluster states. In 2018, Chen et al. [31] realized a deterministic hierarchical remote preparation of any single-qubit state using six-particle partially entangled states. In 2020, Wang et al. [32] used four-particle states as the channel to design a single-qubit state hierarchical CQTP scheme. In 2021, Ma and Wang [33] used two five-particle cluster states to carry out a hierarchical controlled remote preparation of two-qubit states.

In the previous BHCQC schemes, quantum teleportation and remote quantum state preparation were regarded as equally important, lacking control flexibility, and making them unsuitable for specific IoT scenarios. In order to migrate BHCQC to the IoT, this study attempts to introduce the concept of hierarchical control into the hybrid communication protocol, assigning different levels of importance to quantum teleportation and remote quantum state preparation. Thus, this paper proposes a hierarchical controlled hybrid

quantum communication (HCHQC) scheme utilizing a six-qubit entangled state as the quantum channel. Alice wishes to transmit an unknown quantum state to Bob, while simultaneously, Bob desires to prepare a known quantum state on Alice's side remotely. This process is controlled by supervisors Charlie and David. The state Alice wants to transmit to Bob is a less critical task and can be solely controlled by supervisor Charlie. However, the state that Bob intends to prepare for Alice remotely is a more important task and must be jointly controlled by supervisors Charlie and David.

The remaining structure of this paper is as follows: Section 2 presents the application scenarios of the HCHQC protocol in the IoT and provides a detailed description of the HCHQC protocol. In Section 3, we employ IBM's Qiskit Aer quantum computing simulator to conduct simulation experiments to validate the proposed scheme's feasibility. In Section 4, we introduce the presence of amplitude-damping noise and phase-damping noise environments and analyze the factors that impact the fidelity of the HCHQC protocol in specific noise environments. Finally, Section 5 provides a comprehensive summary of this paper.

2. Hierarchical Controlled Hybrid Quantum Communication Scheme in the IoT

2.1. Application Scenario

The Internet of Things (IoT) is a framework that uses internet technology to connect various smart devices and achieve information sharing and interaction. The IoT is widely used in many fields, such as smart homes, intelligent transportation, smart medical care, etc. However, the IoT also faces security issues, such as device identity authentication, data confidentiality and integrity, network defence, etc. To cope with these challenges, the quantum secret communication protocol provides a new solution. The quantum secret communication protocol is a technology that uses quantum mechanics principles, such as quantum superposition, quantum entanglement, quantum no-cloning, etc., to achieve information-secure transmission and encryption. It can effectively resist eavesdropping and interference from third parties and ensure the security and reliability of IoT communication. This paper will discuss a scenario that uses a quantum secret communication protocol to protect IoT communication security.

Device A and device B are edge devices in the IoT that are close to the data source or the user and can process and analyze data locally. Device A is equipped with multiple sensors, which can collect environmental data, such as temperature, pressure, etc. Device B can receive the environmental status of other edge devices and issue corresponding instructions according to the status change. Control node C and control node D are cloud computing devices that are responsible for the authentication and control services of the IoT and ensure the security of the IoT. These devices transmit classical data through classical network channels and exchange quantum information through quantum channels, as shown in Figure 1.

Device A must securely transmit the environmental state data (such as temperature, pressure, etc.) to another device, device B, within the IoT system. This transmission of the environmental state exists at a general level of security within the entire IoT system; thus necessitating authorization solely from control node C. Device A can encode the intended state information into a quantum state, denoted as state a, and employ controlled quantum teleportation to transmit this quantum state to device B. By measuring the received quantum state a, device B can retrieve the environmental state information from device A's side. Conversely, device B controls device A, allowing it to make determinations based on the transmitted environmental state data and issue control commands to device A. Considering the security of the IoT system, this manipulation operates at a higher level of protection throughout the entire IoT system, requiring joint authorization from control nodes C and D. Device B can encode the desired instructions into a quantum state, denoted as state b, and utilizing remote quantum state preparation under the shared control of control nodes C and D, create a quantum state on device A's side that matches the state of quantum state b. Device A can measure the quantum state to decode the instruction.

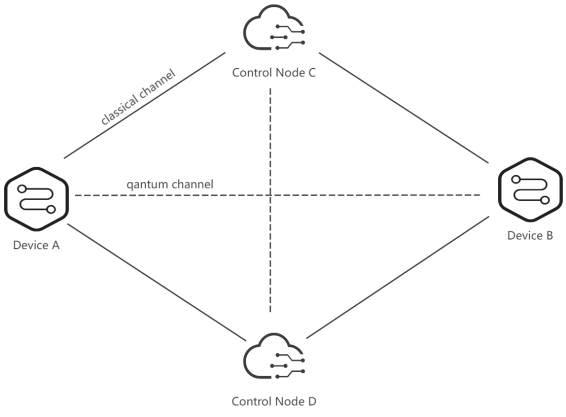


Figure 1. Information exchange and control between different devices in the IoT system through classical and quantum communication channels.

Based on the scenario of using a quantum secret communication protocol to ensure the security of IoT communication, this paper proposes an HCHQC protocol that uses a six-qubit entangled state as the quantum channel and realizes the layered control of two different communication protocols, quantum invisible transmission, and remote quantum state preparation, on one quantum channel. In the communication scheme, device A and device B represent Alice and Bob, respectively, and control node C and control node D represent Charlie and David, respectively. The following Section 2, will detail this hybrid quantum communication scheme that uses a six-qubit entangled state as the quantum channel for the IoT.

2.2. Specific Communication Plan

The communication scheme introduced in this section has the following advantages: (1) flexible communication, which can be controlled according to the task requirements and security levels; (2) quantum resource saving, which can complete two tasks with one six-qubit entangled state; (3) high communication efficiency, which can transmit and reconstruct states with a small amount of classical information and quantum operations. Next, we will explain this scheme in detail.

Assume Alice has an arbitrary unknown single-qubit state, denoted as:

$$|\varphi\rangle_a = \alpha_1 |0\rangle + \alpha_2 |1\rangle \tag{1}$$

where α_1 and α_2 are complex numbers satisfying $|\alpha_1|^2 + |\alpha_2|^2 = 1$.

Alice wants to teleport an unknown single qubit state $|\varphi\rangle$ to Bob using quantum teleportation. Meanwhile, Bob wants to prepare a known single qubit state $|\phi\rangle$ on Alice's side through remote state preparation. The state of $|\phi\rangle$ can be written as:

$$|\phi\rangle_b = \beta_1 |0\rangle + \beta_2 |1\rangle \tag{2}$$

where β_1 and β_2 are real numbers satisfying $|\beta_1|^2 + |\beta_2|^2 = 1$.

At this point, assuming that the quantum channel shared by Alice, Bob, Charlie, and David is a six-qubit entangled quantum state, it can be expressed as:

$$|\psi\rangle_{A_1B_1A_2B_2CD} = \frac{1}{2}(|000000\rangle - |001111\rangle + |110010\rangle - |111101\rangle)_{A_1B_1A_2B_2CD} \tag{3}$$

Among them, qubits A_1 and A_2 belong to Alice, qubits B_1 and B_2 belong to Bob, qubit C belongs to Charlie, and qubit D belongs to David. The quantum state of the whole system can be expressed as follows:

$$\begin{aligned}|\tau\rangle_{aA_1B_1A_2B_2CD} &= |\varphi\rangle_a \otimes |\psi\rangle_{A_1B_1A_2B_2CD} \\ &= \frac{1}{2}(\alpha_1|0\rangle + \alpha_2|1\rangle)_a \\ &\quad \otimes (|000000\rangle - |001111\rangle + |110010\rangle - |111101\rangle)_{A_1B_1A_2B_2CD}\end{aligned} \tag{4}$$

where $\otimes$ represents the tensor product. We now introduce the hybrid quantum communication scheme in four steps.

Step 1:

Alice performs a joint measurement on two qubits a and A_1 she possesses, using the Bell measurement basis given by the following expression:

$$\begin{aligned}|\Psi^1\rangle_{a,A_1} &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\ |\Psi^2\rangle_{a,A_1} &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \\ |\Psi^3\rangle_{a,A_1} &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \\ |\Psi^4\rangle_{a,A_1} &= \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)\end{aligned} \tag{5}$$

Alice can obtain one of the four possible measurement results with equal probability, and the remaining qubits B_1, A_2, B_2, C, and D will collapse into one of the corresponding four states $\langle\Psi^1|\,|\tau\rangle$, $\langle\Psi^2|\,|\tau\rangle$, $\langle\Psi^3|\,|\tau\rangle$, or $\langle\Psi^4|\,|\tau\rangle$, as shown in Table 1.

Table 1. Alice's measurement results and the corresponding state of the remaining qubits.

Alice's Measurement Result	State of the Remaining Qubits
$\|\Psi^1\rangle_{a,A_1} = \frac{1}{\sqrt{2}}(\|00\rangle + \|11\rangle)$	$\langle\Psi^1\|\,\|\tau\rangle = \frac{1}{2\sqrt{2}}(\alpha_1\|00000\rangle - \alpha_1\|01111\rangle + \alpha_2\|10010\rangle - \alpha_2\|11101\rangle)_{B_1A_2B_2CD}$
$\|\Psi^2\rangle_{a,A_1} = \frac{1}{\sqrt{2}}(\|00\rangle - \|11\rangle)$	$\langle\Psi^2\|\,\|\tau\rangle = \frac{1}{2\sqrt{2}}(\alpha_1\|00000\rangle - \alpha_1\|01111\rangle - \alpha_2\|10010\rangle + \alpha_2\|11101\rangle)_{B_1A_2B_2CD}$
$\|\Psi^3\rangle_{a,A_1} = \frac{1}{\sqrt{2}}(\|01\rangle + \|10\rangle)$	$\langle\Psi^3\|\,\|\tau\rangle = \frac{1}{2\sqrt{2}}(\alpha_1\|10010\rangle - \alpha_1\|11101\rangle + \alpha_2\|00000\rangle - \alpha_2\|01111\rangle)_{B_1A_2B_2CD}$
$\|\Psi^4\rangle_{a,A_1} = \frac{1}{\sqrt{2}}(\|01\rangle - \|10\rangle)$	$\langle\Psi^4\|\,\|\tau\rangle = \frac{1}{2\sqrt{2}}(\alpha_1\|10010\rangle - \alpha_1\|11101\rangle - \alpha_2\|00000\rangle + \alpha_2\|01111\rangle)_{B_1A_2B_2CD}$

Step 2:

At the same time, Bob performs projection measurement on his qubit B_2, and the measurement basis is:

$$\begin{aligned}|\Phi^1\rangle_{B_2} &= \beta_1|0\rangle + \beta_2|1\rangle \\ |\Phi^2\rangle_{B_2} &= \beta_2|0\rangle - \beta_1|1\rangle\end{aligned} \tag{6}$$

Since the coefficients β_1 and β_2 are known to Bob, such a set of measurement bases can be obtained. The projection measurements and the states of the remaining qubits are shown in Table 2 below.

Table 2. The measurement results of Alice and Bob and the state of the remaining qubits.

Alice's Measurement Result	Bob's Measurement Result	State of the Remaining Qubits
$\|\Psi^1\rangle_{a,A_1} = \frac{1}{\sqrt{2}}(\|00\rangle + \|11\rangle)$	$\|\Phi^1\rangle_{B_2}$	$\langle\Phi^1\|\langle\Psi^1\|\|\tau\rangle = \frac{1}{2\sqrt{2}}(\alpha_1\beta_1\|0000\rangle - \alpha_1\beta_2\|0111\rangle + \alpha_2\beta_1\|1010\rangle - \alpha_2\beta_2\|1101\rangle)_{B_1A_2CD}$
	$\|\Phi^2\rangle_{B_2}$	$\langle\Phi^2\|\langle\Psi^1\|\|\tau\rangle = \frac{1}{2\sqrt{2}}(\alpha_1\beta_2\|0000\rangle + \alpha_1\beta_1\|0111\rangle + \alpha_2\beta_2\|1010\rangle + \alpha_2\beta_1\|1101\rangle)_{B_1A_2CD}$
$\|\Psi^2\rangle_{a,A_1} = \frac{1}{\sqrt{2}}(\|00\rangle - \|11\rangle)$	$\|\Phi^1\rangle_{B_2}$	$\langle\Phi^1\|\langle\Psi^2\|\|\tau\rangle = \frac{1}{2\sqrt{2}}(\alpha_1\beta_1\|0000\rangle - \alpha_1\beta_2\|0111\rangle - \alpha_2\beta_1\|1010\rangle + \alpha_2\beta_2\|1101\rangle)_{B_1A_2CD}$
	$\|\Phi^2\rangle_{B_2}$	$\langle\Phi^2\|\langle\Psi^2\|\|\tau\rangle = \frac{1}{2\sqrt{2}}(\alpha_1\beta_2\|0000\rangle + \alpha_1\beta_1\|0111\rangle - \alpha_2\beta_2\|1010\rangle - \alpha_2\beta_1\|1101\rangle)_{B_1A_2CD}$
$\|\Psi^3\rangle_{a,A_1} = \frac{1}{\sqrt{2}}(\|01\rangle + \|10\rangle)$	$\|\Phi^1\rangle_{B_2}$	$\langle\Phi^1\|\langle\Psi^3\|\|\tau\rangle = \frac{1}{2\sqrt{2}}(\alpha_1\beta_1\|1010\rangle - \alpha_1\beta_2\|1101\rangle + \alpha_2\beta_1\|0000\rangle - \alpha_2\beta_2\|0111\rangle)_{B_1A_2CD}$
	$\|\Phi^2\rangle_{B_2}$	$\langle\Phi^2\|\langle\Psi^3\|\|\tau\rangle = \frac{1}{2\sqrt{2}}(\alpha_1\beta_2\|1010\rangle + \alpha_1\beta_1\|1101\rangle + \alpha_2\beta_2\|0000\rangle + \alpha_2\beta_1\|0111\rangle)_{B_1A_2CD}$
$\|\Psi^4\rangle_{a,A_1} = \frac{1}{\sqrt{2}}(\|01\rangle - \|10\rangle)$	$\|\Phi^1\rangle_{B_2}$	$\langle\Phi^1\|\langle\Psi^4\|\|\tau\rangle = \frac{1}{2\sqrt{2}}(\alpha_1\beta_1\|1010\rangle - \alpha_1\beta_2\|1101\rangle - \alpha_2\beta_1\|0000\rangle + \alpha_2\beta_2\|0111\rangle)_{B_1A_2CD}$
	$\|\Phi^2\rangle_{B_2}$	$\langle\Phi^2\|\langle\Psi^4\|\|\tau\rangle = \frac{1}{2\sqrt{2}}(\alpha_1\beta_2\|1010\rangle + \alpha_1\beta_1\|1101\rangle - \alpha_2\beta_2\|0000\rangle - \alpha_2\beta_1\|0111\rangle)_{B_1A_2CD}$

Step 3:

Alice needs Charlie's help to complete the task of quantum teleporting the quantum state $|\varphi\rangle$ to Bob using quantum entanglement. Suppose Charlie agrees to assist Alice and Bob. In that case, he needs to perform a von Neumann measurement on his qubit C on the basis $\{|\nu^1\rangle, |\nu^2\rangle\}$ and communicate the measurement result to Bob through the classical communication channel. The measurement basis $\{|\nu^1\rangle, |\nu^2\rangle\}$ is as follows:

$$\begin{aligned} |\nu^1\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |\nu^2\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{aligned} \tag{7}$$

Now suppose a situation exists where, before Charlie performs a single-qubit von Neumann measurement, the measurement results of Alice and Bob are $|\Psi^1\rangle$ and $|\Phi^1\rangle$, respectively, then the remaining qubits will collapse to the state $(\alpha_1\beta_1|0000\rangle - \alpha_1\beta_2|0111\rangle + \alpha_2\beta_1|1010\rangle - \alpha_2\beta_2|1101\rangle)_{B_1A_2CD}$. After Charlie performs the von Neumann measurement on qubit C, if the obtained measurement result is $|\nu^2\rangle$, the state of the remaining qubits will collapse to:

$$\langle\nu^2|\langle\Phi^1|\langle\Psi^1||\tau\rangle = (\alpha_1\beta_1|000\rangle + \alpha_1\beta_2|011\rangle - \alpha_2\beta_1|100\rangle - \alpha_2\beta_2|111\rangle)_{B_1A_2D} \tag{8}$$

We can go one step further and write the equation as follows:

$$\langle v^2| \langle \Phi^1| \langle \Psi^1| |\tau\rangle = (\alpha_1 |0\rangle - \alpha_2 |1\rangle)_{B_1} \otimes (\beta_1 |00\rangle + \beta_2 |11\rangle)_{A_2D} \quad (9)$$

It can be clearly seen that after Bob learns the measurement result of Charlie through the classical channel, he only needs to perform the σ_z gate operation on the qubit B_1 to correct the quantum state. Then he can obtain the quantum state $|\varphi\rangle$. However, the qubits A_2 and D are still entangled, so Alice cannot obtain the state $|\phi\rangle$ prepared remotely by Bob. The other cases and the unitary operations corresponding to each case are shown in Table 3.

Table 3. Charlie's measurement results and Bob's unitary operation corresponding to each result.

State before Charlie Measured	Charlie's Measurement Result	State of the Remaining Qubits	Bob's Operation
$\langle \Phi^1| \langle \Psi^1| |\tau\rangle$	$\|v^1\rangle$	$(\alpha_1 \|0\rangle + \alpha_2 \|1\rangle)B_1 \otimes (\beta_1 \|00\rangle - \beta_2 \|11\rangle)A_2D$	I
	$\|v^2\rangle$	$(\alpha_1 \|0\rangle - \alpha_2 \|1\rangle)B_1 \otimes (\beta_1 \|00\rangle + \beta_2 \|11\rangle)A_2D$	σ_z
$\langle \Phi^2| \langle \Psi^1| |\tau\rangle$	$\|v^1\rangle$	$(\alpha_1 \|0\rangle + \alpha_2 \|1\rangle)B_1 \otimes (\beta_2 \|00\rangle + \beta_1 \|11\rangle)A_2D$	I
	$\|v^2\rangle$	$(\alpha_1 \|0\rangle - \alpha_2 \|1\rangle)B_1 \otimes (\beta_2 \|00\rangle - \beta_1 \|11\rangle)A_2D$	σ_z
$\langle \Phi^1| \langle \Psi^2| |\tau\rangle$	$\|v^1\rangle$	$(\alpha_1 \|0\rangle - \alpha_2 \|1\rangle)B_1 \otimes (\beta_1 \|00\rangle - \beta_2 \|11\rangle)A_2D$	σ_z
	$\|v^2\rangle$	$(\alpha_1 \|0\rangle + \alpha_2 \|1\rangle)B_1 \otimes (\beta_1 \|00\rangle + \beta_2 \|11\rangle)A_2D$	I
$\langle \Phi^2| \langle \Psi^2| |\tau\rangle$	$\|v^1\rangle$	$(\alpha_1 \|0\rangle - \alpha_2 \|1\rangle)B_1 \otimes (\beta_2 \|00\rangle + \beta_1 \|11\rangle)A_2D$	σ_z
	$\|v^2\rangle$	$(\alpha_1 \|0\rangle + \alpha_2 \|1\rangle)B_1 \otimes (\beta_2 \|00\rangle - \beta_1 \|11\rangle)A_2D$	I
$\langle \Phi^1| \langle \Psi^3| |\tau\rangle$	$\|v^1\rangle$	$(\alpha_2 \|0\rangle + \alpha_1 \|1\rangle)B_1 \otimes (\beta_1 \|00\rangle - \beta_2 \|11\rangle)A_2D$	σ_x
	$\|v^2\rangle$	$(\alpha_2 \|0\rangle - \alpha_1 \|1\rangle)B_1 \otimes (\beta_1 \|00\rangle + \beta_2 \|11\rangle)A_2D$	$-i\sigma_y$
$\langle \Phi^2| \langle \Psi^3| |\tau\rangle$	$\|v^1\rangle$	$(\alpha_2 \|0\rangle + \alpha_1 \|1\rangle)B_1 \otimes (\beta_2 \|00\rangle + \beta_1 \|11\rangle)A_2D$	σ_x
	$\|v^2\rangle$	$(\alpha_2 \|0\rangle - \alpha_1 \|1\rangle)B_1 \otimes (\beta_2 \|00\rangle - \beta_1 \|11\rangle)A_2D$	$-i\sigma_y$
$\langle \Phi^1| \langle \Psi^4| |\tau\rangle$	$\|v^1\rangle$	$(-\alpha_2 \|0\rangle + \alpha_1 \|1\rangle)B_1 \otimes (\beta_1 \|00\rangle - \beta_2 \|11\rangle)A_2D$	$i\sigma_y$
	$\|v^2\rangle$	$(-\alpha_2 \|0\rangle - \alpha_1 \|1\rangle)B_1 \otimes (\beta_1 \|00\rangle + \beta_2 \|11\rangle)A_2D$	$-\sigma_x$
$\langle \Phi^2| \langle \Psi^4| |\tau\rangle$	$\|v^1\rangle$	$(-\alpha_2 \|0\rangle + \alpha_1 \|1\rangle)B_1 \otimes (\beta_2 \|00\rangle + \beta_1 \|11\rangle)A_2D$	$i\sigma_y$
	$\|v^2\rangle$	$(-\alpha_2 \|0\rangle - \alpha_1 \|1\rangle)B_1 \otimes (\beta_2 \|00\rangle - \beta_1 \|11\rangle)A_2D$	$-\sigma_x$

Step 4:

To remotely prepare the quantum state $|\phi\rangle$ from Bob to Alice, supervisors Charlie and David are needed. David must perform the qubit D single-qubit von Neumann measurements. Once Alice receives the measurement results from Charlie and David through the classical channel, she can perform the corresponding unitary operation on qubit A_2 to obtain the desired quantum state $|\phi\rangle$ that Bob intends to prepare.

In Step 3's example, if David's measurement result is $|v^1\rangle$, then Alice will obtain the collapsed state of qubit A_2 as $\beta_1 |0\rangle + \beta_2 |1\rangle$. To reconstruct the quantum state $|\phi\rangle$, Alice can perform the unitary operation I on A_2. On the other hand, if David's measurement result is $|v^2\rangle$, then Alice will observe the collapsed state of qubit A_2 as $\beta_1 |0\rangle - \beta_2 |1\rangle$. To reconstruct the quantum state $|\phi\rangle$, Alice can perform the unitary operation σ_z on A_2. The details of David's measurement results, the state of qubit A_2, and Alice's unitary operation are presented in Table 4.

The four unitary operations used in the transmission process are:

$$\begin{aligned} I &= |0\rangle \langle 0| + |1\rangle \langle 1| \\ \sigma_z &= |0\rangle \langle 0| - |1\rangle \langle 1| \\ \sigma_x &= |0\rangle \langle 1| + |1\rangle \langle 0| \\ i\sigma_y &= |0\rangle \langle 1| - |1\rangle \langle 0| \end{aligned} \quad (10)$$

The above steps illustrate how a hybrid quantum communication system is implemented using a six-qubit entangled state as a channel. During the transmission process, Alice sends the state of qubit a to Bob via quantum teleportation, which Charlie supervises. Simultaneously, Bob prepares a qubit A_2 in the same state as qubit b on Alice's side through remote state preparation, which Charlie and David jointly supervise. This approach allows

different communication protocols to be controlled on a quantum channel, improving communication flexibility.

Table 4. David's measurement results and Alice's unitary operation corresponding to each result.

State before David Measured	David's Measurement Result	State of Qubits	Alice's Operation
$\langle \nu^1 \vert \langle \Phi^1 \vert \langle \Psi^1 \vert \vert \tau \rangle$	$\vert \nu^1 \rangle$	$\beta_1 \vert 0 \rangle - \beta_2 \vert 1 \rangle$	σ_z
	$\vert \nu^2 \rangle$	$\beta_1 \vert 0 \rangle + \beta_2 \vert 1 \rangle$	I
$\langle \nu^2 \vert \langle \Phi^1 \vert \langle \Psi^1 \vert \vert \tau \rangle$	$\vert \nu^1 \rangle$	$\beta_1 \vert 0 \rangle + \beta_2 \vert 1 \rangle$	I
	$\vert \nu^2 \rangle$	$\beta_1 \vert 0 \rangle - \beta_2 \vert 1 \rangle$	σ_z
$\langle \nu^1 \vert \langle \Phi^2 \vert \langle \Psi^1 \vert \vert \tau \rangle$	$\vert \nu^1 \rangle$	$\beta_2 \vert 0 \rangle + \beta_1 \vert 1 \rangle$	σ_x
	$\vert \nu^2 \rangle$	$\beta_2 \vert 0 \rangle - \beta_1 \vert 1 \rangle$	$-i\sigma_y$
$\langle \nu^2 \vert \langle \Phi^2 \vert \langle \Psi^1 \vert \vert \tau \rangle$	$\vert \nu^1 \rangle$	$\beta_2 \vert 0 \rangle - \beta_1 \vert 1 \rangle$	$-i\sigma_y$
	$\vert \nu^2 \rangle$	$\beta_2 \vert 0 \rangle + \beta_1 \vert 1 \rangle$	σ_x
$\langle \nu^1 \vert \langle \Phi^1 \vert \langle \Psi^2 \vert \vert \tau \rangle$	$\vert \nu^1 \rangle$	$\beta_1 \vert 0 \rangle - \beta_2 \vert 1 \rangle$	σ_z
	$\vert \nu^2 \rangle$	$\beta_1 \vert 0 \rangle + \beta_2 \vert 1 \rangle$	I
$\langle \nu^2 \vert \langle \Phi^1 \vert \langle \Psi^2 \vert \vert \tau \rangle$	$\vert \nu^1 \rangle$	$\beta_1 \vert 0 \rangle + \beta_2 \vert 1 \rangle$	I
	$\vert \nu^2 \rangle$	$\beta_1 \vert 0 \rangle - \beta_2 \vert 1 \rangle$	σ_z
$\langle \nu^1 \vert \langle \Phi^2 \vert \langle \Psi^2 \vert \vert \tau \rangle$	$\vert \nu^1 \rangle$	$\beta_2 \vert 0 \rangle + \beta_1 \vert 1 \rangle$	σ_x
	$\vert \nu^2 \rangle$	$\beta_2 \vert 0 \rangle - \beta_1 \vert 1 \rangle$	$-i\sigma_y$
$\langle \nu^2 \vert \langle \Phi^2 \vert \langle \Psi^2 \vert \vert \tau \rangle$	$\vert \nu^1 \rangle$	$\beta_2 \vert 0 \rangle - \beta_1 \vert 1 \rangle$	$-i\sigma_y$
	$\vert \nu^2 \rangle$	$\beta_2 \vert 0 \rangle + \beta_1 \vert 1 \rangle$	σ_x
$\langle \nu^1 \vert \langle \Phi^1 \vert \langle \Psi^3 \vert \vert \tau \rangle$	$\vert \nu^1 \rangle$	$\beta_1 \vert 0 \rangle - \beta_2 \vert 1 \rangle$	σ_z
	$\vert \nu^2 \rangle$	$\beta_1 \vert 0 \rangle + \beta_2 \vert 1 \rangle$	I
$\langle \nu^2 \vert \langle \Phi^1 \vert \langle \Psi^3 \vert \vert \tau \rangle$	$\vert \nu^1 \rangle$	$\beta_1 \vert 0 \rangle + \beta_2 \vert 1 \rangle$	I
	$\vert \nu^2 \rangle$	$\beta_1 \vert 0 \rangle - \beta_2 \vert 1 \rangle$	σ_z
$\langle \nu^1 \vert \langle \Phi^2 \vert \langle \Psi^3 \vert \vert \tau \rangle$	$\vert \nu^1 \rangle$	$\beta_2 \vert 0 \rangle + \beta_1 \vert 1 \rangle$	σ_x
	$\vert \nu^2 \rangle$	$\beta_2 \vert 0 \rangle - \beta_1 \vert 1 \rangle$	$-i\sigma_y$
$\langle \nu^2 \vert \langle \Phi^2 \vert \langle \Psi^3 \vert \vert \tau \rangle$	$\vert \nu^1 \rangle$	$\beta_2 \vert 0 \rangle - \beta_1 \vert 1 \rangle$	$-i\sigma_y$
	$\vert \nu^2 \rangle$	$\beta_2 \vert 0 \rangle + \beta_1 \vert 1 \rangle$	σ_x
$\langle \nu^1 \vert \langle \Phi^1 \vert \langle \Psi^4 \vert \vert \tau \rangle$	$\vert \nu^1 \rangle$	$\beta_1 \vert 0 \rangle - \beta_2 \vert 1 \rangle$	σ_z
	$\vert \nu^2 \rangle$	$\beta_1 \vert 0 \rangle + \beta_2 \vert 1 \rangle$	I
$\langle \nu^2 \vert \langle \Phi^1 \vert \langle \Psi^4 \vert \vert \tau \rangle$	$\vert \nu^1 \rangle$	$\beta_1 \vert 0 \rangle + \beta_2 \vert 1 \rangle$	I
	$\vert \nu^2 \rangle$	$\beta_1 \vert 0 \rangle - \beta_2 \vert 1 \rangle$	σ_z
$\langle \nu^1 \vert \langle \Phi^2 \vert \langle \Psi^4 \vert \vert \tau \rangle$	$\vert \nu^1 \rangle$	$\beta_2 \vert 0 \rangle + \beta_1 \vert 1 \rangle$	σ_x
	$\vert \nu^2 \rangle$	$\beta_2 \vert 0 \rangle - \beta_1 \vert 1 \rangle$	$-i\sigma_y$
$\langle \nu^2 \vert \langle \Phi^2 \vert \langle \Psi^4 \vert \vert \tau \rangle$	$\vert \nu^1 \rangle$	$\beta_2 \vert 0 \rangle - \beta_1 \vert 1 \rangle$	$-i\sigma_y$
	$\vert \nu^2 \rangle$	$\beta_2 \vert 0 \rangle + \beta_1 \vert 1 \rangle$	σ_x

3. Experimental Verification

To test the feasibility of our proposed hybrid quantum communication scheme in IoT communication, we conducted a simulation experiment using IBM's Qiskit Aer (0.13.0) quantum computing simulator. Qiskit Aer is an open-source quantum computing software package that can simulate quantum circuits on different noise models and backend devices. We used the statevector-simulator in Qiskit Aer as the backend device, which can give the final state vector after the quantum circuit runs. We built a quantum circuit with seven qubits and sixteen classical bits, completed the four steps in the scheme, and recorded the measurement results and correction operations of Alice, Bob, Charlie, and David. Next, we will show the whole process and results of the experiment.

The complete process of the experiment is shown in Figure 2, where $q_1 \sim q_6$ is a six-qubit quantum channel, and q_0 is a single-qubit quantum state; their initial states are all

$|0\rangle$. Part I shows the quantum circuit that generates arbitrary single-qubit states through U_1 gates. In this experiment, we set the parameters of the U_1 gate to $(\theta = \pi/2, \varphi = 0, \lambda = \pi)$. We can obtain a quantum state of $|\varphi\rangle_a = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ through the unitary operation of the U_1 gate. This qubit is utilized for experimental verification purposes. Part II is the process of preparing quantum channels, showing the use of H gates, X gates, and $CNOT$ gates to generate six-qubit entangled states. Part III shows Alice's process of performing Bell state measurements on qubits a and A_1. In this process, according to different measurement results, Alice will obtain two classical bit information. The measurement results and corresponding classical bit encoding are shown in Table 5 below. The same goes for Bob, Charlie, and David.

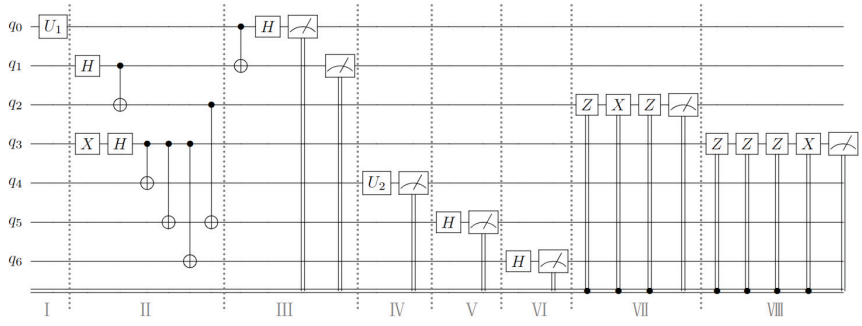


Figure 2. Using the Qiskit Aer quantum computing simulator to implement quantum circuit design for HCHQC protocols.

Table 5. Alice, Bob, David, and Charlie's measurement results and corresponding classical information.

Measurer	Measurement Result	Classical Information
Alice	$\|\Psi^1\rangle_{a,A_1} = \frac{1}{\sqrt{2}}(\|00\rangle + \|11\rangle)$	00
	$\|\Psi^2\rangle_{a,A_1} = \frac{1}{\sqrt{2}}(\|00\rangle - \|11\rangle)$	01
	$\|\Psi^3\rangle_{a,A_1} = \frac{1}{\sqrt{2}}(\|01\rangle + \|10\rangle)$	10
	$\|\Psi^4\rangle_{a,A_1} = \frac{1}{\sqrt{2}}(\|01\rangle - \|10\rangle)$	11
Bob	$\|\Phi^1\rangle_{B_2} = \beta_1 \|0\rangle + \beta_2 \|1\rangle$	0
	$\|\Phi^2\rangle_{B_2} = \beta_2 \|0\rangle - \beta_1 \|1\rangle$	1
Charlie/David	$\|\nu^1\rangle = \frac{1}{\sqrt{2}}(\|0\rangle + \|1\rangle)$	0
	$\|\nu^2\rangle = \frac{1}{\sqrt{2}}(\|0\rangle - \|1\rangle)$	1

Part IV is the circuit where Bob performs projection measurement on his qubit B_2. In this experiment, we set the quantum state that Bob wants to transmit as $|\phi\rangle_b = \frac{1}{\sqrt{3}}|0\rangle + \frac{2}{\sqrt{3}}|1\rangle)$. Therefore, the quantum gate U_2 used for projection measurement can be represented by the following matrix:

$$U_2 = \begin{pmatrix} \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{3}} \\ \frac{2}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \end{pmatrix}$$

Part V and Part VI show the circuits used by Charlie and David to perform the von Neumann measurements. Parts VII and VIII represent the process of reconstructing quantum information by Bob and Alice according to the classical bit encoding by the corresponding unitary operations.

Qiskit Aer is a high-performance quantum computing simulator with realistic noise models. Using several different simulation methods, it provides an interface to run quantum circuits with or without noise. By running the above quantum circuit in the Qiskit Aer

simulator, the probability distribution of the qubit states obtained by Alice and Bob can be obtained. The following Figures 3 and 4 shows the experimental results of 8192 experiments on Qiskit Aer.

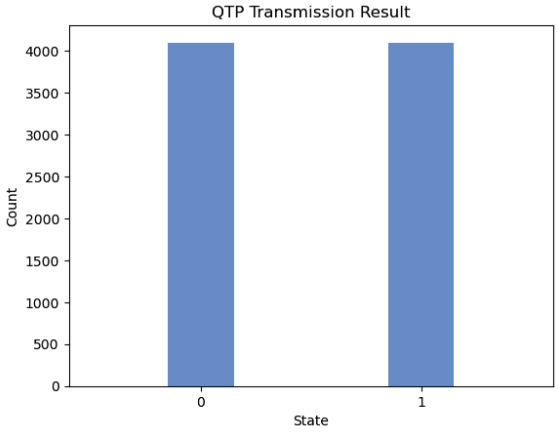


Figure 3. Probability distribution of the results of 8192 experiments of HCHQC protocol on Qiskit Aer when Alice performs QTP to Bob.

Figure 4. Probability distribution of the results of 8192 experiments of HCHQC protocol on Qiskit Aer when Bob performs RSP to Alice.

According to Table 6, the qubit state obtained by Alice and Bob can be written as Equation (11), where $|\varphi\rangle_{B_1}$ represents the QTP simulated state received by Bob calculated based on the actual transmission result, and $|\phi\rangle_{A_2}$ represents the RSP simulated state received by Alice calculated based on the actual transmission result.

$$\begin{aligned} |\varphi\rangle_{B_1} &= \sqrt{0.5002}\,|0\rangle + \sqrt{0.4997}\,|1\rangle \\ |\phi\rangle_{A_2} &= \sqrt{0.3291}\,|0\rangle + \sqrt{0.6708}\,|1\rangle \end{aligned} \tag{11}$$

Fidelity is a metric that assesses the efficacy of quantum communication protocols. It signifies the degree of resemblance between the transmitted quantum state and the original quantum state. Generally speaking, higher fidelity indicates a more successful quantum

communication protocol. In this scheme, we can calculate the fidelity of QTP from Alice to Bob and RSP from Bob to Alice. The calculation formulas are (12) and (13), respectively:

$$\begin{aligned} F_{QTP} = \langle\varphi|_a \rho_1 |\varphi\rangle_a = &(\frac{\sqrt{2}}{2}\langle 0| + \frac{\sqrt{2}}{2}\langle 1|)(\sqrt{0.5002}|0\rangle + \sqrt{0.4997}|1\rangle) \\ &(\sqrt{0.5002}\langle 0| + \sqrt{0.4997}\langle 1|)(\frac{\sqrt{2}}{2}|0\rangle + \frac{\sqrt{2}}{2}|1\rangle) \approx 0.999 \end{aligned} \tag{12}$$

$$\begin{aligned} F_{RSP} = \langle\phi|_b \rho_2 |\phi\rangle_b = &(\frac{1}{\sqrt{3}}\langle 0| + \frac{2}{\sqrt{3}}\langle 1|)(\sqrt{0.3291}|0\rangle + \sqrt{0.6708}|1\rangle) \\ &(\sqrt{0.3291}\langle 0| + \sqrt{0.6708}\langle 1|)(\frac{1}{\sqrt{3}}|0\rangle + \frac{2}{\sqrt{3}}|1\rangle) \approx 0.999 \end{aligned} \tag{13}$$

where F_{QTP} represents the fidelity of quantum teleportation from Alice to Bob, F_{RSP} represents the fidelity of remote state preparation from Bob to Alice, $\rho_1 = |\varphi\rangle_{B_1}\langle\varphi|$ represents the density operator of the QTP simulated state, and $\rho_2 = |\phi\rangle_{A_2}\langle\phi|$ represents the density operator of the RSP simulated state. According to the calculation results of the above formulas, we can see that, in our designed experiment, the fidelity of the quantum state transmitted by Alice to Bob via quantum teleportation (QTP) is as high as 0.999, close to the ideal value of 1, which demonstrates the validity of QTP in the hybrid quantum communication protocol; similarly, the fidelity of the quantum state prepared by Bob for Alice via remote quantum state preparation (RSP) is also 0.999, close to the ideal value of 1, which demonstrates the validity of RSP in the hybrid quantum communication protocol.

Table 6. Transmission result.

Protocal	States	Shots	Frequency(%)
QTP	$\|0\rangle$	4098	50.02%
	$\|1\rangle$	4094	49.97%
RSP	$\|0\rangle$	2696	32.91%
	$\|1\rangle$	5496	67.08%

4. Impact of Quantum Noise on Hybrid Quantum Communication Schemes

Quantum secret communication uses the characteristics of quantum mechanics to achieve high-security information transmission. However, there are also some challenges in applying quantum secret communication in the IoT, such as the interference of quantum noise. Quantum noise refers to the environmental factors or device defects in the quantum channel, which cause the transmitted quantum state to change or be in error. This affects the efficiency and security of quantum secret communication and may even cause communication interruption or cracking. There are several main types of quantum noise in the IoT:

1. Amplitude-damping noise: This particular noise engenders the disappearance or attenuation of photons, thereby diminishing both the signal strength and the signal-to-noise ratio. Amplitude-damping noise primarily arises from factors such as fibre-optic transmission losses, reflection in optical devices, and scattering;
2. Phase-damping noise: This particular noise induces random variations in the phase of photons, thereby disrupting the quantum superposition and entanglement states. Phase-damping noise primarily stems from factors such as temperature fluctuations, mechanical vibrations, and electromagnetic interference;
3. Displacement noise: This noise engenders random displacements in the position of photons, thereby altering their wavelength or frequency. Displacement noise primarily arises from factors such as the Doppler effect, fibre dispersion, and nonlinear effects;

4. Rotational noise: This particular noise induces random rotations in the polarization direction of photons, thereby altering their polarization state. Rotational noise primarily stems from factors such as fibre birefringence, the Faraday effect, and magnetic fields.

In this section, we have chosen two exemplary forms of quantum noise: amplitude-damping noise and phase-damping noise. We shall discuss the fidelity of communication in the IoT communication scenario for the HCHQC protocol in the presence of these two types of quantum noise. The Kraus operator for amplitude-damping noise is expressed as:

$$E_0^A = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\eta_A} \end{bmatrix}, E_1^A = \begin{bmatrix} 0 & \sqrt{\eta_A} \\ 0 & 0 \end{bmatrix} \tag{14}$$

where η_A is the error probability caused by the amplitude-damping noise of the qubit. The Kraus operator for the phase-damping noise is expressed as:

$$E_0^P = \sqrt{1-\eta_P} I, E_1^P = \sqrt{\eta_P} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, E_2^P = \sqrt{\eta_P} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{15}$$

Here, η_P represents the strength of the noise, and I is the identity matrix. In the presence of quantum noise, the input of a pure state will be converted into a mixed state, which can be expressed more conveniently in the form of a density matrix. Therefore, the channel in the pure state can be written as $\rho = |\psi\rangle_{A_1B_1A_2B_2CD} \langle\psi|$ such a form of the density matrix.

Assume that the channel particles are prepared by the supervisor David, who then sends A_1, A_2 to Alice, B_1, B_2 to Bob, and C to Charlie. To simplify the analysis model, assume that the particle C sent by David to Charlie is not affected by noise, the two particles received by Alice are affected by the same noise, and the two particles received by Bob are affected by the same noise. The following formula can describe the influence of noise on the channel:

$$\begin{aligned} \xi^r(\rho) = \sum_{m,n} & \left(E_m^{r,A_1}\right)\left(E_n^{r,B_1}\right)\left(E_m^{r,A_2}\right)\left(E_n^{r,B_2}\right) \\ & \rho\left(E_m^{r,A_1}\right)^\dagger\left(E_n^{r,B_1}\right)^\dagger\left(E_m^{r,A_2}\right)^\dagger\left(E_n^{r,B_2}\right)^\dagger \end{aligned} \tag{16}$$

where $r \in \{A, P\}$. When $r = A$, the formula describes the amplitude-damping noise; when $r = P$, the formula describes the phase-damping noise. The subscript $m, n \in \{0, 1, 2\}$ of E indicates the Kraus operators corresponding to each noise. The second superscript of E indicates which qubit of the channel the corresponding noise operator acts on.

When the quantum channel is affected by these two different noises, it will become the corresponding mixed state:

$$\begin{aligned} \xi^A(\rho) = & \frac{1}{4}\{[|000000\rangle - (1-\eta_A)|001111\rangle + \\ & (1-\eta_A)|110010\rangle - (1-\eta_A)^2|111101\rangle]\times \\ & [\langle 000000| - (1-\eta_A)\langle 001111| + \\ & (1-\eta_A)\langle 110010| - (1-\eta_A)^2\langle 111101|]+ \\ & [-\eta_A(1-\eta_A)|101001\rangle] \times [-\eta_A(1-\eta_A)\langle 101001|]+ \\ & [-\eta_A(1-\eta_A)|010101\rangle] \times [-\eta_A(1-\eta_A)\langle 010101|]+ \\ & [-\eta_A^2|000001\rangle] \times [-\eta_A^2\langle 000001|]\} \end{aligned} \tag{17}$$

$$\begin{aligned}\xi^P(\rho) =& \frac{1}{4}\{[(1-\eta_P)^2 |000000\rangle - (1-\eta_P)^2 |001111\rangle + \\ & (1-\eta_P)^2 |110010\rangle] - (1-\eta_P)^2 |111101\rangle]\times \\ & [(1-\eta_P)^2 \langle 000000| - (1-\eta_P)^2 \langle 001111| + \\ & (1-\eta_P)^2 \langle 110010|] - (1-\eta_P)^2 \langle 111101|]+ \\ & \eta_P^2(2-\eta_P^2) |000000\rangle \langle 000000| + \\ & \eta_P^2(2-\eta_P^2) |111101\rangle \langle 111101|\}\end{aligned} \tag{18}$$

When Alice, Bob, Charlie, and David complete the communication operation, the final state density matrix can be expressed as follows:

$$\rho_{\text{out}}^r = \text{Tr}_{aA_1B_2CD}\left\{U_{ijkl}[\rho_a \otimes \xi^r(\rho)]U_{ijkl}^\dagger\right\} \tag{19}$$

In the above formula, $\rho_a = (\alpha_1 |0\rangle + \alpha_2 |1\rangle)(\alpha_1 \langle 0| + \alpha_2 \langle 1|)$, U_{ijkl} is:

$$\begin{aligned}U_{ijkl} =& \{I_a \otimes I_{A_1} \otimes \sigma_{B_1}^{ijk} \otimes \sigma_{A_2}^{ijkl} \otimes I_{B_2} \otimes I_C \otimes I_D\} \\ & \{I_a \otimes I_{A_1} \otimes I_{B_1} \otimes I_{A_2} \otimes I_{B_2} \otimes I_C \otimes \langle \nu^l|_D\} \\ & \{I_a \otimes I_{A_1} \otimes I_{B_1} \otimes I_{A_2} \otimes I_{B_2} \otimes \langle \nu^k|_C \otimes I_D\} \\ & \{I_a \otimes I_{A_1} \otimes I_{B_1} \otimes I_{A_2} \otimes \langle \Phi^j|_{B_2} \otimes I_C \otimes I_D\} \\ & \{\langle \Psi^i|_{aA_1} \otimes I_{B_1} \otimes I_{A_2} \otimes I_{B_2} \otimes I_C \otimes I_D\}\end{aligned} \tag{20}$$

Here, $i \in \{1,2,3,4\}$, $\langle \Psi^i|_{aA_1}$ represents the result of the Bell state measurement on qubit a, A_1; $j \in \{1,2\}$, $\langle \Phi^j|_{B_2}$ represents the result obtained by custom base projection measurement on qubit B_2; $k,l \in \{1,2\}$, $\langle \nu^k|, \langle \nu^l|$ represent the results obtained by von Neumann measurement of qubits C and D, respectively; while $\sigma_{B_1}^{ijk}, \sigma_{A_2}^{ijkl}$ represent the corresponding unitary operations on qubits B_1, A_2 according to the measurement results. Through the calculation of the above formula, the density matrix of the resulting state obtained by executing the HCHQC protocol under the influence of two noise environments can be obtained:

$$\begin{aligned}\rho_{\text{out}}^A =& \frac{1}{32}\{[\alpha_1(\beta_1 |00\rangle - \beta_2(1-\eta_A) |01\rangle)+ \\ & \alpha_2(\beta_1(1-\eta_A) |10\rangle - \beta_2(1-\eta_A)^2 |11\rangle)]\times \\ & [\alpha_1(\beta_1 \langle 00| - \beta_2(1-\eta_A) \langle 01|)+ \\ & \alpha_2(\beta_1(1-\eta_A) \langle 10| - \beta_2(1-\eta_A)^2 \langle 11|)]+ \\ & \alpha_2^2\beta_1^2\eta_A^2(1-\eta_A)^2 |01\rangle \langle 01| + \\ & \alpha_1^2\beta_2^2\eta_A^2(1-\eta_A)^2 |10\rangle \langle 10| + \\ & \alpha_1^2\beta_1^2\eta_A^4 |00\rangle \langle 00|\}\end{aligned} \tag{21}$$

$$\begin{aligned}\rho_{\text{out}}^A =& \frac{1}{32}\{[\alpha_1(\beta_1(1-\eta_P)^2 |00\rangle - \beta_2(1-\eta_P)^2 |01\rangle)+ \\ & \alpha_2(\beta_1(1-\eta_P)^2 |10\rangle - \beta_2(1-\eta_P)^2 |11\rangle)]\times \\ & [\alpha_1(\beta_1(1-\eta_P)^2 \langle 00| - \beta_2(1-\eta_P)^2 \langle 01|)+ \\ & \alpha_2(\beta_1(1-\eta_P)^2 \langle 10| - \beta_2(1-\eta_P)^2 \langle 11|)]+ \\ & \alpha_1^2\beta_1^2\eta_P^2(2-\eta_P^2) |00\rangle \langle 00| + \\ & \alpha_2^2\beta_2^2\eta_P^2(2-\eta_P^2) |11\rangle \langle 11|\}\end{aligned} \tag{22}$$

With the density matrix of the communication results in the above two noise environments, this can be brought into the formula:

$$F_r = \langle\varphi|_{B_1} \langle\phi|_{A_2} \rho^r_{\text{out}} |\varphi\rangle_{B_1} |\phi\rangle_{A_2} \tag{23}$$

The corresponding communication fidelity can be obtained:

$$\begin{aligned} F_A =& \frac{1}{32}\{[\alpha_1^2\beta_1^2 + \alpha_1^2\beta_2^2(1-\eta_A) + \alpha_2^2\beta_1^2(1-\eta_A) + \alpha_2^2\beta_2^2(1-\eta_A)^2]^2 \\ &+ 2\alpha_1^2\beta_1^2\alpha_2^2\beta_2^2\eta_A^2(1-\eta_A)^2 + \alpha_1^4\beta_1^4\eta_A^4\} \end{aligned} \tag{24}$$

$$\begin{aligned} F_P =& \frac{1}{32}\{(1-\eta_P)^4(\alpha_1^2\beta_1^2 + \alpha_1^2\beta_2^2 + \alpha_2^2\beta_1^2 + \alpha_2^2\beta_2^2)^2 \\ &+ \eta_P^2(2-\eta_P^2)\alpha_1^4\beta_1^4 + \eta_P^2(2-\eta_P^2)\alpha_2^4\beta_2^4\} \end{aligned} \tag{25}$$

According to the formula, it can be found that under amplitude-damping noise and phase-damping noise, the fidelity of the transmission process only depends on the amplitude parameter and noise rate of the initial state. To reflect this relationship more intuitively, we plot the relationship between the fidelity function value and the variable under the two noise environments, as shown in Figures 5–10. It can be seen from the Figures 5, 6, 8 and 9 that as the noise rate η_A, η_P increases, the fidelity F_A, F_P gradually reduces.

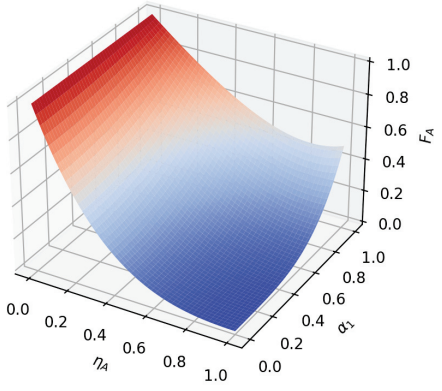


Figure 5. When $\beta_1 = \frac{\sqrt{2}}{2}$, $\beta_2 = \frac{\sqrt{2}}{2}$, the fidelity under amplitude-damping noise environment variation graph with noise rate and α_1.

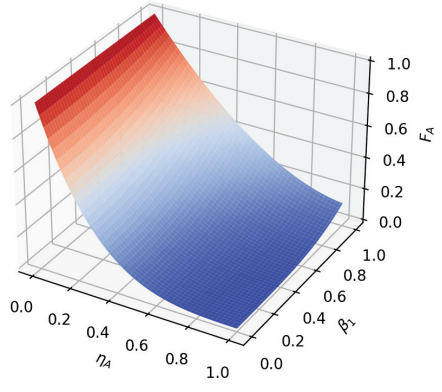


Figure 6. When $\alpha_1 = \frac{1}{2}$, $\alpha_2 = \frac{\sqrt{3}}{2}$, the fidelity under amplitude-damping noise environment variation graph with noise rate and β_1.

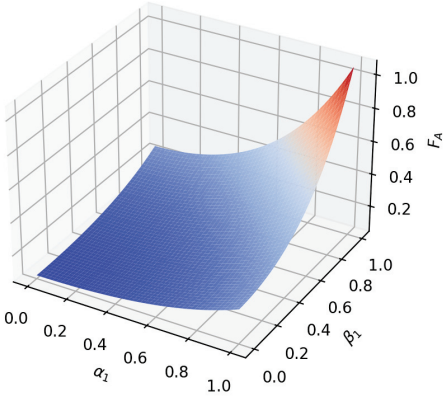


Figure 7. When the noise rate is 0.5, the fidelity changes with α_1 and β_1 in the amplitude-damping noise environment.

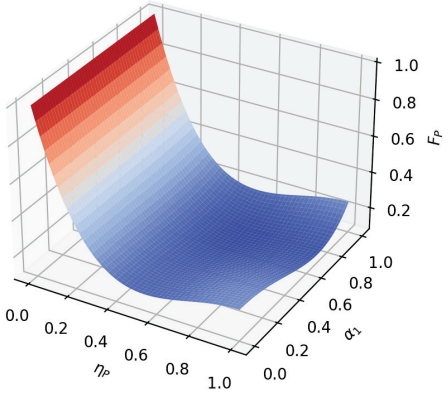


Figure 8. When $\beta_1 = \frac{\sqrt{2}}{2}$, $\beta_2 = \frac{\sqrt{2}}{2}$, the fidelity under phase-damping noise environment variation graph with noise rate and α_1.

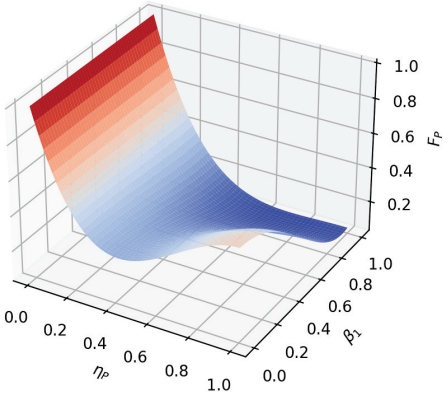


Figure 9. When $\alpha_1 = \frac{1}{2}$, $\alpha_2 = \frac{\sqrt{3}}{2}$, the fidelity under phase-damping noise environment variation graph with noise rate and β_1.

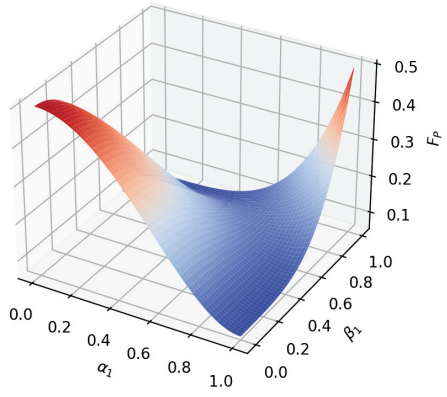


Figure 10. When the noise rate is 0.5, the fidelity changes with α_1 and β_1 in the phase-damping noise environment.

To analyze the influence of the amplitude parameter of the initial state on the fidelity of the hybrid quantum communication protocol, we first set $\alpha_1 = \alpha_2 = \frac{\sqrt{2}}{2}$, $\beta_1 = \beta_2 = \frac{\sqrt{2}}{2}$, $\eta_A = \eta_P$; the function image of fidelity F_A, F_P changing with the noise rate η_r in two noise environments is represented in Figure 11. Then, setting $\alpha_1 = \alpha_2 = \frac{\sqrt{2}}{2}, \beta_1 = 0, \beta_2 = 1$, $\eta_A = \eta_P$, the image of the fidelity F_A, F_P as a function of the noise rate η_r is plotted for the two noise environments, as shown in Figure 12.

In the case shown in Figure 11, for the same noise rate, the fidelity under the amplitude-damping noise environment (the blue line in Figure 11) is always greater than that under the phase-damping noise environment (the yellow line in Figure 11). In the case shown in Figure 12, when the noise rate η_A is less than 0.4486, the fidelity under the amplitude-damping noise environment is higher than that under the phase-damping noise environment; when the noise rate η_A is more significant than 0.4486, the fidelity under the phase-damping noise environment is higher than that under the amplitude-damping noise environment. It can be seen that when the amplitude parameters of the initial state are different, even if the noise rate is the same, there is a significant difference in fidelity for the different noise environments.

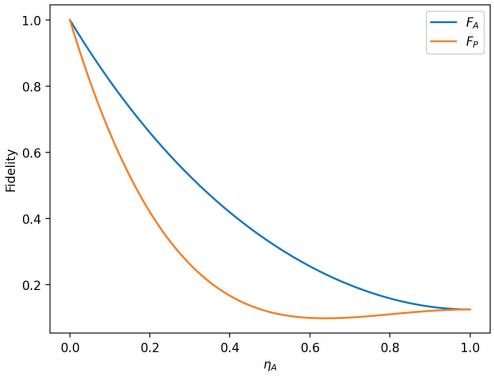


Figure 11. When $\alpha_1 = \alpha_2 = \frac{\sqrt{2}}{2}, \beta_1 = \beta_2 = \frac{\sqrt{2}}{2}, \eta_A = \eta_P$, fidelity variation with noise rate in amplitude-damping noise environment and phase-damping noise environment.

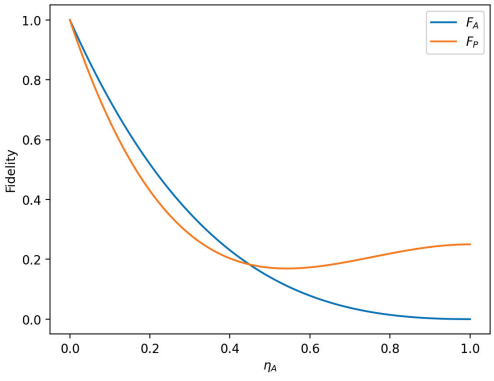


Figure 12. When $\alpha_1 = \alpha_2 = \frac{\sqrt{2}}{2}$, $\beta_1 = 0$, $\beta_2 = 1$, $\eta_A = \eta_P$, fidelity variation with noise rate in amplitude-damping noise environment and phase-damping noise environment.

This section analyzes the impact of amplitude-damping and phase-damping noise on the fidelity of the hierarchical controlled hybrid quantum communication protocol (HCHQC). The results show that, under the same noise intensity, amplitude-damping noise causes more damage to the fidelity than phase-damping noise, and the parameters of the initial state also affect the trend of the fidelity change. In order to improve the reliability and security of the HCHQC protocol in IoT applications, it is possible to use error correction codes or noise estimation circuits to detect and correct noise errors in the quantum channel or to use noise filters to reduce noise effects.

5. Conclusions

This paper proposes a hybrid quantum communication scheme based on six-qubit entangled states for hierarchical control, suitable for specific IoT application scenarios. The scheme realizes the hierarchical control of two communication protocols, quantum teleportation and remote quantum state preparation, on a quantum channel, improving the flexibility and efficiency of communication. This paper describes the principle and steps of the scheme in detail, designs the quantum circuit of the scheme, and performs simulation experiments using IBM's Qiskit Aer quantum computing simulator. The results show that the fidelity of the scheme reaches 0.999, verifying the feasibility of the scheme. This paper also analyzes the factors that affect the fidelity of the scheme in a specific quantum noise environment, providing references for further optimization of the scheme. By combining quantum communication technology with IoT application scenarios, this paper provides a new possibility for IoT communication. In order to improve the efficiency and reliability of quantum communication, we will continue to conduct in-depth research on high-dimensional entangled states, complex communication protocols, noise models and error correction mechanisms in the future and seek better solutions.

Author Contributions: Conceptualization, X.H. and D.L.; methodology, X.H. and D.L.; software, X.H.; validation, X.H. and D.L.; formal analysis, Y.F.; investigation, Y.Z. and Y.J.; resources, D.L.; data curation, X.H., Y.F. and Y.J.; writing—original draft preparation, X.H.; writing—review and editing, D.L., X.H., Y.F., Y.Z., Y.J., J.Z., X.Y. and Y.T.; supervision, D.L.; All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (62172060), the Sichuan Science and Technology Program (2022YFG0316, 2023ZHCG0004), and the National Key R&D Plan: 2022YFB3304303.

Data Availability Statement: Data are contained within the article.

Acknowledgments: Thank you to the editor reviewers for reviewing this article in their busy schedules, and thank you to the teachers for their careful guidance and help.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xie, Y.M.; Lu, Y.S.; Weng, C.X.; Cao, X.Y.; Jia, Z.Y.; Bao, Y.; Wang, Y.; Fu, Y.; Yin, H.L.; Chen, Z.B. Breaking the Rate-Loss Bound of Quantum Key Distribution with Asynchronous Two-Photon Interference. *PRX Quantum* **2022**, *3*, 020315. [CrossRef]
2. Zhou, L.; Lin, J.; Xie, Y.M.; Lu, Y.S.; Jing, Y.; Yin, H.L.; Yuan, Z. Experimental Quantum Communication Overcomes the Rate-Loss Limit without Global Phase Tracking. *Phys. Rev. Lett.* **2023**, *130*, 250801. [CrossRef] [PubMed]
3. Yin, H.L.; Fu, Y.; Li, C.L.; Weng, C.X.; Li, B.H.; Gu, J.; Lu, Y.S.; Huang, S.; Chen, Z.B. Experimental quantum secure network with digital signatures and encryption. *Natl. Sci. Rev.* **2022**, *10*, nwac228. [CrossRef] [PubMed]
4. Liao, Q.; Liu, X.; Ou, B.; Fu, X. Continuous-Variable Quantum Secret Sharing Based on Multi-Ring Discrete Modulation. *IEEE Trans. Commun.* **2023**, *71*, 6051–6060. [CrossRef]
5. Zhou, N.R.; Zhang, T.F.; Xie, X.W.; Wu, J.Y. Hybrid quantum–classical generative adversarial networks for image generation via learning discrete distribution. *Signal Process. Image Commun.* **2023**, *110*, 116891. [CrossRef]
6. Gong, L.H.; Pei, J.J.; Zhang, T.F.; Zhou, N.R. Quantum convolutional neural network based on variational quantum circuits. *Opt. Commun.* **2024**, *550*, 129993. [CrossRef]
7. Rahman, M.S.; Hossam-E-Haider, M. Quantum IoT: A Quantum Approach in IoT Security Maintenance. In Proceedings of the 2019 International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST), Dhaka, Bangladesh, 10–12 January 2019; pp. 269–272. [CrossRef]
8. Mumtaz, S.; Guizani, M. An overview of quantum computing and quantum communication systems. *IET Quantum Commun.* **2021**, *2*, 136–138. [CrossRef]
9. Fernández-Caramés, T.M. From Pre-Quantum to Post-Quantum IoT Security: A Survey on Quantum-Resistant Cryptosystems for the Internet of Things. *IEEE Internet Things J.* **2020**, *7*, 6457–6480. [CrossRef]
10. Cheng, C.; Lu, R.; Petzoldt, A.; Takagi, T. Securing the Internet of Things in a Quantum World. *IEEE Commun. Mag.* **2017**, *55*, 116–120. [CrossRef]
11. Al-Mohammed, H.A.; Yaacoub, E. On The Use of Quantum Communications for Securing IoT Devices in the 6G Era. In Proceedings of the 2021 IEEE International Conference on Communications Workshops (ICC Workshops), Montreal, QC, Canada, 14–23 June 2021; pp. 1–6. [CrossRef]
12. Al-Mohammed, H.A.; Al-Ali, A.; Yaacoub, E.; Qidwai, U.; Abualsaud, K.; Rzewuski, S.; Flizikowski, A. Machine Learning Techniques for Detecting Attackers During Quantum Key Distribution in IoT Networks with Application to Railway Scenarios. *IEEE Access* **2021**, *9*, 136994–137004. [CrossRef]
13. Liu, G.; Han, J.; Zhou, Y.; Liu, T.; Chen, J.; Pryss, R. QSLT: A Quantum-Based Lightweight Transmission Mechanism against Eavesdropping for IoT Networks. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 4809210. [CrossRef]
14. Bennett, C.H.; Brassard, G.; Crépeau, C.; Jozsa, R.; Peres, A.; Wootters, W.K. Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. *Phys. Rev. Lett.* **1993**, *70*, 1895–1899. [CrossRef]
15. Li, D.; Zheng, Y.; Liu, X.; Liu, M. Hierarchical Quantum Teleportation of Arbitrary Single-Qubit State by Using Four-Qubit Cluster State. *Int. J. Theor. Phys.* **2021**, *60*, 1911–1919. [CrossRef]
16. Li, D.; Zheng, Y.; Liu, X.; Zhou, J.; Tan, Y.; Yang, X.; Liu, M. Hierarchical Quantum Information Splitting of an Arbitrary Two-Qubit State Based on a Decision Tree. *Mathematics* **2022**, *10*, 4571. [CrossRef]
17. Zheng, Y.; Li, D.; Liu, X.; Liu, M.; Zhou, J.; Yang, X.; Tan, Y.; Wang, R. Quantum Teleportation of Unknown Seven-Qubit Entangled State Using Four-Qubit Entangled State. *Int. J. Theor. Phys.* **2022**, *61*, 220. [CrossRef]
18. Liu, X.; Li, D.; Zheng, Y.; Liu, M.; Yang, X.; Zhou, J.; Tan, Y.; Wang, R. Quantum Information Splitting of an Arbitrary Five-Qubit State Using Four-Qubit Entangled States. *Int. J. Theor. Phys.* **2022**, *61*, 138. [CrossRef]
19. Karlsson, A.; Bourennane, M. Quantum teleportation using three-particle entanglement. *Phys. Rev. A* **1998**, *58*, 4394–4400. [CrossRef]
20. Chen, X.B.; Zhang, N.; Lin, S.; Wen, Q.Y.; Zhu, F.C. Quantum circuits for controlled teleportation of two-particle entanglement via a W state. *Optics Commun.* **2008**, *281*, 2331–2335. .: 10.1016/j.optcom.2007.12.002 [CrossRef]
21. Na, C.; Dongxiao, Q.; Hong, Y.; Changxing, P. Perfect quantum controlled teleportation via a novel three-particle partially entangled channel. *J. China Univ. Posts Telecommun.* **2015**, *22*, 45–50. [CrossRef]
22. Shi, L.; Zhou, K.; Wei, J.; Zhu, Y.; Zhu, Q. Quantum Controlled Teleportation of Arbitrary Two-Qubit State via Entangled States. *Adv. Math. Phys.* **2018**, *2018*, 4575438. [CrossRef]
23. Chen, J.; Li, D.; Liu, M.; Yang, Y.; Zhou, Q. Quantum Controlled Teleportation of Bell State Using Seven-Qubit Entangled State. *Int. J. Theor. Phys.* **2020**, *59*, 1402–1412. [CrossRef]
24. Lo, H.K. Classical-communication cost in distributed quantum-information processing: A generalization of quantum-communication complexity. *Phys. Rev. A* **2000**, *62*, 012313. [CrossRef]
25. Fang, S.H.; Jiang, M. Bidirectional and Asymmetric Controlled Quantum Information Transmission via Five-qubit Brown State. *Int. J. Theor. Phys.* **2017**, *56*, 1530–1536. [CrossRef]

26. Sang, Z.W. Bidirectional Controlled Quantum Information Transmission by Using a Five-Qubit Cluster State. *Int. J. Theor. Phys.* **2017**, *56*, 3400–3404. [CrossRef]
27. Ma, P.C.; Chen, G.B.; Li, X.W.; Zhan, Y.B. Schemes for Hybrid Bidirectional Controlled Quantum Communication via Multi-qubit Entangled States. *Int. J. Theor. Phys.* **2017**, *57*, 443–452. [CrossRef]
28. Wang, X.W.; Xia, L.X.; Wang, Z.Y.; Zhang, D.Y. Hierarchical quantum-information splitting. *Opt. Commun.* **2010**, *283*, 1196–1199. [CrossRef]
29. Shukla, C.; Pathak, A. Hierarchical quantum communication. *Phys. Lett. A* **2013**, *377*, 1337–1344. [CrossRef]
30. Shukla, C.; Thapliyal, K.; Pathak, A. Hierarchical joint remote state preparation in noisy environment. *Quantum Inf. Process.* **2017**, *16*, 205. [CrossRef]
31. Chen, N.; Yan, B.; Chen, G.; Zhang, M.J.; Pei, C.X. Deterministic hierarchical joint remote state preparation with six-particle partially entangled state*. *Chin. Phys. B* **2018**, *27*, 090304. [CrossRef]
32. Wang, N.N.; Ma, S.Y.; Li, X. Hierarchical controlled quantum communication via the χ state under noisy environment. *Mod. Phys. Lett. A* **2020**, *35*, 2050306. [CrossRef]
33. Ma, S.; Wang, N. Hierarchical Remote Preparation of an Arbitrary Two-Qubit State with Multiparty. *Quantum Inf. Process.* **2021**, *20*, 276. [CrossRef]

sensors

MDPI

Article

A Scheme for Quantum Teleportation and Remote Quantum State Preparation of IoT Multiple Devices

You Fu *, Dongfen Li *, Xiaoyu Hua, Yangyang Jiang, Yonghao Zhu, Jie Zhou, Xiaolong Yang and Yuqiao Tan

College of Computer Science and Cyber Security (Oxford Brookes College), Chengdu University of Technology, Chengdu 610059, China; 2022050888@stu.cdut.edu.cn (X.H.); 2022050894@stu.cdut.edu.cn (Y.J.); 2022050893@stu.cdut.edu.cn (Y.Z.); 2021020861@stu.cdut.edu.cn (J.Z.); 2021050831@stu.cdut.edu.cn (X.Y.); 2021020897@stu.cdut.edu.cn (Y.T.)

* Correspondence: 2022050889@stu.cdut.edu.cn (Y.F.); lidongfen17@cdut.edu.cn (D.L.)

Abstract: With the continuous development of the Internet of Things (IoT) technology, the industry's awareness of the security of the IoT is also increasing, and the adoption of quantum communication technology can significantly improve the communication security of various devices in the IoT. This paper proposes a scheme of controlled remote quantum state preparation and quantum teleportation based on multiple communication parties, and a nine-qubit entanglement channel is used to achieve secure communication of multiple devices in the IoT. The channel preparation, measurement operation, and unitary operation of the scheme were successfully simulated on the IBM Quantum platform, and the entanglement degree and reliability of the channel were verified through 8192 shots. The scheme's application in the IoT was analyzed, and the steps and examples of the scheme in the secure communication of multiple devices in the IoT are discussed. By simulating two different attack modes, the effect of the attack on the communication scheme in the IoT was deduced, and the scheme's high security and anti-interference ability was analyzed. Compared with other schemes from the two aspects of principle and transmission efficiency, it is highlighted that the advantages of the proposed scheme are that it overcomes the single fixed one-way or two-way transmission protocol form of quantum teleportation in the past and can realize quantum communication with multiple devices, ensuring both security and transmission efficiency.

Keywords: IoT security; quantum communication; controlled quantum teleportation; remote state preparation; multi-device communication

Citation: Fu, Y.; Li, D.; Hua, X.; Jiang, Y.; Zhu, Y.; Zhou, J.; Yang, X.; Tan, Y. A Scheme for Quantum Teleportation and Remote Quantum State Preparation of IoT Multiple Devices. *Sensors* **2023**, *23*, 8475. https://doi.org/10.3390/s23208475

Academic Editor: Jian Li

Received: 19 September 2023
Revised: 10 October 2023
Accepted: 13 October 2023
Published: 15 October 2023

1. Introduction

The IoT refers to various devices and objects connected via the Internet, enabling them to communicate and share information. With the continuous progress of technology, the IoT has become an essential part of modern society, and its application scope covers various fields such as home, industry, medical treatment, and transportation. However, the rapid development of the IoT has also brought a series of security issues. Data transmission in the IoT often needs to be carried out in different network environments, including wireless networks, cloud servers, and so on. This increases the risk of the data being eavesdropped, tampered with, or falsified in transit. Traditional encryption methods may not provide adequate protection, as the rise of quantum computing could threaten current encryption algorithms [1–3].

Quantum mechanics has brought new possibilities to the development of the IoT, and so far, various quantum communication and quantum computing technologies have emerged. These include quantum authentication technology [4–6], quantum neural networks [7,8], and quantum signature technology [9,10], which greatly enhance computational efficiency and security. Quantum authentication technology aims to verify quantum channels' security, ensuring communication integrity and confidentiality. It can be used to

verify the presence of potential eavesdroppers or tampering between communication parties. However, quantum authentication technology cannot provide absolute confidentiality as it still relies on classical communication for verification and key distribution. Quantum signature technology is used to ensure the non-repudiation of digital signatures, meaning that signers cannot deny the authenticity of their signatures. Although quantum signature technology can provide a certain level of security, it still relies on classical communication and computation for signature verification and identity authentication.

In contrast, Quantum Teleportation (QT) and Remote Quantum State Preparation (RSP) offer a higher level of security. These protocols utilize the properties of quantum entanglement to achieve secure transmission of information at the quantum level. The security of QT and RSP is based on the protection of quantum entangled states and the non-repudiation of quantum states during transmission. Due to the characteristics of quantum entanglement, even if an eavesdropper intercepts the transmitted quantum states, it cannot copy or steal the information contained within, thereby ensuring the security of the information. The development of QT has been rapid, starting with the concept proposed by Bennett et al. [11] in 1993, and now, there are many improved and novel QT schemes and other quantum communication protocols based on it, for example Quantum Information Splitting (QIS), Quantum Controlled Teleportation (QCT), Bidirectional Quantum Teleportation (BQT), Asymmetric Bidirectional Quantum Teleportation (ABQT), and RSP, which differs from the principles of QT. QCT introduces a third-party supervisor in addition to the communicating parties. Completing the communication requires the supervisor's consent to reconstruct the quantum state sent by the sender.

Afterward, many QCT schemes have also been developed [12–16]. In 2020, Chen et al. [12] proposed a scheme using a seven-qubit entangled state as the quantum channel to transmit Bell states. They considered the scheme's effectiveness and improved its security by incorporating decoy states. In 2021, Zhou et al. [17] implemented a bidirectional quantum teleportation scheme using a six-qubit quantum channel for quantum state transmission, providing valuable insights for enhancing efficiency. In 2022, Mengting Wang and Hai-Sheng Li [18] achieved bidirectional controlled quantum teleportation, contributing to the ongoing development of bidirectional quantum teleportation protocols [17–22]. Asymmetric bidirectional quantum teleportation refers to the scenario where the quantum states transmitted by both parties can be different, allowing for a greater variety of quantum state transmissions. In 2022, Kazemikhah et al. [23] proposed a scheme using an eight-qubit cluster state as the quantum channel for asymmetric bidirectional quantum state transfer. This ABQT scheme has also become an area of interest for researchers [23–25]. In addition, there are various types of quantum teleportation, including quantum information splitting [26,27], cyclic quantum teleportation [28–30], and probabilistic quantum teleportation [31–33]. On the other hand, RSP differs from the principles of quantum teleportation. In RSP, the sender, Alice, knows the quantum state being transmitted, and it is prepared remotely using entangled quantum channels. The concept of remote state preparation was introduced by Bennett et al. [34], adding another consideration to the field of quantum communication. Similar to QT, RSP also encompasses controlled RSP [35,36], bidirectional RSP [37], and various schemes for preparing different types of quantum states.

QT and RSP are also developing, but most current research focuses on one-way or two-way communication. In this paper, a scheme is proposed to realize secure communication between multiple devices in the IoT by using remote quantum state preparation and quantum teleportation. Experiments verified the scheme's feasibility, and quantum computers were used to simulate quantum circuits and measurement results. This technology enables secure communication between multiple devices in the IoT. Compared with other schemes, the advantage of this scheme is that it overcomes the single fixed one-way or two-way transmission protocol form in the past quantum teleportation, realizes the quantum communication of multiple devices in the IoT, and ensures the consideration of security and transmission efficiency.

The structure of this paper is as follows. The Section 2 introduces the theoretical derivation of the scheme and describes the theoretical construction of the channel and the specific details of the transmission steps. The Section 3 introduces the experimental verification of the scheme, which mainly realizes the specific channel-construction process, the transmission process, and the final quantum-state-reconstruction process through the IBM Quantum platform based on a theoretical derivation. The Section 4 mainly introduces the technology and corresponding steps needed in the practical implementation of this scheme. The Section 5 introduces the security analysis of the scheme in the IoT. It analyzes the security of the scheme by simulating two different attacks and deducing the impact of attacks on the communication scheme. In the Section 6, we make some comparisons with other schemes and briefly analyze the scheme principle and transmission efficiency. In the last section, the thesis and the scheme are summarized.

2. Specific Scheme

The scheme aims to achieve controlled RSP and QT. This part mainly expounds upon the scheme from the theoretical part and separately studies the channel-preparation and transmission protocol.

2.1. Controlled Quantum Communication Protocol Based on Multiple Parties

Assume that there are five parties in the process of RSP and QT: Alice, Bob, Charlie, David. and Eve. As the sender, Alice remotely prepares quantum states from Bob and Charlie and sends quantum states to David through teleportation. Eve acts as a supervisor and controls the entire transmission outcome.

2.2. The Process of Channel Preparation

We prepare the channel with nine qubits, three H gates, and six CNOT operations. First, the nine qubits have an initial state of $|0\rangle$, and then, the Hadamard gate and CNOT operations are performed on them to prepare the entangled quantum channel. The entire channel preparation process is as follows:

First, take nine qubits with an initial state of $|0\rangle$ and multiply them together as a tensor product:

$$|C_0\rangle = |0\rangle_1 \otimes |0\rangle_2 \otimes |0\rangle_3 \otimes |0\rangle_4 \otimes |0\rangle_5 \otimes |0\rangle_6 \otimes |0\rangle_7 \otimes |0\rangle_8 \otimes |0\rangle_9 \tag{1}$$

Then, apply the Hadamard gate to the qubits 1, 2, and 3, and the system transforms to $|C_1\rangle$:

$$\begin{aligned} |C_1\rangle &= \frac{|0\rangle + |1\rangle}{\sqrt{2}}_1 \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}}_2 \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}}_3 \otimes |0\rangle_4 \otimes |0\rangle_5 \otimes |0\rangle_6 \otimes |0\rangle_7 \otimes |0\rangle_8 \otimes |0\rangle_9 \\ &= \frac{1}{2\sqrt{2}}(|000000000\rangle \otimes |001000000\rangle \otimes |010000000\rangle \otimes |011000000\rangle \\ &+ \otimes |100000000\rangle \otimes |101000000\rangle \otimes |110000000\rangle \otimes |111000000\rangle)_{123456789} \end{aligned} \tag{2}$$

Next, we need to apply the CNOT operation. The control bit is Qubit 1, and the target bits are Qubit 5 and Qubit 6. The control bit is Qubit 2, and the target bit is 7. The control bit is Qubit 3, and the target bits are 4, 8, and 9.

$$\begin{aligned} |C_2\rangle = \frac{1}{2\sqrt{2}}(&|000000000\rangle \otimes |001100011\rangle \otimes |010000100\rangle \otimes |011100111\rangle \\ &+ \otimes |100110000\rangle \otimes |101111011\rangle \otimes |110011100\rangle \otimes |111111111\rangle)_{123456789} \end{aligned} \tag{3}$$

Now, we have an entangled quantum channel, which is then used for RSP and QT.

2.3. Controlled Remote State Preparation and Quantum Teleportation Scheme

Consider using the above nine-qubit entangled states as a quantum channel for multi-party communication. If Alice acts as the sender, Alice wants to remotely prepare any one-qubit and any two-qubit quantum states from Bob and Charlie, respectively, and transmit arbitrary two-qubit quantum states to David via quantum teleportation. Eve acts as an overseer. Alice, Bob, Charlie, David, and Eve share the nine-qubit entangled channels given in Equation (3), where the qubits (1,2,3) of the channels belong to Alice and are marked with bits a_0, a_1, and a_2 respectively, and Eve owns qubits (4), marked with e. The qubits (5,6) are in David's hands, and they are labeled d_1 and d_2; the qubits (7) and (8,9) belong to Bob and Charlie, respectively, and are labeled b and c_1 and c_2, respectively. We can also rewrite the channel given in Equation (3) as follows:

$$\begin{aligned}|C\rangle_{a_0a_1a_2ed_1d_2bc_1c_2} = \frac{1}{2\sqrt{2}}(&|000000000\rangle \otimes |001100011\rangle \otimes |010000100\rangle \\ &\otimes |011100111\rangle + \otimes |100110000\rangle \otimes |101111011\rangle \\ &\otimes |110011100\rangle \otimes |111111111\rangle)_{a_0a_1a_2ed_1d_2bc_1c_2}\end{aligned}$$

Step 1: Suppose Alice has an unknown quantum state of two qubits, which can be written as follows:

$$|\theta_1\rangle_{A_1A_2} = m\,|00\rangle + n\,|11\rangle$$

where m and n are complex numbers and satisfy $|m|^2 + |n|^2 = 1$. Alice wants to send the state $|\theta_1\rangle_{A_1A_2}$ to David. At the same time, Alice also wants to remotely prepare a single-qubit state and a two-qubit state for Bob and Charlie, respectively, written as:

$$\begin{aligned}|\theta_2\rangle &= a\,|0\rangle + b\,|1\rangle \\ |\theta_3\rangle &= c\,|00\rangle + d\,|11\rangle\end{aligned}$$

where a, b and c, d are also complex numbers and satisfy $|a|^2 + |b|^2 = 1$ and $|c|^2 + |d|^2 = 1$. The quantum state of the entire system can be expressed in the following form:

$$\begin{aligned}|\tau_0\rangle_{A_1A_2a_0a_1a_2ed_1d_2bc_1c_2} &= |\theta_1\rangle_{A_1A_2} \otimes |C\rangle_{a_0a_1a_2ed_1d_2bc_1c_2} \\ &= (m\,|00\rangle + n\,|11\rangle) \otimes (\frac{1}{2\sqrt{2}}(|000000000\rangle \otimes |001100011\rangle \\ &\otimes |010000100\rangle \otimes |011100111\rangle \otimes |100110000\rangle \otimes |101111011\rangle \\ &\otimes |110011100\rangle \otimes |111111111\rangle)))\end{aligned}$$

Step 2: Alice performs a three-qubit measurement of the qubits (A_1, A_2, a_0) based on the following basis:

$$\begin{aligned}|\phi_1\rangle &= \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle) \\ |\phi_2\rangle &= \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle) \\ |\phi_3\rangle &= \frac{1}{\sqrt{2}}(|001\rangle + |110\rangle) \\ |\phi_4\rangle &= \frac{1}{\sqrt{2}}(|001\rangle - |110\rangle)\end{aligned} \tag{4}$$

After taking Equation (4) for the measurement, Alice can obtain one of the four measurements, each with a one-in-four probability. The results of the remaining qubits are shown below. Now, the quantum state will be converted to $|\tau_1\rangle$:

$$\begin{aligned}|\tau_1\rangle = \frac{1}{4}[&|\phi_1\rangle_{A_1A_2a_0} \otimes (m\,|00000000\rangle + m\,|01100011\rangle + m\,|10000100\rangle \\ &+ m\,|11100111\rangle + n\,|00011000\rangle + n\,|01111011\rangle + n\,|10011100\rangle + n\,|11111111\rangle) \\ &+ |\phi_2\rangle_{A_1A_2a_0} \otimes (m\,|00000000\rangle + m\,|01100011\rangle + m\,|10000100\rangle \\ &+ m\,|11100111\rangle - n\,|00011000\rangle - n\,|01111011\rangle - n\,|10011100\rangle - n\,|11111111\rangle) \\ &+ |\phi_3\rangle_{A_1A_2a_0} \otimes (m\,|00011000\rangle + m\,|01111011\rangle + m\,|10011100\rangle + m\,|11111111\rangle \\ &+ n\,|00000000\rangle + n\,|01100011\rangle + n\,|10000100\rangle + n\,|11100111\rangle) \\ &+ |\phi_4\rangle_{A_1A_2a_0} \otimes (m\,|00011000\rangle + m\,|01111011\rangle + m\,|10011100\rangle + m\,|11111111\rangle \\ &- n\,|00000000\rangle - n\,|01100011\rangle - n\,|10000100\rangle - n\,|11100111\rangle)]_{a_1a_2ed_1d_2bc_1c_2}\end{aligned}$$

Step 3: Alice performs a two-qubit measurement of the qubits (a_1, a_2) on the following basis:

$$\begin{aligned}|\psi_1\rangle &= ac\,|00\rangle + ad\,|01\rangle + bc\,|10\rangle + bd\,|11\rangle \\ |\psi_2\rangle &= ad\,|00\rangle - ac\,|01\rangle + bd\,|10\rangle - bc\,|11\rangle \\ |\psi_3\rangle &= bc\,|00\rangle + bd\,|01\rangle - ac\,|10\rangle - ad\,|11\rangle \\ |\psi_4\rangle &= bd\,|00\rangle - bc\,|01\rangle - ad\,|10\rangle + ac\,|11\rangle\end{aligned} \tag{5}$$

The measurement basis given by Equation (5) is linearly independent, and after measurement, the remaining unmeasured qubits $ed_1d_2bc_1c_2$ will be transformed into $|\tau_2\rangle$; the equation for $|\tau_2\rangle$ is shown in Equation (A1), Appendix A.

Step 4: Eve makes the von Neumann measurement with the base $|+\rangle, |-\rangle$ on the qubit e. This is the last measurement, and Eve tells David the measurement result through the classical channel; the measurement basis is as follows:

$$\begin{aligned}|+\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |-\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\end{aligned} \tag{6}$$

Now, we can imagine a situation where, if Alice's measurement is $|\phi_3\rangle$ and $|\psi_4\rangle$, then after Alice sends the measurement to David, Bob, and Charlie over the classical channel, the remaining unmeasured qubits $ed_1d_2bc_1c_2$ will collapse into the following form:

$$\begin{aligned}|\tau^2\rangle = \frac{1}{4}&|\phi_3\rangle \otimes |\psi_4\rangle \otimes (mbd\,|011000\rangle - mbc\,|111011\rangle - mad\,|011100\rangle + mac\,|111111\rangle \\ &+ nbd\,|000000\rangle - nbc\,|100011\rangle - nad\,|000100\rangle + nac\,|100111\rangle)_{ed_1d_2bc_1c_2}\end{aligned}$$

Eve only needs to make one von Neumann measurement and, then, tell David the result; if her measurement is $|-\rangle$, then the remaining unmeasured qubits $d_1d_2bc_1c_2$ will collapse into $|\tau^3\rangle$:

$$\begin{aligned}|\tau^3\rangle &= \frac{1}{4\sqrt{2}}|\phi_3\rangle \otimes |\psi_4\rangle \otimes |-\rangle \otimes (mbd\,|11000\rangle + mbc\,|11011\rangle - mad\,|11100\rangle - mac\,|11111\rangle \\ &\quad + nbd\,|00000\rangle + nbc\,|00011\rangle - nad\,|00100\rangle - nac\,|00111\rangle) \\ &= \frac{1}{4\sqrt{2}}|\phi_3\rangle \otimes |\psi_4\rangle \otimes |-\rangle \otimes (m\,|11\rangle + n\,|00\rangle) \otimes (b\,|0\rangle - a\,|1\rangle) \otimes (d\,|00\rangle - c\,|11\rangle)\end{aligned}$$

Step 5: After Alice and Eve send the measurement results, David does not know the quantum state Alice wants to transmit due to quantum teleportation, so he needs to wait until Alice and Eve send the measurement results to David via classical channels, and then, he needs a specific unitary operation to transform the qubits he holds. While Bob and Charlie are using remote quantum state preparation, they both know what quantum state Alice is preparing, so Bob and Charlie need to use the appropriate unitary operation to achieve Alice's quantum state preparation at their location. Some unitary operators are shown in Equation (7):

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \tag{7}$$

For the example above, David uses the operator $X \otimes X$ to restore the qubits d_1d_2 to the original quantum state sent by Alice. Bob and Charlie use the operators ZX and $ZX \otimes X$ for qubits b and c_1c_2, respectively, to obtain the initial quantum state prepared by Alice at Bob and Charlie, respectively. The complete multi-party controlled quantum teleportation and remote quantum state preparation schemes are completed at this point. The detailed measurement results of Alice and Eve and the unitary operations required by Bob, Charlie, and David are shown in Table A1.

3. Experimental Verification

Currently, some researchers are utilizing the IBM Quantum platform for experimental quantum communication schemes. The IBM Quantum platform, initiated and developed by IBM, is one of the leading platforms in the field of quantum computing. It provides researchers with quantum computers with different numbers of qubits and simulated quantum computers to meet the simulation requirements of quantum communication systems at various scales and complexities. Additionally, the IBM platform provides a wealth of quantum computing programming interfaces and tools, such as Qiskit, which facilitate the development and experimentation of quantum communication simulations. Through these tools, users can design and implement quantum communication protocols, simulate quantum channel transmissions, and test quantum-error-correction schemes.

The proposed scheme employs a combination of two protocols: quantum teleportation and remote quantum state preparation. It utilizes various quantum gate circuits, and with the aid of the IBM Quantum platform, it can effectively achieve the channel preparation, remote quantum state preparation, and quantum state transmission required for this communication scheme.

3.1. Experimental Verification of Controlled Quantum Communication Protocols Based On Multiple Parties

Our experiments were carried out on the IBM Quantum platform, and IBM's ibmq-qasm-simulator was used for the simulation experiments. The experiment is divided into two parts: the experimental channel preparation and the experimental communication protocol implementation. The experimental environment is shown in Table A2. We implemented the channel preparation of the above theory and the whole process of the protocol through experiments. The simulation experiment was designed to be carried out in an ideal state without quantum computing attacks. The actual IoT multi-device communication and communication security analysis are described in detail in Sections 4 and 5.

3.2. Experiments on Channel Preparation

We need to prepare the channel circuit:

$$\begin{aligned} |C\rangle_{a_0a_1a_2ed_1d_2bc_1c_2} = \frac{1}{2\sqrt{2}}(&|000000000\rangle \otimes |001100011\rangle \otimes |010000100\rangle \otimes |011100111\rangle \\ &+ \otimes |100110000\rangle \otimes |101111011\rangle \otimes |110011100\rangle \otimes |111111111\rangle)_{a_0a_1a_2ed_1d_2bc_1c_2} \end{aligned}$$

q_2 to q_10 represent the qubits (a_0, a_1, a_2, e, d_1, d_2, b, c_1, c_2), and the initial state from q_2 to q_{10} is $|0\rangle$.

First, q_2, q_3 and q_4 are prepared using the H gate, and then, six CNOT operations are used to complete the channel preparation. Figure 1 shows the entire process of channel preparation. Figure 2 shows the probability obtained by measuring each quantum state. A total of 8192 shots were made, and the probability of collapsing into eight possible quantum states was basically equal. The circuit and measurement experiment verified the entanglement degree and reliability of the channel.

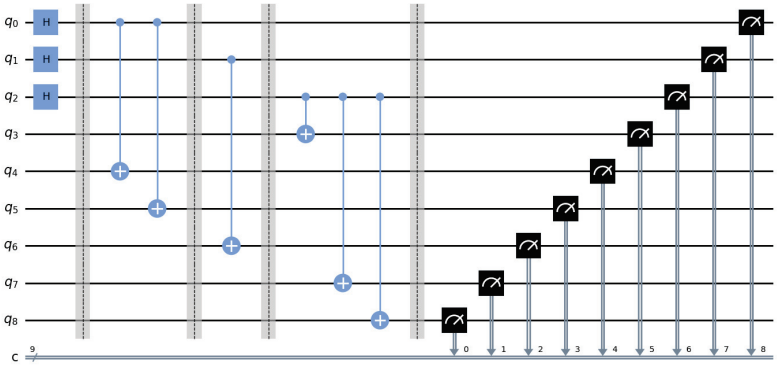


Figure 1. This circuit builds a nine-qubit entanglement channel. From left to right, q_2 to q_10 represent the qubits (a_0, a_1, a_2, e, d_1, d_2, b, c_1, c_2), and the initial state from q_2 to q_{10} is $|0\rangle$. The first part is the H gate, the middle parts 2, 3, and 4 are the CNOT operation, and the fifth part is the measurement operation. The number pointed by the arrow of the measurement mark represents the classical bit number. The diagram shows the circuit diagram process of the channel preparation experiment.

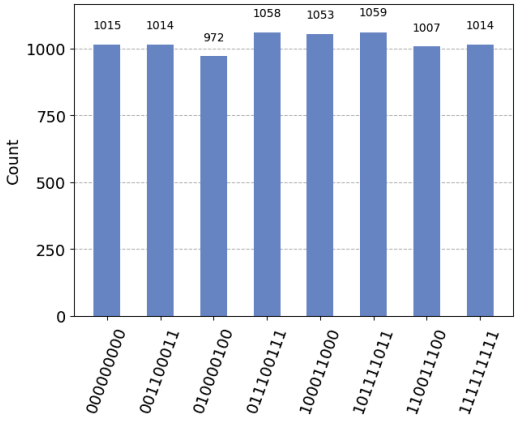


Figure 2. After three H gates and six CNOT operations, the channel system is entangled; the horizontal coordinate represents the corresponding quantum state that may collapse, and the vertical coordinate represents the number of collapses of each quantum state. The figure shows that the collapse probability of each quantum state is basically the same, which reflects the rationality of the entangled channel.

3.3. Experimental Quantum Teleportation and Remote Quantum State Preparation Process

A total of 11 particles are required for the entire transmission process. q_0 and q_1 belong to Alice, while q_2 through q_{10} are channel particles. Eleven classical registers are used during the circuit to store the measured values.

The transmission process is divided into three parts, including the initialization of the quantum state, the measurement operation performed separately, and the corresponding unitary operation at the end.

The first is the initialization phase. Since we want to transmit arbitrary quantum states, we must build a quantum gate that initializes the coefficients. For Alice, her transmission to Bob, Charlie, and David requires quantum state initialization, respectively, in order to achieve any quantum state transmission process. Here, we used Qiskit's Operator function to do this. Assuming the coefficients from Alice to David are $m = \sqrt{\frac{6}{7}}$ and $n = \frac{1}{\sqrt{7}}$, then the U gate can be expressed as:

$$\frac{1}{\sqrt{7}}\begin{bmatrix}\sqrt{6} & 1\\ 1 & -\sqrt{6}\end{bmatrix}$$

Similarly, suppose we need to initialize the quantum states to be passed to Bob and Charlie, whose coefficients are:

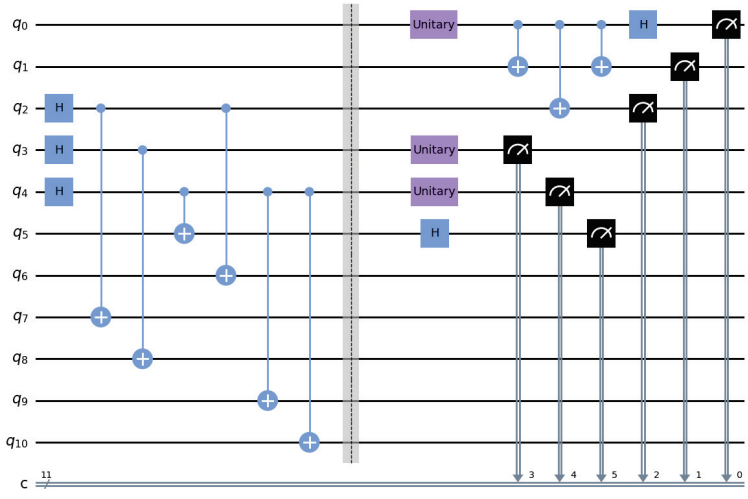
$$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}, \frac{1}{\sqrt{11}}\begin{bmatrix}\sqrt{10} & 1\\ 1 & -\sqrt{10}\end{bmatrix}$$

After initializing each coefficient, we can take the measurement step; Alice takes two measurements, namely the three-qubit measurement and the two-qubit measurement, while Eve takes only one von Neumann measurement; in the three-qubit measurement, we measure A_1, A_2 and a_0, corresponding to q_0, q_1, and q_2, and in the two-qubit measurement, we measure a_1 and a_2, corresponding to the circuit qubits q_3 and q_4. Eve performs a von Neumann measurement of the particle e corresponding to q_5.

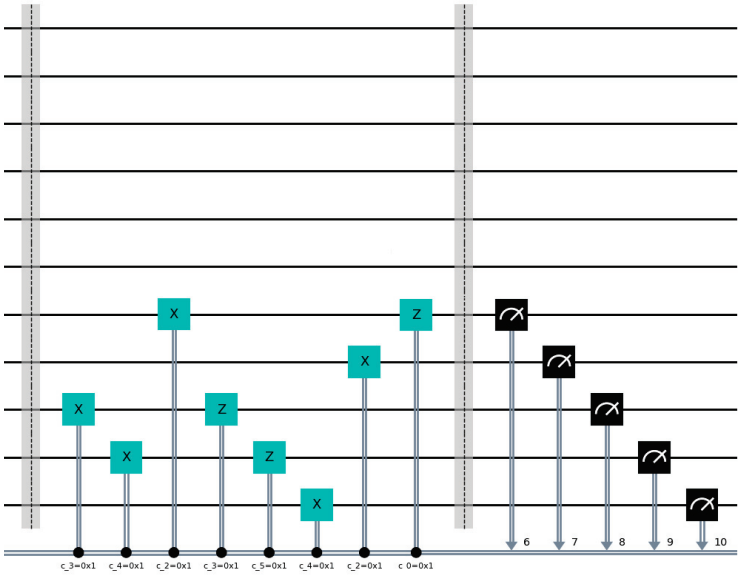
Bob, Charlie, and David then need to perform the appropriate unitary operations on their particles b, c_1, c_2, d_1, and d_2 to achieve the final process of remote state preparation and quantum teleportation. The whole circuit diagram is shown in Figure 3, including the channel preparation, the preparation and initialization of the quantum state to be transmitted, and the circuit for the final unitary operation.

For the coefficients mentioned above, the quantum state $\sqrt{\frac{6}{7}}|00\rangle + \frac{1}{\sqrt{7}}|11\rangle$ should be six-times more likely to measure $|00\rangle$ than $|11\rangle$. Similarly, for the quantum states $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ and $\sqrt{\frac{10}{11}}|00\rangle + \frac{1}{\sqrt{11}}|11\rangle$, the former is equally likely to measure $|0\rangle$ and $|1\rangle$, while the latter is ten-times more likely to measure $|00\rangle$ than $|11\rangle$. We tested the above process in the IBM Quantum platform qasm-simulator. To make the experimental results more accurate and reduce the statistical error, the scheme's performance was evaluated comprehensively. In this article, parameter "shots" was set to 65,536; too large a value may cause performance problems. The specific measurement results of the experiment are shown in Figure 4.

The experiment showed that the measured results of Bob, Charlie, and David after restoring the quantum state through their respective unitary operations were basically consistent with the theoretically calculated probability values, and the entire experimental process of the transmission protocol and the preparation protocol is complete.



(**a**)



(**b**)

Figure 3. This is a circuit diagram of controlled remote state preparation and quantum teleportation using a nine-qubit entangled channel. From q_0 to q_{10} represent the qubits (A_1, A_2, a_1, a_2, e, d_1, d_2, b, c_1, c_2). Marked by the dividing line, the first part of Figure (**a**) is the channel preparation; the second part is the preparation and initialization of the quantum state to be transmitted; the first part of Figure (**b**) is the corresponding unitary operation; the second part is the measurement process. "H" indicates the H gate; the light blue symbol "+" indicates the target bit; the light blue dot represents the control bit; both the Init gate and Unitary gate are initialization operations for quantum states to enable any quantum state transmission. The black mark represents the measurement operation; the X and Z identifiers represent X operations and Z operations, whose matrix form is indicated in Equation (7). The two diagrams show the complete protocol process.

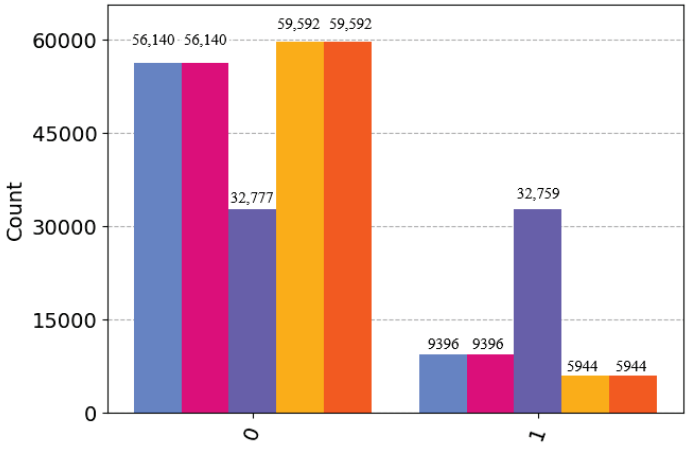


Figure 4. The 0 and 1 represent $|0\rangle$ and $|1\rangle$. The first two bars of 0 and 1, respectively, are David's experimental results; the middle bar is Bob's experimental results; the last two bars are Charlie's experimental results, which are very close to the expected results.

4. Application Analysis of Quantum Secure Communication Based on Multi-Devices IoT

The preparation based on controlled quantum teleportation and remote quantum states has many application scenarios:

Quantum secure communication: This can be used to enable secure communication in the IoT. Through the transmission and preparation of quantum states, the secure transmission of encryption keys can be achieved, thus ensuring the confidentiality and integrity of communication. This has important implications regarding sensitive data transfer and privacy protection, such as financial transactions, medical records, and personal authentication.

Quantum sensing networks: IoT sensor networks can use QT and RSP to enable efficient sensor data transmission and processing. By transmitting quantum states to remote nodes, distributed perception and collaborative processing can be achieved, thereby improving the performance and capability of sensor networks. This has potential applications in areas such as environmental monitoring, intelligent transportation, and agriculture.

Distributed quantum computing: QT and RSP can enable distributed quantum computing in the IoT. By preparing quantum states remotely, different devices can work together on quantum computing tasks, thus enabling the acceleration and expansion of distributed computing. This has potential applications in optimization problem-solving, machine learning, and large-scale data processing.

Quantum authentication and authentication in the IoT: Quantum teleportation and remote quantum state preparation can be applied to IoT's authentication and authentication process. By taking advantage of the uniqueness and non-repudiation of quantum states, a more-secure and -reliable authentication mechanism can be achieved, thereby preventing deception and forgery.

This part mainly introduces the secure communication process of quantum teleportation and remote quantum state preparation in the IoT. In the actual multi-party communication of the IoT, the five-qubit QT and RSP based on the nine-qubit quantum channel have a broad application prospect. This part introduces the application scenario analysis of the five-qubit QT based on the nine-qubit quantum channel transmission and the RSP in the actual multi-party communication of the IoT and uses QT to achieve secure quantum information transmission in the multiple devices communication of the IoT. Suppose there are five devices: Devices A (Alice), B (Bob), C (Charlie), D (David), and E (Eve), and Alice wants to transmit the two-qubit state to David via QT and prepare one qubit and two qubits for Bob and Charlie, respectively. As the controller, Eve monitors and controls the

transmission, aiming to ensure the security of the transmission. Figure 5 shows the use of quantum channels, which are used to transmit and measure quantum states, and classical channels, which are used to transmit unitary operations.

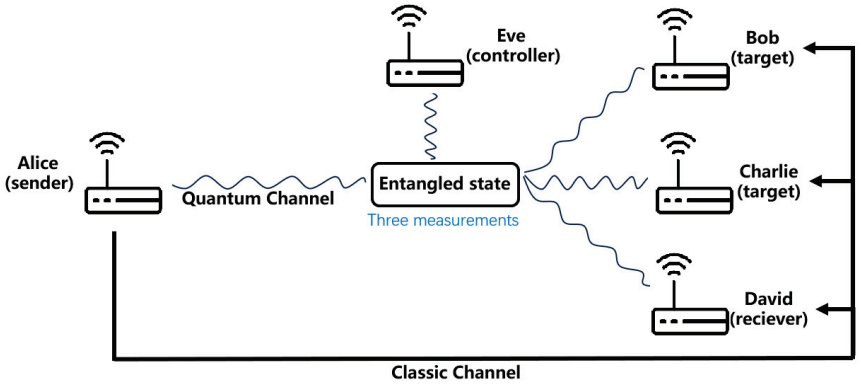


Figure 5. Alice represents the sending device; David represents the receiving device; Bob and Charlie represent the target device for the preparation of the quantum state; Eve is the control device. The figure shows the use of quantum channels, which are used to transmit and measure quantum states, and classical channels, which are used to transmit unitary operations.

Step 1: Prepare the devices:

In practical IoT multi-party communication for QT and remote quantum state preparation, each device must be equipped with a quantum computer to generate, manipulate, and measure qubits. These devices can be nodes or end devices in the IoT, such as sensors, smart devices, or communication terminals.

Step 2: Qubit channel preparation:

In order to realize QT and remote quantum state preparation, a quantum channel must be prepared first. The initial quantum states can be entangled together to establish entangled quantum channels by applying appropriate quantum gate operations. This entangled quantum channel will be used to transmit information about quantum states.

Step 3: Initialize the quantum state of the transmission:

As the sender, Alice uses her quantum computer to prepare the two-qubit state to transmit to David. This quantum state can be initialized using appropriate unitary operations such as U gates.

Step 4: Measurement base selection and measurement:

In QT and RSP, Alice needs to choose the measurement basis given in Formula (5) to measure the tensor product of the quantum state and quantum channel that she has. However, this measurement process does not immediately provide classical bit information. To obtain classical bit information, the quantum states of Bob and Charlie also need to be two-qubit-measured against the tensor product of the channel quantum state, while Eve needs to be single-qubit-measured against the corresponding qubit in the channel.

Step 5: Quantum state transmission and preparation:

Alice sends David the quantum state information with the corresponding unitary operation through the classical communication channel. David can use his quantum computer to perform the corresponding operation to recover the quantum state information sent by Alice. At the same time, Bob and Charlie can also use their quantum computer to perform unitary operations on the quantum states they measure in the channel to achieve the quantum states Alice prepared at their place.

Step 6: Security assurance:

As a controller, Eve can monitor and control communication links. If the communication link is not secure, Eve can interrupt the communication link to ensure the security of the transmission. In the subsequent communication process, Eve does not perform the corresponding single-qubit measurement, thus protecting the security of quantum information.

By building entangled quantum channels and using appropriate measurement bases, Alice can send states to Bob, Charlie, and David and prevent eavesdropping by adding a control device, Eve. QT and RSP enable secure information transmission in the IoT multi-device communication. This method can be applied to multi-party communication scenarios in the IoT to ensure the confidentiality and integrity of quantum information in communication, while also helping to protect sensitive information and improve the security of the IoT system. However, for larger-scale and complex IoT systems, further research and development of quantum security protocols and technologies are needed to deal with potential attacks and security threats.

5. Security Analysis

In this section, we explore the security of this solution in the IoT. Suppose there is an eavesdropper except for Device A (Alice), Device B (Bob), Device C (Charlie), Device D (David), and controller Device E (Eve). In that case, the eavesdropper has several attack methods, namely intercept–replace–resend attack and intercept–measure–resend attack; we verified the security of this scheme by studying these two attack methods. Since this scheme has two methods, namely remote quantum state preparation and QT, we discuss the security of Alice's quantum state preparation at Bob and Charlie and the security of Alice's quantum state sending to David through QT.

Since there is a substitution attack or measurement attack during the remote quantum state preparation, the receiver to perform remote preparation will know the prepared quantum state in advance, and the prepared quantum state does not match the target quantum state, indicating that the protocol has an attack, so the attack will be detected. We need to add a way to improve security for the transmission process that Alice sends to David. After Alice sends her measurement to David over the classical channel, in the process, we can add a sequence of quantum states $|01\rangle_{ij}, |10\rangle_{ij}$ to combine it with the measurement Alice sends to David and perform the appropriate CNOT operation. If she wants to send her measurement results to David, she needs to select any state in a sequence and insert it at any point in transmitting the quantum state, and then, Alice sends the quantum state to David through the classical channel. David only needs to use the quantum state sent by Alice and the measurement results to know the quantum state sent by Alice and whether the QT protocol is eavesdropped.

5.1. Intercept–Replace–Resend Attack

Suppose there is an intercept–replace–resend attack between Alice, Bob, and Charlie since Alice uses remote quantum state preparation. In that case, Bob and Charlie know in advance the quantum state Alice will prepare in their place, so once the replace–resend occurs, Bob and Charlie cannot restore the remaining unmeasured qubits to the quantum state Alice wants to prepare by using the corresponding unitary operator, so it can be found that the enemy attacks the communication. Still, in this process, if the classical channel has been eavesdropped, information will be leaked. The enemy will directly use the information.

Alice sends her measurements to David, assuming that the eavesdropper is trying to intercept Alice's quantum state and the quantum state Alice intended to transmit to David in the first place. However, Alice has changed the conversion rule of this quantum state, so when David receives the false quantum state replaced by the eavesdropper, he first checks the insertion position and conversion rule told by Alice in advance and then performs the corresponding CNOT operation to obtain the converted quantum state. The quantum state sent by Alice over the classical channel is then compared with the quantum

state obtained by David, who also realizes that the information he received is wrong, and the eavesdropper's attack fails.

5.2. Intercept–Measure–Resend Attack

Similarly, since Bob and Charlie know in advance the quantum state to be prepared by Alice and the quantum state is informed by Alice through the classical channel, Bob and Charlie will find it invalid if an attack occurs during the remote quantum state preparation process. When Alice sends information to David via QT, suppose there is an eavesdropper who tries to intercept the quantum state Alice sends to David when Alice sends the measurement result to the receiver, David, and performs an appropriate measurement on that quantum state, then the information Alice is trying to obtain is sent to David. However, the eavesdropper's interception will not succeed because Alice has already performed a proper transformation of the quantum state before sending the measurement results. Since the eavesdropper does not know Alice's conversion rules, no matter how it performs the measurement, it will not obtain any information about Alice's transmission.

For example, if the measurements of Alice are $|\phi_1\rangle$ and $|\psi_1\rangle$and the measurement of Eve is $|+\rangle$, then David's result is $(m|00\rangle + n|11\rangle)_{d_1d_2}$. If Alice chooses the sequence $|01\rangle_{ij}$, then d_1, d_2 are the control bits, and i and j are the target bits, then the quantum state can be rewritten as:

$$(m|00\rangle + n|11\rangle)|01\rangle(a|0\rangle + b|1\rangle)(c|00\rangle + d|11\rangle)_{d_1d_2ijbc_1c_2}$$

If the CNOT operation is performed, d_1 controls i, and d_2 controls j, then the quantum state can be rewritten as:

$$(m|0001\rangle + n|1110\rangle)(a|0\rangle + b|1\rangle)(c|00\rangle + d|11\rangle)_{d_1d_2ijbc_1c_2}$$

Then, Alice transmits this quantum state to David through the classical channel, where d_1d_2ij is David's, and the conversion rule is known only to Alice and David. Assuming that the eavesdropper performs a single particle measurement on it, the entangled quantum state sequence will be affected. If the measurement attack occurs, the quantum state sequence will be affected. David would have detected it.

As another example, if Alice's measurement result is $|\phi_2\rangle$ and $|\psi_3\rangle$ and Eve's measurement result is $|-\rangle$, through the previous analysis, We can obtain David's result as $(m|00\rangle - n|11\rangle)_{d_1d_2}$, and if Alice chooses $|11\rangle_{ij}$, we also take d_1, d_2 as the control bits and i and j as the target bits. The quantum state can be rewritten as:

$$(m|00\rangle - n|11\rangle)|11\rangle \otimes (b|0\rangle - a|1\rangle) \otimes (c|00\rangle - d|11\rangle)_{d_1d_2ijc_1c_2}$$

The same as above, we perform the CNOT operation, and the quantum state changes:

$$(m|0011\rangle - n|1100\rangle) \otimes (b|0\rangle - a|1\rangle) \otimes (c|00\rangle - d|11\rangle)_{d_1d_2ijc_1c_2}$$

Similarly, if the eavesdropper performs a single-quantum measurement or double-qubit measurement on the quantum state sent by Alice and the sequence of quantum states sent together with Alice, the original transmission form will be changed, leading to the discovery of the attack.

The results of the above two tests showed that neither an intercept–measure–resend attack nor an intercept–replace–resend attack can obtain valid information while transmitting quantum states. Therefore, our method is proven to be safe and reliable. By taking additional steps to ensure the security of the transmitted quantum state, we can further improve the security of QT and enable it to be used for a wide range of secure communication applications. Moreover, the security of QT is not limited to detecting and preventing eavesdropping attacks. It can also be used to distribute keys between two parties and, then, for secure communication. In this case, Alice and Bob would use QT to transmit the shared key, which would then be used to encrypt and decrypt the messages sent between them.

Since any attempt to intercept the key will be detected, the technology provides a high degree of security for sensitive communications of IoT multiple devices, greatly improving the ability to prevent quantum computing attacks.

6. Protocol Comparison and Analysis

In this section, we compare and analyze related protocols from the aspects of quantum channel selection, whether there is a supervisor, and resource utilization. The formula for transmission efficiency was constructed based on the number of particles transmitted, the number of channel particles, the number of classical bit particles used, and the number of auxiliary particles required:

$$\eta = \frac{c}{p+q+t}$$

c is the number of particles transmitted or prepared; p is the number of particles required by the channel; q is the number of classical bits used in the transmission process; t is the number of auxiliary particles used. Five particles are transmitted in our scheme, while the channel uses nine particles. In the whole transmission process, classic bits use six due to the need to use classical bits to perform the corresponding unitary operation of the receiver. Our efficiency is 33.3%. The formula for transmission efficiency mentioned in this article and other comparative studies do not consider information transmission outside of the protocol. Therefore, this formula applies only within the specific transmission protocol context.

Table 1 shows the schemes for comparison in this paper.

Table 1. Comparison of transmission efficiency results, including the transmission channel, the number of particles transmitted, and whether a controller was added.

Protocol	Teleported Particles	Quantum Channel	Controller (Yes/No)	Efficiency
[38]	Four-qubit	Eight-qubit	No	25%
[39]	Three-qubit	Five-qubit	No	30%
[24]	Five-qubit	Nine-qubit	No	23.8%
[19]	Three-qubit	Eight-qubit	No	18.75%
[18]	Three-qubit	Five-qubit	No	30%
proposed	Five-qubit	Nine-qubit	Yes	33.3%

The protocol used in [38] adopted an eight-qubit channel to realize bidirectional transmission of any two-qubit state, but the transmission efficiency formula given in [38] is inconsistent with the transmission efficiency calculation formula given by us. Here, according to the transmission efficiency calculation formula adopted in this paper, $\eta = \frac{4}{8+8+0} = 25\%$, the efficiency of [38] should be 25%. In [39], they used a five-qubit channel to transmit three-qubit quantum states in a bidirectional controlled manner, with an efficiency of 30%. In [24], consistent with our scheme, a nine-qubit channel was also adopted for communication, and a total of five-qubit quantum states were also transmitted bi-directionally. The paper [19] used an 8-qubit channel to transmit a 3-qubit channel, with a transmission efficiency of 3/16, which is relatively low. The transmission form of the paper [18] was the same as that of the paper [39], but the transmission efficiency was not improved. However, we reduced the amount of classical bits, and it is a controlled quantum teleportation with relatively higher efficiency.

Finally, different from the scheme adopted in the above paper, our scheme aimed to realize a multi-receiver multi-party communication, which is also one of the differences from the above schemes. Through the comparison and analysis, the advantages of this scheme were higher efficiency and multi-party communication.

7. Conclusions

This paper proposed a multi-party-based scheme for controlled remote quantum state preparation and quantum teleportation, which realizes multi-device communication in the IoT using a nine-qubit entanglement channel. The scheme was explained and verified in detail from the two aspects of theory and experiment, and the safety and efficiency of the scheme were analyzed. This paper's main contributions and discoveries were as follows: a novel nine-qubit entangled channel was constructed, which can simultaneously achieve controlled remote quantum state preparation and quantum teleportation, expanding the possibility and diversity of quantum communication. The channel preparation, measurement operation, and unitary operation of the scheme were successfully simulated on the IBM Quantum platform, and 8192 shots verified the entanglement degree and reliability of the channel. This paper not only analyzed the steps and methods of the scheme in the multi-device communication of the IoT, but also deduced the effect of the attack on the communication scheme by simulating two different attack modes and analyzed that the scheme has high security and anti-interference ability in the multi-device communication of the IoT. Compared with other schemes, this paper briefly analyzed two aspects of principle and transmission efficiency and pointed out the advantages of this scheme. This scheme can provide a reference for future secure communication of IoT devices and multi-party quantum transmission protocols. Although QT and RSP have advantages in terms of security, they still face other implementation challenges, such as the reliability of the entanglement distribution and the impact of noise and errors. This scheme only considered the ideal noise-free quantum communication scheme. An improvement of this paper may be to discuss the influence of noise and take corresponding measures to deal with the influence of noise.

Author Contributions: Conceptualization, D.L.; methodology, Y.F. and D.L.; validation, Y.F., X.H., Y.J., and Y.Z.; writing—original draft preparation, Y.F.; writing—review and editing, Y.F., J.Z., Y.T., and X.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (62172060), the Sichuan Science and Technology Program (2022YFG0316, 2023ZHCG0004), and the National Key R&D Plan (2022YFB3304303).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

IoT	Internet of Things
QT	Quantum Teleportation
QCT	Quantum Controlled Teleportation
QKD	Quantum Key Distribution
QSC	Quantum Superdense Coding
QIS	Quantum Information Splitting
ABQT	Asymmetric Bidirectional Quantum Teleportation
RSP	Remote State Preparation

Appendix A

$$
\begin{aligned}
|\tau_2\rangle = \frac{1}{4}[&|\phi_1\rangle \otimes |\psi_1\rangle \otimes (mac\,|000000\rangle + mad\,|100011\rangle + mbc\,|000100\rangle + mbd\,|100111\rangle \\
&+nac\,|011000\rangle + nad\,|111011\rangle + nbc\,|011100\rangle + nbd\,|111111\rangle) \\
&+ |\phi_2\rangle \otimes |\psi_1\rangle \otimes (mac\,|000000\rangle + mad\,|100011\rangle + mbc\,|000100\rangle + mbd\,|100111\rangle \\
&-nac\,|011000\rangle - nad\,|111011\rangle - nbc\,|011100\rangle - nbd\,|111111\rangle) \\
&+ |\phi_3\rangle \otimes |\psi_1\rangle \otimes (mac\,|011000\rangle + mad\,|111011\rangle + mbc\,|011100\rangle + mbd\,|111111\rangle \\
&+nac\,|000000\rangle + nad\,|100011\rangle + nbc\,|000100\rangle + nbd\,|100111\rangle) \\
&+ |\phi_4\rangle \otimes |\psi_1\rangle \otimes (mac\,|011000\rangle + mad\,|111011\rangle + mbc\,|011100\rangle + mbd\,|111111\rangle \\
&-nac\,|000000\rangle - nad\,|100011\rangle - nbc\,|000100\rangle - nbd\,|100111\rangle) \\
&+ |\phi_1\rangle \otimes |\psi_2\rangle \otimes (mad\,|000000\rangle - mac\,|100011\rangle + mbd\,|000100\rangle - mbc\,|100111\rangle \\
&+nad\,|011000\rangle - nac\,|111011\rangle + nbd\,|011100\rangle - nbc\,|111111\rangle) \\
&+ |\phi_2\rangle \otimes |\psi_2\rangle \otimes (mad\,|000000\rangle - mac\,|100011\rangle + mbd\,|000100\rangle - mbc\,|100111\rangle \\
&-nad\,|011000\rangle + nac\,|111011\rangle - nbd\,|011100\rangle + nbc\,|111111\rangle) \\
&+ |\phi_3\rangle \otimes |\psi_2\rangle \otimes (mad\,|011000\rangle - mac\,|111011\rangle + mbd\,|011100\rangle - mbc\,|111111\rangle \\
&+nad\,|000000\rangle - nac\,|100011\rangle + nbd\,|000100\rangle - nbc\,|100111\rangle) \\
&+ |\phi_4\rangle \otimes |\psi_2\rangle \otimes (mad\,|011000\rangle - mac\,|111011\rangle + mbd\,|011100\rangle - mbc\,|111111\rangle \\
&-nad\,|000000\rangle + nac\,|100011\rangle - nbd\,|000100\rangle + nbc\,|100111\rangle) \\
&+ |\phi_1\rangle \otimes |\psi_3\rangle \otimes (mbc\,|000000\rangle + mbd\,|100011\rangle - mac\,|000100\rangle - mad\,|100111\rangle \\
&+nbc\,|011000\rangle + nbd\,|111011\rangle - nac\,|011100\rangle - nad\,|111111\rangle) \\
&+ |\phi_2\rangle \otimes |\psi_3\rangle \otimes (mbc\,|000000\rangle + mbd\,|100011\rangle - mac\,|000100\rangle - mad\,|100111\rangle \\
&-nbc\,|011000\rangle - nbd\,|111011\rangle + nac\,|011100\rangle + nad\,|111111\rangle) \\
&+ |\phi_3\rangle \otimes |\psi_3\rangle \otimes (mbc\,|011000\rangle + mbd\,|111011\rangle - mac\,|011100\rangle - mad\,|111111\rangle \\
&+nbc\,|000000\rangle + nbd\,|100011\rangle - nac\,|000100\rangle - nad\,|100111\rangle) \\
&+ |\phi_4\rangle \otimes |\psi_3\rangle \otimes (mbc\,|011000\rangle + mbd\,|111011\rangle - mac\,|011100\rangle - mad\,|111111\rangle \\
&-nbc\,|000000\rangle - nbd\,|100011\rangle + nac\,|000100\rangle + nad\,|100111\rangle) \\
&+ |\phi_1\rangle \otimes |\psi_4\rangle \otimes (mbd\,|000000\rangle - mbc\,|100011\rangle - mad\,|000100\rangle + mac\,|100111\rangle \\
&+nbd\,|011000\rangle - nbc\,|111011\rangle - nad\,|011100\rangle + nac\,|111111\rangle) \\
&+ |\phi_2\rangle \otimes |\psi_4\rangle \otimes (mbd\,|000000\rangle - mbc\,|100011\rangle - mad\,|000100\rangle + mac\,|100111\rangle \\
&-nbd\,|011000\rangle + nbc\,|111011\rangle + nad\,|011100\rangle - nac\,|111111\rangle) \\
&+ |\phi_3\rangle \otimes |\psi_4\rangle \otimes (mbd\,|011000\rangle - mbc\,|111011\rangle - mad\,|011100\rangle + mac\,|111111\rangle \\
&+nbd\,|000000\rangle - nbc\,|100011\rangle - nad\,|000100\rangle + nac\,|100111\rangle) \\
&+ |\phi_4\rangle \otimes |\psi_4\rangle \otimes (mbd\,|011000\rangle - mbc\,|111011\rangle - mad\,|011100\rangle + mac\,|111111\rangle \\
&-nbd\,|000000\rangle + nbc\,|100011\rangle + nad\,|000100\rangle - nac\,|100111\rangle)]_{ed_1d_2bc_1c_2}
\end{aligned}
\tag{A1}
$$

Appendix B

Table A1. The measurement results of Alice and Eve, the state of the remaining qubits, and the specific unitary operator, where ZX indicates that the Z operation is performed before the X operation is performed.

Alice's R	Eve's R	Remaining Unmeasured Qubits	David's Gate	Bob's Gate	Charlie's Gate
$\|\phi_1\rangle \otimes \|\psi_1\rangle$	$\|+\rangle$	$(m\|00\rangle + n\|11\rangle) \otimes (a\|0\rangle + b\|1\rangle) \otimes (c\|00\rangle + d\|11\rangle)$	$I \otimes I$	I	$I \otimes I$
	$\|-\rangle$	$(m\|00\rangle + n\|11\rangle) \otimes (a\|0\rangle + b\|1\rangle) \otimes (c\|00\rangle - d\|11\rangle)$	$I \otimes I$	I	$Z \otimes I$
$\|\phi_2\rangle \otimes \|\psi_1\rangle$	$\|+\rangle$	$(m\|00\rangle - n\|11\rangle) \otimes (a\|0\rangle + b\|1\rangle) \otimes (c\|00\rangle + d\|11\rangle)$	$Z \otimes I$	I	$I \otimes I$
	$\|-\rangle$	$(m\|00\rangle - n\|11\rangle) \otimes (a\|0\rangle + b\|1\rangle) \otimes (c\|00\rangle - d\|11\rangle)$	$Z \otimes I$	I	$Z \otimes I$
$\|\phi_3\rangle \otimes \|\psi_1\rangle$	$\|+\rangle$	$(m\|11\rangle + n\|00\rangle) \otimes (a\|0\rangle + b\|1\rangle) \otimes (c\|00\rangle + d\|11\rangle)$	$X \otimes X$	I	$I \otimes I$
	$\|-\rangle$	$(m\|11\rangle + n\|00\rangle) \otimes (a\|0\rangle + b\|1\rangle) \otimes (c\|00\rangle - d\|11\rangle)$	$X \otimes X$	I	$Z \otimes I$
$\|\phi_4\rangle \otimes \|\psi_1\rangle$	$\|+\rangle$	$(m\|11\rangle - n\|00\rangle) \otimes (a\|0\rangle + b\|1\rangle) \otimes (c\|00\rangle + d\|11\rangle)$	$ZX \otimes X$	I	$I \otimes I$
	$\|-\rangle$	$(m\|11\rangle - n\|00\rangle) \otimes (a\|0\rangle + b\|1\rangle) \otimes (c\|00\rangle - d\|11\rangle)$	$ZX \otimes X$	I	$Z \otimes I$
$\|\phi_1\rangle \otimes \|\psi_2\rangle$	$\|+\rangle$	$(m\|00\rangle + n\|11\rangle) \otimes (a\|0\rangle + b\|1\rangle) \otimes (d\|00\rangle - c\|11\rangle)$	$I \otimes I$	I	$ZX \otimes X$
	$\|-\rangle$	$(m\|00\rangle + n\|11\rangle) \otimes (a\|0\rangle + b\|1\rangle) \otimes (d\|00\rangle + c\|11\rangle)$	$I \otimes I$	I	$X \otimes X$
$\|\phi_2\rangle \otimes \|\psi_2\rangle$	$\|+\rangle$	$(m\|00\rangle - n\|11\rangle) \otimes (a\|0\rangle + b\|1\rangle) \otimes (d\|00\rangle - c\|11\rangle)$	$Z \otimes I$	I	$ZX \otimes X$
	$\|-\rangle$	$(m\|00\rangle - n\|11\rangle) \otimes (a\|0\rangle + b\|1\rangle) \otimes (d\|00\rangle + c\|11\rangle)$	$Z \otimes I$	I	$X \otimes X$
$\|\phi_3\rangle \otimes \|\psi_2\rangle$	$\|+\rangle$	$(m\|11\rangle + n\|00\rangle) \otimes (a\|0\rangle + b\|1\rangle) \otimes (d\|00\rangle - c\|11\rangle)$	$X \otimes X$	I	$ZX \otimes X$
	$\|-\rangle$	$(m\|11\rangle + n\|00\rangle) \otimes (a\|0\rangle + b\|1\rangle) \otimes (d\|00\rangle + c\|11\rangle)$	$X \otimes X$	I	$X \otimes X$
$\|\phi_4\rangle \otimes \|\psi_2\rangle$	$\|+\rangle$	$(m\|11\rangle - n\|00\rangle) \otimes (a\|0\rangle + b\|1\rangle) \otimes (d\|00\rangle - c\|11\rangle)$	$ZX \otimes X$	I	$ZX \otimes X$
	$\|-\rangle$	$(m\|11\rangle - n\|00\rangle) \otimes (a\|0\rangle + b\|1\rangle) \otimes (d\|00\rangle + c\|11\rangle)$	$ZX \otimes X$	I	$X \otimes X$
$\|\phi_1\rangle \otimes \|\psi_3\rangle$	$\|+\rangle$	$(m\|00\rangle + n\|11\rangle) \otimes (b\|0\rangle - a\|1\rangle) \otimes (c\|00\rangle + d\|11\rangle)$	$I \otimes I$	ZX	$I \otimes I$
	$\|-\rangle$	$(m\|00\rangle + n\|11\rangle) \otimes (b\|0\rangle - a\|1\rangle) \otimes (c\|00\rangle - d\|11\rangle)$	$I \otimes I$	ZX	$Z \otimes I$
$\|\phi_2\rangle \otimes \|\psi_3\rangle$	$\|+\rangle$	$(m\|00\rangle - n\|11\rangle) \otimes (b\|0\rangle - a\|1\rangle) \otimes (c\|00\rangle + d\|11\rangle)$	$Z \otimes I$	ZX	$I \otimes I$
	$\|-\rangle$	$(m\|00\rangle - n\|11\rangle) \otimes (b\|0\rangle - a\|1\rangle) \otimes (c\|00\rangle - d\|11\rangle)$	$Z \otimes I$	ZX	$Z \otimes I$
$\|\phi_3\rangle \otimes \|\psi_3\rangle$	$\|+\rangle$	$(m\|11\rangle + n\|00\rangle) \otimes (b\|0\rangle - a\|1\rangle) \otimes (c\|00\rangle + d\|11\rangle)$	$X \otimes X$	ZX	$I \otimes I$
	$\|-\rangle$	$(m\|11\rangle + n\|00\rangle) \otimes (b\|0\rangle - a\|1\rangle) \otimes (c\|00\rangle - d\|11\rangle)$	$X \otimes X$	ZX	$Z \otimes I$
$\|\phi_4\rangle \otimes \|\psi_3\rangle$	$\|+\rangle$	$(m\|11\rangle - n\|00\rangle) \otimes (b\|0\rangle - a\|1\rangle) \otimes (c\|00\rangle + d\|11\rangle)$	$ZX \otimes X$	ZX	$I \otimes I$
	$\|-\rangle$	$(m\|11\rangle - n\|00\rangle) \otimes (b\|0\rangle - a\|1\rangle) \otimes (c\|00\rangle - d\|11\rangle)$	$ZX \otimes X$	ZX	$Z \otimes I$
$\|\phi_1\rangle \otimes \|\psi_4\rangle$	$\|+\rangle$	$(m\|00\rangle + n\|11\rangle) \otimes (b\|0\rangle - a\|1\rangle) \otimes (d\|00\rangle - c\|11\rangle)$	$I \otimes I$	ZX	$ZX \otimes X$
	$\|-\rangle$	$(m\|00\rangle + n\|11\rangle) \otimes (b\|0\rangle - a\|1\rangle) \otimes (d\|00\rangle + c\|11\rangle)$	$I \otimes I$	ZX	$X \otimes X$
$\|\phi_2\rangle \otimes \|\psi_4\rangle$	$\|+\rangle$	$(m\|00\rangle - n\|11\rangle) \otimes (b\|0\rangle - a\|1\rangle) \otimes (d\|00\rangle - c\|11\rangle)$	$Z \otimes I$	ZX	$ZX \otimes X$
	$\|-\rangle$	$(m\|00\rangle - n\|11\rangle) \otimes (b\|0\rangle - a\|1\rangle) \otimes (d\|00\rangle + c\|11\rangle)$	$Z \otimes I$	ZX	$X \otimes X$
$\|\phi_3\rangle \otimes \|\psi_4\rangle$	$\|+\rangle$	$(m\|11\rangle + n\|00\rangle) \otimes (b\|0\rangle - a\|1\rangle) \otimes (d\|00\rangle - c\|11\rangle)$	$X \otimes X$	ZX	$ZX \otimes X$
	$\|-\rangle$	$(m\|11\rangle + n\|00\rangle) \otimes (b\|0\rangle - a\|1\rangle) \otimes (d\|00\rangle + c\|11\rangle)$	$X \otimes X$	ZX	$X \otimes X$
$\|\phi_4\rangle \otimes \|\psi_4\rangle$	$\|+\rangle$	$(m\|11\rangle - n\|00\rangle) \otimes (b\|0\rangle - a\|1\rangle) \otimes (d\|00\rangle - c\|11\rangle)$	$ZX \otimes X$	ZX	$ZX \otimes X$
	$\|-\rangle$	$(m\|11\rangle - n\|00\rangle) \otimes (b\|0\rangle - a\|1\rangle) \otimes (d\|00\rangle + c\|11\rangle)$	$ZX \otimes X$	ZX	$X \otimes X$

Table A2. The environment required for the experiment and the corresponding version.

Environment	Edition
Programming Language	Python 3.7
Programming IDE	VS Code
Experimental Platform	IBM Quantum Lab
Quantum Simulator	ibmq-qasm-simulator

References

1. Lohachab, A.; Lohachab, A.; Jangra, A. A comprehensive survey of prominent cryptographic aspects for securing communication in post-quantum IoT networks. *Internet Things* **2020**, *9*, 100174. [CrossRef]
2. Chawla, D.; Mehra, P.S. A Survey on Quantum Computing for Internet of Things Security. *Procedia Comput. Sci.* **2023**, *218*, 2191–2200.
3. Hassan, W.H. Current research on Internet of Things (IoT) security: A survey. *Comput. Netw.* **2019**, *148*, 283–294.
4. Chen, G.; Wang, Y.; Jian, L.; Zhou, Y.; Liu, S. Quantum identity authentication based on the extension of quantum rotation. *EPJ Quantum Technol.* **2023**, *10*, 11. [CrossRef]

5. McLeod, J.; Majumdar, R.; Das, S. Challenges and future directions in the implementation of quantum authentication protocols. In Proceedings of the International Conference on Computational Science, London, UK, 21–23 June 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 164–170.
6. Rao, B.D.; Jayaraman, R. A novel quantum identity authentication protocol without entanglement and preserving pre-shared key information. *Quantum Inf. Process.* **2023**, *22*, 92. [CrossRef]
7. Gong, L.H.; Pei, J.J.; Zhang, T.F.; Zhou, N.R. Quantum convolutional neural network based on variational quantum circuits. *Opt. Commun.* **2024**, *550*, 129993. [CrossRef]
8. Zhou, N.R.; Zhang, T.F.; Xie, X.W.; Wu, J.Y. Hybrid quantum–classical generative adversarial networks for image generation via learning discrete distribution. *Signal Process. Image Commun.* **2023**, *110*, 116891. [CrossRef]
9. Huang, Y.; Xu, G.; Song, X. An improved efficient identity-based quantum signature scheme. *Quantum Inf. Process.* **2022**, *22*, 36. [CrossRef]
10. Tang, G.; Duong, D.H.; Joux, A.; Plantard, T.; Qiao, Y.; Susilo, W. Practical post-quantum signature schemes from isomorphism problems of trilinear forms. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, 30 May–3 June 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 582–612.
11. Bennett, C.H.; Brassard, G.; Crépeau, C.; Jozsa, R.; Peres, A.; Wootters, W.K. Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. *Phys. Rev. Lett.* **1993**, *70*, 1895. [CrossRef]
12. Chen, J.; Li, D.; Liu, M.; Yang, Y.; Zhou, Q. Quantum controlled teleportation of bell state using seven-qubit entangled state. *Int. J. Theor. Phys.* **2020**, *59*, 1402–1412. [CrossRef]
13. Kirdi, M.E.; Slaoui, A.; Hadfi, H.E.; Daoud, M. Efficient Quantum Controlled Teleportation of an Arbitrary Three-Qubit State Using Two GHZ Entangled States and One Bell Entangled State. *J. Russ. Laser Res.* **2023**, *44*, 121–134. [CrossRef]
14. Ma, Y. Convenient quantum controlled teleportation of two-qubit pure state with seven-qubit entangled state. *Laser Phys. Lett.* **2023**, *20*, 095204. [CrossRef]
15. Hou, K.; Bao, D.; Zhu, C.; Yang, Y. Controlled teleportation of an arbitrary two-qubit entanglement in noises environment. *Quantum Inf. Process.* **2019**, *18*, 104. [CrossRef]
16. Liu, X.F.; Li, D.F.; Zheng, Y.D.; Yang, X.L.; Zhou, J.; Tan, Y.Q.; Liu, M.Z. Experimental realization of quantum controlled teleportation of arbitrary two-qubit state via a five-qubit entangled state. *Chin. Phys. B* **2022**, *31*, 050301. [CrossRef]
17. Zhou, R.G.; Li, X.; Qian, C.; Ian, H. Quantum bidirectional teleportation 2↔2 or 2↔3 Qubit teleportation protocol via 6-Qubit entangled state. *Int. J. Theor. Phys.* **2020**, *59*, 166–172. [CrossRef]
18. Wang, M.; Li, H.S. Bidirectional quantum teleportation using a five-qubit cluster state as a quantum channel. *Quantum Inf. Process.* **2022**, *21*, 44. [CrossRef]
19. Yang, B. Bidirectional hierarchical quantum teleportation based on an eight-qubit entangled state. *Laser Phys. Lett.* **2023**, *20*, 095201. [CrossRef]
20. Zhou, R.G.; Xu, R.; Lan, H. Bidirectional quantum teleportation by using six-qubit cluster state. *IEEE Access* **2019**, *7*, 44269–44275. [CrossRef]
21. Zhou, R.G.; Zhang, Y.N. Bidirectional quantum controlled teleportation of three-qubit state by using GHZ states. *Int. J. Theor. Phys.* **2019**, *58*, 3594–3601. [CrossRef]
22. Tabatabaei, L.S.; Vakili, B. Bi-directional quantum teleportation of GHZ-like states. *arXiv* **2023**, arXiv:2302.11300.
23. Kazemikhah, P.; Tabalvandani, M.B.; Mafi, Y.; Aghababa, H. Asymmetric bidirectional controlled quantum teleportation using eight qubit cluster state. *Int. J. Theor. Phys.* **2022**, *61*, 17. [CrossRef]
24. Choudhury, B.S.; Samanta, S. Asymmetric bidirectional 3 $\Longleftrightarrow$ 2 qubit teleportation protocol between Alice and Bob via 9-qubit cluster state. *Int. J. Theor. Phys.* **2017**, *56*, 3285–3296. [CrossRef]
25. Kaur, S.; Gill, S. Asymmetric Controlled Quantum Teleportation Via Eight-Qubit Entangled State in a Noisy Environment. *Int. J. Theor. Phys.* **2023**, *62*, 31. [CrossRef]
26. Yang, Y.; Li, D.; Liu, M.; Chen, J. Quantum information splitting of arbitrary two-qubit state via a five-qubit cluster state and a bell-state. *Int. J. Theor. Phys.* **2020**, *59*, 187–199. [CrossRef]
27. Liu, X.; Li, D.; Zheng, Y.; Liu, M.; Yang, X.; Zhou, J.; Tan, Y.; Wang, R. Quantum Information Splitting of an Arbitrary Five-Qubit State Using Four-Qubit Entangled States. *Int. J. Theor. Phys.* **2022**, *61*, 220. [CrossRef]
28. Rahmawati, R.; Purwanto, A.; Subagyo, B.A.; Taufiqi, M.; Hatmoko, B.D. Symmetric and asymmetric cyclic quantum teleportation with different controller for each participant. *Int. J. Theor. Phys.* **2022**, *61*, 244. [CrossRef]
29. Li, Y.; Qiao, Y.; Sang, M.; Nie, Y. Controlled cyclic quantum teleportation of an arbitrary two-qubit entangled state by using a ten-qubit entangled state. *Int. J. Theor. Phys.* **2019**, *58*, 1541–1545. [CrossRef]
30. Zhou, R.G.; Ling, C. Asymmetric cyclic controlled quantum teleportation by using nine-qubit entangled state. *Int. J. Theor. Phys.* **2021**, *60*, 3435–3459. [CrossRef]
31. Kirdi, M.E.; Slaoui, A.; Hadfi, H.E.; Daoud, M. Improving the probabilistic quantum teleportation efficiency of arbitrary superposed coherent state using multipartite even and odd j-spin coherent states as resource. *Appl. Phys. B* **2023**, *129*, 94. [CrossRef]
32. Li, H.; Li, J.; Chen, X. Probabilistic quantum teleportation of shared quantum secret. *Chin. Phys. B* **2022**, *31*, 090303. [CrossRef]
33. Javed, S.; Pandey, R.K.; Yadav, P.S.; Prakash, R.; Prakash, H. Probabilistic Quantum Teleportation via 3-Qubit Non-maximally Entangled GHZ State by Repeated Generalized Measurements. *Int. J. Theor. Phys.* **2022**, *62*, 11. [CrossRef]

34. Bennett, C.H.; DiVincenzo, D.P.; Shor, P.W.; Smolin, J.A.; Terhal, B.M.; Wootters, W.K. Remote state preparation. *Phys. Rev. Lett.* **2001**, *87*, 077902. [CrossRef] [PubMed]
35. Jin, R.H.; Wei, W.S.; Zhou, P. Hierarchical controlled remote preparation of an arbitrary m-qudit state with four-qudit cluster states. *Quantum Inf. Process.* **2023**, *22*, 113. [CrossRef]
36. Mandal, M.K.; Choudhury, B.S.; Samanta, S. Cyclic controlled remote state preparation protocol initiated by a mentor for qubits. *Opt. Quantum Electron.* **2022**, *54*, 602. [CrossRef]
37. Chen, X.B.; Sun, Y.R.; Xu, G.; Jia, H.Y.; Qu, Z.; Yang, Y.X. Controlled bidirectional remote preparation of three-qubit state. *Quantum Inf. Process.* **2017**, *16*, 244. [CrossRef]
38. Sadeghi Zadeh, M.S.; Houshmand, M.; Aghababa, H. Bidirectional teleportation of a two-qubit state by using eight-qubit entangled state as a quantum channel. *Int. J. Theor. Phys.* **2017**, *56*, 2101–2112. [CrossRef]
39. Sang, M. Bidirectional quantum teleportation by using five-qubit cluster state. *Int. J. Theor. Phys.* **2016**, *55*, 1333–1335. [CrossRef]

sensors

MDPI

Article

IoTSim: Internet of Things-Oriented Binary Code Similarity Detection with Multiple Block Relations

Zhenhao Luo, Pengfei Wang *, Wei Xie, Xu Zhou and Baosheng Wang

College of Computer, National University of Defense Technology, Changsha 410073, China
* Correspondence: pfwang@nudt.edu.cn

Abstract: Binary code similarity detection (BCSD) plays a crucial role in various computer security applications, including vulnerability detection, malware detection, and software component analysis. With the development of the Internet of Things (IoT), there are many binaries from different instruction architecture sets, which require BCSD approaches robust against different architectures. In this study, we propose a novel IoT-oriented binary code similarity detection approach. Our approach leverages a customized transformer-based language model with disentangled attention to capture relative position information. To mitigate out-of-vocabulary (OOV) challenges in the language model, we introduce a base-token prediction pre-training task aimed at capturing basic semantics for unseen tokens. During function embedding generation, we integrate directed jumps, data dependency, and address adjacency to capture multiple block relations. We then assign different weights to different relations and use multi-layer Graph Convolutional Networks (GCN) to generate function embeddings. We implemented the prototype of IoTSim. Our experimental results show that our proposed block relation matrix improves IoTSim with large margins. With a pool size of 10^3, IoTSim achieves a recall@1 of 0.903 across architectures, outperforming the state-of-the-art approaches Trex, SAFE, and PalmTree.

Keywords: IoT security; binary code similarity detection; vulnerability detection

Citation: Luo, Z.; Wang, P.; Xie, W; Zhou, X; Wang, B. IoTSim: Internet of Things-Oriented Binary Code Similarity Detection with Multiple Block Relations. *Sensors* **2023**, *23*, 7789. https://doi.org/10.3390/s23187789

Academic Editor: Jian Li

Received: 7 August 2023
Revised: 3 September 2023
Accepted: 6 September 2023
Published: 11 September 2023

1. Introduction

Nowadays, the wide usage of Internet of Things (IoT) devices in various fields, including smart medical care and smart homes, has significantly improved people's lives. According to a report by IoT statistics [1], the number of connected IoT devices is expected to exceed 29 billion by 2030. However, the rapid growth in demand for IoT devices has led to the development of IoT firmware heavily relying on third-party components (TPCs), often without necessary security audits. Although this way improves development efficiency and reduces costs, it also exposes the firmware to vulnerabilities and weaknesses, making them attractive targets for attackers. Numerous security issues [2–7] indicate the fragility of the current IoT ecosystem, raising public concern about IoT security risks.

However, detecting these vulnerabilities in IoT firmware is challenging due to the following reasons. Firstly, many IoT firmware images only provide binary files, making source code unavailable for security analysis. Secondly, IoT firmware originates from different instruction set architectures (ISAs), necessitating extensive reverse engineering expertise and specialized knowledge for security analysis. Thirdly, analyzing more than 29 billion IoT devices and discovering their vulnerabilities significantly burdens researchers. Therefore, there is an urgent need for an accurate and automated technique to identify these vulnerable TPCs and vulnerable functions. As a result, binary code similarity detection (BCSD) has become an active research focus for detecting vulnerabilities hidden in IoT devices.

Binary code similarity detection is a fundamental technique in computer security that can detect similarities between two binary code snippets. It is widely used for various

applications, including vulnerability detection [8–22], malware analysis [23–28], and binary patch analysis [29–31]. Figure 1 shows an example of BCSD usage in firmware security analysis. Given an IoT firmware without symbol tables, the BCSD approaches extract its functions and match them with functions in the vulnerability database based on function similarity to detect vulnerable functions in the firmware. For example, function `sub_5686c` is matched with function `BN_hex2bn`, which is associated with the vulnerability in the Common Vulnerabilities and Exposures (CVE) database. Furthermore, according to the BCSD results, the function name, the symbol table, the project name, and even the source code can be restored. This can locate vulnerabilities in massive IoT firmware and provide reverse analysts with critical information, e.g., source code and symbol tables.

Prior to the utilization of machine learning (ML) in Binary Code Similarity Detection (BCSD) tasks, traditional BCSD approaches [20,21,32,33] heavily rely on specific features, including control flow graphs (CFGs), the count of basic blocks, and string constants. However, determining the weights of these syntactic features requires extensive experience and expert knowledge, and they may vary with different compilers and optimization options. Moreover, traditional graph-based methods such as graph isomorphism matching are excessively time-consuming for analyzing large-scale firmware, leading to relatively lower accuracy and scalability.

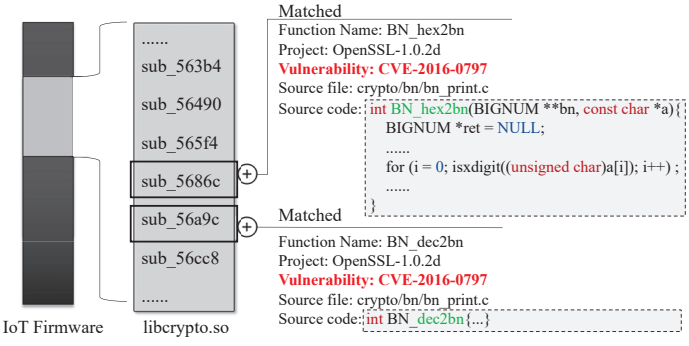


Figure 1. An example of using BCSD in IoT firmware analysis.

In recent years, researchers have increasingly adopted learning-based approaches to tackle BCSD tasks, and the current state-of-the-art BCSD approaches [11,13,14,34,35] are predominantly based on machine learning (ML) techniques. These approaches typically involve the disassembly of binary code into either assembly language or intermediate representation (IR). Subsequently, trained neural networks are utilized to extract the semantic information from functions and embed them into high-dimensional representations. Finally, function matching is performed based on the similarity of these embeddings. Gemini [36] manually selects statistical features of basic blocks and employs graph neural networks (GNNs) to generate function embeddings for function matching. More recently, the field of natural language processing (NLP) has achieved significant advancements in semantic extraction. Consequently, NLP techniques have been introduced into BCSD tasks by methods such as jTrans [11], PalmTree [34], and SAFE [35]. These techniques aim to automatically extract semantics and generate function embeddings for calculating function similarity. Despite the progress made by learning-based models, they still have limitations when it comes to new application scenarios of the BCSD for IoT firmware.

P1: IoT firmware originates from different architectures such as `x86`, `arm`, and `mips`. This leads to the inclusion of instructions from different architectures, and poses significant risks of encountering out-of-vocabulary (OOV) challenges. OOV problems are widely recognized in the NLP field. When a word has not been encountered during training, it is referred to as an OOV word, and the word embedding model becomes incapable of generating semantic representations for such words. The OOV issue in BCSD for IoT firmware is further exacerbated by the existence of various instruction sets, address offsets, and registers.

Figure 2 illustrates the assembly code compiled from the same source code but targeting different architectures (i.e., `x86-64` and `arm`). Notably, the assembly code generated for these two architectures exhibits stark dissimilarities. As a consequence, considerable differences arise at the lexical and syntactic levels, resulting in severe OOV challenges.

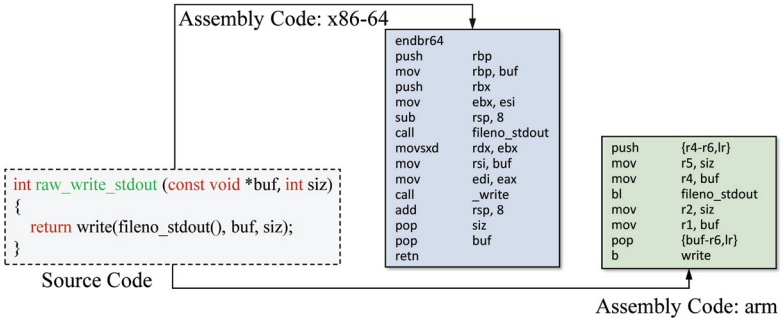


Figure 2. Assembly code on the different architectures. Both assembly sequences are from the same source code in the OpenSSL project. However, their opcodes, operands, and calling conventions are very different due to the different architectures.

For OOV words, existing approaches [11,34,35,37] use normalization based on predefined rules to deal with string literals, immediate numbers, and address offsets. However, when these special words are replaced with dummy tokens, the essential semantics may be compromised. Moreover, when confronted with unknown expressions found in IoT firmware, these approaches still encounter OOV challenges. For example, despite being trained on multiple ISAs, SAFE [35] still suffers from OOV issues and does not exhibit satisfactory performance in cross-architecture BCSD tasks.

P2: Existing approaches extract superficial features from Control Flow Graphs (CFGs), such as adjacency matrices, which overlooks critical semantic information. A CFG represents all possible paths during execution, which consists of basic blocks. Multiple relations exist between connected basic blocks: (1) edges between basic blocks are directed, (2) address adjacent basic blocks often have a stronger mutual influence, and (3) data flow dependencies may exist between basic blocks. Adjacency matrices alone cannot capture these in-depth multiple relations. Neglecting these semantic and structural nuances makes it challenging to build an accurate model for BCSD tasks for IoT firmware. Previous approaches [19,36] primarily concentrate on connections between basic blocks, disregarding various associations like data dependencies, which results in low performance.

To solve the aforementioned problems, in this paper, we present `IoTSim`, a novel deeply customized cross-architecture approach for binary code similarity detection in IoT firmware, which supports vulnerability discovery and firmware component analysis. We use an NLP-based model DeBERTa [38] to capture basic block semantics, and use GNNs to capture the control and data flow information, to generate binary function semantic embeddings. To resolve the challenges in problem **P1**, we lift assembly code into an intermediate representation, namely microcode, to mitigate differences in instruction sets, registers, and calling conventions. During the training phase, we do not use normalization to deal with string literals and immediate numbers, in order to preserve important semantics. For OOV issues in testing, we tokenize these OOV words into their base-tokens and generate semantic embeddings. To this end, we propose a newly designed pre-training task, namely Base-Token Prediction (BTP), so that base-tokens include the basic semantics of their types. Furthermore, in binary code, the attention weight of an instruction pair depends on not only their contextual instructions but also their relative positions. Adjacent instructions tend to have stronger dependencies. Therefore, we utilize relative position embeddings based on disentangled attentions to capture content-to-content (`c2c`), content-to-position (`c2p`), and position-to-content (`p2c`) attentions for higher precision.

To address problem **P2**, we propose using directed adjacency, data dependency, and address matrices to represent relations between basic blocks. For multiple relations between connected basic blocks, first, we determine the direction relations based on the predecessors and successors of the basic block. Then, according to the address and connection of the basic blocks, the larger address distances between blocks generally indicate weaker mutual influence. Furthermore, variables in binary code are required to be defined before being used. Based on the def-use chains at the basic block level, we can determine the data dependencies between basic blocks. Thus, we generate an abundant data-based control flow graph (DCFG) to represent the CFG deeply. Section 5.3 shows that DCFGs can improve recall@1 by $15.5 \pm 6.1\%$ compared to CFGs solely in the function level BCSD tasks with 10^3 candidates.

In summary, we have made the following contributions:

- We propose a novel deeply cross-architecture approach using NLP techniques for IoT-oriented binary code similarity detection tasks. To resolve problem **P1**, we lift assembly code into microcode and propose a newly designed pre-training task to mitigate OOV issues;
- To resolve problem **P2**, we consider multiple relations between basic blocks to generate DCFGs to capture rich contextual information between basic blocks. We then use a GNN model to integrate basic block embeddings based on DCFGs for generating function embeddings;
- We implement `IoTSim` which can be used for vulnerability detection and firmware component analysis in the real world. We evaluate `IoTSim` with extensive experiments. The experiments show that `IoTSim` outperforms the state-of-the-art approaches such as Trex, SAFE, GMN, and PalmTree.

The remaining sections of the paper are structured as follows: Section 2 provides an overview of the relevant literature. Section 3 clarifies key points of this paper. In Section 4, we present a detailed description of the design of `IoTSim`. Section 5 conducts extensive experiments to evaluate the performance of `IoTSim`. Lastly, Section 6 concludes the paper and provides final remarks.

2. Related Work

In this section, we provide a concise overview of the relevant literature concerning binary code similarity detection. We discuss both mono-architecture approaches and cross-architecture approaches.

Mono-architecture Approaches: Extensive progress has been made in research on detecting binary code similarity for mono-architecture binaries. For example, various approaches [20,23,28,39,40] utilize syntax, structural, and statistical features to match similar binary functions. Tracelet [41] decomposes binary functions into continuous traces and measures the similarity between two traces by constraint solving and data dependencies. BLEX [42] executes functions under a randomized environment and compares their similarity based on the corresponding I/O values collected. BinSim [23] proposes a hybrid fine-grained approach using system call sliced segment to identify binary code similarities with symbolic execution. Similarly, CoP [43] employs symbolic execution and a theorem prover to compare binary code similarity by matching the longest common sub-sequence with basic blocks as elements. Nonetheless, due to their computational complexity, these approaches may not be practical when dealing with extensive function repositories. Drawing inspiration from Natural Language Processing (NLP) techniques, many researchers [11,14,22,34,44,45] introduce language models to extract the semantics of binary code for BCSD tasks. Ding et al. [14] propose Asm2Vec, which represents binary functions as high-dimensional embeddings and utilizes the Distributed Memory Model of Paragraph Vectors (PV-DM) model [46] to extract semantics from binary functions for function embedding generation. Li et al. [34] employ PalmTree, a transformer-based NLP model using BERT [47], to measure the similarity of binary code in assembly language. Additionally, jTrans [11] presents a transformer-based approach that incorporates control

flow information through jump-aware representation. These approaches involve representing binary functions as high-dimensional embeddings, facilitating the search for similar candidate functions within extensive function repositories. However, in the IoT scenarios, firmware images and executable files from different architectures require BCSD approaches to support finding similar functions across various architectures.

Cross-architecture Approaches: With the urgent need for cross-architectural BCSD approaches, recent research has focused on cross-architecture BCSD tasks. Traditional methods typically involve selecting architecture-robust features, such as statistical, syntactic, and structural features, to compute the similarity of binary code. These include BinDiff [32], DiscovRE [48], Esh [15], GitZ [49], Genius [19], and Gemini [36]. Esh [15] and GitZ [49] decompose binary procedures into comparable fragments using data flow analysis and use a statistical framework to detect similar binary fragments. Genius [19] and Gemini [36] adopt machine learning and consider statistical features as attributes of CFGs to graph embeddings for BCSD tasks. Both of them rely on hand-crafted features, which necessitate rich experience and domain knowledge to match similar functions. Trex [13] proposes a transformer-based model using micro-traces to capture execution semantics of functions for cross-architecture BCSD tasks. VulHawk [50] integrates basic block features and CFGs to detect vulnerabilities across architectures. SAFE [35] trains a Word2Vec model [51] using binaries from various architectures, including `x86` and `arm`, to detect binary code across different architectures. However, in our experiments, we observe that despite being trained using multiple architecture binaries, SAFE still suffers from a large number of OOV issues in the experiments, which severely affects its performance.

Furthermore, there are multiple relations (e.g., directed jumps, data dependency, and address adjacency) in binary functions, which contain important semantic information for BCSD tasks. The existing approaches use single basic block relations and do not deeply consider the combination of multiple relations, which may miss critical semantic information to distinguish dissimilar functions with minor differences. Facing high pool size scenarios, it is difficult for them to achieve good performance in numerous candidate functions.

3. Problem Definition

In this section, we aim to clarify key points of this paper to enhance the clarity of the presentation.

Detection Granularity. This paper focuses on measuring the similarity of two binaries at the function level. In the analysis of IoT firmware, giving the similarity and symbol tables of functions can help researchers understand binary code and greatly reduce the manual workload.

Similarity Explanation. In the existing BCSD literature [14,27,52], there are four types of function similarity: (1) literal identity, (2) syntactic equivalence, (3) functional equivalence, and (4) the same or logic similar source code. Due to the usage of different compilers and optimization options in IoT firmware, there may still be literal and syntactic differences among binary functions originating from the same source code. The same functionality has various implementations (e.g., bubble sort and quick sort), but they cannot share the symbol tables. Consequently, types (1), (2) and (3) are not suitable for vulnerability discovery and firmware component analysis with symbol table supplementation. Our focus lies on type (4), which involves binary functions that may exhibit syntactic differences but share similar functional logic in their source code.

Binary Code Similarity Detection. BCSD approaches are employed to calculate the similarity between two binary functions. In IoT firmware analysis, binary functions are compiled using diverse compilers (e.g., `GCC` and `Clang`) with various optimization options (e.g., `O0`, `O1`, `O2`, `O3`, and `Os`) on multiple architectures (e.g., `arm`, `mips`, and `x86`). As a result, even when two functions originate from the same source code, they may have different instructions and structures. Therefore, an effective IoT-oriented BCSD approach should be robust across architectures, compilers, and optimization options. Furthermore, it also supports working on large pools of candidate binary functions. In practice, given

a query function, BCSD approaches need to calculate the similarity between the query function and a large pool of candidate functions, selecting the most similar one [11]. This requires accurately identifying the most similar functions from the candidate pool while distinguishing irrelevant ones.

4. Design

To address the problems mentioned in Section 1, we propose a BCSD approach named `IoTSim`, for IoT-oriented BCSD tasks. Figure 3 shows the overview of `IoTSim`, including three modules to implement its functionality.

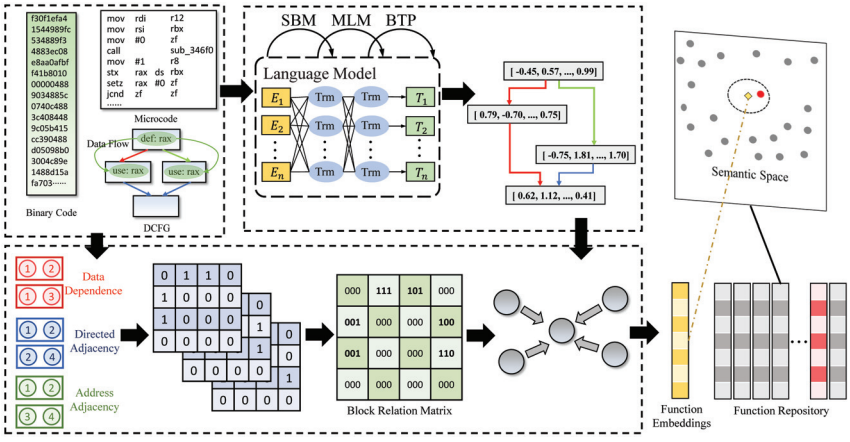


Figure 3. The overview of IoTSim, which consists of a preprocessor, a block semantic model, and a function embedding model.

Preprocessor. Firstly, IoT binaries are disassembled and their binary code is transformed into microcode. By analyzing the def-use chains, we obtain data dependencies based on defined variables and used variables for each basic block. Subsequently, CFGs and data dependencies are integrated to generate data-based control flow graphs (DCFGs). Based on the various relations between basic blocks in DCFGs, we create the address matrix, the directed matrix, and the data dependencies.

Block Semantic Model. This module utilizes an NLP model to produce semantic embeddings for basic blocks using microcode sequences. We employ a language model based on DeBERTa [38] to construct basic block embeddings. To optimize model parameters, we utilize two pre-training tasks: (1) the masked language model (MLM) task helps the model learn semantic relations between microcode; and (2) the base-token prediction (BTP) task assists the model in learning and complementing the semantics of base-tokens.

Function Embedding Model. This module combines basic block embeddings and graph features to generate function embeddings. We establish a block relation matrix that incorporates multiple relations such as directed adjacency, data dependencies, and address adjacency. Leveraging the block relation matrix and basic block embeddings, we utilize graph neural networks (GNNs) to capture control-flow and data-flow relations for generating function embeddings. During training, we use the normalized temperature-scaled cross-entropy loss (NT-Xent) [53] to optimize the model parameters, making the similar functions' embeddings closer in the semantic space.

4.1. Preprocessor

The preprocessor generates the function features used by the Block Semantic Model and Function Embedding Model, including instructions, control-flow, and data-flow features.

We initially disassemble binary files and convert their binary code into microcode to address the cross-architecture differences. The microcode is an architecture-agnostic inter-

mediate representation from IDA Pro [54]. Figure 4 illustrates the conversion of assembly code from diverse architectures into microcode. Notably, there are significant dissimilarities in assembly snippets when comparing the same source code across different architectures (e.g., `x86` and `arm`). Their registers (e.g., `eax` and `W0`), opcodes (e.g., `jz` and `B.EQ`), and calling conventions look completely different. After converting to microcode, they have the same opcodes and calling conventions. Consequently, this conversion mitigates the differences introduced by instruction sets and calling conventions. Within the microcode, intricate instruction nests can be observed, as demonstrated in Ln. 7 of the microcode (x86) in Figure 4, which are compounded by multiple semantics and are susceptible to OOV issues. To address such OOV problems, we analyze sequences of microcode instructions and split these nests into individual instructions. We consider an instruction with multiple operands nested within it to be an instruction nest. For example, the instruction of Ln.7 in microcode (x86) is an instruction nest, which nests three opcodes `call`, `xdu`, and `add`. After analysis, we split it into three sub-instructions of Ln.11-13 in detailed microcode. These instructions are semantically equivalent to the original instruction nest and can reduce the OOV issues caused by instruction nesting.

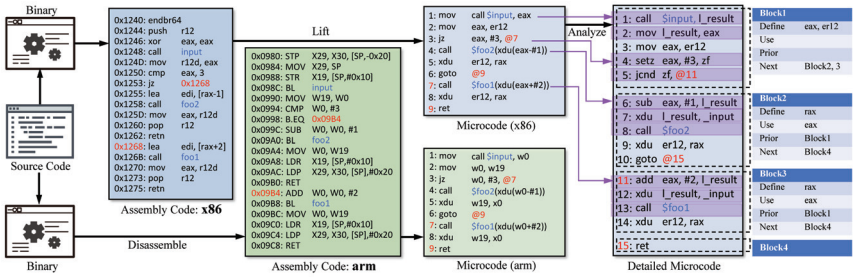


Figure 4. An example control flow of a binary function. The left is the linear layout assembly code with jump addresses, and the right is the corresponding control-flow graph.

To capture the data dependency among basic blocks, we use def-use chains to capture defined variables and used variables for each basic block to generate Data-based Control Flow Graphs (DCFGs). In binary code, variables should be defined before being used. The def-use chains enable us to monitor the usage and definition of variables. Here, we focus on defining and using variables at the basic block level. For example, in optimized microcode, `Block3` uses `eax`, which is defined by `Block1`. This indicates a data dependency between `Block1` and `Block3`. We integrate CFGs based on prior and next blocks of basic blocks, and data dependencies based on the definition and use of variables between basic blocks, to construct DCFGs for binary functions.

4.2. Block Semantic Model

In this section, we employ a customized transformer-based language model with disentangled attention to learn microcode semantics and generate semantic embeddings at the basic block level. Compared with instructions, basic blocks have richer semantics; and compared with functions, basic blocks do not have complex multi-branch structures, which facilitates semantic extraction. We consider instructions as words and basic blocks as sentences. Considering the important influence of instruction content and position on block semantics, we adopt disentangled attention to capturing relative position semantics. Subsequently, we utilize two self-supervised pre-training tasks, namely Masked Language Model (MLM) and Base-Token Prediction (BTP), to train our model.

4.2.1. Language Model

The transformer-based architectures have shown encouraging results in NLP tasks. Our model is built upon the DeBERTa [38] model, one of the state-of-the-art NLP models. Figure 5 illustrates the architecture of our language model, comprising stacked transformer

blocks for generating embeddings. For each input sequence, we utilize a tokenizer to convert microcode into token objects. These tokens are then processed through transformer stacks to produce hidden states, with the final hidden states representing the semantics of the microcode.

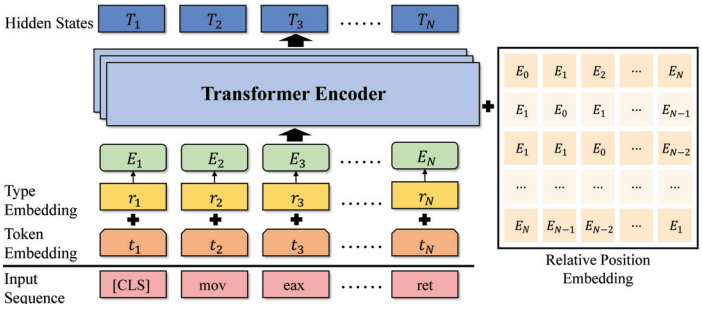


Figure 5. The architecture of the language model.

Tokenizer. The problem of out-of-vocabulary (OOV) is widely recognized in NLP. In the context of BCSD for IoT firmware, the OOV issue is further amplified due to the presence of various instruction sets. Even when converting binary code into microcode, the OOV issue can still arise from string literals, address offsets, registers, and function names. To mitigate the OOV issues, Inter-BIN [55] adopts a character-level tokenizer, which has a smaller vocabulary and rarely suffers from OOV issues. However, a single character is less meaningful and may lead to incorrect semantics. Also, it requires more computing resources for each character computation. And, other approaches [11,35,37] replace special words (e.g., string literals and constant values) with dummy tokens, which may lose important semantics.

We perform tokenization of the microcode at the opcode/operand level. To handle low-frequency operands, we replace them with their base-tokens instead of using dummy tokens. The microcode provides a basic type for each operand. For example, the addresses `0x400C8D` and `0x400A30` are both specific address offsets that may result in out-of-vocabulary (OOV) issues, while microcode categorizes them into the address type (`mop_a`), which represents their common basic semantics. This is particularly useful for addressing OOV issues, as it allows us to map OOV words to their corresponding base-tokens based on their basic types. To construct our vocabulary, we initially iterate through all the microcode and filter out infrequently occurring tokens with frequencies lower than 100. Subsequently, we include all basic types in our vocabulary as base-tokens. During the tokenization process, any OOV words are mapped to their corresponding base-tokens according to their basic types.

Relative Position Embedding. The positions of instructions play a crucial role in determining basic block semantics. Recent studies [38,47,56,57] have shown that relative positions are more effective than absolute positions for NLP tasks. Inspired by DeBERTa [38], in each transformer encoder layer, we utilize disentangled attentions in each transformer encoder layer to generate relative position embeddings that capture content-to-content (`c2c`), content-to-position (`c2p`), and position-to-content (`p2c`) attentions. The cross attention score between tokens i and j is calculated as follows:

$$\begin{aligned} A_{i,j} &= \{\boldsymbol{H}_i, \boldsymbol{P}_{i|j}\} \times \{\boldsymbol{H}_j, \boldsymbol{P}_{j|i}\}^\intercal \\ &= \underbrace{\boldsymbol{H}_i \boldsymbol{H}_j^\intercal}_{c2c} + \underbrace{\boldsymbol{H}_i \boldsymbol{P}_{j|i}^\intercal}_{c2p} + \underbrace{\boldsymbol{P}_{i|j} \boldsymbol{H}_j^\intercal}_{p2c} + \underbrace{\boldsymbol{P}_{i|j} \boldsymbol{P}_{j|i}^\intercal}_{p2p} \end{aligned} \tag{1}$$

where H_i and $P_{i|j}$ represent its content and relative position with the token j, respectively. In microcode, the position-to-position (p2p) term does not provide much additional information, we do not consider it in implementation.

The formulation for the standard self-attention calculation [58] is as follows:

$$\begin{aligned} Q = HW_q, \quad K = HW_k, \quad V = HW_v \\ H_o = \text{softmax}(\frac{A}{\sqrt{d}})V \end{aligned} \tag{2}$$

where $H \in R^{N\times d}$ is the input hidden vectors, $H_o \in R^{N\times d}$ represents the output of self-attentions, Q, K, and V are three matrices, and W_q, W_k, and W_v are their projection matrices. We put the content input H and the relative position P into Equation (2), where W^c and W^r represent content position matrix and relative position projection matrix, respectively.

$$\begin{aligned} &Q_c = HW_q^c, \quad K_c = HW_k^c, \quad V = HW_v^c, \\ &Q_r = PW_q^r, \quad K_r = PW_k^r \end{aligned} \tag{3}$$

According to Equations (1) and (3), the elements $\tilde{A}_{i,j}$ of attention matrix $\tilde{A}$ are calculated as follows:

$$\tilde{A}_{i,j} = Q_i^c K_j^{c\intercal} + Q_i^c {K_{\delta(i,j)}^r}^\intercal + K_j^c {Q_{\delta(j,i)}^r}^\intercal \tag{4}$$

where $\delta(i,j)$ represent the maximum relative distance from token i to token j, and the output H_o of disentangled attentions can be formulated as:

$$H_o = \text{softmax}(\frac{\tilde{A}}{\sqrt{3d}})V_c \tag{5}$$

We feed the output H_o into a fully connected feed-forward network to obtain the transformer encoder layer output.

4.2.2. Pre-Training Tasks

For large-scale training our model, we incorporate the Masked Language Model (MLM) task, similar to other NLP model training approaches, but with domain-specific adaptations. Specifically, we propose a novel pre-training task, Base-Token Prediction (BTP), to improve the semantic understanding of base-tokens in `IoTSim`.

Masked Language Model. This model is employed to perform fill-in-the-blank tasks, enabling the model to utilize the tokens surrounding a mask token for predicting the masked token [47]. Through the MLM task, the model learns the connections between tokens.

Given a token sequence $X = \{x_i | i \in (0,n)\}$, we randomly select 15% of the token sequence to be replaced. Among the selected tokens, 80% of them are substituted by the `[MASK]` token, 10% are replaced with other tokens, and 10% remain unchanged. This replacement process yields a masked sequence denoted as $\tilde{X}$. Subsequently, we input $\tilde{X}$ into the block semantic model and feed the output into an MLM head to reconstruct X by predicting the masked tokens $\tilde{x}$ conditioned on $\tilde{X}$. The loss function is as follows:

$$\mathcal{L}_{MLM}(\theta_1, \theta_2) = -\log p_{\{\theta_1,\theta_2\}}(X|\tilde{X}) \tag{6}$$

where θ_1 and θ_2 are the parameters of the block semantic model and the MLM head, respectively.

Base-Token Prediction. In our model, the tokenizer is utilized to replace OOV operands with their base-tokens. To establish semantic connections between tokens and their respective base-tokens, we propose a task called Base-Token Prediction (BTP). This task aims to facilitate the learning of common semantics among basic types of tokens by their corresponding base-tokens.

Given a sequence of tokens denoted as $\boldsymbol{X} = \{x_i | i \in (0, n)\}$, we randomly select 15% of the token sequence to substitute them with their base-tokens. As a result, we obtain a replaced sequence denoted as $\tilde{\boldsymbol{X}}_{\boldsymbol{b}}$. Figure 6 illustrates an example of the BTP task, where tokens such as `eax` (register), `#2` (immediate number), and `$foo2` (function name) are selected. These tokens are replaced with their respective base-tokens; for instance, `#2` is substituted with `mop_n`, which represents immediate number constants.

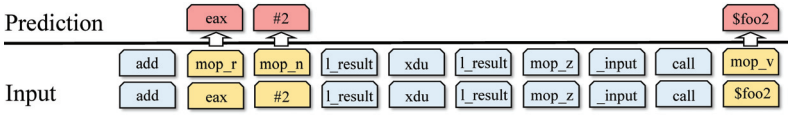


Figure 6. Base-token prediction

We feed $\tilde{\boldsymbol{X}}_{\boldsymbol{b}}$ into the block semantic model and feed the output into a BTP head to reconstruct $\boldsymbol{X}$ by predicting the replaced tokens $\tilde{x}$ conditioned on $\tilde{\boldsymbol{X}}$. The loss function is as follows:

$$\mathcal{L}_{BTP}(\theta_1, \theta_3) = -\log p_{\{\theta_1, \theta_3\}}(\boldsymbol{X} | \tilde{\boldsymbol{X}}) \tag{7}$$

where θ_1 and θ_3 are the parameters of the block semantic model and the BTP head, respectively.

The loss function of the block semantic model is the combination of loss functions:

$$\mathcal{L} = \mathcal{L}_{MLM} + \mathcal{L}_{BTP} \tag{8}$$

4.3. Function Embedding Model

The objective of the function embedding model is to generate semantic function embeddings by integrating basic block embeddings and graph features. Binary functions encompass both control-flow information and data-flow information. Control-flow information describes the potential execution paths of a function, while data-flow information describes the data dependencies between basic blocks. Hence, these semantic and structural features are crucial in the generation of function embeddings.

We first generate directed matrices, data dependency matrices, and address matrices according to DCFGs. Then, we combine these matrices into a block relation matrix. Finally, we integrate the block relation matrix and block embeddings to generate function embeddings using Graph Convolutional Networks (GCNs) [59].

4.3.1. Block Relation Matrix

In binary functions, basic blocks have multiple relations. First, connected basic blocks are directed. Second, basic blocks with adjacent addresses indicate relatively more dependencies. Third, data flow dependencies exist between connected basic blocks or not. These relations result in different mutual influences between basic blocks. Capturing these multiple relations is critical to building an accurate function embedding model for BCSD tasks.

Figure 7 shows an example of constructing a block relation matrix. In the Figure, the DCFG is composed of 4 basic blocks, and there are different relations between these basic blocks. For example, ① `Block1` and `Block2` are adjacent and have data dependencies, ② `Block2` and `Block4` have a directed edge, and ③ `Block2` and `Block3` have no relations. We first base on directed edges, data dependencies, and address adjacency of the DCFG to generate its directed matrix, data dependency matrix, and address matrix, respectively. Then, we combine the above three matrices to construct the block relation matrix. The combination formula is as follows:

$$m_B = (m_A << 2) + (m_C << 1) + m_D$$

where m_B, m_A, m_C, and m_D represent the corresponding elements in the block relation matrix, directed matrix, address matrix, and data dependency matrix, respectively, and << denotes the left shift operation. The constructed block relation matrix distinguishes

different relations between basic blocks. For example, the relation from `Block1` to `Block2` is represented as `111`, the relation from `Block2` to `Block4` is represented as `100`, and the relation from `Block2` to `Block3` is represented as `000`, which indicates no relations.

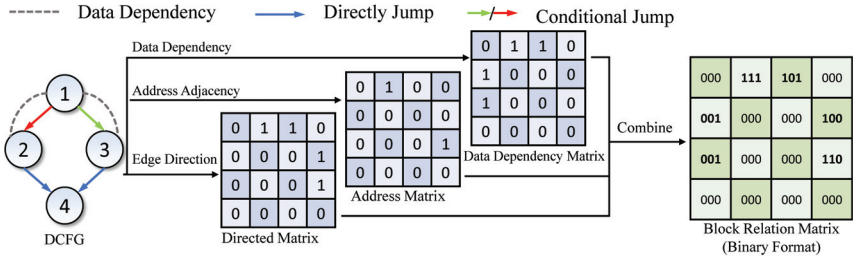


Figure 7. Block relation matrix.

4.3.2. Function Embedding Generation

A binary function consists of basic blocks that exhibit mutual influences. In order to generate function embeddings, we combine the basic block embeddings produced by the Block Semantic Model with the block relation matrix. To deeply combine graph information and basic block embeddings, we utilize Graph Convolutional Networks (GCNs) [59] for embedding generation. Within this framework, we view binary functions as attributed graphs, where the basic blocks serve as nodes and their corresponding embeddings serve as node attributes. The block relation matrix indicates different relations between nodes in the graph. We feed basic block embeddings and the block relation matrix into a multi-layer GCN. $\boldsymbol{X}^{(\ell)}$ represents the node embeddings of the ℓ-th layer, and $\boldsymbol{B}$ represents the block relation matrix. The ℓ-th layer GCN's output $X^{(\ell+1)}$ is computed as follows:

$$\begin{aligned} \boldsymbol{A} &= \mathcal{F}(\boldsymbol{B}) \\ \tilde{\boldsymbol{A}} &= \boldsymbol{A} + \boldsymbol{I_N} \\ \tilde{D}_{ii} &= \sum \tilde{A}_{ij} \\ \boldsymbol{X}^{(\ell+1)} &= \sigma(\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{X}^{(\ell)}\boldsymbol{W}^{(\ell)}) \end{aligned} \tag{9}$$

where $\boldsymbol{A}$ is a weighted adjacency matrix which encodes the relations of $\boldsymbol{B}$. Here, we use a learnable function $\mathcal{F}(\boldsymbol{B})$ to encode different relation types into numeric weights. $\tilde{\boldsymbol{A}}$ represents a weighted adjacency matrix that includes self-connections. $\boldsymbol{I_N}$ is an identity matrix. $\tilde{D}_{ii}$ denotes the degree matrix of each node, while $W^{(\ell)}$ represents a trainable weight matrix specific to each layer. The activation function σ is the rectified linear unit $ReLU(\cdot)$. To comprehensively learn the semantic and structural aspects of CFGs, we adopt a 16-layer GCN to propagate block semantics using the block relation matrix. Finally, by applying mean pooling on the outputs of the final GCN layer, we generate function embeddings.

4.3.3. Model Training

In the BCSD tasks, the main objective of the Function Embedding Model is to map similar functions to nearby regions in the embedded space. To measure the similarity between functions, we utilize the cosine similarity score of function embeddings. By leveraging Debugging with Attributed Record Formats (DWARF) information, it becomes feasible to automatically generate ground truth for function similarity based on function names and source files. Therefore, supervised training is employed to optimize the Function Embedding Model.

Function Ground truth. We automatically construct a function ground truth based on function names and the source files. Given a function f, we pick the functions whose names are the same as f from the same project, and label them as a similar function group $\boldsymbol{F}_{sim}$. We randomly sample the functions whose names differ from f, and label them as a similar

function group $\boldsymbol{F}_{dissim}$ against f. These functions of $\boldsymbol{F}_{sim}$ and $\boldsymbol{F}_{dissim}$ can be compiled by different compilers (e.g., `GCC` and `Clang`) with different optimization levels (`O0`, `O1`, `O2`, `O3`, and `Os`) on any architectures (e.g, `x86`, `arm`, and `mips`), which increases the diversity of the dataset.

Training Objective. During the training process, we aim to maximize the cosine similarity scores between embeddings of similar function pairs while minimizing those between embeddings of dissimilar function pairs. The number of dissimilar function pairs in the real world is much greater than that of similar function pairs. Given a binary function f_i, we pick one of the corresponding similar functions f_j from the dataset $\mathcal{D}_F$ based on its function name and source code. The function f_j can come from different architectures, compilers, and optimization levels against function f_i. The Function Embedding Model encodes both f_i and f_j to generate the function embeddings e_i and e_j. We randomly sample $N-1$ functions from $\mathcal{D}_F$ to construct negative samples against the function f_i. We use the normalized temperature-scaled cross-entropy loss (NT-Xent) [53] as the training loss:

$$\mathcal{L}_{Function} = -\log \frac{\exp(\cos(e_i, e_j)/\tau)}{\sum_{k=1}^{N} \mathbb{1}_{[k \neq i]} \exp(\cos(e_i, e_k)/\tau)} \tag{10}$$

where $\cos(\cdot)$ denotes the cosine similarity score function, $\mathbb{1}$ is the indicator, and τ is a hyper-parameter which controls the temperature.

5. Evaluation

In this section, we evaluate `IoTSim` and answer the following research questions:

- RQ.1: can `IoTSim` effectively identify similar function pairs when given functions from different compilers, architectures, and optimization levels?
- RQ.2: how much does DCFG contribute to the performance of `IoTSim`?
- RQ.3: what are the applications of `IoTSim` in practice?

5.1. Implementation and Setup

We utilized Python v3.8.5 and PyTorch [60] to implement the `IoTSim` framework. We employed the DeBERTa model based on Transformers [61] and Graph Convolutional Networks (GCNs) relying on PyTorch Geometric [62]. By default, our DeBERTa model consists of six layers, and the output embeddings are set to a dimension of 256. In the Function Embedding Model, we employ 16-layer GCNs for generating embeddings, and the NT-Xent loss function adopts a hyper-parameter τ with a value of 0.1. The model training and evaluation experiments were conducted on a desktop computer operating Windows 10, equipped with an Intel Core i9-10920X CPU, 128 GB RAM, and one NVIDIA RTX 3090 GPU. The model training process lasted for one week, during which we retained the best-performing checkpoints for evaluation.

5.1.1. Baselines

To provide a comprehensive comparison, we select the following state-of-the-art approaches as baselines for evaluation. The chosen baseline approaches are as follows:

- Graph Matching Networks (GMN). Marcelli et al. [63] show that a GMN based on CFGs has natural advantages in cross-architecture scenarios;
- PalmTree [34], one of the state-of-the-art BCSD methods, employs pre-trained models using the BERT model to generate semantic embeddings for binary code (https://github.com/palmtreemodel/PalmTree, accessed on 10 March 2023);
- SAFE [35] uses a word2vec model [51] and a recurrent neural network to generate function embeddings (https://github.com/facebookresearch/SAFEtorch, accessed on 10 March 2023);
- Trex [13], the state-of-the-art BCSD approach, uses transfer-learning-based models that utilize micro-traces to generate function embeddings for comparing similar functions (https://github.com/CUMLSec/trex, accessed on 10 March 2023).

These selected baselines are representative BCSD approaches involving NLP techniques, CFG, and micro-traces. By comparing IoTSim with these state-of-the-art approaches, we aim to conduct a comprehensive evaluation of its performance improvements.

5.1.2. Benchmarks

To evaluate IoTSim in depth and detail, we used the following two datasets:

Dataset-1 is a function dataset, which is used to evaluate the performance of IoTSim at the function level. We construct Dataset-1 using including seven projects, i.e., Linux-source, CoreUtils, OpenSSL, DiffUtils, FindUtils, Libmicrohttpd, and SQLite. These projects are compiled using two compilers (GCC and Clang) with five optimization levels (O0, O1, O2, O3, and Os) on three architectures (arm, mips, and x86). Dataset-1 consists of 913,508 functions, which are further divided into three subsets (XO, XC, and XA) to fulfill different tasks. We also use 10-fold cross-validation to split Dataset-2 into three disjoint subsets of functions for training, validation, and testing, respectively.

Dataset-2 is designed specifically for evaluating IoTSim's capability in vulnerability detection. This dataset builds upon embedded firmware and vulnerabilities in previous works [50,63], including 20 firmware images from three vendors (D-Link, TP-Link, and NetGear) and 48 vulnerable functions from the OpenSSL project, as shown in Table 1.

Table 1. The vulnerabilities involved in Dataset-2.

CVE	Vulnerable Function	Confirmed #
CVE-2016-6303	MDC2_Update	10
CVE-2016-2182	BN_bn2dec	14
CVE-2021-23840	EVP_DecryptUpdate	17
CVE-2015-1789	X509_cmp_time	3
CVE-2016-0798	SRP_VBASE_get_by_usr	4

5.1.3. Metrics

The performance of IoTSim and the baselines is measured using the following metrics:

- Recall represents the ratio of correctly matched functions to the total number of function pairs with similar functions. A high recall suggests a low false-negative rate;
- Precision denotes the ratio of correctly matched functions to the total number of function pairs predicted as similar. High precision indicates a low false-positive rate;
- MRR stands for Mean Reciprocal Rank, which is a relative score that calculates the average or mean of the inverse of the ranks at which the first relevant function is retrieved for a set of queries.

5.2. Evaluation on Multiple Scenarios

In this subsection, we conduct experiments to evaluate the performance of IoTSim and other baselines at the function level. All evaluations in this subsection are conducted on Dataset-1. We perform IoTSim and baselines under multiple scenarios, including cross-architecture, cross-compiler, and cross-optimization scenarios, to thoroughly assess their performance. In a real-world BCSD task at the function level, a queried function is typically compared to numerous candidate functions to calculate similarity and retrieve the optimal result. Hence, we employ multiple pool sizes to evaluate IoTSim's performance in different scenarios and discuss the effects of pool sizes on BCSD approaches. Furthermore, to evaluate the contribution of Data-based Control Flow Graphs (DCFGs) to IoTSim, we configure IoTSim_{CFG} using Control Flow Graphs (CFGs) instead of DCFGs.

Tables 2–4 report the recall@1 and MRR results for each approach across different compilers (XC), optimization levels (XO), architectures (XA), and combined scenarios (i.e., XO + XC, XO + XA and All) in different pool sizes (10, 100 and 1000). IoTSim consistently outperforms all the baselines in terms of average recall@1 and MRR by considerable margins.

Table 2. BCSD results on multiple scenarios at the function level (poolsize = 10).

	Recall@1							MRR						
Models	**XO**	**XA**	**XC**	**XO + XC**	**XO + XA**	**XA + XC**	**All**	**XO**	**XA**	**XC**	**XO + XC**	**XO + XA**	**XA + XC**	**All**
SAFE	0.942	0.100	0.937	0.920	0.090	0.103	0.085	0.965	0.287	0.963	0.952	0.278	0.293	0.278
PalmTree	0.892	-	0.866	0.814	-	-	-	0.934	-	0.919	0.886	-	-	-
GMN	0.473	0.516	0.349	0.301	0.333	0.374	0.296	0.616	0.655	0.525	0.488	0.506	0.542	0.479
Trex	0.895	0.800	0.938	0.872	0.680	0.744	0.624	0.930	0.877	0.964	0.916	0.791	0.842	0.753
IoTSim	0.980	0.986	0.969	0.972	0.980	0.962	0.963	0.988	0.991	0.981	0.983	0.987	0.977	0.978
IoTSim_{CFG}	0.948	0.956	0.922	0.930	0.945	0.912	0.899	0.965	0.972	0.950	0.954	0.965	0.944	0.936

Table 3. BCSD results on multiple scenarios at the function level (poolsize = 10^2).

	Recall@1							MRR						
Models	**XO**	**XA**	**XC**	**XO + XC**	**XO + XA**	**XA + XC**	**All**	**XO**	**XA**	**XC**	**XO + XC**	**XO + XA**	**XA + XC**	**All**
SAFE	0.839	0.014	0.806	0.745	0.010	0.008	0.006	0.886	0.056	0.866	0.821	0.051	0.053	0.050
PalmTree	0.732	-	0.638	0.545	-	-	-	0.800	-	0.738	0.660	-	-	-
GMN	0.279	0.319	0.142	0.106	0.132	0.164	0.113	0.359	0.405	0.230	0.190	0.218	0.250	0.197
Trex	0.750	0.521	0.790	0.681	0.376	0.426	0.316	0.811	0.639	0.859	0.764	0.499	0.559	0.448
IoTSim	0.948	0.947	0.909	0.921	0.946	0.906	0.912	0.962	0.963	0.935	0.941	0.960	0.931	0.936
IoTSim_{CFG}	0.890	0.903	0.830	0.846	0.871	0.808	0.803	0.913	0.926	0.872	0.881	0.901	0.854	0.848

Table 4. BCSD results on multiple scenarios at the function level (Poolsize = 10^3).

	Recall@1							MRR						
Models	**XO**	**XA**	**XC**	**XO + XC**	**XO + XA**	**XA + XC**	**All**	**XO**	**XA**	**XC**	**XO + XC**	**XO + XA**	**XA + XC**	**All**
SAFE	0.687	0.002	0.618	0.523	0.001	0.001	0.001	0.749	0.010	0.702	0.616	0.007	0.007	0.007
PalmTree	0.590	-	0.406	0.335	-	-	-	0.648	-	0.497	0.424	-	-	-
GMN	0.193	0.190	0.066	0.042	0.053	0.073	0.038	0.230	0.237	0.098	0.071	0.085	0.112	0.070
Trex	0.627	0.280	0.603	0.465	0.186	0.198	0.133	0.673	0.375	0.678	0.549	0.263	0.286	0.208
IoTSim	0.901	0.903	0.833	0.849	0.905	0.812	0.832	0.922	0.924	0.868	0.880	0.926	0.851	0.867
IoTSim_{CFG}	0.823	0.825	0.740	0.744	0.781	0.688	0.684	0.851	0.858	0.782	0.789	0.818	0.739	0.734

For example, in the XC experiment with a pool size of 10, `IoTSim` achieves a recall@1 of 0.981 and a MRR of 0.970, which represents improvements of 4.6%, 13.2%, and 4.5% over SAFE, PalmTree, and Trex, respectively. Since IoT firmware originates from diverse architectures in the real world, we also evaluate IoTSim and other baselines in the cross-architecture scenario. In the XA experiment with a pool size of 100, IoTSim surpasses its closest competitor baseline (Trex [13]) by 0.426 for the recall@1, and over 50% for the MRR. We observe that SAFE achieves only a recall@1 of 0.014, while PalmTree fails in the cross-architecture scenario. PalmTree specifically focuses on the `x86` mono-instruction set and is unable to handle functions from different architectures. Although SAFE trains its model on different instruction sets, it remains challenging to establish semantic relations between instructions from diverse architectures and embed similar functions from different architectures into comparable embeddings. This limitation is also acknowledged in the Github issues (https://github.com/gadiluna/SAFE/issues/4, accessed on 30 March 2023), which shows current SAFE hardly supports cross-architecture BCSD tasks.

`IoTSim` addresses the challenge of different architectures by converting binary code into microcode. To tackle out-of-vocabulary (OOV) issues, `IoTSim` substitutes OOV words with their base tokens, preserving their semantics. This allows `IoTSim` to capture the basic semantics of OOV words, alleviating the problem of lost semantics due to OOV issues.

The experimental results, presented in Tables 2–4, indicate a decline in recall@1 and MRR for BCSD approaches as the pool size increases. In order to further investigate the impact of the pool size on the performance of BCSD approaches, we conduct experiments with pool sizes ranging from 10^0 to 10^4. Figures 8 and 9 illustrate the results of these experiments with a variety of pool sizes (1, 10, 50, 100, 500, 1000, 5000, and 10,000) under various experimental settings. For the sake of observation, we use a logarithmic x-axis in Figures 8 and 9. As the pool size increases, the performance of all BCSD approaches decreases. Compared to `IoTSim`, all baselines' relative performance worsens as the pool size increases. `IoTSim` does not display sharp drops in its performance, while the baselines'

performance generally declines more rapidly once poolsize over 10^2. For example, when the pool size is 10^0, the recall@1 achieved by SAFE and `IoTSim` is 0.989 and 0.994, respectively, in the XO experiments. When the pool size is 10^4, `IoTSim` achieves a recall@1 of 0.842 (−15.3%), demonstrating greater stability compared to other baselines. In contrast, SAFE only achieves a recall@1 of 0.569 (−42.5%) when the pool size is 10^4. This suggests that our approach is not affected by the pool size as much as other baselines.

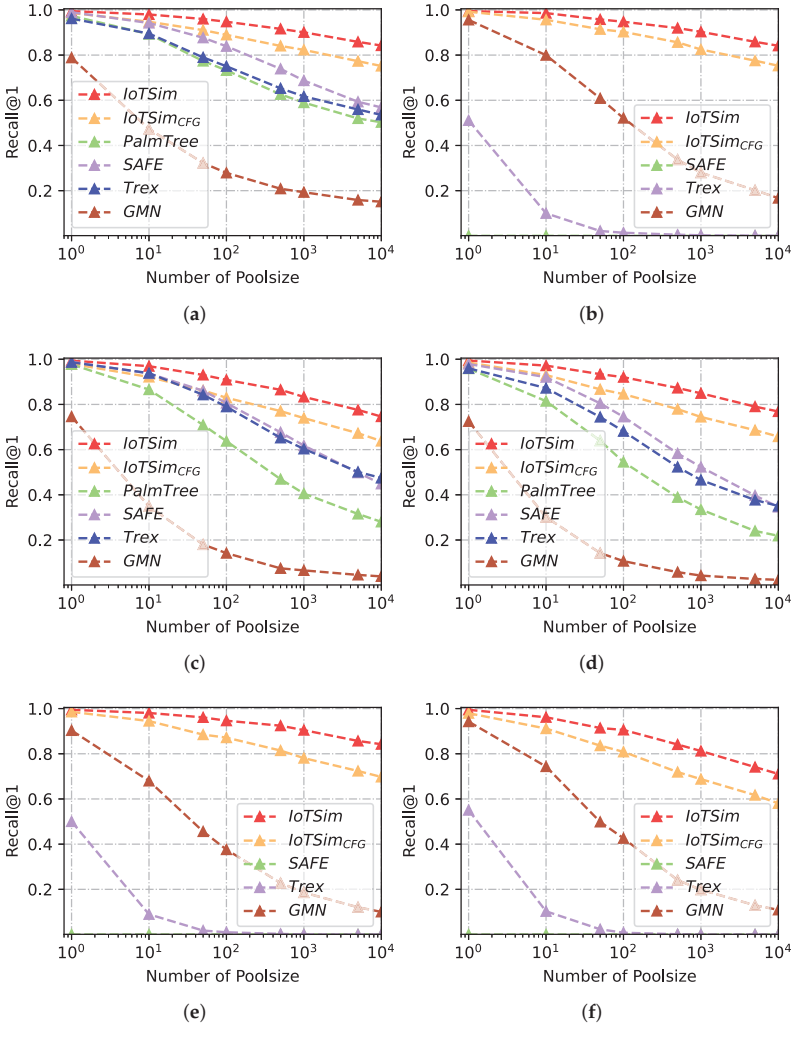


Figure 8. Recall@1 results of multiple scenarios with different poolsizes. (**a**) Recall@1, XO. (**b**) Recall@1, XA. (**c**) Recall@1, XC. (**d**) Recall@1, XO + XC. (**e**) Recall@1, XO + XA. (**f**) Recall@1, XA + XC.

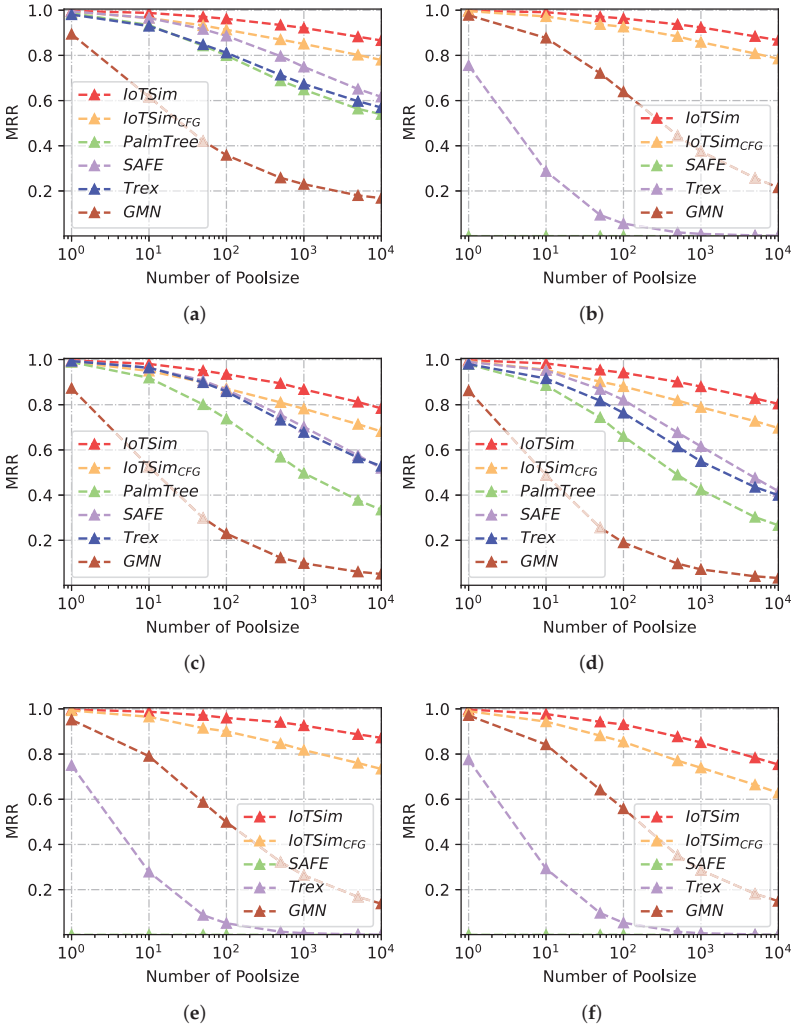


Figure 9. MRR results of multiple scenarios with different poolsizes. (**a**) MRR, XO. (**b**) MRR, XA. (**c**) MRR, XC. (**d**) MRR, XO + XC. (**e**) MRR, XO + XA. (**f**) MRR, XA + XC.

5.3. Ablation Study

We conduct experiments on `IoTSim` to evaluate the contributions of our proposed block relation matrix.

In binary functions, the relations between basic blocks are various. Previous approaches use control flow graphs as function structure features, which mainly consider connected relations without data dependence, edge direction, and address adjacency. We propose DCFGs to capture functions' structures deeply and generate block relation matrix integrating data dependence, directed adjacency, and address adjacency. In Tables 2–4, `IoTSim` outperforms `IoTSim`$_{CFG}$ in terms of average recall@1 and MRR by considerable margins. Figures 8 and 9 also show `IoTSim` achieves higher performance than `IoTSim`$_{CFG}$ and is less impacted by poolsize than `IoTSim`$_{CFG}$. For example, `IoTSim` improves the recall@1 by 21.6% over `IoTSim`$_{CFG}$ in the `All` scenario (poolsize = 10^3). With the help of the DCFGs, `IoTSim` captures more accurate data and control flow structures of functions, which makes `IoTSim` identify similar functions and distinguish dissimilar functions with a higher recall@1 and MRR.

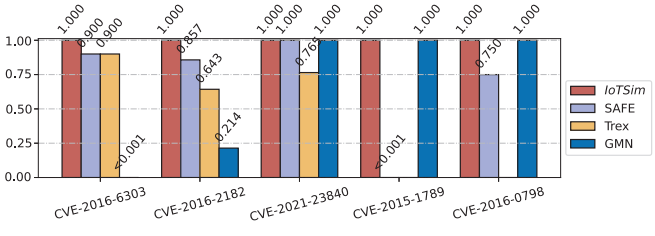
5.4. Applications

We evaluate IoTSim and baselines in two practical applications: vulnerability detection and component analysis.

5.4.1. Vulnerability Detection

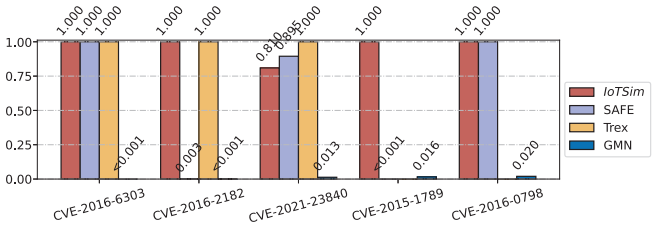
Vulnerability detection is a crucial application in the field of computer security. Within the IoT context, routers play a vital role in facilitating communication between connected IoT devices. In this subsection, we gather 20 firmware images of routers from three vendors, namely D-Link, TP-Link, and NetGear. We identify five known vulnerabilities in OpenSSL from the CVE database. These firmware images and vulnerabilities are then used to evaluate the performance of IoTSim and other baselines in the vulnerability detection tasks. In total, there are 47,090 functions, including 48 related vulnerable functions. In order to construct the vulnerability repository, we utilize IoTSim to generate function embeddings for each vulnerable and patched functions. During the vulnerability detection phase, we use all functions in the firmware libraries as function queries and search for the most similar function in the built vulnerability repository.

Figure 10 shows the results of the recall, precision, and F1-score of each CVE vulnerability. We compare IoTSim with other baselines. It is clear that for most of the CVEs, IoTSim's performance is significantly higher than the state-of-the-art approaches, e.g., SAFE and Trex. For instance, in the case of CVE-2016-2182 from the OpenSSL project, our method achieves a recall of 100%, successfully identifying all 14 vulnerable functions. In contrast, SAFE and Trex achieve recall values of 85.7% and 64.3%, respectively. Furthermore, it is worth noting that SAFE and Trex fail to obtain any recall for CVE-2015-1789 due to their reliance on capstone and objdump, which cannot deal with complex binary formats. For example, objdump (version 2.34) cannot extract function features from ELF files without a section table, which makes them fail to perform BCSD task in these binaries. This reflects the difficulty of the BCSD task in IoT scenarios, and the necessity of our approach. The results demonstrate that IoTSim can be effectively deployed as a reliable tool for detecting vulnerabilities in IoT scenarios.

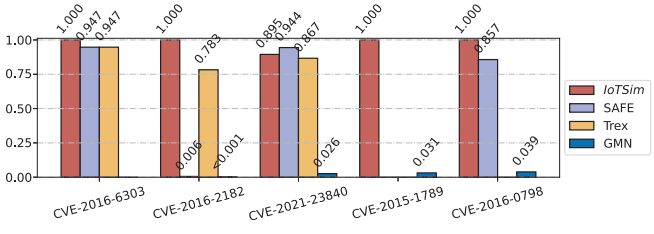


(**a**)

Figure 10. *Cont.*



(**b**)



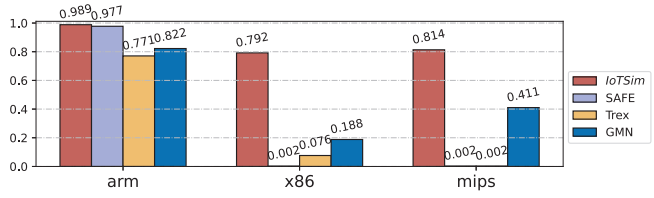
(**c**)

Figure 10. Results of real-world vulnerability detection. (**a**) Recall. (**b**) Precision. (**c**) F1-score.

5.4.2. Component Analysis

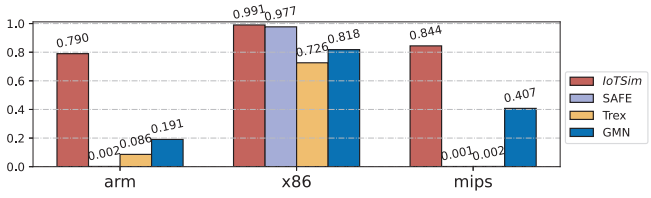
BCSD approaches can provide component analysis for unknown executable binary files and match symbol tables and source code for reference, making it easier for reverse engineers to analyze unknown binaries in IoT security. In this section, we use the `OpenSSL` project, which is widely used in IoT firmware, as the benchmark to evaluate the performance of our proposed approach and other baselines on component analysis. Given an input binary file, the BCSD approaches compare it with our labeled binaries that contain debug information to match symbol tables.

Figure 11 shows recall results in component analysis with labeled binaries from different architectures (i.e., `x86`, `arm`, and `mips`). `IoTSim` achieves high recalls in all component analysis experiments. For example, `IoTSim` obtains a recall of 0.814 when the input files are from `mips` and the labeled binaries are from `arm`, which means more than 80% functions in the input binaries can be correctly matched to their symbol tables and their source code. This significantly reduces the manual burden when analyzing unknown binary files. Compared with the state-of-the-art approaches, `IoTSim` achieves the average recall of 0.874, which improves the recall by 3.0×, 0.8× and 2.1×, compared to SAFE, GMN, and Trex.

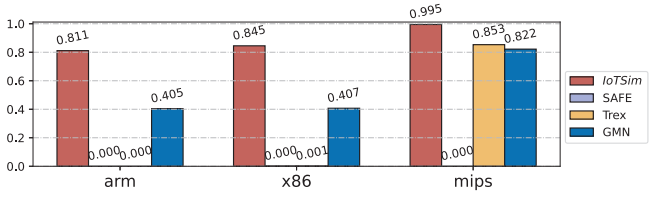


(**a**)

Figure 11. *Cont.*



(**b**)



(**c**)

Figure 11. Results of the component analysis on different architectures. (**a**) arm. (**b**) x86. (**c**) mips.

6. Conclusions

In this paper, we propose a novel IoT-oriented binary code similarity detection approach, called `IoTSim`. Our approach leverages a customized transformer-based language model with disentangled attention to generate embeddings for basic blocks. To address OOV challenges, we introduce a pre-training task called BTP that captures basic semantics for unseen tokens. To help `IoTSim` understand multiple relations between basic blocks, we integrate directed jumps, data dependency, and address adjacency to build block relation matrix. We then assign different weights to different relations in block relation matrix and use multi-layer GCN to generate function embeddings.

We implemented a prototype of `IoTSim` and conducted experiments to evaluate its performance. The experiment results show that `IoTSim` surpasses the state-of-the-art approaches Trex, SAFE, and PalmTree. Additionally, we observe that data-based control flow graphs have positive effects for `IoTSim`. In real-world applications, `IoTSim` proves valuable in helping researchers detect vulnerabilities and identify components in unknown binaries within various IoT firmware. These findings demonstrate that our proposed BCSD approach contributes to practical applications in security analysis within the IoT ecosystem, relieving researchers from the burdensome task of security analysis.

Author Contributions: Methodology, Z.L. and P.W.; software, Z.L.; formal analysis, Z.L., P.W. and B.W.; data curation, Z.L.; writing—original draft preparation, Z.L.; writing—review and editing, Z.L., P.W., W.X., X.Z. and B.W.; supervision, B.W.; project administration, B.W.; funding acquisition, P.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National University of Defense Technology Research Project (ZK20-17, ZK20-09), the National Natural Science Foundation China (62272472, 61902405), the HUNAN Province Natural Science Foundation (2021JJ40692), and the National Key Research and Development Program of China under Grant No. 2021YFB0300101.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Available upon request.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BCSD	Binary Code Similarity Detection
IoT	Internet of Things
OOV	Out-of-Vocabulary
TPC	Third-Parity Components
ISA	Instruction Set Architecture
CVE	Common Vulnerabilities and Exposures
ML	Machine Learning
CFG	Control Flow Graph
IR	Intermediate Representation
GNN	Graph Neural Network
NLP	Natural Language processing
GCN	Graph Convolutional Network

References

1. Lionel Sujay Vailshery. IoT Connected Devices Worldwide 2019–2023. 2023. Available online: https://news.sophos.com/en-us/2022/05/04/attacking-emotets-control-flow-flattening/ (accessed on 3 March 2023).
2. Zhao, B.; Ji, S.; Lee, W.H.; Lin, C.; Weng, H.; Wu, J.; Zhou, P.; Fang, L.; Beyah, R. A large-scale empirical study on the vulnerability of deployed iot devices. *IEEE Trans. Dependable Secur. Comput.* **2020**, *19*, 1826–1840. [CrossRef]
3. Wang, Q.; Ji, S.; Tian, Y.; Zhang, X.; Zhao, B.; Kan, Y.; Lin, Z.; Lin, C.; Deng, S.; Liu, A.X.; et al. MPInspector: A Systematic and Automatic Approach for Evaluating the Security of IoT Messaging Protocols. In Proceedings of the 30th USENIX Security Symposium (USENIX Security 21), Virtual, 11–13 August 2021; pp. 4205–4222.
4. Costin, A.; Zaddach, J. Iot malware: Comprehensive survey, analysis framework and case studies. *BlackHat USA* **2018**, *1*, 1–9.
5. Luo, Z.; Wang, B.; Tang, Y.; Xie, W. Semantic-based representation binary clone detection for cross-architectures in the internet of things. *Appl. Sci.* **2019**, *9*, 3283. [CrossRef]
6. Sun, H.; Wang, X.; Buyya, R.; Su, J. CloudEyes: Cloud-based malware detection with reversible sketch for resource-constrained internet of things (IoT) devices. *Softw. Pract. Exp.* **2017**, *47*, 421–441. [CrossRef]
7. Kolias, C.; Kambourakis, G.; Stavrou, A.; Voas, J. DDoS in the IoT: Mirai and other botnets. *Computer* **2017**, *50*, 80–84. [CrossRef]
8. Feng, Q.; Wang, M.; Zhang, M.; Zhou, R.; Henderson, A.; Yin, H. Extracting conditional formulas for cross-platform bug search. In Proceedings of the 2017 ACM Asia Conference on Computer and Communications Security, Abu Dhabi, United Arab Emirates, 2–6 April 2017; pp. 346–359. [CrossRef]
9. Pewny, J.; Garmany, B.; Gawlik, R.; Rossow, C.; Holz, T. Cross-architecture bug search in binary executables. In Proceedings of the 2015 IEEE Symposium on Security and Privacy, San Jose, CA, USA, 17–21 May 2015; pp. 709–724.
10. Gao, J.; Yang, X.; Fu, Y.; Jiang, Y.; Sun, J. Vulseeker: A semantic learning based vulnerability seeker for cross-platform binary. In Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, Montpellier, France, 3–7 September 2018; pp. 896–899. [CrossRef]
11. Wang, H.; Qu, W.; Katz, G.; Zhu, W.; Gao, Z.; Qiu, H.; Zhuge, J.; Zhang, C. jTrans: Jump-Aware Transformer for Binary Code Similarity. *arXiv* **2022**, arXiv:2205.12713.
12. Lin, J.; Wang, D.; Chang, R.; Wu, L.; Zhou, Y.; Ren, K. EnBinDiff: Identifying Data-only Patches for Binaries. *IEEE Trans. Dependable Secur. Comput.* **2021**, *20*, 343–359. [CrossRef]
13. Pei, K.; Xuan, Z.; Yang, J.; Jana, S.; Ray, B. Trex: Learning execution semantics from micro-traces for binary similarity. *arXiv* **2020**, arXiv:2012.08680.
14. Ding, S.H.; Fung, B.C.; Charland, P. Asm2Vec: Boosting static representation robustness for binary clone search against code obfuscation and compiler optimization. In Proceedings of the IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 19–23 May 2019; pp. 472–489. [CrossRef]
15. David, Y.; Partush, N.; Yahav, E. Statistical similarity of binaries. In Proceedings of the 37th ACM SIGPLAN Conference on Programming Language Design and Implementation, Santa Barbara, CA, USA, 13–17 June 2016; pp. 266–280.
16. Yang, S.; Cheng, L.; Zeng, Y.; Lang, Z.; Zhu, H.; Shi, Z. Asteria: Deep Learning-based AST-Encoding for Cross-platform Binary Code Similarity Detection. In Proceedings of the 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Taipei, Taiwan, 21–24 June 2021; pp. 224–236.
17. David, Y.; Partush, N.; Yahav, E. Firmup: Precise Static Detection of Common Vulnerabilities in Firmware. In Proceedings of the ACM SIGPLAN Notices, Mumbai, India, 15–17 January 2018; ACM: New York, NY, USA, 2018; Volume 53, pp. 392–404.
18. Shirani, P.; Collard, L.; Agba, B.L.; Lebel, B.; Debbabi, M.; Wang, L.; Hanna, A. Binarm: Scalable and efficient detection of vulnerabilities in firmware images of intelligent electronic devices. In Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, Hamburg, Germany, 12–14 July 2018; Springer: Cham, Switzerland, 2018; pp. 114–138.

19. Feng, Q.; Zhou, R.; Xu, C.; Cheng, Y.; Testa, B.; Yin, H. Scalable Graph-based Bug Search for Firmware Images. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security—CCS'16, Vienna, Austria, 24–28 October 2016; ACM: New York, NY, USA, 2016; pp. 480–491. [CrossRef]
20. Pewny, J.; Schuster, F.; Bernhard, L.; Holz, T.; Rossow, C. Leveraging semantic signatures for bug search in binary programs. In Proceedings of the 30th Annual Computer Security Applications Conference, New Orleans, LA, USA, 8–12 December 2014; pp. 406–415.
21. Chandramohan, M.; Xue, Y.; Xu, Z.; Liu, Y.; Cho, C.Y.; Kuan, T.H.B. BinGo: Cross-Architecture cross-os binary search. In Proceedings of the ACM SIGSOFT Symposium on the Foundations of Software Engineering, Seattle, WA, USA, 13–18 November 2016; pp. 678–689. [CrossRef]
22. Ahn, S.; Ahn, S.; Koo, H.; Paek, Y. Practical Binary Code Similarity Detection with BERT-based Transferable Similarity Learning. In Proceedings of the 38th Annual Computer Security Applications Conference, Austin TX USA, 5–9 December 2022; pp. 361–374.
23. Ming, J.; Xu, D.; Jiang, Y.; Wu, D. Binsim: Trace-based semantic binary diffing via system call sliced segment equivalence checking. In Proceedings of the 26th USENIX Security Symposium (USENIX Security 17), Vancouver, BC, Canada, 16–18 August 2017; pp. 253–270.
24. Cesare, S.; Xiang, Y.; Zhou, W. Control flow-based malware variantdetection. *IEEE Trans. Dependable Secur. Comput.* **2013**, *11*, 307–317. [CrossRef]
25. Hu, X.; Chiueh, T.C.; Shin, K.G. Large-scale malware indexing using function-call graphs. In Proceedings of the 16th ACM Conference on Computer and Communications Security, Chicago, IL, USA, 9–13 November 2009; pp. 611–620.
26. Bayer, U.; Comparetti, P.M.; Hlauschek, C.; Kruegel, C.; Kirda, E. Scalable, behavior-based malware clustering. In Proceedings of the NDSS, Citeseer, San Diego, CA, USA, 8–11 February 2009; Volume 9, pp. 8–11.
27. Farhadi, M.R.; Fung, B.C.; Charland, P.; Debbabi, M. Binclone: Detecting code clones in malware. In Proceedings of the 2014 Eighth International Conference on Software Security and Reliability (SERE), San Francisco, CA, USA, 30 June–2 July 2014; pp. 78–87.
28. Jang, J.; Woo, M.; Brumley, D. Towards automatic software lineage inference. In Proceedings of the 22nd USENIX Security Symposium (USENIX Security 13), Washington, DC, USA, 14–16 August 2013; pp. 81–96.
29. Xu, Z.; Chen, B.; Chandramohan, M.; Liu, Y.; Song, F. SPAIN: Security patch analysis for binaries towards understanding the pain and pills. In Proceedings of the 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE), Buenos Aires, Argentina, 20–28 May 2017; pp. 462–472.
30. Huang, H.; Youssef, A.M.; Debbabi, M. Binsequence: Fast, accurate and scalable binary code reuse detection. In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, Abu Dhabi, United Arab Emirates, 2–6 April 2017; pp. 155–166.
31. Kargén, U.; Shahmehri, N. Towards robust instruction-level trace alignment of binary code. In Proceedings of the 2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE), Urbana-Champaign IL, USA, 30 October–3 November 2017; pp. 342–352.
32. Zynamics. BinDiff. 2021. Available online: https://www.zynamics.com/bindiff.html (accessed on 20 February 2023).
33. Gao, D.; Reiter, M.K.; Song, D. Binhunt: Automatically finding semantic differences in binary programs. In Proceedings of the International Conference on Information and Communications Security, Chongqing, China, 19–21 November 2021; Springer: Berlin/Heidelberg, Germany, 2008; pp. 238–255.
34. Li, X.; Yu, Q.; Yin, H. PalmTree: Learning an Assembly Language Model for Instruction Embedding. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual, 15–19 December 2021; pp. 3236–3251.
35. Massarelli, L.; Di Luna, G.A.; Petroni, F.; Baldoni, R.; Querzoni, L. Safe: Self-attentive function embeddings for binary similarity. In Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, Hamburg, Germany, 12–14 July 2019; Springer: Cham, Switzerland, 2019; pp. 309–329.
36. Xu, X.; Liu, C.; Feng, Q.; Yin, H.; Song, L.; Song, D. Neural Network-based Graph Embedding for Cross-Platform Binary Code Similarity Detection. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security—CCS'17, Dallas, TX, USA, 30 October–3 November 2017; pp. 363–376. [CrossRef]
37. Zuo, F.; Li, X.; Young, P.; Luo, L.; Zeng, Q.; Zhang, Z. Neural Machine Translation Inspired Binary Code Similarity Comparison beyond Function Pairs. In Proceedings of the 2019 Network and Distributed System Security Symposium, San Diego, CA, USA, 24–27 February 2019. [CrossRef]
38. He, P.; Liu, X.; Gao, J.; Chen, W. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv* **2020**, arXiv:2006.03654.
39. Khoo, W.M.; Mycroft, A.; Anderson, R. Rendezvous: A search engine for binary code. In Proceedings of the 2013 10th Working Conference on Mining Software Repositories (MSR), San Francisco, CA, USA, 18–19 May 2013; pp. 329–338. [CrossRef]
40. Myles, G.; Collberg, C. K-gram based software birthmarks. In Proceedings of the 2005 ACM Symposium on Applied Computing, Santa Fe, NM, USA, 13–17 March 2005; pp. 314–318.
41. David, Y.; Yahav, E. Tracelet-based code search in executables. In Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), Edinburgh, UK, 9–11 June 2014; pp. 349–360. [CrossRef]
42. Egele, M.; Woo, M.; Chapman, P.; Brumley, D. Blanket execution: Dynamic similarity testing for program binaries and components. In Proceedings of the 23rd USENIX Security Symposium (USENIX Security 14), San Diego, CA, USA, 20–22 August 2014; pp. 303–317.

43. Luo, L.; Ming, J.; Wu, D.; Liu, P.; Zhu, S. Semantics-based obfuscation-resilient binary code similarity comparison with applications to software and algorithm plagiarism detection. *IEEE Trans. Softw. Eng.* **2017**, *43*, 1157–1177. [CrossRef]
44. Duan, Y.; Li, X.; Wang, J.; Yin, H. DeepBinDiff: Learning Program-Wide Code Representations for Binary Diffing. In Proceedings of the 27rd Symposium on Network and Distributed System Security (NDSS), San Diego, CA, USA, 24–27 February 2020. [CrossRef]
45. Yu, Z.; Cao, R.; Tang, Q.; Nie, S.; Huang, J.; Wu, S. Order Matters: Semantic-Aware Neural Networks for Binary Code Similarity Detection. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 1145–1152. [CrossRef]
46. Le, Q.; Mikolov, T. Distributed representations of sentences and documents. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 1188–1196.
47. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the NAACL HLT 2019—2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4171–4186.
48. Eschweiler, S.; Yakdan, K.; Gerhards-Padilla, E. discovRE: Efficient Cross-Architecture Identification of Bugs in Binary Code. In Proceedings of the 2016 Network and Distributed System Security Symposium, San Diego, CA, USA, 21–24 February 2016; pp. 21–24. [CrossRef]
49. David, Y.; Partush, N.; Yahav, E. Similarity of binaries through re-optimization. In Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation, Barcelona Spain, 18–23 June 2017; pp. 79–94.
50. Luo, Z.; Wang, P.; Wang, B.; Tang, Y.; Xie, W.; Zhou, X.; Liu, D.; Lu, K. VulHawk: Cross-architecture Vulnerability Detection with Entropy-based Binary Code Search. In Proceedings of the Network and Distributed Systems Security (NDSS) Symposium, 27 February–3 March 2023; Volume 2023.
51. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. *arXiv* **2013**, arXiv:1310.4546.
52. Ding, S.H.; Fung, B.C.; Charland, P. Kam1n0: MapReduce-based Assembly Clone Search for Reverse Engineering. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining—KDD'16, San Francisco, CA, USA, 13–17 August 2016; pp. 461–470. [CrossRef]
53. Hjelm, R.D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Bachman, P.; Trischler, A.; Bengio, Y. Learning deep representations by mutual information estimation and maximization. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
54. Rays, H. IDA Pro. 2021. Available online: https://www.hex-rays.com/products/ida/ (accessed on 20 February 2023).
55. Song, Q.; Zhang, Y.; Wang, B.; Chen, Y. Inter-BIN: Interaction-based Cross-architecture IoT Binary Similarity Comparison. *IEEE Internet Things J.* **2022**, *9*, 20018–20033. [CrossRef]
56. Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.; Le, Q.V.; Salakhutdinov, R. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv* **2019**, arXiv:1901.02860.
57. Shaw, P.; Uszkoreit, J.; Vaswani, A. Self-attention with relative position representations. *arXiv* **2018**, arXiv:1803.02155.
58. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
59. Ming Chen, Z.W.; Zengfeng Huang, B.D.; Li, Y. Simple and Deep Graph Convolutional Networks. In Proceedings of the 37th International Conference on Machine Learning, Virtual, 13–18 July 2020.
60. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Proceedings of the Annual Conference on Neural Information Processing Systems 2019, Vancouver, BC, Canada, 8–14 December 2019; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019.
61. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. Transformers: State-of-the-Art Natural Language Processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Association for Computational Linguistics, Virtual, 16–20 November 2020; pp. 38–45.
62. Fey, M.; Lenssen, J.E. Fast Graph Representation Learning with PyTorch Geometric. In Proceedings of the ICLR Workshop on Representation Learning on Graphs and Manifolds, New Orleans, LA, USA, 6–9 May 2019.
63. Marcelli, A.; Graziano, M.; Ugarte-Pedrero, X.; Fratantonio, Y.; Mansouri, M.; Balzarotti, D. How Machine Learning Is Solving the Binary Function Similarity Problem. In Proceedings of the Usenix Security 2022, San Diego, CA, USA, 20–22 August 2022; pp. 83–101.

sensors

MDPI

Article

Decentralized Policy-Hidden Fine-Grained Redaction in Blockchain-Based IoT Systems

Hongchen Guo [1], Xiaolong Tao [2], Mingyang Zhao [2], Tong Wu [2,*], Chuan Zhang [2], Jingfeng Xue [1] and Liehuang Zhu [2]

[1] School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China; guohongchen@bit.edu.cn (H.G.); xuejf@bit.edu.cn (J.X.)
[2] School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China; ac_xiaolongt@163.com (X.T.) mingyangz@bit.edu.cn (M.Z.); chuanz@bit.edu.cn (C.Z.); liehuangz@bit.edu.cn (L.Z.)
* Correspondence: tracy_tongw@163.com

Abstract: Currently, decentralized redactable blockchains have been widely applied in IoT systems for secure and controllable data management. Unfortunately, existing works ignore policy privacy (i.e., the content of users' redaction policies), causing severe privacy leakage threats to users since users' policies usually contain large amounts of private information (e.g., health conditions and geographical locations) and limiting the applications in IoT systems. To bridge this research gap, we propose PFRB, a policy-hidden fine-grained redactable blockchain in decentralized blockchain-based IoT systems. PFRB follows the decentralized settings and fine-grained chameleon hash-based redaction in existing redactable blockchains. In addition, PFRB hides users' policies during policy matching such that apart from successful policy matching, users' policy contents cannot be inferred and valid redactions cannot be executed. Some main technical challenges include determining how to hide policy contents and support policy matching. Inspired by Newton's interpolation formula-based secret sharing, PFRB converts policy contents into polynomial parameters and utilizes multi-authority attribute-based encryption to further hide these parameters. Theoretical analysis proves the correctness and security against the chosen-plaintext attack. Extensive experiments on the FISCO blockchain platform and IoT devices show that PFRB achieves competitive efficiency over current redactable blockchains.

Keywords: blockchain-based IoT systems; fine-grained redaction; policy hiding; decentralization

Citation: Guo, H.; Tao, X.; Zhao, M.; Wu, T.; Zhang, C.; Xue, J.; Zhu, L. Decentralized Policy-Hidden Fine-Grained Redaction in Blockchain-Based IoT Systems. *Sensors* **2023**, *23*, 7105. https://doi.org/10.3390/s23167105

Academic Editor: Jian Li

Received: 9 July 2023
Revised: 31 July 2023
Accepted: 9 August 2023
Published: 11 August 2023

1. Introduction

The Internet of Things (IoT) is defined as connecting all objects through information-sensing devices such as radio frequency identification to the Internet, enabling intelligent recognition and management. The concept of IoT entails integrating sensors into various objects, including power grids and railways, enabling data collection and communication for seamless connectivity and interaction with the physical world [1–3]. However, due to the inherent decentralization of IoT device deployment [4,5], achieving secure data management poses challenges.

To address these challenges, blockchain has been widely applied in IoT systems as a decentralized data management platform, providing traceability and integrity to secure IoT systems [6–8]. According to a report by Morder Intelligence (https://www.mordorintelligence.com/industry-reports/blockchain-iot (accessed on 31 July 2023)), the market size of blockchain-based IoT systems is expected to grow from USD 568.51 million in 2023 to USD 3436.54 million by 2028. Despite these benefits, immutable blockchains present two limitations in current blockchain-based IoT systems. Firstly, the immutability of blockchain violates the right to be forgotten in the General Data Protection Regulation (GDPR) (https://gdpr-info.eu/art-17-gdpr/ (accessed on 31 July 2023)), severely limiting the development and implementation of blockchain-based IoT systems. GDPR mandates

that users should be able to erase their personal data (e.g., health and transportation data) from data management systems. However, with immutable blockchains, once data are appended to the blockchain, no modifications can be made [9]. This conflict between immutability and GDPR could result in high fines for blockchain-based IoT systems. Secondly, IoT devices are vulnerable to network attacks and can be used to spread improper information (e.g., indecent surveillance videos) in blockchain-based IoT systems. If this improper information cannot be redacted, it may negatively impact the blockchain-based IoT application ecology [10–12].

To address these limitations, some redactable blockchains have been proposed, adopting chameleon hashes [13–18]. Specifically, in 2017, Ateniese et al. [13] introduced the first redactable blockchain by using the chameleon hash technique. Their work extended researchers' insight into immutable blockchains and made blockchains compliant with data regulation (e.g., GDPR). Unfortunately, their work only supports all-or-nothing redaction privileges, i.e., they either cannot redact any data or can redact all data. Clearly, it is infeasible in practice since real-world blockchain applications contain a large number of devices that own different attributes. To achieve fine-grained redaction, Derler et al. [13], Ma et al. [18], and Xu et al. [19] utilized attribute-based fine-grained access control and chameleon hash to introduce the policy-based chameleon hash technique (PCH), subsequently using PCH to achieve fine-grained redactable blockchains. Following PCH, Mao et al. [18] further introduced a decentralized policy-based chameleon hash technique (DPCH) and used it to design a decentralized fine-grained redactable blockchain. However, in existing fine-grained redactable blockchains, users' policies are public to all users, making them unsuitable for some policy-sensitive IoT systems, such as IoT-based smart healthcare and smart transportation [20]. In these IoT systems, users' policies contain sensitive private information, e.g., users' health conditions and geographical locations [21]. For instance, in an IoT-based smart healthcare application, users use their IoT devices (e.g., smartwatches) to record their health data and utilize blockchain-based IoT systems to manage the data. In the application, users' policies usually contain information about users' health conditions, such as sensitive health details [22]. Obviously, if this private information is leaked, users may face discrimination and even potential attacks [23]. Therefore, there is a great need to design decentralized policy-hidden redactable blockchains in blockchain-based IoT systems. We make a comparison between our proposed solution and existing works, as shown in Table 1. We address the issue of policy disclosure in a decentralized, fine-grained setting.

Table 1. Comparison between PFRB and existing redactable blockchain schemes.

	Fine-Grained	**Decentralization**	**Policy-Hidden**
AMVA17 [13]	×	×	×
DSSS19 [16]	✓	×	×
DMT19 [14]	×	✓	×
TLL20 [24]	✓	×	×
PVM21 [25]	✓	×	×
JSZ21 [26]	×	×	×
XNMX21 [27]	✓	×	×
XNMX22 [15]	✓	×	×
RBDS22 [18]	✓	✓	×
Ours	✓	✓	✓

To solve the above problem, we propose a policy-hidden fine-grained redactable blockchain scheme (named PFRB) in decentralized blockchain-based IoT systems. In PFRB, users can enjoy blockchain services while hiding their policies. A technical challenge in designing PFRB is how to hide policy contents and support policy matching. PFRB

draws inspiration from Newton's interpolation formula-based secret sharing and gracefully converts policy contents into polynomial parameters. Particularly, we summarize the main contributions as follows:

- We propose a policy-hidden fine-grained redactable blockchain scheme (named PFRB) for blockchain-based IoT systems. With decentralized settings, PFRB enables users to achieve fine-grained data redaction without compromising policy privacy.
- PFRB leverages multi-authorized attribute-based encryption and Newton's interpolation formula-based secret sharing to construct a decentralized secret sharing for policy hiding. Then, based on the constructed secret sharing, PFRB further enriches chameleon hashes to achieve decentralized policy-hidden fine-grained redactable blockchains.
- Security analysis proves the security of PFRB under the chosen-plaintext attack in the random oracle model. Experimental results show that PFRB has competitive efficiency over recent fine-grained redactable blockchain schemes.

Organization. The remainder of this paper is structured as follows. In the next section, we review existing works on redactable blockchains to introduce the research gap. In Section 3, we provide an introduction to the preliminary information, followed by an overview of the PFRB system in Section 4. Next, in Section 5, we provide insight into the detailed construction of PFRB. Section 6 presents a formal correctness analysis and security analysis. Performance evaluation of PFRB is shown in Section 7. Finally, concluding remarks are given in Section 8.

2. Related Work

This section provides a systematic review of the current literature on redactable blockchain techniques and introduces the research gap.

In 2017, Ateniese et al. [13] introduced the first redactable blockchain by using the chameleon hash technique. Their work extended researchers' insight into immutable blockchains and made blockchains compliant with the data regulation (e.g., GDPR and CCPA). Unfortunately, their work only supports all-or-nothing redaction privileges, i.e., they either cannot redact any data or can redact all data. Clearly, it is infeasible in practice since real-world blockchain applications contain a large number of devices that own different attributes.

To address the above limitation, Derler et al. [16] proposed a policy-based chameleon hash (PCH) and used PCH to implement fine-grained redactable blockchains. PCH combines chameleon hashes, ephemeral trapdoors, and linear secret sharing matrix-based attribute-based encryption to associate transactions with access policies, enabling rewriting only when editors' attributes satisfy the policy. Followed by Derler et al.'s work, Tian et al. [24] and Xu et al. [19] extent redactable blockchains with accountability by introducing digital signatures, respectively. Due to the utilization of centralized attribute-based encryption schemes, the above schemes can only be applied in consortium blockchains, and cannot support generally decentralized settings. However, in practical blockchain applications, especially distributed IoT systems, decentralized systems are more general. To achieve decentralized redactable blockchains, Ma et al. [18] introduced the first decentralized policy-based chameleon hash (DPCH) by linear secret sharing matrix-based multi-authority attribute-based encryption and used DPCH to achieve decentralized fine-grained blockchain redaction.

However, redaction policies in the above redactable blockchains are open to all participants, which is infeasible in practical blockchain-based IoT systems since policies in IoT systems usually contain users' private information. For instance, in IoT-based smart healthcare applications, patients use their smart swatches to collect their health conditions (such as heart rate and personal temperature) and specify policies to allow their private doctors to edit. Evidently, these policies usually contain sensitive information regarding patients' health conditions and geographical locations. Once this private information is leaked, adversaries can launch attacks on these patients such as robbery and discrimination.

Thus, the protection of policy privacy in decentralized redactable blockchains represents an urgent matter.

3. Preliminary

In this section, we introduce the building blocks of PFRB, i.e., multi-authority attribute-based encryption, Newton's interpolation formula-based secret sharing, and chameleon hash.

3.1. Multi-Authority Attribute-Based Encryption

A multi-authority attribute-based encryption (MA-ABE) [28] system consists of arbitrary numbers of attribute authorities and users. A set of global public parameters is defined in the system. Users can select an attribute authority and obtain their corresponding decryption key. The authorization authority performs the appropriate attribute key generation algorithm and returns the result to the user. The encryption process uses the global public parameters and a set of attributes to generate the ciphertext. The decryption process uses the decryption key for the attribute set to perform decryption.

Definition 1 (MA-ABE): A multi-authority attribute-based encryption ABE_{MC} involves three types of entities: authorities, data owners, and data users. It includes five algorithms:

- Global Setup $(\lambda) \rightarrow (GP)$: This algorithm accepts a secure parameter λ as input and produces a public global parameter GP as output.
- Authority Setup $(GP) \rightarrow (PK, SK)$: In this step, the algorithm takes the public global parameter GP as input and generates a public key PK and a secret key SK as output. It is crucial to keep the secret key SK confidential, while the public key PK is intended for publication.
- Encryption $(M, (A, \rho), GP, \{PK\}) \rightarrow (CT)$: The algorithm accepts several inputs, including a message M, an $n \times \ell$ access matrix A with ρ mapping its rows to attributes, the global parameter GP, and the public keys of the relevant authorities PK. It then produces a ciphertext CT as output.
- KeyGen $(ID, i, SK, GP) \rightarrow (K_{i,ID})$: The algorithm generates a key $K_{i,ID}$ for attribute i associated with an authority using the inputs: a global identifier ID, the attribute i, the secret key SK, and the public global parameter GP.
- Decryption $(CT, \{K_{i,ID}\}, GP) \rightarrow (M)$: The algorithm decrypts the ciphertext CT using the input parameters: the key $K_{i,ID}$ for ID and attribute i, as well as the global parameter GP. The result of the decryption process is the message M.

3.2. Newton's Interpolation Formula-Based Secret Sharing

In this paper, Newton's interpolation formula is primarily used for key recovery. Also, due to the introduction of polynomial-oriented secret sharing, PFRB achieves higher efficiency than traditional linear secret sharing matrix-based works. The transaction issuer hides the key within the zeroth term of a polynomial, ensuring that only users who meet the policy requirements can reconstruct the polynomial and access the hidden key.

- Secret Generation: Assume that there are $(n+1)$ points represented as (x_0, y_0), (x_1, y_1), $\ldots$, (x_n, y_n). Here, x_i is called the interpolation point, and y_i is called the interpolation value. Given an interpolation polynomial $f(x)$, for each $i = 0, 1, 2, \ldots, n$, y_i is represented as $y_i = f(x_i)$. The Newton's basis $n_i(x)$ is defined as follows:

$$n_i(x) = \begin{cases} 1, & \text{if } i = 0, \\ \prod_{j=0}^{i-1}(x - x_j), & \text{otherwise.} \end{cases}$$

Based on $n_i(x)$, the Newton's interpolation polynomial $Q_n(x)$ can be defined as follows:

$$Q_n(x) = K_0 + K_1(x - x_0) + \cdots + K_n \prod_{i=0}^{n-1}(x - x_i) = \sum_{i=0}^{n} K_i n_i(x).$$

Specifically, based on x_0, $Q_n(x)$ can be estimated as follows:

$$Q_n(x_0) = \sum_{i=0}^{n} K_i n_i(x) = K_0 = f(x_0) = f[x_0].$$

where $f[x_0]$ represents the zeroth order divided difference. Similarly, based on x_1, the Newton's interpolation polynomial $Q_n(x)$ is estimated as follows:

$$Q_n(x_1) = K_1(x - x_0) + K_0 = K_1(x - x_0) + f[x_0] = f[x_1].$$

Thus, K_1 can be estimated as follows:

$$K_1 = f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0}.$$

where $f[x_0, x_1]$ represents the first order divided difference. Without loss of generality, K_i can be defined as follows:

$$K_i = \frac{f[x_1, x_2, \ldots, x_i] - f[x_0, x_1, \ldots, x_{i-1}]}{x_i - x_0}.$$

where $f[x_1, x_2, \ldots, x_i]$ denotes the i-th order divided difference, respectively. For more details, the reader can refer to previous literature.

- Secret Construction: We can reconstruct the secret with Newton's parameters as follows:

$$s = Q_n(0) = \sum_{i=0}^{n} K_i n_i(0).$$

3.3. Chameleon Hash

A chameleon hash $\mathcal{CH}$ typically encompasses the following five algorithms:

- Setup$(1^\lambda) \rightarrow pp$: The probabilistic setup algorithm takes a security parameter λ as input and generates a public parameter pp as output. The public parameter pp is used in subsequent algorithms and protocols to ensure the security and functionality of the system.
- KeyGen$(pp) \rightarrow (pk, sk)$: The probabilistic key generation algorithm takes the public parameter pp as input and generates a public-secret key pair (pk, sk) as output. The public key pk is used for encryption or other public operations, while the secret key sk is kept confidential and used for decryption or other sensitive operations.
- Hash$(pk, m) \rightarrow (h, r)$: The probabilistic hash algorithm takes the public key pk and a message $m \in \mathcal{M}$ as input. It then produces an output of 1 if the tuple (h, r) is considered valid according to the algorithm's criteria. If the tuple is not valid, the output will be 0.
- Verify$(pk, m, h, r) \rightarrow \{0, 1\}$: The deterministic verification algorithm takes the public key pk, message m, hash value h, and randomness value r as input. It then determines whether the tuple (h, r) is valid according to the defined criteria. If the tuple is valid, the algorithm outputs 1. Otherwise, if the tuple is not valid, the output will be 0.
- Adapt$(sk, m, m', h, r) \rightarrow r'$: The deterministic adaptation algorithm takes the secret key sk, message $m \in \mathcal{M}$, hash value h, and randomness value r as input. It then generates an adapted randomness value r' as output.

4. System Overview

In this section, we present the system model, brief definition, and security model of PFRB.

4.1. System Model

As shown in Figure 1, the PFRB (Privacy-Preserving Redactable Blockchain) system involves four types of entities:

- Authorities: The authorities are all trusted. One of them initializes the system, and they can all generate attribute-value pairs.
- Transaction Owner: The transaction owner is also trusted and wants to place a deal or some data on the blockchain. They hash the data and attempt to add the transaction to the blockchain.
- Transaction Modifier: The transaction modifier is a user who wants to modify a transaction in the blockchain. They retrieve the attribute-value pairs and try to match the transaction to modify it.
- Blockchain Participants: The blockchain participants are users of the redactable blockchain. They verify each transaction published by the transaction owners or the transaction modifiers.

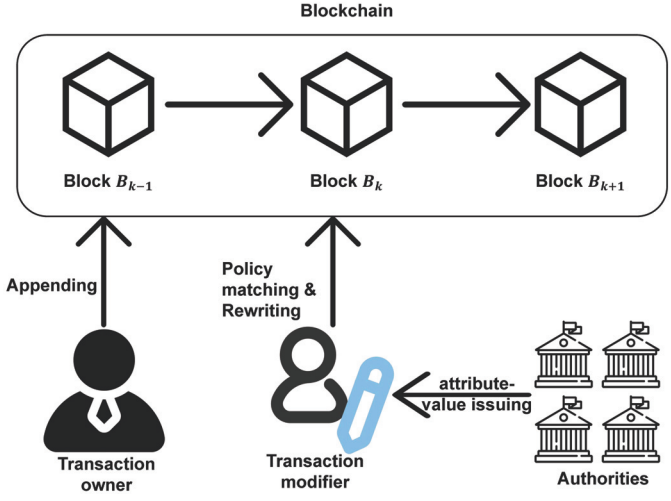


Figure 1. PFRB system model.

4.2. Definition of PFRB

We next provide a brief definition of PFRB and summarize the notations in Table 2.

- Setup $(1^{\lambda}) \rightarrow (pp_s, pk_s, sk_s)$: Given a security parameter, the Setup algorithm outputs a public parameter pp_s, public key pk_s, and secret key sk_s. Then, the authority publishes pp_s and pk_s to all users.
- RKGen $(msk, \rho) \rightarrow (dk_\rho, \{\Delta_{i,R_j}\})$: The RKGen algorithm takes msk and ρ as input, where ρ is the attribute set of the modifier. The algorithm outputs the decryption key dk_ρ and the Lagrange coefficients of ρ, $\{\Delta_{i,R_j}\}$.
- ModSetup $(sk_s, id) \rightarrow (sk_{id}, \sigma_{id})$: The ModSetup algorithm takes the secret key sk_s and the global identifier id as input. It outputs the modifier's secret key sk_{id} and the modifier's signature σ_{id}.
- AuthSetup $(\theta) \rightarrow (pk_\theta, sk_\theta)$: The AuthSetup algorithm takes the authority θ as input and outputs the authority's public key pk_θ and secret key sk_θ.
- ModKeyGen $(pk_s, id, \sigma_{id}, sk_\theta, A) \rightarrow sk_{id,A} / \perp$: The ModKeyGen algorithm takes the public key pk_s, the modifier's global identifier id, the modifier's signature σ_{id}, the authority's secret key sk_θ, and an attribute A as input. It generates the secret key $sk_{id,A}$ for the modifier's attribute A if the request is legal; otherwise, it outputs nothing.
- Hash $(pk_s, \{pk_\theta\}, m, R) \rightarrow (pk^{etd}, h, r, c)$: The Hash algorithm is designed to take the following inputs: the public key pk_s, a group of authorities' public keys pk_θ, the message

m to be encrypted, and the policy R of the target receiver. It generates four outputs: a public key pk^{etd} (a public component of the ephemeral trapdoor), a hash value h, a randomness value r, and a ciphertext c. The ciphertext c plays the crucial role of securely sealing the secret component sk^{etd}, guaranteeing its confidentiality.

- Verify $(pk_s, pk^{etd}, m, h, r) \rightarrow \{0,1\}$: The Verify algorithm can be executed by any entity within the system. It accepts the following inputs: the public key pk_s, the public component pk^{etd} of the ephemeral trapdoor, the message $m \in M$, the hash value h, and the randomness value r. The algorithm then determines whether the tuple (h, r) is valid according to its defined criteria. If the tuple is deemed valid, the algorithm outputs 1. However, if the tuple is found to be invalid, the output will be 0.
- Adapt $(sk_{id}, \{sk_{id,A}\}, c, m, m', h, r) \rightarrow \mathrm{r}'$: The Adapt algorithm is executed by the transaction modifier. It takes inputs such as the secret component sk^{etd}, a set of secret keys $sk_{id,A}$, the ciphertext c, messages m and m', the hash value h, and the randomness value r. The output of the algorithm is a new randomness value r'.

Table 2. Notations used in PFRB.

Notations	Descriptions	Notations	Descriptions
(p, G, G_0, e, g)	description of bilinear group	λ	security parameter
(n, p, q, e, d)	description of RSA parameter	$H_{ID,G}$, $H_{U,G}$, H_0, H_{b,Z_p}	description of hash function
$e(g, g)$	description of the bilinear function	pp_s	public parameter
pk_s	public key	sk_s	secret key
msk	master secret key	ρ	an attribute set of modifier
dp_ρ	the decryption key	Δ_{i,R_j}	Lagrange coefficient of ρ
id	global identifier	sk_{id}	modifier's secret key
σ	modifier's signature	θ	an authority
(pk_θ, sk_θ)	authority's own public key the secret key	A	an attribute
$sk_{id,A}$	the secret key of the modifier's attribute A	sk^{etd}	the secret component of the ephemeral trapdoor
R	the target receiver of the transaction	pk^{etd}	public component of an ephemeral trapdoor

4.3. Security Model

In our scheme, we assume that all authorities and the data owner are trusted entities, and communications between them are secure. The transaction owner generates mutable transactions honestly, and the authorities preserve the secret key honestly. However, other entities, such as chain participants, can act as adversaries and collaborate to launch the chosen-plaintext attack. The security of PFRB is defined as the indistinguishability and the collision resistance under the chosen-plaintext attack in the random oracle model as follows.

- Setup: The challenger runs the Setup algorithm and shares the public parameters PK with the adversary.
- Phase 1: The challenger allows the adversary to request private keys from the encryption oracle $\mathcal{O}_E$ by their attributes $S_1, \ldots, S_{q1}$.
- Challenge: The adversary selects and uploads two messages, M_0 and M_1, of equal length. The adversary also presents a challenge access structure, denoted as A, which none of the previously generated attribute sets can satisfy. The challenger randomly chooses a coin flip outcome, encrypts either M_0 or M_1 under the challenge access structure A, and provides the resulting ciphertext CT^* to the adversary.
- Phase 2: Phase 1 is repeated, but with the additional constraint that none of the sets of attributes $S_{q1+1}, S_{q1+2}, \ldots, S_q$ satisfy the access structure associated with the given challenge. This restriction ensures that the adversary cannot find any new sets of attributes that fulfill the challenge access structure.
- Guess: Based on the above experiment, the adversary outputs a guess, b_0, of b.

We say that the adversary $\mathcal{A}$ wins the above game if the guess $b^{'}$ equals b. Specifically, PFRB is secure against the chosen-plaintext attack (CPA) if any probabilistic polynomial-time adversary $\mathcal{A}$ only has a negligible advantage to win the game as follows.

$$\mathrm{Adv}_{\mathcal{A}}^{CPA}(\lambda) = |\Pr[b' = b|\mathcal{A}^{\mathcal{O}_E}(M_0, M_1, CT^*)]|.$$

5. Proposed Scheme

In this section, we present detailed construction of PFRB. The workflow of PFRB is shown in Figure 2. There are eight algorithms: Setup, RkGen, ModSetup, AuthSetup, ModKeyGen, Hash, Verify and Adapt.

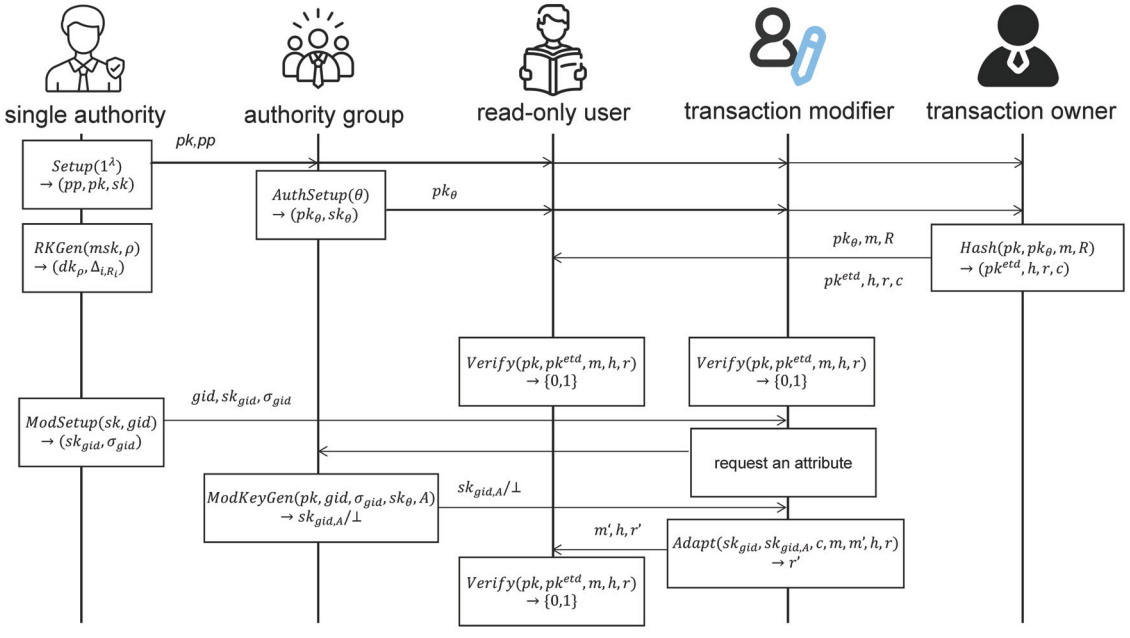


Figure 2. Detailed workflow in PFRB.

- Setup $(1^{\lambda}) \rightarrow (pp_s, pk_s, sk_s, mpk, msk)$:
  - Given a security parameter λ, generate the bilinear group description (p, G, G_0, e, g).
  - Running $RSAKeyGen(1^{\lambda})$, and then get the first set of RSA parameter (n_0, p_0, q_0, e_0, d_0)
  - Choose three random exponents $\alpha \in Z_p, \beta, \gamma \in G$, a mapping function, and four hash functions as follows.

$$H_{b,Z_p} : \{0,1\} \rightarrow Z_p,$$
$$H_{ID,G} : \{0,1\} \times ID \rightarrow G,$$
$$H_0 : \{0,1\}^* \rightarrow Z_n^*,$$
$$H_{U,G} : U \rightarrow G,$$
$$F_m : U \rightarrow U_\theta.$$

  - Set $Y_\beta = e(g,g)^\beta, Y_\gamma = e(g,g)^\gamma$. Choose n, d and $2n+2$ random values $t_1, \ldots, t_{n+1}$, $d_1, \ldots, d_{n+1} \in Z_p$ and set $T_i = g^{t_i}, D_i = g^{d_i}$ for each i from 1 to $n+1$.
  - Choose a symmetric encoding method $F_{en} : \{0,1\}^* \rightarrow G_0$, and the corresponding decoding method $F_{en}^{-1} : G_0 \rightarrow \{0,1\}^*$.

- Calculate the system public parameter pp_s, master secret key msk, master public key mpk, system public key pk_s, and system secret key sk_s as:

$$pp_s = (1^\lambda, p, G, G_0, e, g, H_{b,Z_p}, H_{ID,G}, H_0, H_{U,G}, F_m, F_{en}, F_{en}^{-1}),$$
$$mpk = (Y_\beta, Y_\gamma, \{T_i\}, \{D_i\}), msk = (\beta, \gamma),$$
$$pk_s = (H_0, n_0, e_0, g^\alpha), sk_s = (d_0, \alpha).$$

  The function $H_{ID,G}$ receives a bit $b \in \{0,1\}$ and an input $id \in ID$, producing a hash value in G. In our system, the attributes are named using the format "[attribute-id]@[authority-id]". To extract only the authority ID from the attribute name, we use a mapping function called F_m. This function helps us retrieve the authority ID while ignoring the attribute ID.

- RKGen $(msk, \rho) \rightarrow (dk_\rho, \{\Delta_{i,R_j}\})$: ρ is an attribute set of modifier.
  - Randomly generate a d-1 degree polynomial q_2 with $q_2(0) = \gamma$
  - Calculate the values of $dk_i = g^{\frac{q_2(i)}{d_i}}$ for each i in ρ, and store them as $dk_\rho = \{dk_i\}$.
  - Compute a set of Lagrange coefficients Δ_{i,R_j}, where each Lagrange coefficient might satisfies the policy, and R_j belongs to ρ.
- ModSetup $(sk_s, id) \rightarrow (sk_{id}, \sigma_{id})$:
  - The authorities compute their secret key $sk_{id} = d_0$ and their signature $\sigma_{id} = H_{ID,G}(1, id)^\alpha$. Return sk_{id} and σ_{id}.
- AuthSetup $(\theta) \rightarrow (pk_\theta, sk_\theta)$:
  - The authority Chooses two random values $a_\theta, b_\theta \in Z_p$. Calculate the secret key sk_θ as (a_θ, b_θ) and public key pk_θ as $(e(g,g)^{a_\theta}, g^{b_\theta})$. Return pk_θ and sk_θ.
- ModKeyGen $(pk_s, id, \sigma_{id}, sk_\theta, A) \rightarrow sk_{id,A} / \perp$
  - If $e(g, \sigma) \neq e(g^\alpha, H_{ID,G}(1, id))$, return $\perp$.
  - Generate a random value $t \in Z_p$. Compute $sk_{id,A,0} = g^{a_\theta} H_{ID,G}(0, id)^{b_\theta} H_{U,G}(A)^t$ and $sk_{id,A,1} = g^t$.
  - Output the corresponding secret key of the attribute A as follows.

$$sk_{id,A} = (id, A, sk_{id,A,0}, sk_{id,A,1}).$$

- $Hash(pk_s, \{pk_\theta\}, m, R) \rightarrow (pk^{etd}, h, r, c)$: R is a policy of target receiver.
  - Run the RSA key generator RSAKeyGen(1^λ) and generate another set of RSA parameter$(n_1, p_1, q_1, e_1, d_1)$.
  - Choose $H_1 : \{0,1\}^* \rightarrow Z_{n_1}^*, r_0 \in Z_{n_0}^*$ and $r_1 \in Z_{n_1}^*$. Then compute two hash values $h_0 = H_0(m) r_0^{e_0}$ and $h_1 = H_1(m) r_1^{e_1}$
  - Choose a random sequence $r_t \in \{0,1\}^\lambda$. Then, run $SE.KeyGen(1^\lambda)$ to generate a key k. Subsequently, utilize k to generate a ciphertext $c_t \leftarrow SE.Enc(k, d_1)$.
  - Run the symmetric encryption algorithm and compute $k_c \leftarrow F_{en}(k, r_t)$ and $z \leftarrow H_{b,Z_p}(r_t, G_n(x))$. Calculate $c_0 = k_c e(g,g)^z$. Choose a set of random numbers $u_r, t_r, r_{1,r}, r_{2,r} \in Z_p$. Compute

$$T_r = g^{t_r}, U_r = g^{u_r}, R_{1,r} = g^{r_{1,r}}, R_{2,r} = g^{r_{2,r}}.$$

    Compute $P_{i,r} = D_{i,r}^{r_{1,r}}$, $E_{i,r} = T_{i,r}^{r_{2,r}}$ where $i \in R$.
  - Compute

$$K_{1,r} = e(R_{1,r}, T_r) Y_\gamma^{r_{1,r}},$$
$$K_{2,r} = e(R_{2,r}, U_r) Y_\beta^{r_{2,r}},$$
$$V_r = z \oplus H(e(R_{1,r}, T_r)) \oplus H(e(R_{2,r}, U_r)).$$

– Return the public key pk^{etd}, random value r, hash value h, and ciphertext c as follows.

$$pk^{etd} = (H_1, n_1, e_1), h = (h_0, h_1), r = (r_0, r_1),$$
$$c = (c_0, T_r, U_r, \{P_{i,r}\}, \{E_{i,r}\}, W_r, \{K_{1,r}\}, \{K_{2,r}\}, V_r).$$

- Verify $(pk_s, pk^{etd}, m, h, r) \rightarrow \{0,1\}$:Parse $h = (h_0, h_1)$ and $r = (r_0, r_1)$
  - Return 1 if $h_0 = H_0(m) r_0^{e_0} \ mod \ n_0$ and $h_1 = H_1(m) r_1^{e_1} \ mod \ n_1$; otherwise, return 0;
- Adapt$(sk_{id}, \{sk_{id,A}\}, c, m, m', h, r) \rightarrow r'$:
  - If m equals m', then output $r' = r$.
  - Enumerate all sets of combinations of attributes and compute

$$g_{T,1} = K_1 / \prod_{i,R_j} e(P_i, dk_i)^{\Delta_{i,R_j}(0)}, with \ R_j \subset R, |R_j| \geq d,$$
$$g_{T,2} = K_2 / e(\prod_{i,S} E_{i,r}, W_r),$$
$$z = V \oplus H(g_{T,1}) \oplus H(g_{T,2}),$$
$$k_c = c_0 / e(g,g)^z, (k, r_t) \leftarrow F_{en}^{-1}(k_c).$$

  - Run $d_1 \leftarrow SE.Dec(c_t, k)$. Compute

$$r'_0 = (h_0 (H_0(m')^{-1}))^{d_0} mod \ n_0,$$
$$r'_1 = (h_1 (H_1(m')^{-1}))^{d_1} mod \ n_1.$$

  - Return the randomness $r' = (r'_0, r'_1)$.

6. Theoretical Analysis

In this section, we theoretically analyze the correctness and security of PFRB. Then, we discuss some promising applications of PFRB.

6.1. Correctness Analysis

In this section, we will provide detailed proof of the correctness of the proposed scheme in this paper. The scheme presented in this paper builds upon Ma et al's scheme (i.e., RBDS22) [18] while incorporating additional improvements. It is worth noting that if RBDS22 is correct and the calculations of $K_{1,r}$ and $K_{2,r}$ in this scheme are correct, then the proposed scheme in this paper is also correct. We will now present the specific proof of the correctness of $K_{1,r}$ and $K_{2,r}$ as follows:

In the Hash part, we have $K_{1,r} = e(R_{1,r}, T_r) Y_\gamma^{r_{1,r}}$,$K_{2,r} = e(R_{2,r}, U_r) Y_\beta^{r_{2,r}}$. $K_{1,r}$ can be transformed as follows:

$$K_{1,r} = e(R_{1,r}, T_r) Y_\gamma^{r_{1,r}} = e(R_{1,r}, T_r) e(g,g)^{\gamma, r_{1,r}}.$$

Similarly, $K_{2,r}$ can be transformed as follows:

$$K_{2,r} = e(R_{2,r}, U_r) Y_\beta^{r_{2,r}} = e(R_{2,r}, U_r) e(g,g)^{\beta, r_{2,r}}.$$

In the Adapt part, we have: $g_{T,1} = K_{1,r} / \prod_{i,R_j} e(P_{i,r}, dk_i)^{\Delta i, R_j(0)}$, $g_{T,2} = K_{2,r} / e\left(\prod_{i,s} E_{i,r}, W_r\right)$.

Transform these two parts as follows:

$$K_{1,r} = g_{T,1} \cdot \prod_{i,R_j} e(P_{i,r}, dki)^{\Delta i,R_j(0)} = g_{T,1} \cdot \prod_{i,R_j} e(P_{i,r}, dki^{\Delta i,R_j(0)})$$
$$= g_{T,1} \cdot \prod_{i,R_j} e(D_{i,r}^{r_{1,r}}, dki^{\Delta i,R_j(0)}) = g_{T,1} \cdot \prod_{i,R_j} e(g^{d_{i,r}}, dki^{\Delta i,R_j(0)})$$
$$= g_{T,1} \cdot \prod_{i,R_j} e(g^{d_{i,r}}, g^{\frac{q_2(i)}{d_i}\Delta_{i,R_j}(0)}) = g_{T,1} \cdot e(g,g)^{\gamma, r_{1,r}}.$$

Similarly, we have $K_{2,r} = g_{T,2} \cdot e(g,g)^{\beta, r_{2,r}}$.

When assuming that the two instances of $K_{1,r}$ and $K_{2,r}$ are equal, we can deduce that $e(R_{1,r}, T_r) = g_{T,1}$ and $e(R_{2,r}, U_r) = g_{T,2}$. In this case, the modifier can obtain the desired decryption key z from V as follows: $z = V \oplus H(g_{T,1}) \oplus H(g_{T,2})$.

From the above, it is evident that when the calculations of $K_{1,r}$ and $K_{2,r}$ are correct, the modifier can successfully recover the key z, thereby reducing the correctness of this paper's scheme to the correctness of RBDS22. As RBDS22 is known to be correct, it follows that this paper's scheme is also correct.

6.2. Security Analysis

Theorem 1. *PFRB is secure against the chosen-plaintext attack.*

Proof. To prove the security of the encryption method against the chosen-plaintext attack, we need to demonstrate that the adversary cannot distinguish the generated ciphertext, even if it can choose a set of plaintexts and observe their encryption forms.

Considering the message need to be encrypted m, the challenger first generates a randomness r_t. Calculates the K_c by running $F_{en}(m, r_t)$. then let $z = H_{b,Z_p}(r_t, G_n(x))$, where $G_n(x)$ is an invariant polynomial which $\mathcal{A}$ cannot get. Computing $c_0 = k_c e(g,g)^z$. Return c_0 to the adversary $\mathcal{A}$.

Recall the above encryption process, we know that $e()$, g, F_{en}, $H_{b,Z_p}()$, λ are fixed. message m are values chosen by $\mathcal{A}$, and r_t are one-time random value chosen by $\mathcal{C}$. It is obvious that C_0 is a random value associated with r_t, which means although the adversary $\mathcal{A}$ can continuously submit requests to $\mathcal{C}$ and receive their corresponding ciphertexts, the ciphertexts appear random to $\mathcal{A}$. Therefore, $\mathcal{A}$ is incapable of distinguishing plaintexts from ciphertexts and it can only randomly guess b' from 0,1. Hence, we can know that

$$|\mathsf{Adv}_{\mathcal{A}}[b' = b] - \frac{1}{2}| \leq \epsilon,$$

where ϵ is negligible. Thus, PFRB is of CPA security, and Theorem 1 is proven. □

Also, PFRB can resist user collusion attacks if the soundness of Newton's interpolation formula-based secret sharing is hard. As shown in the detailed construction, PFRB relies on Newton's interpolation formula-based secret sharing to achieve controllable redaction. Specifically, Newton's interpolation formula-based secret sharing is used to specify policies and generate secret keys. namely, the user collusion resistance of PFRB can be reduced to the soundness of Newton's interpolation formula-based secret sharing. As proven in prior works, Newton's interpolation formula-based secret sharing holds soundness. Thus, PFRB is secure against the user collusion attack.

6.3. Application Discussion

The emergence of decentralized policy-hidden fine-grained redactable blockchain (PFRB) technology has opened new avenues for secure data management and privacy preservation. In this section, we explore the potential benefits of integrating PFRB with smart healthcare, smart industry, and artificial intelligence (AI) systems.

6.3.1. Application of PFRB in Smart Healthcare

Currently, with policy privacy, PFRB can be widely applied in smart healthcare applications to collect and manage medical data. For instance, PFRB ensures that nobody can infer users' private information from users' redaction policies. This benefit can impel users' enthusiasm for using portable devices (e.g., mobile phones and smart swatches) to collect data associated with their health condition, such as heart rate and temperature [29]. Clearly, these data hold high value for medical data analytic applications.

6.3.2. Application of PFRB in Smart Industry

In the industry environment, due to the harsh environment and heavy data collection tasks, IoT devices have been deployed to replace humans' work and securely transmit their data through blockchains [30]. In this case, policies from IoT devices usually contain large amounts of commercially sensitive data, such as data types and factory addresses [31]. Protecting this information is crucial to increase the wide implementation of blockchain-based IoT systems in the smart industry. Matching these practical requirements, PFRB is a promising solution to blockchain-based IoT systems in the smart industry.

6.3.3. Application of PFRB in AI

Recently, some cases have been proposed to prove the practical feasibility of combining the advantages of blockchains, AI, and IoT systems, such as swarm learning and machine/deep-learning-based IoT systems [32]. Specifically, IoT systems are responsible for collecting data for AI models, and blockchains are responsible for managing data and training AI models. The trained models can then provide rich services for IoT devices, such as fault detection and inference [33]. Similarly, in these promising applications, redaction policies from IoT devices contain much private information of IoT owners, such as policies for personal temperatures in medical analytics containing users' health conditions. PRFB addresses the policy leakage problem and can further impel the development of combining blockchains, IoT systems, and AI in real-world applications.

7. Performance Evaluation

In this section, we conduct experiments on the FISCO blockchain platform to evaluate the practical efficiency of PFRB.

7.1. Experimental Settings

Configuration: The experiment was conducted on a personal computer running Windows 10 (x64) with an Intel i7 8550U processor clocked at 1.80GHz and 8GB of memory. The implementation was done using the JPBC library in Java 8, and the MNT224 curve, known for its type-III properties and offering a 96-bit security level, was selected for pairing operations. Additionally, a 2048-bit RSA group was used for the Chameleon hash, providing a security level of 112 bits. Our scheme was deployed on the FISCO BCOS platform, which is an open-source platform customized for the financial industry. It is built upon the BCOS platform and incorporates module upgrades and functionality customization. The "Arbitration Chain" in FISCO leverages blockchain decentralization, tamper resistance, and trustworthiness. Real-time preserved data are securely stored on the blockchain using distributed data storage and encryption algorithms, ensuring the authenticity, legality, and relevance of the evidence.

Parameter design: The data owner randomly selects a news headline and uploads it to the service node as a transaction. In our experiment, we assume that each user can choose up to 100 attributes, such as gender, age, region, education level, occupation, etc., for access permissions. The access control policies for transactions are determined based on information such as the time and location of the news, and up to 100 policies can be set. It is important to note that the traditional scheme lacks policy protection and is vulnerable to security issues such as privacy leaks. This paper presents the experimental perfor-

mance of key generation, hashing, and adaptation and chooses recent related works [16,18] as comparisons.

Dataset: The implementation of our proposed solution utilizes the MNH9 dataset, which is derived from a real-world open dataset provided by the Australian Broadcasting Corporation. The MNH9 dataset consists of millions of news headlines.

7.2. Experimental Results

In this paper, we present a superior solution that excels in key generation, hashing, verification, and adaptation compared to traditional methods.

Figure 3 illustrates a significant improvement in the key generation aspect. For all three approaches, the required time increases linearly with the number of attributes, and both our proposed scheme and RBDS22 outperform the traditional approach by requiring considerably less time. This noteworthy enhancement is primarily attributed to the higher efficiency of our scheme, which leverages traditional linear secret sharing matrices at the underlying level, resulting in a more stable and efficient process.

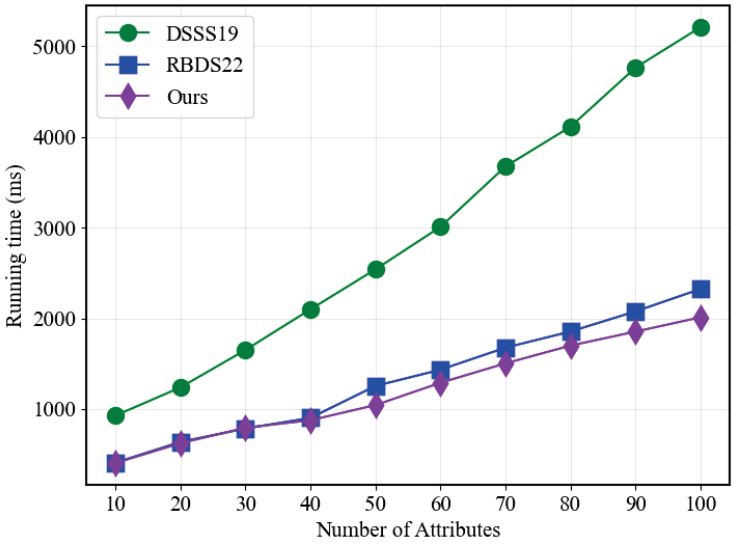


Figure 3. Key generation performance.

Moving on to Figure 4, the most noticeable efficiency improvement of the proposed solution is in the hashing part when compared to the traditional scheme. The traditional approach's required time increases exponentially with the number of policies, while the proposed solution and RBDS22 demonstrate linear growth. This advantage can be attributed to the use of polynomial functions generated from point values, which have proven to be significantly more efficient than using traditional linear secret sharing matrices schemes.

Figure 5 reveals that the adaptation algorithm part, which employs Newton's interpolation to reconstruct polynomials, presents challenges due to the enumeration of attribute-value pairs owned by the user. While the required time for both the traditional scheme and the proposed solution increases exponentially with the number of policies, the proposed scheme still outperforms the traditional approach significantly. Moreover, RBDS22 shows linear growth in the required time, and the proposed solution performs better than RBDS22, especially when the number of policies is low.

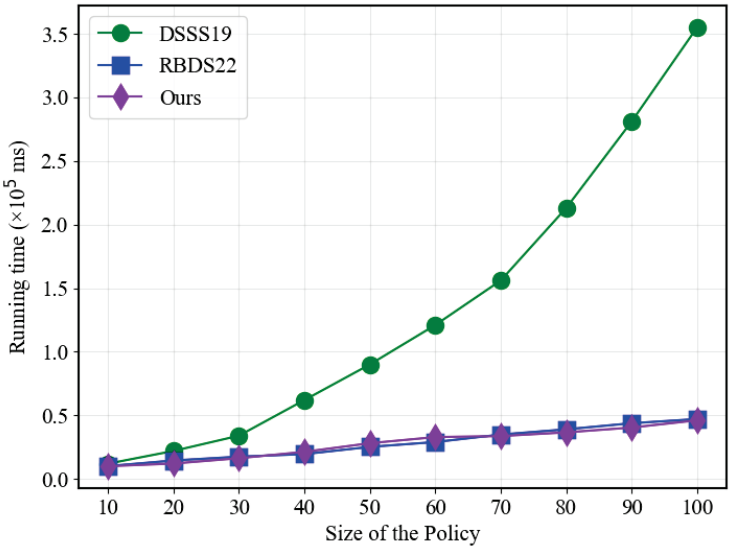


Figure 4. Hash performance with different policy sizes.

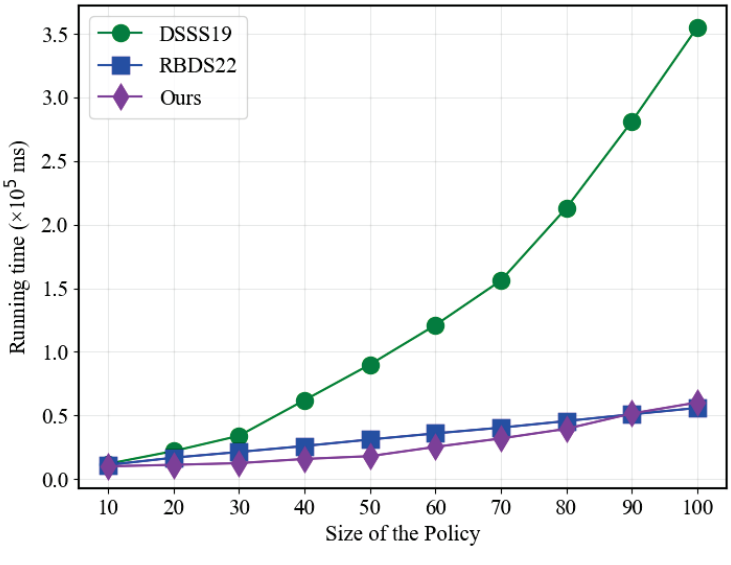


Figure 5. Adaption performance with different policy sizes.

Figure 6 demonstrates that in the Verify algorithm part, the response time of all three schemes increases linearly with the number of requests, with no significant difference in the time used. This similarity is mainly due to the similarity in the calculation operations performed in the Verify algorithm part of all three schemes. Therefore, the time used by all three schemes is similar in this regard.

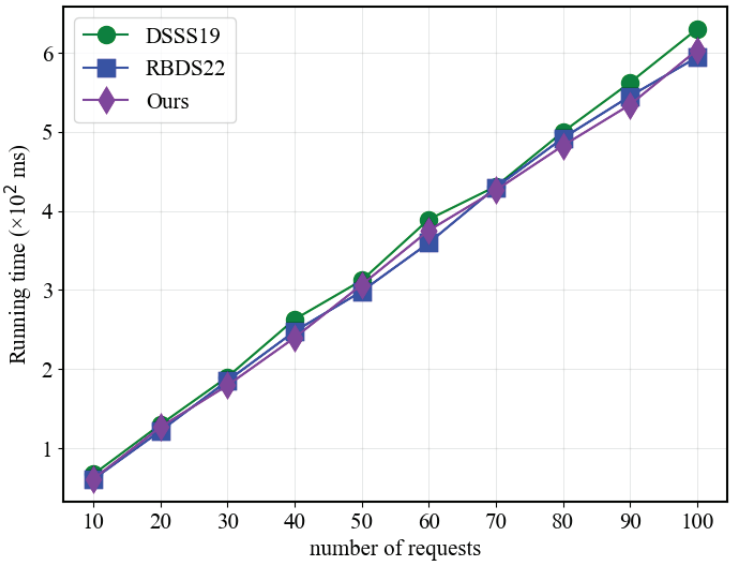


Figure 6. Verify performance with different request numbers.

Figures 7 and 8 display that when the number of policies is fixed, the response time of all three schemes in the hashing and adaptation algorithm parts increases linearly with the number of requests. However, the proposed scheme requires significantly less time than the traditional scheme and slightly less time than RBDS22 when the number of policies is relatively small. This advantage can be attributed to the proposed scheme's utilization of MA-ABE and Newton's interpolation, which enables it to achieve higher efficiency compared to traditional ABE and linear secret sharing matrices.

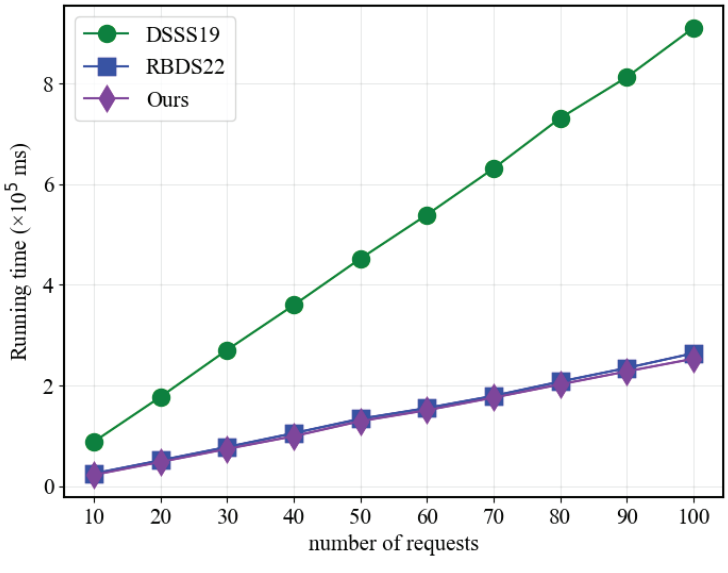


Figure 7. Hash performance with different user numbers.

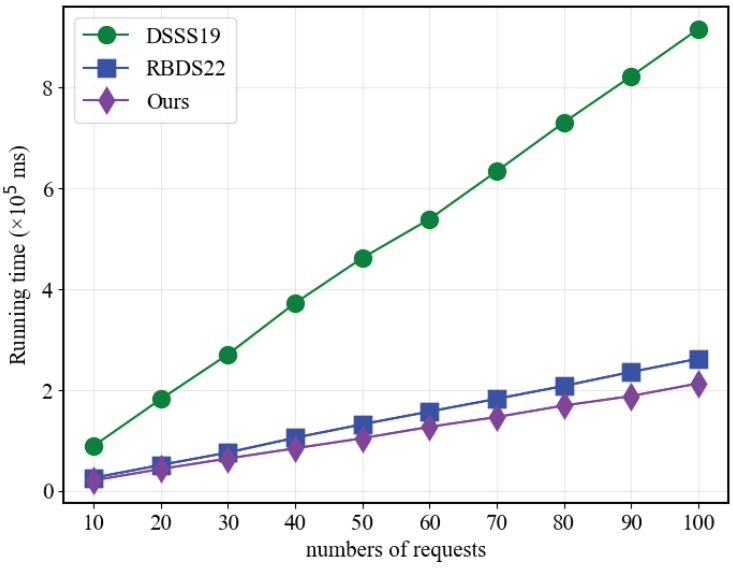


Figure 8. Adaption performance with different user numbers.

8. Conclusions

In this paper, we propose a policy-hidden fine-grained redactable blockchain (named PFRB) in decentralized blockchain-based IoT systems. Considering existing redactable blockchains, PFRB supports decentralized settings and fine-grained chameleon hash-based redaction. In addition, PFRB ensures that apart from successful policy matching, anyone cannot infer users' policy contents and execute any valid redaction. PFRB draws inspiration from Newton's interpolation formula-based secret sharing to convert policy contents into polynomial parameters. PFRB then utilizes multi-authority attribute-based encryption to hide these parameters further. Theoretical analysis proves that PRFB is secure against the chosen-plaintext attack. Extensive experiments on the FISCO blockchain platform and IoT devices show that PFRB achieves competitive efficiency over current redactable blockchains. For future work, we will focus on providing more comprehensive privacy protection (e.g., data privacy) and richer functionalities (such as accountability and revocation) in decentralized IoT systems. In addition, we will also focus on achieving richer data analytic mechanisms by combining various machine learning schemes.

Author Contributions: Conceptualization, H.G.; Formal analysis, M.Z.; Funding acquisition, C.Z. and J.X.; Methodology, H.G. and X.T.; Project administration, L.Z.; Software, X.T.; Supervision, T.W., C.Z., J.X. and L.Z.; Validation, T.W.; Writing—original draft, H.G., X.T. and M.Z.; Writing—review and editing, H.G., X.T., M.Z., T.W. and C.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by National Natural Science Foundation of China (Grant Nos. 62202051, 62102027, and 62172042), China Postdoctoral Science Foundation (Grant Nos. 2021M700435 and 2021TQ0042), Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies (Grant No. 2022B1212010005), Shandong Provincial Key Research and Development Program (Grant No. 2021CXGC010106), Major Scientific and Technological Innovation Projects of Shandong Province (2020CXGC010116), and Beijing Institute of Technology Research Fund Program for Young Scholars.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chunka, C.; Banerjee, S.; Sachin Kumar, G. A secure communication using multifactor authentication and key agreement techniques in internet of medical things for COVID-19 patients. *Concurr. Comput. Pract. Exp.* **2023**, *35*, e7602. [CrossRef]
2. Ahmad, U.; Chaudhary, J.; Ahmad, M.; Naz, A.A. Survey on internet of things (IoT) for different industry environments. *Ann. Emerg. Technol. Comput. (AETiC)* **2019**, *3*, 28–43. [CrossRef]
3. Sinha, B.B.; Dhanalakshmi, R. Recent advancements and challenges of Internet of Things in smart agriculture: A survey. *Future Gener. Comput. Syst.* **2022**, *126*, 169–184. [CrossRef]
4. Liu, Y.; Hao, X.; Ren, W.; Xiong, R.; Zhu, T.; Choo, K.K.R.; Min, G. A Blockchain-Based Decentralized, Fair and Authenticated Information Sharing Scheme in Zero Trust Internet-of-Things. *IEEE Trans. Comput.* **2022**, *72*, 501–512. [CrossRef]
5. Kouicem, D.E.; Imine, Y.; Bouabdallah, A.; Lakhlef, H. Decentralized Blockchain-Based Trust Management Protocol for the Internet of Things. *IEEE Trans. Dependable Secur. Comput.* **2022**, *19*, 1292–1306. [CrossRef]
6. Mathur, S.; Kalla, A.; Gür, G.; Bohra, M.; Liyanage, M. A Survey on Role of Blockchain for IoT: Applications and Technical Aspects. *Comput. Netw.* **2023**, *227*, 109726. [CrossRef]
7. Hao, X.; Ren, W.; Fei, Y.; Zhu, T.; Choo, K.K.R. A blockchain-based cross-domain and autonomous access control scheme for internet of things. *IEEE Trans. Serv. Comput.* **2022**, *16*, 773–786. [CrossRef]
8. Bothra, P.; Karmakar, R.; Bhattacharya, S.; De, S. How can applications of blockchain and artificial intelligence improve performance of Internet of Things?–A survey. *Comput. Netw.* **2023**, *224*, 109634. [CrossRef]
9. Zhang, C.; Zhao, M.; Zhu, L.; Zhang, W.; Wu, T.; Ni, J. FRUIT: A blockchain-based efficient and privacy-preserving quality-aware incentive scheme. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 3343–3357. [CrossRef]
10. Moonie, H. Man's "Right to Be Forgotten" Case Stalls After He Is Found on the Bitcoin Blockchain. 2016. Available online: https://medium.com/@hankmoonie/mans-right-to-beforgotten-case-stalls-after-he-is-found-on-the-bitcoin-blockchain-1a32c4fc0963 (accessed on 31 July 2023).
11. Tian, G.; Wei, J.; Kutylowski, M.; Susilo, W.; Huang, X.; Chen, X. VRBC: A Verifiable Redactable Blockchain With Efficient Query and Integrity Auditing. *IEEE Trans. Comput.* **2023**, *72*, 1928–1942. [CrossRef]
12. Shen, J.; Chen, X.; Liu, Z.; Susilo, W. Verifiable and Redactable Blockchains With Fully Editing Operations. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 3787–3802. [CrossRef]
13. Ateniese, G.; Magri, B.; Venturi, D.; Andrade, E. Redactable blockchain–or–rewriting history in bitcoin and friends. In Proceedings of the 2017 IEEE European Symposium on Security and Privacy (EuroS&P), Paris, France, 26–28 April 2017; pp. 111–126.
14. Deuber, D.; Magri, B.; Thyagarajan, S.A.K. Redactable blockchain in the permissionless setting. In Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), Francisco, CA, USA, 19–23 May 2019; pp. 124–138.
15. Xu, S.; Ning, J.; Ma, J.; Huang, X.; Deng, R.H. K-time modifiable and epoch-based redactable blockchain. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 4507–4520. [CrossRef]
16. Derler, D.; Samelin, K.; Slamanig, D.; Striecks, C. Fine-Grained and Controlled Rewriting in Blockchains: Chameleon-Hashing Gone Attribute-Based. *Cryptol. ePrint Arch.* **2019**.
17. Jia, M.; Chen, J.; He, K.; Du, R.; Zheng, L.; Lai, M.; Wang, D.; Liu, F. Redactable Blockchain From Decentralized Chameleon Hash Functions. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 2771–2783. [CrossRef]
18. Ma, J.; Xu, S.; Ning, J.; Huang, X.; Deng, R.H. Redactable blockchain in decentralized setting. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 1227–1242. [CrossRef]
19. Xu, S.; Huang, X.; Yuan, J.; Li, Y.; Deng, R.H. Accountable and Fine-Grained Controllable Rewriting in Blockchains. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 101–116. [CrossRef]
20. Zhang, C.; Zhao, M.; Zhu, L.; Wu, T.; Liu, X. Enabling Efficient and Strong Privacy-Preserving Truth Discovery in Mobile Crowdsensing. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 3569–3581. [CrossRef]
21. Hu, C.; Zhang, C.; Lei, D.; Wu, T.; Liu, X.; Zhu, L. Achieving Privacy-Preserving and Verifiable Support Vector Machine Training in the Cloud. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 3476–3491. [CrossRef]
22. Zhang, C.; Zhao, M.; Xu, Y.; Wu, T.; Li, Y.; Zhu, L.; Wang, H. Achieving fuzzy matching data sharing for secure cloud-edge communication. *China Commun.* **2022**, *19*, 257–276. [CrossRef]
23. Zhang, C.; Hu, C.; Wu, T.; Zhu, L.; Liu, X. Achieving Efficient and Privacy-Preserving Neural Network Training and Prediction in Cloud Environments. In *IEEE Transactions on Dependable and Secure Computing*; Early Access; IEEE: New York City, NY, USA, 2022.
24. Tian, Y.; Li, N.; Li, Y.; Szalachowski, P.; Zhou, J. Policy-based chameleon hash for blockchain rewriting with black-box accountability. In Proceedings of the Annual Computer Security Applications Conference, Austin, TX, USA, 7–11 December 2020; pp. 813–828.
25. Panwar, G.; Vishwanathan, R.; Misra, S. ReTRACe: Revocable and traceable blockchain rewrites using attribute-based cryptosystems. In Proceedings of the 26th ACM Symposium on Access Control Models and Technologies, Virtual, 16–18 June 2021; pp. 103–114.
26. Jia, Y.; Sun, S.F.; Zhang, Y.; Liu, Z.; Gu, D. Redactable blockchain supporting supervision and self-management. In Proceedings of the Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security, Virtual, 7–11 June 2021; pp. 844–858.

27. Xu, S.; Ning, J.; Ma, J.; Xu, G.; Yuan, J.; Deng, R.H. Revocable policy-based chameleon hash. In Proceedings of the Computer Security–ESORICS 2021: 26th European Symposium on Research in Computer Security, Darmstadt, Germany, 4–8 October 2021; pp. 327–347.
28. Chase, M. Multi-authority attribute based encryption. In Proceedings of the Theory of Cryptography: 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, 21–24 February 2007; pp. 515–534.
29. Chae, Y.; Wang, S.; Kim, S.M. Exploiting WiFi Guard Band for Safeguarded ZigBee. In Proceedings of the Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems, SenSys, Shenzhen, China, 4–7 November 2018; pp. 172–184.
30. Wang, S.; Kim, S.M.; He, T. Symbol-Level Cross-Technology Communication via Payload Encoding. In Proceedings of the 38th IEEE International Conference on Distributed Computing Systems, Vienna, Austria, 2–5 July 2018; pp. 500–510.
31. Wu, C.; Li, X.; Zuo, F.; Luo, L.; Du, X.; Di, J.; Zeng, Q. Use It-No Need to Shake It!: Accurate Implicit Authentication for Everyday Objects with Smart Sensing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2022**, *6*, 146:1–146:25. [CrossRef]
32. Zhang, J.; Li, Y.; Xiao, W. Integrated Multiple Kernel Learning for Device-Free Localization in Cluttered Environments Using Spatiotemporal Information. *IEEE Internet Things J.* **2021**, *8*, 4749–4761. [CrossRef]
33. Zhang, J.; Li, Y.; Xiao, W.; Zhang, Z. Online Spatiotemporal Modeling for Robust and Lightweight Device-Free Localization in Nonstationary Environments. *IEEE Trans. Ind. Inform.* **2023**, *19*, 8528–8538. [CrossRef]

sensors

MDPI

Article

Multi-Layered Filtration Framework for Efficient Detection of Network Attacks Using Machine Learning

Muhammad Arsalan Paracha [1], Muhammad Sadiq [2], Junwei Liang [2,*], Muhammad Hanif Durad [3] and Muhammad Sheeraz [3]

[1] Critical Infrastructure Protection and Malware Analysis Lab, Department of Computer and Information Sciences, Pakistan Institute of Engineering and Applied Sciences, Islamabad 44000, Pakistan; arsalan_18@pieas.edu.pk
[2] Shenzhen Institute of Information Technology, Shenzhen 518109, China; muhammad.sadiq@szu.edu.cn
[3] Department of Computer and Information Sciences, Pakistan Institute of Engineering and Applied Sciences, Islamabad 44000, Pakistan; hanif@pieas.edu.pk (M.H.D.); msheeraz_18@pieas.edu.pk (M.S.)
* Correspondence: jwliang@sziit.edu.cn

Abstract: The advancements and reliance on digital data necessitates dependence on information technology. The growing amount of digital data and their availability over the Internet have given rise to the problem of information security. With the increase in connectivity among devices and networks, maintaining the information security of an asset has now become essential for an organization. Intrusion detection systems (IDS) are widely used in networks for protection against different network attacks. Several machine-learning-based techniques have been used among researchers for the implementation of anomaly-based IDS (AIDS). In the past, the focus primarily remained on the improvement of the accuracy of the system. Efficiency with respect to time is an important aspect of an IDS, which most of the research has thus far somewhat overlooked. For this purpose, we propose a multi-layered filtration framework (MLFF) for feature reduction using a statistical approach. The proposed framework helps reduce the detection time without affecting the accuracy. We use the CIC-IDS2017 dataset for experiments. The proposed framework contains three filters and is connected in sequential order. The accuracy, precision, recall and F1 score are calculated against the selected machine learning models. In addition, the training time and the detection time are also calculated because these parameters are considered important in measuring the performance of a detection system. Generally, decision tree models, random forest methods, and artificial neural networks show better results in the detection of network attacks with minimum detection time.

Keywords: intrusion detection system; network attacks; machine learning; network security; security information and event management; CIC-IDS2017; anomaly detection

Citation: Paracha, M.A.; Sadiq, M.; Liang, J.; Durad, M.H.; Sheeraz, M. Multi-Layered Filtration Framework for Efficient Detection of Network Attacks Using Machine Learning. *Sensors* **2023**, *23*, 5829. https://doi.org/10.3390/s23135829

Academic Editor: Jian Li

Received: 10 May 2023
Revised: 12 June 2023
Accepted: 13 June 2023
Published: 22 June 2023

1. Introduction

In today's digital world, cybersecurity is becoming an essential need for military and government organizations, as well as for small enterprises and even individuals. Threat prevention is the epitome of digital security, which requires threat detection and threat management capabilities [1]. Security information and event management (SIEM) is being implemented by a large number of organizations and becoming a standardized approach to handle information security issues [2]. Due to the recent rise in cyberattacks and the strict security regulations required by governments, organizations have been investing in the security domain [3]. The core of any SIEM solution is the detection capability of the system. Information security experts have developed multiple network intrusion detection tools and techniques for the detection and prevention of evolving network attacks [4].

A computer network is a set of computers connected with each other for resource sharing. Any unauthorized action on the hardware or software of the systems connected with the network is called a network attack. In other words, a network attack is any action

that compromises the confidentiality, integrity, or availability of a system. An attack can be classified into one of two types [3]: active attacks and passive attacks. In active attacks, the attacker attempts to alter system resources or affect normal operations. Such attacks can lead to data modification, data loss, data theft, and denial of service. In passive attacks, the attacker attempts to learn important information without affecting the system resources and normal operations. Such attacks can lead to sensitive information being revealed [5].

Denial-of-service (DoS) attacks are the most common type of network attack. In this attack, the attacker tries to engage the system's resources to deny the service to legitimate users. It is an automated and continuous process that aimst o compromise a system by sending a large number of bogus requests. When these requests are generated from multiple systems, such attacks are known as distributed denial-of-service (DDoS) attacks. In brute-force attacks, the attacker tries multiple combinations to guess the credentials. These attacks are computationally extensive and require huge resources.

Attacks can also be categorized with respect to the origin of the attack, which can be inside attacks or outside attacks [6]. Inside attacks are attacks initiated by an entity within the network or organization and are commonly known as insider attacks. The insider is an authorized user or entity that has access to system resources but uses them in an incorrect way. Outside attacks are attacks initiated by an entity from outside the network or organization and are commonly known as outsider attacks. An outsider is an unauthorized user or entity that performs illegitimate actions.

Broadly, network intrusion detection is categorized into two main types: signature-based detection and anomaly-based detection [7,8]. Static methodology is used in signature-based detection, wherein network traffic is compared with predefined rules, whereas anomaly-based detection is a dynamic method that consists of machine-learning-based models to detect unwanted or malicious behavior over the network [9]. Signature-based approaches directly define the abnormal or malicious behavior in the form of rules. In this case, known threats can be identified very quickly and efficiently. On the other hand, the main objective of anomaly-based approaches is to define the normal or expected behavior so that any variation from normal, or unexpected behavior, can be identified. Therefore, anomaly-based detection techniques can also detect unknown or zero-day attacks [10]. With this clear advantage, anomaly detection would be the preferred methodology among the information security researchers [6,11].

Intrusion detection systems (IDS) serve as essential components in network defense, providing protection against a variety of network attacks. However, the focus of previous research has primarily been on improving the accuracy of IDS while relatively overlooking the crucial aspect of efficiency with respect to time. This oversight limits the effectiveness of IDS in rapidly detecting and responding to network threats.

To address this gap, we propose a multi-layered filtration framework (MLFF) for the efficient detection of network attacks. The primary objective of our research is to minimize detection time without compromising the accuracy of the system. By employing statistical methods for feature reduction, the MLFF systematically reduces the number of features needed, optimizing the efficiency of the intrusion detection process. Through experimentation on the CIC-IDS2017 dataset, we assess the effectiveness of the proposed framework by evaluating its accuracy, precision, recall, and F1 score, in addition to its training time and detection time.

This paper presents an approach to systematically reduce the number of features using statistical methods and proposes a multi-layered framework for feature reduction. The major contributions of this paper are summarized below:

- It provides a multi-layered filtration framework for feature reduction to systematically reduce the number of features using statistical methods.
- It provides a mechanism to effectively reduce the detection time without compromising the accuracy of the detection system.
- It shows the accuracy, precision, recall, F1 score and detection time against selected machine learning models for CIC-IDS2017.

The rest of the paper is organized as follows. Section 2 covers the related work and provides information about available datasets and the evaluation metrics for the detection system. Section 3 provides the methodology and the proposed framework. Section 4 deals with the results and provides a discussion; these aspects are performed through experiments and comparative analysis. Finally, Section 5 covers the conclusions and recommendations for future work.

2. Related Work

Research on the detection of network attacks has been conducted using different publicly available datasets. These datasets play a vital role in the validation of the detection approach and are used as benchmarks [7]. The initial work of creating a dataset for an IDS was carried out by DARPA (Defence Advanced Research Project Agency); they generated the KDD98 (Knowledge Discovery and Data Mining) dataset in 1998. This was created by modeling a small US Air Force base network connected to the Internet. It had 41 features that were categorized as normal or abnormal [8]. This dataset plays an important contribution in the research of IDS. However, in [12], the author criticized the accuracy and capability of KDD98 to contemplate realistic environments. Although KDD98 had multiple reported problems, even then, it was being used by the research community [13,14].

Ref. [15] identified numerous issues in the KDD98 dataset, due to which a new dataset NSL-KDD was published in 2009 [16]. The dataset was created by eliminating duplicate records to overcome issues of bias in machine learning models.

Several other IDS datasets have been created. In 2007, a dataset named CAIDA was proposed [17]. This dataset contains network traffic of distributed denial-of-service (DDoS) attacks. This dataset lacks attack diversity. A labeled dataset for flow-based intrusion detection was also proposed in 2009 [18]. The dataset was based on a honeypot deployed over the Internet to maximize exposure to attacks. In 2015, Moustafa and Slay proposed a dataset called UNSW-NB15, which addresses the issue of the unavailability of a network benchmark dataset [19]. The dataset was generated by simulating network attacks and suggesting nine different attack families.

Multiple datasets were reviewed, and in [20], the author proposed 11 characteristics to evaluate a dataset. On the basis of these 11 characteristics, the Canadian Institute for Cybersecurity (CIC) released a dataset called CIC-IDS2017 [21]. The CIC-IDS2017 dataset is now widely being used among security researchers[22–27].

Multiple techniques were combined by Yulianto et al. [28] to improve the performance of IDS using the CICIDS-2017 dataset. The hybrid feature selection method was used by Tama et al. [29] and reduced the number of features to 37 with an accuracy of 96.46%.

Gupta et al. [30] suggested that the class imbalance problem can be tackled with the help of ensemble algorithms. Deep neural network, eXtreme gradient boosting, and random forest algorithms were used in three different stages and achieved an accuracy of 92% for the CICIDS-2017 dataset. Doaa et al. [31] also worked on feature reduction along with ensemble learning techniques. The results show 99% accuracy with 30 features from the CICIDS-2017 dataset.

A context-aware feature extraction method was proposed by Shams et al. [32] for convolutional neural networks (CNN); these authors concluded that CNNs showed better results as compared to an ordinary neural network. Birnur et al. [33] proposed an approach using optimal feature selection and finding multivariate outliers for the improvement of the performance of an IDS. The NSL-KDD dataset was used for experiments.

In [34], the author proposed a hybrid optimization scheme to improve the rate of precision in the detection of an intrusion. Qureshi et al. [35] proposed a transfer learning technique to train deep neural networks. The original and extracted features were combined to improve the performance of an intrusion detection system.

Venkatesan et al. [36] suggested a model for intrusion detection and worked on feature selection using a modeling approach. Similarly, in [37,38], the authors worked on reducing the number of network parameters, resulting in decreased time and cost

In [39], dimensionality reduction was carried out using PCA, and subsequently, SVM was employed for the detection of DDoS attacks in SDN. Ref. [40] also worked on SDN and proposed an allocation-based approach using a multi-criteria decision-making (MCDM) strategy for a multi-domain SDN-enabled IoT network.

In [41], the authors proposed a detection framework based on 16 features for DDoS attack detections. However, this method was not designed to cater to imbalanced data. Yi et al. [42] proposed a multi-objective evolutionary convolutional neural network for an IDS. However, these authors' results show that the detection performance was affected by the use of very few neurons and layers.

Although much work has been conducted using the available datasets for intrusion detection systems, the major focus remains on the improvement of the accuracy of the detection. In addition to the accuracy, the detection time is also an important factor, especially in the case of AIDS deployed in an inline mode. Our main focus is to propose a framework for feature reduction to effectively reduce the detection time without affecting the accuracy of the system.

Evaluation Metrics

A confusion matrix is used to evaluate the performance of an intrusion detection system. Table 1 shows a confusion matrix for a binary class classifier.

Table 1. Confusion Matrix.

		Predicted Class	
		Attack	Normal
Actual Class	Attack	*TP*	*FN*
	Normal	*FP*	*TN*

True positives (*TP*) represent the number of attacks that are correctly predicted as attacks, and true negatives (*TN*) represent the number of normal events or instances that are correctly predicted as normal behavior. False positives (*FP*) represent the number of normal instances that are incorrectly predicted as attacks, and false negatives (*FN*) represent the number of attacks that are incorrectly classified as a normal instance. In a good detection system, *TP* and *TN* should remain high, and *FP* and *FN* should remain low [43].

The accuracy, precision, recall, and F1-score are the performance metrics used to evaluate intrusion detection systems. They are derived from the information given in the confusion matrix and calculated as per the following formulas [7]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \times 100 \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \times 100 \quad (3)$$

$$F1Score = \frac{Precision * Recall}{Precision + Recall} \times 2 \quad (4)$$

Accuracy measure how accurate the detection system is at detecting normal and attack traffic. It is the percentage of all correctly predicted instances against all instances. Precision is the accuracy of positive predictions. Recall is the measure of the true-positive rate and is also called the detection rate or sensitivity. F1 score is the harmonic mean of precision and recall. In addition to these performance metrics, we also calculated the training time and the detection time and compared these values against different ML and DL models. In the case of an intrusion detection system, the training time is not considered a very

important factor as the model is trained only once. However, the detection time is of prime importance with respect to an intrusion detection system. Both the training time and the detection time are calculated using the formulas below. Table 2 shows a list of notations and their explanations.

$$TrainingTime = TrEndTime - TrStartTime \quad (5)$$

$$DetectionTime = PrEndTime - PrStartTime \quad (6)$$

Table 2. List of notations and their explanations.

Notations	Explanaiton
TP	True positive
TN	True negative
FP	False positive
FN	False negative
Ho	Null hypothesis
Ha	Alternate hypothesis
t	Calculated value of test statistic
r	Correlation coefficient
n	Sample size
d	Difference between the two ranks of each observation
R	Coefficient of determination
TrstartTime	Time when training starts
TrEndTime	Time when training ends
PrstartTime	Time when testing/prediction starts
PrEndTime	Time when testing/prediction ends

3. Methodology

The dataset CICIDS2017 was selected as it has been widely used among the research ca ommunity and is also publically available. The dataset was generated on a real network that contained an attacker network and victim network. On the attacker side, four machines were connected, having Kali and Windows 8.1 operating systems. The victim side was protected with a firewall and contained multiple machines with Windows, Linux, and Macintosh operating systems. Multiple services, including domain controller and domain name system (DNS), were also running on the servers so that an attacker could perform real attacks [21]. Almost 50 GB of captured traffic in PCAP files were provided. Along with PCAP files, 8 CSV files were also provided, along with a set of 84 features and a label. The dataset consisted of a total of 15 classes. One of the classes was related to "benign" or normal traffic, and the other 14 corresponded to different "attack" classes. These attack classes were DDoS, Portscan, Bot, Infiltration, Web Attack Brute Force, Web Attack XSS, Web Attack SqlInjection, FTP-Pataor, SSH-Patator, DoS Slowloris, DoS SlowHttp, DoS Hulk, DoS Goldeneye, and Heartbleed.

3.1. Proposed Framework Architecture

The proposed multi-layered filtration framework (MLFF) for attack detection is demonstrated in Figure 1. It consists of multiple phases of preprocessing, three layers of feature reduction, the creation of the final dataset, training models on a training dataset, and validating the results on a test dataset.

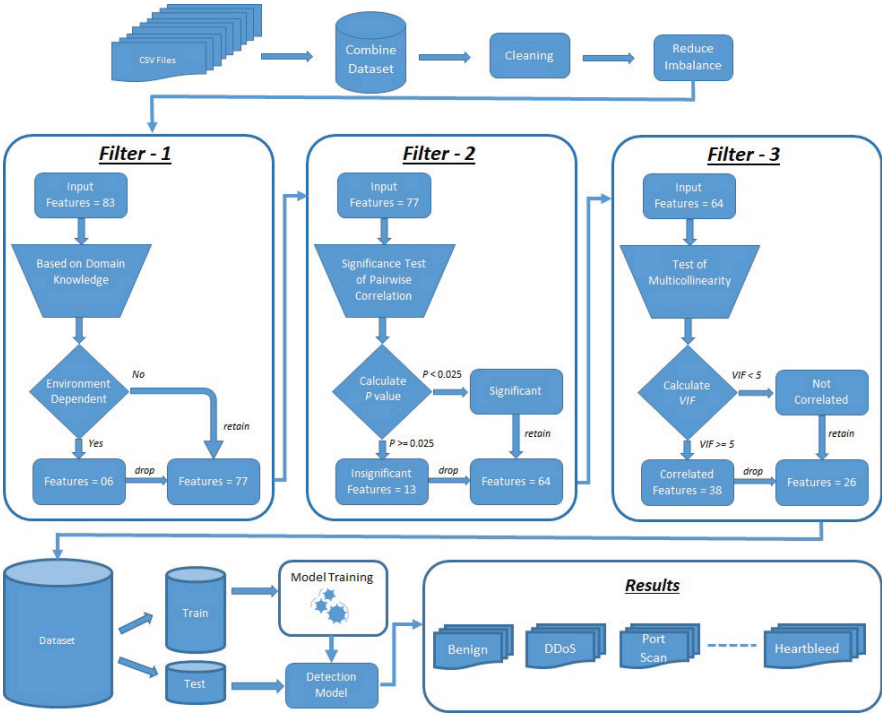


Figure 1. Proposed Framework Architecture.

3.1.1. Merge

As a first step, the 8 CSV files were merged together in order to create a single file that contained the traffic from all classes. This combined dataset was generated from 8 files that were collected from traffic captured from Monday to Friday. A total of 2,520,798 instances were created. Table 3 shows the complete distribution of data among the classes.

Table 3. Distribution among Classes.

Label	Instances
BENIGN	2,095,057
DoS Hulk	172,846
DDoS	128,014
PortScan	90,694
DoS Goldeneye	10,286
FTP-Pataor	5931
DoS Slowloris	5385
DoS SlowHttp	5228
SSH-Patator	3219
Bot	1948
Web Attack Brute-Force	1470
Web Attack XSS	652
Infiltration	36
Web Attack SqlInjection	21
Heartbleed	11

3.1.2. Cleaning

After that, the dataset was cleaned from NaN (missing) and infinity values; duplicate values and white spaces were also checked and removed.

3.1.3. Reducing the Imbalance Problem

The results clearly show that the dataset is huge and highly imbalanced. Applying a machine learning model directly may lead to inefficiencies. To overcome the problem of imbalanced data, instances from 4 classes (i.e. Benign, DoS Hulk, DDoS, and Portscan) were removed using the Pandas dataframe.sample method, which returns random samples according to the specified percentage. A total of 31,213 rows were removed because of duplication. After that, a total of 207,908 instances were retained in the dataset. The distribution of these instances among 15 classes is shown in Figure 2.

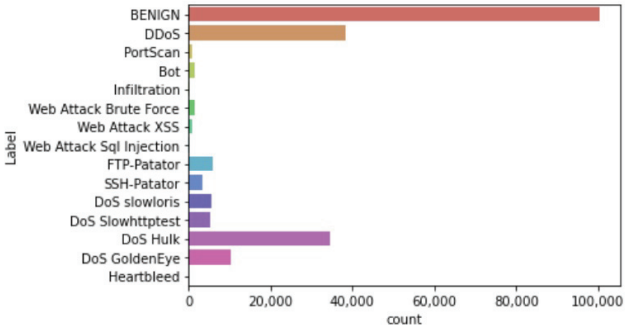


Figure 2. Final dataset distribution.

3.1.4. Filter-1

The first filter was a manual filter in which features were dropped on the basis of domain knowledge. Some features are purely environment-dependent; for example, an IP address can be changed according to the network configurations. Similarly, port numbers can also vary in different scenarios. When the sender wants to communicate with the receiver, it uses a random port number as a source port. Therefore, it is necessary to eliminate such features. If we train our model without removing these features, the model may perform well on test data; however, it will not achieve paramount results in real-world networks.

A total of 83 features and a label are present in the dataset, and of them only 6 features, which were named FlowID, Source IP, Source Port, Destination IP, Destination Port, and Timestamp, were totally dependent on the architecture and the time of the test performed. Therefore, these features were removed from the dataset, and 77 features were left. Table 4 shows the names of the feature dropped in the first layer.

Table 4. Features dropped in the first layer.

S. No.	Feature Name
1	FlowID
2	Source IP
3	Source Port
4	Destination IP
5	Destination Port
6	Timestamp

3.1.5. Filter-2

The main objective of our study is to minimize the detection time while maintaining accuracy so that the framework can be used in real networks. In this layer, first, we identified the insignificant features, and then, we dropped these features from the dataset. Therefore, in order to identify the insignificant features, a statistical approach of testing

the significance using the *p*-value method was adopted [44]. That is, we wished to test the hypotheses as follows:

$$Null\ Hypothesis\ H_0 : p = 0\ (insignificant) \tag{7}$$

$$Alternate\ Hypothesis\ H_a : p \neq 0\ (insignificant) \tag{8}$$

The significance of correlation coefficients can be checked with the help of a t-test [45]. In general, we tested the degree of the deviation of the correlation coefficient from zero.

$$t = \frac{r\sqrt{n-2}}{1-r^2} \tag{9}$$

Here, *r* is the correlation coefficient calculated from the sample, and n is the sample size. With t and the sample size, we can calculate the p value. If the p value is greater or equal to the significance level, which was set to an alpha equal to 0.025 in our case, we retain the null hypothesis and conclude that the variable is insignificant. If the p value is less than an alpha of 0.025, the null hypothesis is rejected, and we conclude that the variable is significant. Thus, we will not drop it [46].

Because, in our case, the relationship among the features is non-linear, Spearman's rank correlation coefficient method [47] was used to find the value of r.

$$r_s = 1 - \frac{6\sum d^2}{n(n^2-1)} \tag{10}$$

where,

$$d = rank\ X - rank\ Y \tag{11}$$

The results show that 13 features are insignificant and can be dropped because they do not contribute to the learning of the model. After this test, the number of features was reduced to 64. Table 5 shows the names of the features dropped in the second layer.

Table 5. Features dropped in the second layer.

S. No	Feature Name
1	Fwd Packet Length Mean
2	Bwd PSH Flags
3	Fwd URG Flags
4	Bwd URG Flags
5	CWE Flag Count
6	Avg Fwd Segment Size
7	Fwd Avg Bytes/Bulk
8	Fwd Avg Packets/Bulk
9	Fwd Avg Bulk Rate
10	Bwd Avg Bytes/Bulk
11	Bwd Avg Packets/Bulk
12	Bwd Avg Bulk Rate
13	Idle Std

3.1.6. Filter-3

The main objective of this filter is to test whether independent variables in a model are correlated among themselves. During this test, we found and removed those independent variables that were highly correlated to avoid the problem of overfitting the model. We selected the variation inflation factor (VIF) method for the detection of multicollinearity [48,49] among independent variables and also calculated the tolerance rate.

$$VIF = \frac{1}{1-R^2} \tag{12}$$

$$Tolerance = 1 - R^2 \quad (13)$$

Here, R^2 is the coefficient of determination, which indicates the amount of proportional change in the dependent variable due to the change in the independent variable.

In this case, we selected variables one by one, calculated the VIF against the included variables in a model, ran tests for multicollinearity, and kept the variables that have VIF less than 5. After completing the test iteration, 38 variables were dropped because of their high collinearity with others.

3.1.7. Dataset after Filtration

After passing the parameters through the filtration framework, a new dataset was created with only 26 selected parameters. The dataset was split into training and test data with a ratio of 70/30. The models were trained with training data and evaluated on test data. The training time and the detection time were also recorded against the selected ML models.

4. Results and Discussion

This section describes the results obtained after the implementation of a multi-layered feature reduction framework. All the implementation was carried out in Python on Jupyter Notebook (Anaconda3). A Dell OptiPlex 7060 PC with an Intel Core i7-7800 CPU@ 3.40 GHz with 32 GB RAM was used to conduct the experiment.

4.1. Selected Parameters

The final selected parameters, variance inflation factor (VIF), and tolerance against each selected feature are shown in Table 6. Among 26 selected parameters, packet length is identified as an important parameter as usually, attack traffic consists of irregular packet length. Flow bytes/s is the packet flow per second. The time between two packets in a flow in a forward and backward direction is also considered an important parameter. Moreover, the number of packets per second in both directions is also selected.

Table 6. Names of the selected feature.

S. No	Feature Name	VIF	Tolerance
1	Fwd Packet Length Min	1.381	0.724
2	Fwd Packet Length Std	1.145	0.872
3	Bwd Packet Length Min	2.048	0.488
4	Flow Bytes/s	1.078	0.927
5	Flow IAT Min	1.487	0.672
6	Fwd IAT Min	1.927	0.518
7	Bwd IAT Total	3.210	0.311
8	Bwd IAT Std	3.440	0.290
9	Fwd Packets/s	1.161	0.860
10	Bwd Packets/s	1.090	0.917
11	Min Packet Length	2.256	0.443
12	Packet Length Variance	1.350	0.740
13	FIN Flag Count	2.027	0.493
14	SYN Flag Count	1.335	0.748
15	ACK Flag Count	3.256	0.307
16	URG Flag Count	1.731	0.577
17	ECE Flag Count	1.000	0.999
18	Down/Up Ratio	1.625	0.615
19	Subflow Fwd Bytes	1.093	0.914
20	Subflow Bwd Bytes	1.064	0.939
21	Init Win bytes forward	1.916	0.521
22	Init Win bytes backward	1.098	0.910
23	min seg size forward	1.383	0.72

Table 6. *Cont.*

S. No	Feature Name	VIF	Tolerance
24	Active Mean	1.266	0.789
25	Active Std	1.394	0.717
26	Idle Min	2.522	0.396

There are different types of flags in a packet header. Each flag has a distinct role in communication. Attackers manipulate the value of these flags to launch attack traffic. The SYN flag is used to initiate a TCP connection. ACK is used to recognize the successful delivery of a packet. The URG flag is used to prioritize the packet. The FIN flag specifies the end of a TCP session. ECE is used to send the congestion indication. The value of these flags and the total number of bytes in both the forward and backward directions sent in an initial window are also selected for the model.

The active mean is the mean time for which the flow remains active, and active std is the standard deviation time before the flow becomes idle. Idle Min is the minimum time a flow remains idle before going active. All these parameters contributed to the efficient detection of network attacks.

4.2. Comparison of Results

The main focus of our research is to develop an efficient attack detection system that maintains a high detection rate with a minimum detection time. In this paper, we have calculated the accuracy, precision, recall and F1-score along with the training time and detection time against the selected machine learning models. We compared the results with the findings presented by [24]. The results are also compared with the findings of other researchers and are shown in Table 7.

Table 7. Comparison of results of the proposed framework and previous studies.

Solution	Model	Accuracy	Precision	Recall	F1-Score	Training Time (Seconds)	Detection Time (Seconds)
Proposed	DT	0.9927	0.9927	0.9927	0.9927	1.02	0.02
	RF	0.9942	0.9938	0.9942	0.9939	13.7	0.78
	SVM	0.8911	0.8886	0.8911	0.8812	914	201
	NB	0.3614	0.7070	0.3614	0.3218	0.56	0.21
	KNN	0.9795	0.9795	0.9795	0.9795	0.13	91.3
	LR	0.8184	0.8053	0.8184	0.8041	100	0.01
	ANN	0.9815	0.9824	0.9815	0.9816	84.6	0.03
[24]	DT	0.9949	0.9943	0.9949	0.9942	1.23	1.12
	RF	0.9930	0.9909	0.9930	0.9912	9.38	6.76
	SVM	0.7521	0.9916	0.7521	0.7660	343	33.1
	NB	0.9886	0.9901	0.9886	0.9885	1.07	0.15
	KNN	0.9952	0.9949	0.9952	0.9949	11.13	7.92
	ANN	0.9928	0.9937	0.9928	0.9917	53.78	48.03
[21]	KNN	-	0.96	0.96	0.96	-	-
	RF	-	0.98	0.97	0.97	-	-
	ANN	-	0.98	0.98	0.98	-	-
[25]	CNN + GRU	0.9017	0.9234	0.9124	0.9205	741	17.3
[26]	RF	-	0.9572	0.9458	0.9415	-	
	DBN + SVM	-	0.9774	0.9767	0.9768	-	-
[27]	RF + AutoEncoder	-	-	-	0.9950	-	-

Table 7. *Cont.*

Solution	Model	Accuracy	Precision	Recall	F-1 Score	Training Time (Seconds)	Detection Time (Seconds)
[28]	AdaBoost + SMOTE	0.8147	0.8169	0.9576	0.8817	-	-
[30]	CSE-IDS	0.92	-	-	-	274	0.005
[31]	RF+NB+KNN+SVM	0.997	-	-	-	-	-
[32]	CAFE-CNN	0.992	-	-	-	-	-
[42]	MECNN	0.997	0.991	0.791	83.31	-	-

Figure 3 shows the accuracy and detection time of the proposed framework against selected models. As our main focus is on the reduction of detection time, the results show that in the case of the decision tree (DT) model, the detection time is 0.02 s, as compared to 1.12 s in [24]. Additionally, we still manage to maintain an accuracy of 99.27% as compared to 99.49% in [24]. In addition, random forest (RF) also produces 99.42% accuracy as compared to 99.30% in [24]. The detection time of random forest is calculated as 0.78 s as compared to 6.76 s in [24]. The results of the artificial neural network (ANN) show 98.15% accuracy, which is a little less as compared to 99.28% in [24]. However, the detection time has been significantly reduced, from 48.03 s [24] to 0.03 s.

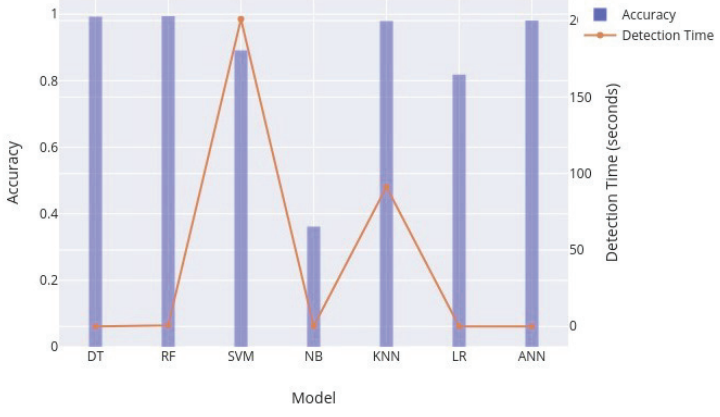


Figure 3. Models with accuracy and detection time of proposed framework.

The results of a K-nearest neighbor (K-NN) classifier show an accuracy of 97.95%, but the detection time is 91.3 s, which makes it impractical for use in an intrusion detection system. Similarly, a support-vector machine (SVM) architecture has a detection time of 201 s, which is also considerably high. The detection time in the case of logistic regression (LR) is very low, at only 0.01 s, but the accuracy is 81.84%. The naïve Bayes (NB) classifier shows an accuracy of only 36.14% with a detection time of 0.21 s. The results clearly show that in our approach, the random forest classifier gives better accuracy with less detection time, and the decision tree classifier gives almost the same accuracy with significantly less detection time.

As the RF model outperforms all others, the confusion matrix for the random forest model is shown in Figure 4. We can observe that Web Attack Brute-Force and Web Attack XSS have many false positives and false negatives in common. This is due to the high degree of similarity between these attacks.

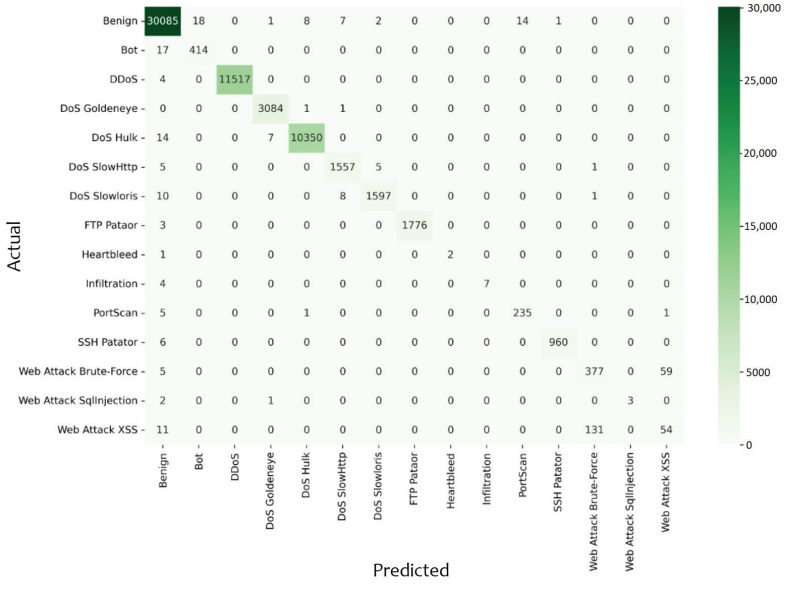


Figure 4. Confusion matrix for random forest model.

We have also calculated the accuracy over 2, 5, 7, and 10 folds of cross-validation for our top 3 models, i.e., decision tree, random forest, and ANN. The results of the average accuracy against each fold are shown in Figure 5. The results show that the decision tree and random forest models perform better with 10-fold cross-validation, and the ANN performs best with 2-fold cross-validation. However, after the analysis of the results of both holdout and cross-validation, it is clear that holdout validation gives better results. The CIC-IDS2017 dataset used in our experiments is a relatively large dataset with a substantial number of samples. This allowed us to split the dataset into separate training and testing sets, ensuring a sufficient amount of data for model training and evaluation. Additionally, our focus was primarily on reducing detection time without compromising accuracy, and holdout validation provided a straightforward and efficient way to assess these metrics. This allowed us to directly compare our results with existing benchmarks and demonstrate the effectiveness of our proposed multi-layered filtration framework (MLFF).

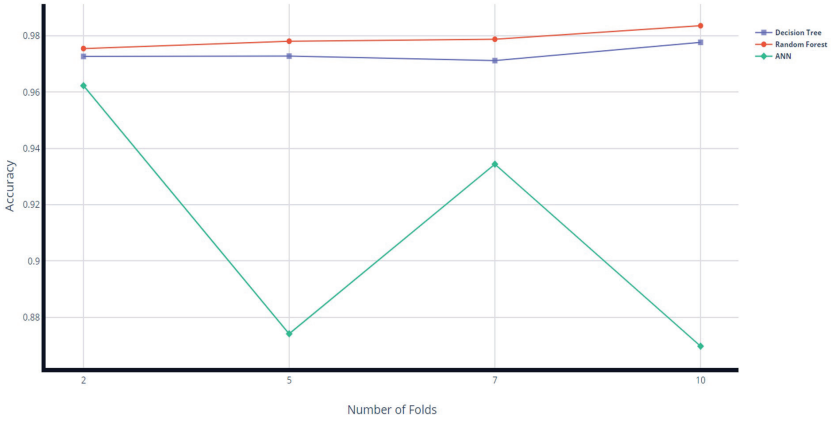


Figure 5. Results of different folds of cross-validation.

4.3. Ablation Study

Our proposed multi-layered filtration framework consists of three essential filters: filter 1 (F1), filter 2 (F2), and filter 3 (F3). In this subsection, we carry out an ablation study to validate their effectiveness by removing or changing the order of these filters. F1 is a mandatory filter and cannot be removed because it is based on environmental parameters, as discussed earlier. However, F2 and F3 can be reordered or removed one by one. Table 8 shows a comparison of the results obtained from this process. For this ablation study, we only selected the random forest ML algorithm because the results in Table 7 clearly show that RF outperforms all other models with respect to accuracy and time.

Table 8. Comparison of the proposed framework with different filters.

Filters	Number of Features	Accuracy	Detection Time (s)
F1	77	99.29	1.50
F1 + F2	64	99.30	0.99
F1 + F3	28	99.40	0.82
F1 + F2 + F3	26	99.42	0.78
F1 + F3 + F2	26	99.39	0.79

From Figure 6, we can see that each proposed filter plays an important part in reducing the number of features and improving the performance in terms of accuracy and detection time. Furthermore, it can be clearly observed that the best performance is achieved when all three filters are used altogether in the sequence proposed in the multi-layered filtration framework (MLFF).

Using the proposed framework, we managed to maintain a high detection rate while ensuring the minimum detection time by reducing the selected parameters. However, this approach has its own limitations; for example, in the case of high traffic volume, the system cannot be used at an adequate speed. Additionally, this study only focuses on the network attacks used in the CICIDS2017 dataset

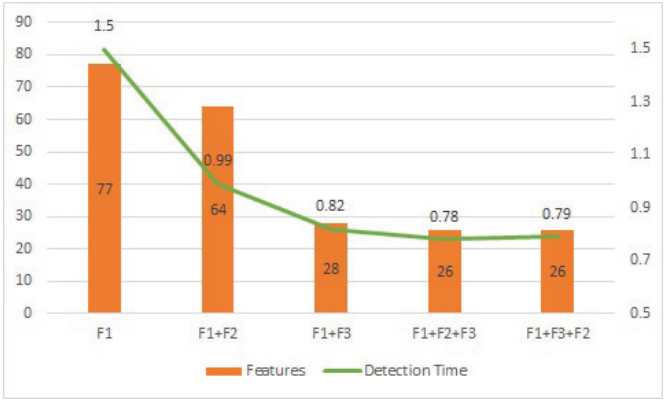


Figure 6. Number of Features and Detection Time against a combination of different filters.

5. Conclusions

This paper proposed a multi-layered filtration framework (MLFF) for the efficient detection of network attacks. Using this framework, the number of features were systematically reduced without compromising the performance of the intrusion detection system. The main aim of this paper is to minimize the detection time without affecting the accuracy of the system. The proposed framework contains three filters that are connected in such a way that the output of the first filter becomes the input of the second filter and the output of the second filter becomes the input of the third filter. A total of 26 features were selected out of 83 features. The model was trained using only the selected features, and detection was performed on the test data. The accuracy, precision, recall, and F1-score, along with the training time and detection time, were calculated against the selected machine learning models. The results were compared with the available benchmarks and demonstrated a significant improvement after the implementation of the proposed framework.

Random forest (RF) produced 99.42% accuracy, and the detection time was calculated as 0.78 s. The results clearly showed that random forest outperformed all other models. In addition, the decision tree model and artificial neural networks also performed well. The experiment showed that the detection time was reduced significantly without compromising the accuracy of the system. Because we have managed to reduce the detection time, the proposed framework can be deployed in intrusion detection systems running in real networks.

In order to further enhance the accuracy and timing of the proposed multi-layered filtration framework (MLFF) for the efficient detection of network attacks, future research can focus on leveraging advanced datasets, i.e., cicids2018, to optimize the model. Firstly, expanding the scope of the attack spectrum by incorporating additional attacks from multiple layers would be a valuable direction. By including a wider range of attack types and techniques, the MLFF can improve its ability to detect sophisticated and multi-faceted attacks.

Secondly, future work should explore the utilization of increased computing resources to boost the performance of the MLFF. More powerful hardware, such as high-performance servers or specialized processing units, can significantly accelerate the training and detection processes. This would result in reduced detection times and enable near-real-time analysis of network traffic data. Additionally, the integration of multiple detection frameworks into a centralized security information event management system will improve the security echo system.

Author Contributions: Conceptualization, M.A.P. and M.H.D.; methodology, M.A.P.; software, M.A.P.; validation, M.S. (Muhammad Sadiq), J.L. and M.H.D.; formal analysis, M.A.P., M.S. (Muhammad Sadiq); investigation, M.A.P.; resources, M.H.D.; data curation, M.A.P. and M.S. (Muhammad

Sadiq); writing—original draft preparation, M.A.P.; writing—review and editing, M.S. (Muhammad Sadiq), J.L. and M.H.D.; visualization, M.A.P.; supervision, M.H.D.; project administration, M.H.D.; funding acquisition, M.S. (Muhammad Sadiq), J.L. and M.H.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research work was supported by the funding of the Guangdong Provincial Research Platform and Project under Grant No. 2022KQNCX233, the Foundation of State Key Laboratory of Public Big Data under Grant No. PBD2022-14, the Shenzhen Science and Technology Program under Grant No. RCBS20221008093252092, and the Ministry of Planning, Development and Special Initiatives through the Higher Education Commission of Pakistan under the National Center for Cyber Security, CIPMA Lab, PIEAS.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

IDS	Intrusion detection system
AIDS	Anomaly-based intrusion detection system
SIEM	Security information and event management
DARPA	Defence Advanced Research Project Agency
DDOS	Distributed denial of service
PCAP	Packet capture
XSS	Cross-site scripting
FTP	File transfer protocol
SSH	Secure socket shell

References

1. Li, Y.; Liu, Q. A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments. *Energy Rep.* **2021**, *7*, 8176–8186. [CrossRef]
2. Sheeraz, M.; Paracha, M.A.; Haque, M.U.; Durad, M.H.; Mohsin, S.M.; Band, S.S.; Mosavi, A. Effective Security Monitoring Using Efficient SIEM Architecture. *Hum.-Centric Comput. Inf. Sci.* **2023**, *13*, 1–18.
3. Latha, S.; Prakash, S.J. A survey on network attacks and Intrusion detection systems. In Proceedings of the 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6–7 January 2017; pp. 1–7.
4. Singh, R.; Srivastav, G. Novel Framework for Anomaly Detection Using Machine Learning Technique on CIC-IDS2017 Dataset. In Proceedings of the 2021 International Conference on Technological Advancements and Innovations (ICTAI), Tashkent, Uzbekistan, 10–12 November 2021; pp. 632–636.
5. Uma, M.; Padmavathi, G. A Survey on Various Cyber Attacks and their Classification. *Int. J. Netw. Secur.* **2013**, *15*, 390–396.
6. William Stallings, L.B. *Computer Security: Principles and Practice*; Pearson: London, UK, 2015.
7. Thapa, N.; Liu, Z.; Kc, D.B.; Gokaraju, B.; Roy, K. Comparison of machine learning and deep learning models for network intrusion detection systems. *Future Internet* **2020**, *12*, 167. [CrossRef]
8. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J. Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity* **2019**, *2*, 20. [CrossRef]
9. Aljuhani, A. Machine learning approaches for combating distributed denial of service attacks in modern networking environments. *IEEE Access* **2021**, *9*, 42236–42264. [CrossRef]
10. Nawaz, M.; Paracha, M.A.; Majid, A.; Durad, H. Attack Detection From Network Traffic using Machine Learning. *VFAST Trans. Softw. Eng.* **2020**, *8*, 1–7.
11. Zhou, Y.; Cheng, G.; Jiang, S.; Dai, M. Building an efficient intrusion detection system based on feature selection and ensemble classifier. *Comput. Netw.* **2020**, *174*, 107247. [CrossRef]
12. Creech, G.; Hu, J. A semantic approach to host-based intrusion detection systems using contiguousand discontiguous system call patterns. *IEEE Trans. Comput.* **2013**, *63*, 807–819. [CrossRef]
13. Ji, S.Y.; Jeong, B.K.; Choi, S.; Jeong, D.H. A multi-level intrusion detection method for abnormal network behaviors. *J. Netw. Comput. Appl.* **2016**, *62*, 9–17. [CrossRef]

14. Duque, S.; bin Omar, M.N. Using data mining algorithms for developing a model for intrusion detection system (IDS). *Procedia Comput. Sci.* **2015**, *61*, 46–51. [CrossRef]
15. McHugh, J. Testing intrusion detection systems: A critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. *ACM Trans. Inf. Syst. Secur. (TISSEC)* **2000**, *3*, 262–294. [CrossRef]
16. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 8–10 July 2009; pp. 1–6.
17. Hick, P.; Aben E.; Claffy K. ; Polterock, J. The CAIDA UCSD "DDoS Attack 2007" Dataset. 2007. Available online: https://www.caida.org/catalog/datasets/ddos-20070804_dataset/ (accessed on 5 May 2021).
18. Sperotto, A.; Sadre, R.; Van Vliet, F.; Pras, A. A labeled data set for flow-based intrusion detection. In Proceedings of the International Workshop on IP Operations and Management, Venice, Italy, 29–30 October 2009; Springer: Berlin/Heidelberg, Germany, 2009; pp. 39–50.
19. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the 2015 military communications and information systems conference (MilCIS), Canberra, ACT, Australia, 10–12 November 2015; pp. 1–6.
20. Gharib, A.; Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. An evaluation framework for intrusion detection dataset. In Proceedings of the 2016 International Conference on Information Science and Security (ICISS), Pattaya, Thailand, 19–22 December 2016; pp. 1–6.
21. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *Int. Conf. Inf. Syst. Secur. Priv. (Icissp)* **2018**, *1*, 108–116.
22. Gamage, S.; Samarabandu, J. Deep learning methods in network intrusion detection: A survey and an objective comparison. *J. Netw. Comput. Appl.* **2020**, *169*, 102767. [CrossRef]
23. Ho, S.; Al Jufout, S.; Dajani, K.; Mozumdar, M. A novel intrusion detection model for detecting known and innovative cyberattacks using convolutional neural network. *IEEE Open J. Comput. Soc.* **2021**, *2*, 14–25. [CrossRef]
24. Maseer, Z.K.; Yusof, R.; Bahaman, N.; Mostafa, S.A.; Foozy, C.F.M. Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset. *IEEE Access* **2021**, *9*, 22351–22370. [CrossRef]
25. Bakhshi, T.; Ghita, B. Anomaly Detection in Encrypted Internet Traffic Using Hybrid Deep Learning. *Secur. Commun. Netw.* **2021**, *2021*, 1–6. [CrossRef]
26. Zhang, H.; Li, Y.; Lv, Z.; Sangaiah, A.K.; Huang, T. A real-time and ubiquitous network attack detection based on deep belief network and support vector machine. *IEEE/CAA J. Autom. Sin.* **2020**, *7*, 790–799. [CrossRef]
27. Abdulhammed, R.; Musafer, H.; Alessa, A.; Faezipour, M.; Abuzneid, A. Features dimensionality reduction approaches for machine learning based network intrusion detection. *Electronics* **2019**, *8*, 322. [CrossRef]
28. Yulianto, A.; Sukarno, P.; Suwastika, N.A. *Improving Adaboost-Based Intrusion Detection System (IDS) Performance on CIC IDS 2017 Dataset*; Journal of Physics: Conference Series; IOP Publishing: Bristol, UK, 2019; Volume 1192, p. 012018.
29. Tama, B.A.; Comuzzi, M.; Rhee, K.H. TSE-IDS: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system. *IEEE Access* **2019**, *7*, 94497–94507. [CrossRef]
30. Gupta, N.; Jindal, V.; Bedi, P. CSE-IDS: Using cost-sensitive deep learning and ensemble algorithms to handle class imbalance in network-based intrusion detection systems. *Comput. Secur.* **2022**, *112*, 102499. [CrossRef]
31. Mhawi, D.N.; Aldallal, A.; Hassan, S. Advanced feature-selection-based hybrid ensemble learning algorithms for network intrusion detection systems. *Symmetry* **2022**, *14*, 1461. [CrossRef]
32. Shams, E.A.; Rizaner, A.; Ulusoy, A.H. A novel context-aware feature extraction method for convolutional neural network-based intrusion detection systems. *Neural Comput. Appl.* **2021**, *33*, 13647–13665. [CrossRef]
33. Uzun, B.; Ballı, S. A novel method for intrusion detection in computer networks by identifying multivariate outliers and ReliefF feature selection. *Neural Comput. Appl.* **2022**, *34*, 17647–17662. [CrossRef]
34. Velliangiri, S.; Karthikeyan, P. Hybrid optimization scheme for intrusion detection using considerable feature selection. *Neural Comput. Appl.* **2020**, *32*, 7925–7939. [CrossRef]
35. Qureshi, A.S.; Khan, A.; Shamim, N.; Durad, M.H. Intrusion detection using deep sparse auto-encoder and self-taught learning. *Neural Comput. Appl.* **2020**, *32*, 3135–3147. [CrossRef]
36. Venkatesan, S. Design an Intrusion Detection System based on Feature Selection Using ML Algorithms. *Math. Stat. Eng. Appl.* **2023**, *72*, 702–710.
37. Sadiq, M.; Shi, D. Attentive occlusion-adaptive deep network for facial landmark detection. *Pattern Recognit.* **2022**, *125*, 108510. [CrossRef]
38. Sadiq, M.; Shi, D.; Liang, J. A robust occlusion-adaptive attention-based deep network for facial landmark detection. *Appl. Intell.* **2022**, *52*, 9320–9333. [CrossRef]
39. Ali, J.; Roh, B.h.; Lee, B.; Oh, J.; Adil, M. A Machine Learning Framework for Prevention of Software-Defined Networking controller from DDoS Attacks and dimensionality reduction of big data. In Proceedings of the 2020 International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Republic of Korea, 21–23 October 2020; pp. 515–519. [CrossRef]

40. Ali, J.; Jhaveri, R.H.; Alswailim, M.; Roh, B.h. ESCALB: An effective slave controller allocation-based load balancing scheme for multi-domain SDN-enabled-IoT networks. *J. King Saud Univ.-Comput. Inf. Sci.* **2023**, *35*, 101566. [CrossRef]
41. Kshirsagar, D.; Kumar, S. A feature reduction based reflected and exploited DDoS attacks detection system. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *13*, 393–405. [CrossRef]
42. Chen, Y.; Lin, Q.; Wei, W.; Ji, J.; Wong, K.C.; Coello, C.A.C. Intrusion detection using multi-objective evolutionary convolutional neural network for Internet of Things in Fog computing. *Knowl.-Based Syst.* **2022**, *244*, 108505. [CrossRef]
43. Rosay, A.; Cheval, E.; Carlier, F.; Leroux, P. Network Intrusion Detection: A Comprehensive Analysis of CIC-IDS2017. In Proceedings of the 8th International Conference on Information Systems Security and Privacy. SCITEPRESS-Science and Technology Publications, Online, 9–11 February 2022; pp. 25–36.
44. Wonu, N.; Victor-Edema, U.A.; Ndimele, S.C. Test of significance of correlation coefficient in science and educational research. *Int. J. Math. Stat. Stud.* **2018**, *9*, 53–68.
45. Keysers, C.; Gazzola, V.; Wagenmakers, E.J. Using Bayes factor hypothesis testing in neuroscience to establish evidence of absence. *Nat. Neurosci.* **2020**, *23*, 788–799. [CrossRef]
46. Rudolf Freund, W.W., P.S. *Regression Analysis*; Elsevier: Amsterdam, The Netherlands, 2006.
47. Zar, J.H. Significance testing of the Spearman rank correlation coefficient. *J. Am. Stat. Assoc.* **1972**, *67*, 578–580. [CrossRef]
48. Shrestha, N. Detecting multicollinearity in regression analysis. *Am. J. Appl. Math. Stat.* **2020**, *8*, 39–42. [CrossRef]
49. Tamura, R.; Kobayashi, K.; Takano, Y.; Miyashiro, R.; Nakata, K.; Matsui, T. Mixed integer quadratic optimization formulations for eliminating multicollinearity based on variance inflation factor. *J. Glob. Optim.* **2019**, *73*, 431–446. [CrossRef]

sensors

MDPI

Article

Ensemble-Learning Framework for Intrusion Detection to Enhance Internet of Things' Devices Security

Yazeed Alotaibi * and Mohammad Ilyas

Department of Electrical Engineering and Computer Science, Florida Atlantic University, 777 Glades Road, Boca Raton, FL 33431, USA; ilyas@fau.edu
* Correspondence: yalotaibi2021@fau.edu

Abstract: The Internet of Things (IoT) comprises a network of interconnected nodes constantly communicating, exchanging, and transferring data over various network protocols. Studies have shown that these protocols pose a severe threat (Cyber-attacks) to the security of data transmitted due to their ease of exploitation. In this research, we aim to contribute to the literature by improving the Intrusion Detection System (IDS) detection efficiency. In order to improve the efficiency of the IDS, a binary classification of normal and abnormal IoT traffic is constructed to enhance the IDS performance. Our method employs various supervised ML algorithms and ensemble classifiers. The proposed model was trained on TON-IoT network traffic datasets. Four of the trained ML-supervised models have achieved the highest accurate outcomes; Random Forest, Decision Tree, Logistic Regression, and K-Nearest Neighbor. These four classifiers are fed to two ensemble approaches: voting and stacking. The ensemble approaches were evaluated using the evaluation metrics and compared for their efficacy on this classification problem. The accuracy of the ensemble classifiers was higher than that of the individual models. This improvement can be attributed to ensemble learning strategies that leverage diverse learning mechanisms with varying capabilities. By combining these strategies, we were able to enhance the reliability of our predictions while reducing the occurrence of classification errors. The experimental results show that the framework can improve the efficiency of the Intrusion Detection System, achieving an accuracy rate of 0.9863.

Keywords: ensemble learning; machine learning; internet of things; security; intrusion detection system

Citation: Alotaibi, Y.; Ilyas, M. Ensemble-Learning Framework for Intrusion Detection to Enhance Internet of Things' Devices Security. *Sensors* **2023**, *23*, 5568. https://doi.org/10.3390/s23125568

Academic Editor: Jian Li

Received: 16 May 2023
Revised: 1 June 2023
Accepted: 6 June 2023
Published: 14 June 2023

1. Introduction

The Internet of Things (IoT) has grown exponentially due to technology's evolution. The IoT facilitates people's lives by providing and enhancing connectivity that supports the automation aspect of several human services. Millions of interconnected devices use the IoT to communicate, transfer, share, collect, and analyze data from several domains [1]. While involving the internet as the main factor in the technology's fields, it opens a new platform for cybercriminals [2]. Therefore, enhancing security and employing artificial intelligence innovations and technologies led to a protected and reliable IoT infrastructure.

In addition, the IoT architecture contains three main layers: the application, network, and user experience levels. The Perception layer has responsibility for every task, from utilizing the sensors to gathering the information, but it is also susceptible to multiple attacks due to its central role. Physical attacks on sensor-equipped equipment, unauthorized entry into the infrastructure, and other forms of physical attack are prevalent. The Network layer allows the devices fitted with sensors to communicate and exchange data with the gateways and other IoT devices via wireless technologies such as Wi-Fi, 3G, and 4G. The most common attacks faced by the Network layer are distributed denial-of-service (DDoS), denial-of-service (DOS), Man of the Middle, information theft, and gateways attacks [1,3].

Despite IoT's high dependence on the Internet as its primary communication network, the Internet of Things faces a new generation of innovative cybercriminals [4]. The IoT

consists of a network of interconnected nodes that constantly communicate with one another and process information over a variety of network protocols. Because of the ease with which these protocols can be exploited, it has been discovered that they pose a considerable risk to the confidentiality of the transferred data [5].

Researchers have developed and invented new security technologies based on AI technologies, such as ML, to detect and prevent such attacks. Unlike traditional firewalls and detection techniques, ML can handle big data environments [6,7]. Moreover, the ML is a well-known solution to address detection attacks and classification issues. ML might be considered the most suitable solution to secure and stabilize IoT network traffic [8]. ML has several techniques, such as regression, and classification [9]. In addition to preventing and detecting novel attacks, ML provides an appropriate strategy for securing IoT networks with its collection of supervised, unsupervised, and reinforcement techniques and algorithms [10].

One of the recent machine learning techniques is the prevention technique, such as the Intrusion detection system (IDS).The primary function of the IDS is to identify and report the normal or abnormal behavior of the traffic messages [1,11]. In addition, the IoT object is vulnerable to attacks because they rely on wireless communication protocols [12]. Attacks on the IoT affect all IoT network components, as opposed to the typical local network, where attacks affect specific nodes [13].

Furthermore, it is still unclear which machine learning techniques are more reliable and efficient for building a dynamic IDS. IoT IDS have been the focus of extensive study and research, with numerous machine learning and deep learning approaches being applied to a variety of datasets to determine their efficacy [14]. Time is a critical component of an effective IoT attack prevention system, so it's essential to consider ways to speed up the process of building and training the system. This can be completed by decreasing the computational time required by intrusion detection systems [11].

While the ML model scenarios do not show accurate and satisfactory results, the ensemble methods were employed to address this issue since it is important to combine several methods to avoid the instability aspect [15]. The primary objective of the ensemble is to enhance efficiency by combining the classification of several ML base classifiers [16]. When the performance of the ML classifiers is considered weak or low, the ensemble method is applied to combine the weak classifiers to construct a robust predictive model and enhance the performance [17]. Ensemble learning has been applied to several datasets, and it is communally used to construct intrusion detection approaches [17].

In addition, the ML detection system's capability depends on the quality of the database, so collecting or generating a credible dataset from the IoT communications environment at various levels that involves realistic attacks and regular traffic is a necessary step. Recently, a group of datasets, TON-IoT [18–25], which were collected from multiple IoT devices, have been tested and authenticated in a cyber lab to be more reliable to be applied to ML approaches.

In this study, we propose an IDS built with algorithms that use machine learning specifically for detecting intrusion attacks in IoT network traffic. Our approach's main goal is to enhance the accuracy of attack detection. We constructed and evaluated four supervised machine learning models to classify IoT network traffic as normal or abnormal. In addition, we used two ensemble techniques, voting and stacking, to merge these supervised models. Through the use of ensemble learning, we facilitate collaboration and improve classification performance between learning mechanisms with distinct capabilities. Effectiveness in classification tasks is enhanced by the ensemble learning method, which encourages the cooperation and reinforcement of several learning systems. Therefore, the paper's contributions are as follows:

1. Proposing a binary classification of IoT device network traffic as normal or abnormal.
2. Applying feature selection methods to improve the IDS performance of IoT network devices. Hence, we examined multiple ML algorithms to determine the most accurate

and efficient learners for building an efficient IDS to detect attacks on IoT devices within IoT network data.

3. Constructing and assessing four supervised models using data preprocessing and feature selection methods. This group of models consists of Random Forests, Decision Trees, Logistic Regression, and K-Nearest Neighbor. In addition, by combining the four supervised ML models, we employ two ensemble methods to improve the efficiency of the proposed model. The performance evaluation includes accuracy, precision, recall, and F1-score metrics.
4. The model we constructed improved the performance of the detection technique compared with the most recent study that used the same dataset.

This paper is organized as follows: Section 2 highlights previous work on machine learning-based and ensemble-based models for attack categorization, and Section 3 provides a brief background of the used techniques, methods, and several machine learning algorithms. Section 4 illustrates the methodology in detail. Section 5 covers our findings, comparisons, discussion, and evaluation method. Finally, Section 6 describes and outlines the conclusion of the entire study and the planned future work.

2. Related Work

The IoT transforms our daily activities by providing a mechanism for managing physical objects on the edge. IoT is one of the remarkable innovations in our era to make and support systems that are smart by improving system performance and reducing human involvement by automating several domains of services [26]. While the IoT environment includes several devices and objects, it allows these objects to communicate and be interconnected. These connected IoT objects communicate and exchange data via the internet and can be remotely accessible, which puts them at high risk of being targeted by malicious attacks and unauthorized access [27].

Enhancing the security aspect will protect the IoT architecture from attacks. Preventing these attacks requires detection techniques to expose intrusions or malicious activities within the IoT network. The IDS is an efficient technique for detecting attacks and malicious activities. The ensemble machine learning technique is considered one technique that provides an efficient and accurate solution to detecting and preventing attacks or malicious activities [20]. Recently, security detection technologies based on artificial intelligence innovations have become the primary technologies and modern techniques. Thus, artificial intelligence techniques driven by extensive data analysis can detect anomalies and malicious activities more accurately and comprehensively [6].

In the research paper [6,28–30], an ensemble-based model is used to detect and prevent attacks by applying several machine learning classification algorithms such as SVM, J48, and DT. Several feature selection methods are applied in proposed model to select the features that are predicted to be the most relevant, such as swarm optimization. The KDD99 dataset has been used to select the nine features that were predicted to be the most relevant. Furthermore, they proposed a model with an accuracy higher than 90%. Another study in ref. [29] implemented different algorithms, LR, Gradient boosting, and DT, and applied Gradient Boosting to the stacking classifier of ensemble learning. While the most relevant features must be selected among the CICIDS2018 dataset, Chi-square analysis was performed to zero in on these 23 most important features. As a result, the suggested model outperformed seven other classifiers, with a detection rate of 98.8% and a F-measure score of 97.1% [28,31].

Another research paper [32,33] compared seven different algorithms. The BoT-IoT dataset was used with machine learning algorithms for binary and multiclass prediction. While the proposed model produced and presented the RF as the best algorithm in binary prediction, the KNN showed that it gave the highest accuracy in multiclass prediction. The research in ref. [34] highlighted that the ensemble machine learning, neural networks, and kernel methods had been applied to detect anomalies and malicious activities in the IoT

architecture. The ensemble machine learning methods also outperform them in accuracy and detection rates.

While IoT is dealing with vast amounts of data, it is required to employ AI techniques to protect and secure the IoT environment and maintain the speed of exchanging and transforming the data [35]. Moreover, several public and private sectors use the internet to exchange data. When the data is transferred unencrypted via the internet, it will cause privacy issues and lead to data being hacked. While encryption is an essential aspect of transmitting data over the network. Thus, improving the Caesar cipher method creatively by applying an arithmetic model to switch to ciphertext has been accomplished [36,37].

Another research paper [38] highlighted that the authentication aspect is one of the most essential and significant factors in the security aspect. While the data has to be accessed by an authentic key to grant access, enhancing the authentication aspect of the image by promoting the security of the invisible watermarking data .Furthermore, research in ref. [39] has stated that data confidentiality is a critical issue in the cloud computing domain. While it depends entirely on the internet, it has several security issues. The most important issue out of them is the access control issue, which leads to data exploitation, and the time to access the data will be high.

The IoT has created a connected network of devices that involves heterogeneous objects from various aspects. While in the IoT, the network has no unified protocols or standards. Therefore, it is challenging to allow full security measures for those devices. While traditional security protocols provide optimal protection for the IoT architecture against threats on the Internet. Thus, constructing an efficient IDS relying on Deep Learning (DL) techniques and technologies is promising. Research papers [40,41] highlighted that applying novel DL techniques and frameworks, besides enhancing the existing DL models, presents remarkable results and strengthens the IDS performance. While it has an accurate and high prediction rate, it can assist in the maintenance aspect of the IoT network.

An IDS is an acronym for an intrusion detection system, which is focused on behavior detection to handle abnormal traffic. While IDS deals with the network's behavior in terms of attacks, it is established to learn from this behavior pattern using several techniques. Thus, the system will recognize any violated network traffic that can lead to an attack from this pattern [42,43].

There are two different layers of the Botnet attack classification, which are network-based or hot-based [34,44,45]. While the authors emphasized that the hot-based type is considered less realistic, the authors relied on the network layer to propose a comprehensive IoT attack detection and classification model. The suggested model used six supervised Machine Learning methods to develop an IDS: the ensemble learning technique was in three of them, the neural network used two, and the kernel used one. While they have relied on two standard attack detection datasets to observe their proposed model, NSL-KDD and distilled-Kitsune-2018, the proposed model produced results that were considered the best by 1–20% from any other prior work.

The recent anomaly detection studies [46–50] highlighted that the ensemble learning model is applied to enhance the performance of the current anomaly detection techniques. While the ensemble model combines and uses multiple algorithms for efficient predictive performance, it outperforms and is better than the single learning algorithm. Furthermore, the research in [50] presented an intrusion detection model to detect attacks and anomalies by applying single and ensemble-based learning implemented on the UNSW-NB15 dataset. The model achieved 99% accuracy. Although few studies address the imbalanced data issue in IoT anomaly detection [51], the research in [49] presented an ensemble detection model to detect outliers to address the imbalanced data issue by extracting the in-depth features through a stacked autoencoder (SAE) and inserting them into a probabilistic neural network of the ensemble for single and multiple outlier detection. Thus, reliance on the SAE enhances performance and stability. Table 1 highlights and summarizes the recent research on machine learning and ensemble techniques in terms of their approach and the types of datasets that have been relied on.

Table 1. State-of-the-Art Learning Based approach.

Paper	Dataset	Potential Approach	Classification Type	Attacks
[8]	BoT-IoT	Learning model (RF, and CNN)	Binary	Botent IoT malware, and IoT network attacks traffic.
[52]	Distilled-Kitsune-2018	Ensemble-based learning	Binary	OS scan, ARP, SSDP, SSL, and Maria attacks.
[53]	KDD cup'99 and NSL-KDD	Naïve Base, KNN, RF, and DT algorithms.	Binary	DoS, Sybil and Spoofing, Man-in-the-middle, Hole ataacks.
[54]	NSL-KDD	Ensemble-based learning	Multi-Class	DDoS Attacks, and IoT network attacks traffic.
[55]	Network Traffic	Ensemble-based learning	Binary /Multi-Class	Data theft, DoS, and spam
[56]	NSL-KDD	Ensemble-based learning	Binary	DDoS Attacks, and IoT network attacks traffic.
[57]	IoT Network Traffic	K-means	Binary	Ping flood Attacks

The comparison process of recent machine learning models that relied on the ensemble method is necessary to emphasize that the proposed techniques and approaches are significantly improving performance and building an efficient IDS. Table 2 presents recent ML ensemble methods to discuss and highlight the recent prior work that will be used in the results section.

Table 2. Recent Machine based learning applying Ensemble methods proposed model.

Paper	Year	Dataset	Objectives	Accuracy	Approach
[45]	2022	TON_IoT	Comparison process of ensemble learning for multiclass IoT network attacks.	91–96%	RF, XGBoos, LGBM, and CatBoost
[46]	2022	TON_IoT	Ensemble learning with binary and multi-class classification to enhance IDS system performance.	86%	LR, LDA, RF, NB, SVM,and LSTM
[47]	2022	CICIDS2017	A Binary ensemble-based learning for improving the performance of IDS system	88%	NB(M), DT, and LR
The proposed Model	2023	TON_IoT	A Binary ensemble-based learning for improving the performance of IDS system with a Comparison process of ensemble learning	98%	RF, ET, KNN, and SVC, Stacking method

While Tables 2 and 3 present a comparison process between the proposed model and the prior work. To the best of our knowledge, the proposed work's novelty is that, besides the analysis, the comparison to prior research, the preprocessing process with its applied methods, the selected feature based on the applied feature selection methods, the applied ML supervised models (RF, DT, LR, KNN), and the comparison between the two ensemble approaches. The proposed approach produces significantly improved

performance. A recall is 98.60 percent, precision is 98.2 percent, the F1 score is 98.61 percent, and accuracy is 98.63 percent. According to these measurements, advancements have been made relative to earlier studies' findings. The suggested model is an ensemble of several different algorithms, including the Random Forest (RF), K-Nearest Neighbors (KNN), Logistic Regression (LR), and Decision Tree (DT). By combining several algorithms into an "ensemble", the model takes advantage of their strengths to achieve better results than would be possible with a single algorithm alone. Table 3 presents the recent Ensemble-learning based methods compared with the proposed approach.

Table 3. Recent Ensemble-learning based methods compared with the proposed approach.

Paper	Year	Dataset	Recall	Precision	F1-Score	Accuracy	Approach
[53]	2022	TONIoT	91–97%	89–97%	89–97%	89–97%	RF, XGBoos, LGBM, and CatBoost
[42]	2022	TONIoT	85%	87%	86%	86%	LR, LDA, RF, NB, SVM, and LSTM
[28]	2022	CICIDS2017	Nil	Nil	Nil	88.92%	NB(M), DT, and LR
The proposed Model	2023	TONIoT	98.60%	98.20%	98.61%	98.63%	RF, ET, KNN, and SVC Stacking method

3. Background Studies

3.1. Intrusion Detection System (IDS)

An IDS is an acronym for an intrusion detection system, which is focused on behavior detection to handle anomalies or attacks [42]. While IDS deals with the network's behavior in terms of attacks, it is established to learn from this behavior pattern using several techniques. Thus, the system will recognize any violated network traffic from this pattern that can lead to an attack on the infrastructure [42].

3.2. Supervised Models for IDS

While recent researchers have implemented various machine learning algorithms and techniques to build an efficient IDS, there are some challenges. It is considered risky for the research community to share network data because it may contain confidential or sensitive information. Thus, the lack of training data will impact the ML implementation to build an IDS system [42].

3.3. Ensemble Learning (EL)

The ensemble learning method is a way to view all the learning techniques and algorithms simultaneously rather than deploy them individually [17]. Recently, EL has been used for many predictions and forecasting applications, so it has been applied to address several complex issues. EL relies on a set of combined classifiers or predictors instead of single classifiers, so these sets of classifiers are trained and learned from the conducted patterns to address the same issue and get better results [17].

In addition, the ensemble methods that could be applied to IDS approaches are as follow:

3.3.1. Voting

Is every lower-level classifier for its prediction casts a vote, so the winner is the prediction with the most votes [58] .

3.3.2. Stacking

A learning method is employed to combine the predictions in the voting process. The resulting meta-level classifier is then utilized to construct the final prediction from the predictions of the basis classifiers [58] .

3.3.3. Boosting

Starts with the original data set and uses a learning algorithm to construct a classifier. A new classifier is built using the same learning process after increasing the weights of the mistakenly classified tasks. The method is repeated several times. The classifiers are then combined using weighted voting [58].

In the methodology section, we explained the ensemble learning approaches that applied to our proposed model with the selected ensemble methods.

4. Methodology

4.1. Experimental Environment

The framework was implemented using Jupyter Notebook as the development environment. Within the Anaconda environment, Jupyter Notebook is a widely used tool. Python was used to create the actual code for the scheme. Python was selected because of how effectively it works, how well it scales, and how stable it is. In addition, Python offers a variety of useful metrics for evaluation, which were utilized here effectively.

While the proposed model includes several methods and algorithms, the study's dataset, feature selection techniques, classification algorithms, ensemble approach, and the selected ensemble methods are all detailed here.

We propose a binary classification of network traffic from the Internet of Things devices as normal or abnormal. We implemented feature selection algorithms to enhance the IDS performance in the IoT devices. Therefore, since our goal is to construct an effective IDS that can detect attacks on Internet of Things devices within the IoT network dataset, we examined a variety of machine learning algorithms to choose which models were the most accurate and efficient learner. Using the scaling of the data and the selection of features, we construct and evaluate four supervised models. RF, DT, LR, and KNN are the models that are included in the proposed model. In addition, to improve the effectiveness of our method for attack detection,we employ two ensemble methods to improve the efficiency of the proposed model. The performance evaluation includes accuracy, precision, recall, and F1-score metrics.

Furthermore, the main objective is to use a variety of supervised machine learning models to combine and input them into the ensemble model, which offers a solution for enhancing the IDS performance besides predicting malicious and normal IoT network traffic, where 0 denotes normal, and 1 means abnormal, using the features type, date, ts, time, label, longitude, altitude, and label. However, in the result section, we will illustrate the final result for all ML classifiers.

Moreover, Figure 1 illustrates the first workflow of the supervised ML classifiers applied to the IDS model. While it presents the preprocessing process and the methods used to fulfill all required steps to complete the preprocessing requirement, the proposed model used data cleaning, label encoding, and data normalization processes. In addition, Figure 1 illustrates the feature selection methods applied to the proposed model: Mutual information (MI), Pearson Coefficient Correlation (PCC), and K-Best feature. After we selected the proposed model features based on the feature selection methods, we trained the model to conduct the results. Finally, we will discuss the ML models, the preprocessing process, and feature selection methods applied to the proposed model.

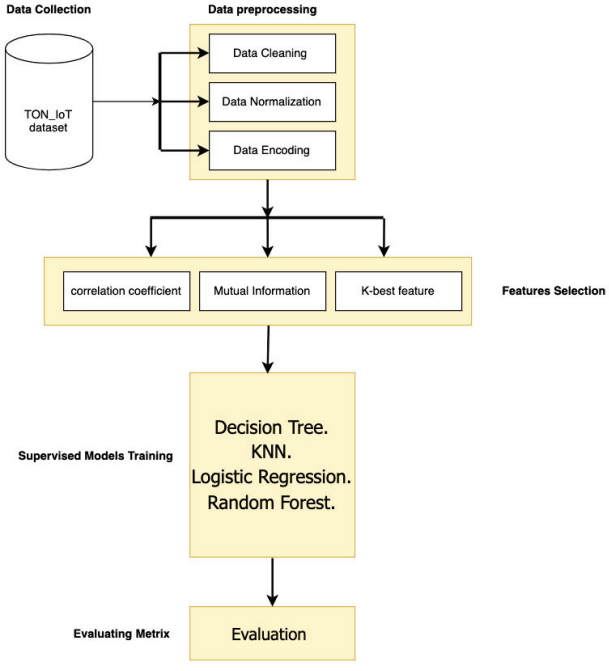


Figure 1. Illustration the workflow of the supervised ML classifiers that are applied to IDS.

4.1.1. Random forest (RF)

RF is described as a Random decision forest and a Machine learning algorithm that can be applied to several methods, such as ensemble learning. The samples for the forest decision [42] were generated by the RF, which is structured by constructing an unconnected tree and then centralizing it. Although RF is applied to deal with high-dimensional data, a significant difficulty with RF is that it has to cope with vast or huge data while being a viable technique to enhance accuracy by applying and merging numerous decision trees DTs [28]. Although it utilized memory and demanded additional time since it depends on a computational process, it gave high-accuracy results because it deals with dimensional data to improve performance.

4.1.2. Decision Tree (DT)

DT is a well-known Machine learning ML classification used to build an Intrusion detection system. DT contains three essential factors: the decision node, the branch, and leaf node. After the model was learned and trained with the dataset, the DT established and formed the decision tree [42]. DT is characterized by the fact that it can handle numerical and categorical features and recognize nonlinear relationships. Despite the simplicity of implementation of the DT, DT requires and consumes more storage capacity, which is considered an issue [59]. In our experiment, the DT presented high-accuracy results with straightforward implementation while requiring more space in memory. The mathematical representation of functionality is as follows:

$$G(D) = \Sigma_{I=1}^{C}(P(i) * (1 - P(i)) \quad (1)$$

The Gini impurity is a measure of the level of impurity or uncertainty in a training dataset (D) that is constructed from a set of class labels (C) and the fractions ($p(i)$) of samples that

are assigned to each class label (I) in the set. The Gini impurity is zero when there is just one class label in the set.

4.1.3. K-Nearest Neighbors (KNN)

KNN refers to K-Nearest Neighbors, where K is the number of the selected sample points. KNN is a supervised machine learning algorithm applied to a set of data points or a single one to make a classification or prediction. While KNN deals with regressions and classification issues, the KNN works based on the hypothesis that a point can be located near another with similarity [60]. Furthermore, the voting majority generates the class point, which appears most frequently in specified data points. When KNN is defined as a suitable method to handle multi-classification tasks, KNN has an issue: the values of K will be different from one dataset to another [60]. While KNN has influenced our model positively since it deals with complex decisions borderline, it consumes memory and requires more time since it relies on computational processes.

Moreover, KNN uses the Euclidean distance to calculate the spread between the nearest points to measure the KNN represented in the equation as [60]:

$$Euclideandistance = \sqrt{((a1 - a2)^2 + (b1 - b2))} \quad (2)$$

Which, $(a1-a2)$ is the first, and $(b1-b2)$ is the second.

4.1.4. Logistic Regression (LR)

The LR algorithm can be used to classify a set of discrete variables. The logistic sigmoid is the basis for logistic regression (LR). This technique predicts the probability value of a test sample that can be plotted to discrete types of two or more by transforming absolute values into values between 0 and 1 [42]. The corresponding transactions are classified as negative when the model predicts a less than 50 result. When the outcome is anticipated to be greater than 50, the corresponding transactions are classified as positive. is the equivalent mathematical expression [42]. The mathematical representation of functionality is as follows:

$$g(z) = \frac{(1)}{(1 + e^{-2})} \quad (3)$$

An estimated probability between 0 and 1 is represented by the function $g(z)$, where z is the input value, e is the base of the logarithm of nature, and z is the result.

4.2. Ensemble Learning (EL)

Instead of applying learning techniques and algorithms one at a time, EL allows you to view them simultaneously by combining a group of classifiers or predictors. These groups of classifiers are trained and learned from prior patterns of behavior to address complex issues [17,61]. The workflow of the proposed ensemble classifier approach is described in Figure 2 for any application.

The supervised model used the preprocessed dataset, as shown in Figure 1 in Section 4. While Figure 2 indicates that the stacking and voting ensemble classifiers were examined, it represents the second workflow of the proposed model. We used ensemble classifier prediction approaches to improve our system's accuracy. We aimed to enhance classification accuracy by merging many classifiers into a single ensemble platform. We will give each classifier's findings and explain and compare the two approaches in the results section. Figure 2 illustrates the workflow of the proposed ensemble classifiers.

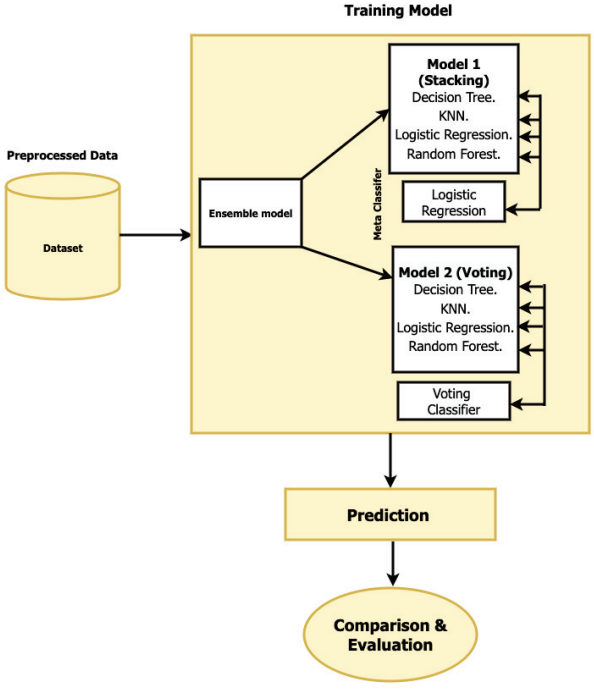


Figure 2. Illustration the workflow of the proposed ensemble classifiers.

Furthermore, We build and assess four supervised models using data scaling and the feature selection. This group of models consists of Random Forests, Decision Trees, Logistic Regression, and K-Nearest Neighbors to improve IDS detection capability. We employ two ensemble methods to enhance the attack detection efficiency of our scheme. This system uses ensemble classifiers based on classification algorithms. We chose stacking and voting as ensemble techniques since their predictions are weighted depending on the relevance of the individual classifiers. The weighted probabilities are then added together to get the total probability.

4.3. TON-IoT Network Dataset

The TON IoT network traffic dataset is acquired from real-world IoT network device (in-home) scenarios [18–25]. The TON-IoT dataset network was generated from UNDW Canberra Cyber, the Australian Defense Force Academy (ADFA), housed in the School of Engineering and Information Technology (SEIT) at the University of New South Wales (UNSW) in Canberra [20]. The TON-IoT datasets, collected from multiple IoT devices through IoT architecture, have been tested and authenticated in a cyber lab to be more reliable to be applied and implemented through machine learning techniques.

In this study, however, we present a binary classification of normal and abnormal IoT device network traffic. The proposed model presents a solution for improving IDS performance by constructing an efficient ensemble framework of supervised machine learning models.

Moreover, several datasets have been used for IDS domain development, such as BOT IOT (2018), UNSWNB (2015), and KDD-CUP99, the most common datasets. Still, these datasets have shown that a lack of various sensors and newly updated attacks may lead to a need for more data. Consequently, TON-IoT was selected for several IDS approaches, while it is publicly available and the most recent work to build an IDS. The TON means the

telemetry data, system log operating, and network traffic of the IoT network from which the dataset was collected. Thus, the dataset has multiple dataset files containing sensor telemetry data, so Table 4 presents details of the selected files in the proposed model.

Table 4. Details the selected files in the proposed model.

Datasets	Number of Features	Number of Instances	Features' Names	Types of Attacks
TON_IoT (IoT_Fridge).	7	59,944	Ts, date, time, fridge_temperature, temp_condition,type, and label.	Normal. Backdoor. DDoS. Injection. Password Ransomware. Scanning. XSS.
TON_IoT (IoT_Garage_Door).	15	409,963	ts, date, time, door_state, sph1e_signal, temp_c1diti1, l1gitude, light_status, current_temperature, thermostat_status, temperature, pressure, humidity, type, and lable	
TON_IoT (IoT_Thermostat).	7	52,774	Ts, date, time, current_temperature, thermostat_status, type and label.	
TON_IoT (IoT_GPS_Tracker).	7	58,960	Ts, date, time, latitude, longitude, type and label.	
TON_IoT (IoT_Motion_Light).	7	59,488	Ts, date, time, motion_status, light_status, type and label.	
TON_IoT (IoT_Weather).	8	59,260	Ts, date, time, temperature, pressure, humidity, type and label.	
Combined_dataset	20	700,389	ts, date, time, fridge_temperature, temp_condition, label, type, door_state, sph1e_signal, temp_c1diti1, l1gitude, light_status, current_temperature, thermostat_status, temperature, pressure, humidity, latitude, longitude, moti1_status	

In the proposed model, we have combined all the datasets mentioned above into one dataset called the combined dataset to establish the experiment and presented model. In contrast, the TON-IoT dataset contains telemetry data in all dataset files representing various IoT devices and realistic IoT network traffic with different attack scenarios. In addition, Table 5 Presents statistical details of the attack types in the combined dataset.

Table 5. Presents statically details of the attacks types [42].

Data No.	Attack Types							Attacks Totals	Normal Totals
	Scanning	DDOS	Ransomware	Backdoor	Injection	XSS	Password		
1	Nil	5000	2902	5000	5000	2042	5000	24,944	35,000
2	529	5000	2902	5000	5000	1156	5000	24,587	35,000
3	61	Nil	2264	5000	5000	449	5000	17,774	35,000
4	550	5000	2833	5000	5000	577	5000	23960	35,000
5	1775	5000	2264	5000	5000	449	5000	24,488	35,000
6	529	5000	2865	5000	5000	866	5000	24,260	35,000
7	3444	25,000	13,990	30,000	30,000	5539	30,000	185,494	190,710

1. TON-IoT (IoT-Fridge). 2. TON-IoT (IoT-Garage-Door). 3. TON-IoT (IoT-Thermostat). 4. TON-IoT (IoT-GPS-Tracker). 5. TON-IoT (IoT-Motion-Light). 6. TON-IoT (IoT-Weather). 7. Combined data.

4.4. Data Preprocessing

4.4.1. Data Cleaning

Several individual dataset files exist within the TON-IoT dataset network; we merged them into one single document called the combined dataset. Even though the new, larger

file resulted from the merge, we still had to clean up some unexpected information, so random data as '-' has been removed from several features. Therefore, the median value process was utilized to recalculate values for each feature in place of the extracted and missing values. The median was chosen as the measurement method instead of the mean because it is more robust against outlier errors [32].

4.4.2. Label Encoding

In addition, many ML classifiers work exclusively with continuous values; hence, features with categorical values must be transformed into continuous ones. Since the combined dataset has several categorical features required to be transformed into numerical values, we employed the Label Encoding approach. Label Encoding was chosen because it does not increase the number of features or the computational complexity of the modeling process. Both are important considerations given the time needed for classification [32,42]. The natural values of true, false, off, and low were consequently converted to zero or one.

4.4.3. Data Standardization and Normalization

Moreover, most classification algorithms function more effectively when features are of comparable magnitude, since this helps to lessen the bias toward traits with high multiplicity values in the prediction results [32]. Although the combined dataset has a variety of attribute values, the fact that several features contain values in a wide range (from zero to hundreds to thousands) would negatively impact the model's performance and produce erroneous results. Therefore, standardization and normalization for scaling the features are two methods for dealing with this problem. As a result, we used min-max scaling for several features in our investigation and provided a model. The following equation represents the Min-Max scaling method [32]:

$$a_{normlized} = \frac{(^{a}-^{a}min)}{(^{a}max -^{-a} min)} \quad (4)$$

The dataset maintains attribute 'a' values. 'amax' and 'amin' indicate attribute 'a"s greatest and lowest values. Data separation preceded feature scaling. To preserve the prediction accuracy of trained models, the normalized data in the training set was kept hidden from the test data [32].

4.4.4. Data Splitting

It is necessary to note that the data has been partitioned per the feature mapping. The dataset was split into training and testing subsets using stratification and randomization, with the same 70:30% split of class types as the original dataset. Predictive models were developed using the training data, and the best intrusion detection model was selected based on its performance in the testing set.

4.5. Features Selections

Additionally, experiments are run to evaluate various feature selection algorithms and determine which ones are most effective for detecting anomalies. The training process can be simplified through feature selection by eliminating irrelevant or redundant data points from the collection [32]. Mutual information, Pearson Coefficient Correlation, and K-Best are the most popular and common feature selection methods that are applied for various IDS models:

4.5.1. Mutual Information (MI)

Is the one that has been modified the most frequently in practice. A reduction in uncertainty for one variable is predicted by this formula when the other variable has a known value. The value of mutual information indicates a stronger relationship between the two variables if it is greater than zero [32]. A result of 0 from a calculation including

both variables suggests that they are unrelated. Thus, the mathematical representation of the MI is as follows [32]:

$$IG(X|Y) = H(X) - H(X|Y) \tag{5}$$

where H(x) is the the process of the independent variable x, given by

$$H(X) = -\Sigma_i P(x_i) log_2(P(x_i)) \tag{6}$$

as well as the formula for H(x | y):

$$H(x|y) = -\Sigma_i P(y_i) \Sigma_i P(x_i|y_i) log_2(P(x_i|y_i)) \tag{7}$$

4.5.2. Pearson Coefficient Correlation (PCC)

Represents a standard to evaluate the degree of a statistical association between two independent variables [32]. Its value, which can range from −1 to 1, and its sign, which can be negative or positive, indicate the type of significance between two attribute vectors. This formula is used to determine PCC [32]:

$$r = \frac{\Sigma(x - x^-)(y - y^-)}{\sqrt{\Sigma(x - x^-)^2 + \Sigma(y - y^-)^2}} \tag{8}$$

The symbol r denotes Pearson's correlation coefficient (PCC). The variable x represents the conditional feature, and the value y represents the decision feature. In addition, we refer to the sample means for the x and y characteristics as x and y, respectively.

4.5.3. K-Best Feature

Each feature's value is computed, then the chi-squared function is applied to those values, and finally, the top feature's value is ranked according to the k we choose. The chi-squared statistic [62] is used in feature selection to determine whether or not two features are independent of a given class. This equation is written mathematically as [32]:

$$x^2(f, c) = \frac{m * (PQ - RT)^2}{(P + x) + (P + Q) + (P + R) + (T + Q)} \tag{9}$$

where f stands for a single feature, c for a class, M for the entire dataset's total number of occurrences, P is the frequency, f occurs in c, and Q is the frequency neither f nor c does. The frequency with which class c occurs without feature f, denoted by R, and class f without class c, represented by T, are inversely proportional.

Furthermore, feature selection uses chi-squared statistics. This statistic determines if two class-related features are independent. The chi-squared test determines a feature's importance to the target class. This equation's numerical address can determine the feature's importance. Thus, the highest-scoring features are modeled. Chi-squared feature selection improves model efficiency in several ways: Removing unneeded or duplicate features reduces dimensionality during feature selection. This reduces overfitting and improves model performance. Selecting the most relevant attributes helps researchers determine which data features best predict the target class. Readability may help uncover data patterns and correlations. Removing unnecessary characteristics might help the model focus on data values, improving generalization. This enhances the model's pattern recognition and prediction. Chi-squared-based feature selection helps us select the most relevant characteristics, minimize dimensionality, improve interpretability, and increase model generalization [63].

After we applied the above selection method to the combined dataset, we decided to select the most repeated features given high scores by the selection algorithms, and we called the selected models features (Final selection). Table 6 illustrates the features that each method has selected.

Table 6. Illustrates the features that each method has selected and the final selection.

Feature Selection Technique	Selected Features
Mutual information (MI)	type, date, ts, time, and lable
Pearson Coefficient Correlation (PCC)	date, l1gitude, light_status, longitude, moti1_status, sph1e_signal, temp_condition, and type, lable.
K-Best feature	type, date, ts, time, logtiude, and iltitude.
Final selection	type, date, ts, time, lable, logtiude, iltitude nad lable

5. Results

This section compares the learning models and the experimental results with traditional performance metrics like F1-score, recall, accuracy, and precision. The models created after the ML bais models algorithms have been trained and evaluated based on standard key evaluation parameters [64]:

1. True positive (TP): The method correctly identified and categorized malicious attempts across many samples [58].
2. True Negative (TN): The algorithm accurately categorizes the proportion of normal samples [58].
3. True positive (TP): The method correctly identified and categorized malicious attempts across many samples [64].
4. True Negative (TN): The algorithm accurately categorizes the proportion of normal samples [64].

The performance measures generated from the parameters above are described as follows:

5.1. Accuracy

The accuracy metric is necessary to determine how well the model is doing. On the other hand, it is only usable with data that is evenly distributed [58,64]. It is computed as the ratio of correctly predicted occurrences to the total test samples. The mathematical representation of the equation is as follows [64]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

5.2. Precision

The percentage of accurate predictions that turned out to be accurate [58]. In other words, the proportion of positively identified samples that were correctly classified (TP) compared with the balance of positively identified samples that were mistakenly classified (but were still positively identified) [64]. The mathematical representation of the equation is as follows:

$$Precision = \frac{TP}{FP + TP} \tag{11}$$

5.3. Recall

The percentage of the positive sample count that has been correctly categorized and identified [64]. The equation is as follows:

$$Recall = \frac{TP}{FN + TP} \tag{12}$$

5.4. F1-Score

The F1-score, with its usual range of 1.0 to 0.0, can be used to determine the harmonic mean of precision and recall [64]. The F-1 score improves as the degree of accuracy and precision increases. The equation is as follows [58]:

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (13)$$

While we got our data from the TON-IoT dataset, as stated, we started the prepatory process to build the supervised machine learning models. In addition, four effective classifiers were employed to detect normal and abnormal IoT network traffic. LR, k-NN, RF, and DT were used. The primary goal is to combine them into the ensemble to enhance the IDS performance.

Moreover, Table 7 illustrates the evaluation results of the supervised models. LR has achieved the highest accuracy, 98.42%; the KNN has 98.28% accuracy; the RF has 98.15% accuracy; and the DT has 97.44% accuracy, which is considered the lower one, respectively.

Table 7. Evaluation results of the Supervised models.

Algorithm	Accuracy	Recall	Precision	F1-Score
LR	98.42%	98.42%	98.47%	98.42%
KNN	98.28%	98.28%	98.29%	98.30%
DT	97.44%	97.44%	97.44%	97.44%
RF	98.15%	98.15%	98.16%	98.15%

Our research aims to develop a binary classification based on an ensemble-based learning model to boost the IDS's efficiency further. The ensemble-based learning approach combined four supervised ML models (LR, KNN, DT, and RF). These models are accurate and well-balanced in terms of data, while LR, KNN, and RF stand out as having the highest accuracy. The advantages of the ensemble learning approach become apparent when considering how different learning strategies can mutually benefit one another. Stacking and voting are two methods of ensemble classifier models. Table 8 highlights the ensemble classifiers and the performance metrics for the two ensemble classifier models.

Table 8. Evaluation results of the ensemble classifiers.

Algorithm	Accuracy	Recall	Precision	F1-Score
Stacking	98.64%	98.60%	98.66%	98.61%
Voting	96.63%	96.59%	96.51%	96.60%

We chose stacking and voting as ensemble techniques since their predictions are weighted depending on the relevance of the individual classifiers. The weighted probabilities are then added together to get the total probability. Furthermore, stacking involves two steps. There are four model-based learners (LR, KNN, DT, and RF) in the first step; the next stage is to include one model in a meta-learner (Logistic Regression). Stacking uses these two stages to learn and identify the best approach to merging base and meta-learner models. Stacking outperformed Voting in all measures metrics.

However, Figure 3 illustrates the comparison process between the ensemble methods and the supervised ML models, which emphasizes that the proposed stacking model outperforms the Voting ensemble classifier, although the supervised ML models have achieved high accuracy. Stacking has the highest F1 scores (98.64%), Accuracy (98.64%), Recall (98.60%), and precision (98.20%).

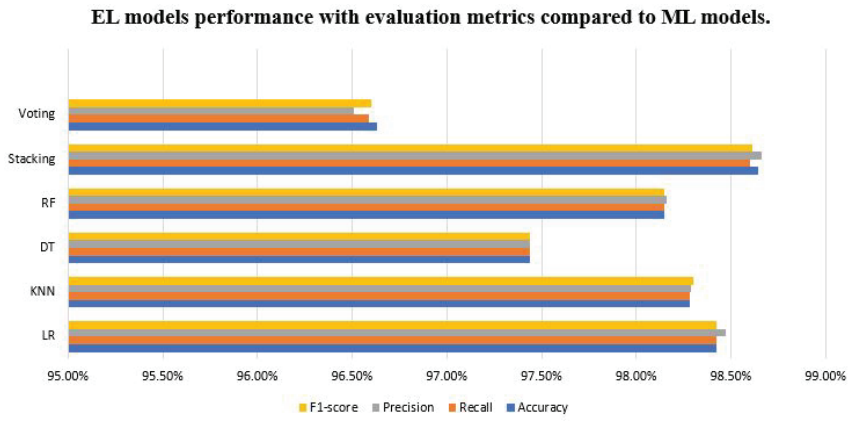


Figure 3. Compares the proposed and supervised ML models regarding evaluation metrics.

Table 3 presents recent machine-based learning models that used ensemble approaches in refs. [17,28,42] and were compared with our proposed scheme. Both of them used the same dataset. The proposed model performed better when measured against evaluation metrics for the same dataset. While the proposed IDS can detect attacks on IoT network traffic, the detection performance is improved based on the conducted results. The suggested ensemble models for binary classification showed performance and evaluation metrics improvements compared with the prior models.

5.5. Discussion

Table 3 compared the performance metrics of different approaches on the TON-IoT and CICIDS2017 datasets. Each approach utilizes different machine learning algorithms.

Moreover, the research [53] used the TON-IoT dataset, the Random Forest (RF), XGBoost (XGBoos), LightGBM (LGBM), and CatBoost algorithms were employed. The approach achieved recall, precision, F1-score, and accuracy in the range of 91% to 97%, 89% to 97%, 89% to 97%, and 89% to 97%, respectively. In the study [42] on the same TON-IoT dataset, the Logistic Regression (LR), Linear Discriminant Analysis (LDA), RF, Naive Bayes (NB), Support Vector Machine (SVM), and Long Short-Term Memory (LSTM) algorithms were utilized. The reported metrics for this approach were 85% recall, 87% precision, 86% F1-score, and 86% accuracy. In addition, the study [28] employed the CICIDS2017 dataset and used the Naive Bayes (NB), Decision Tree (DT), and Logistic Regression (LR) algorithms. Although specific values are provided for recall, precision, and F1-score, the reported accuracy was 88.92%.

The proposed models on the TON-IoT dataset employed the Random Forest (RF), K-Nearest Neighbors (KNN), Logistic Regression (LR), and Decision tree (DT) algorithms. The stacking and voting methods used in our model to generate these results. The evaluation metrics for our ensemble stacking model were 98.60% recall, 98.20% precision, 98.61% F1-score, and 98.63% accuracy. Our models are compared with recently proposed models that utilized the same dataset and similar approaches. Our models performed better at detecting attacks and presented an enhancement in the IDS performance, as shown in Tables 6–8. Precision is 98.2, recall is 98.60, the F1 score is 98.61, and accuracy is 98.63.

Moreover, stacking and voting were selected as ensemble methods because their predictions are weighted according to the importance of the individual classifiers. After assigning weights, probabilities can be totaled to generate an overall prediction. The proposed stacking model performs better than the proposed ensemble classifier and is highly accurate, although supervised ML models are also highly accurate. F1 (98.64%), Accuracy (98.60%), Recall (98.2%), and Precision (98.1%) are all best achieved by the stacking method. There are two phases of the Stack. In the first phase, we employ four model-based learners

(Logistic Regression, K-Nearest Neighbor, Distance-based, and Radial Basis Function). These two steps allow stacking to learn and determine the optimal method for combining base and meta-learner models. Compared with Voting, the Stacking method performed better on all measures and metrics.

These evaluation metrics have shown improvement over prior research. The proposed model uses RF, KNN, LR, and DT algorithms. Combining multiple algorithms into an "ensemble", the model uses their strengths to generate better outcomes than a single algorithm. The proposed model stacks many distinct models and integrates their predictions with a meta-model. The model can detect patterns and make more accurate predictions by combining basic model results.

Furthermore, we relied on the ensemble since the ensemble learning approach allows all learning techniques and algorithms to be evaluated simultaneously rather than individually. There has been an increase in the usage of EL for prediction and forecasting in recent years, and this has allowed it to be put to use in solving a number of challenging tasks. EL relies on a set of combined classifiers or predictors instead of single classifiers, so these sets of classifiers are trained and learned from the conducted patterns to address the same issue and get better results.

While we proposed a binary classification of IoT device network traffic as normal or abnormal, by combining the four supervised ML models, we employed two ensemble methods to improve the efficiency of the proposed model. The proposed method primarily focuses on the Internet of Things (IoT); however, it is noteworthy that the fundamental techniques and approaches are transferable to other network environments. The principles of supervised machine learning, ensemble classifiers, and enhancing detection efficacy are not exclusive to the Internet of Things (IoT) and can be customized to various network contexts.

Hence, the proposed approach is tailored to optimize Intrusion Detection System (IDS) efficacy in the Internet of Things (IoT). However, this approach's fundamental principles and methodologies can potentially be extrapolated and implemented in other network contexts. While we evaluated the proposed model in the IoT network traffic dataset, the proposed methodology may provide valuable perspectives and methods that have the potential to enhance intrusion detection across diverse network environments, extending beyond the boundaries of the Internet of Things.

6. Conclusions

This study proposes an IDS for IoT networks that employs machine learning to identify malicious behavior in IoT network data. The primary objective of this approach is to enhance the efficiency of intrusion and attack detection. This research compares and contrasts four supervised machine learning algorithms—Random Forest, Decision Tree, Logistic Regression, and K-Nearest Neighbor— to classify normal and abnormal Internet of Things network data. Logistic Regression and K-Nearest Neighbor classifiers provided the most optimal findings. Ensemble approaches, including voting and stacking, are used to improve classification by combining all supervised models. By working together, various ensemble learning methods can improve classification accuracy.

Ensemble learning has the advantage of combining various learning methods to support and reinforce one another in a classification task. This means that the efficiency of the machine learning models is greatly improved by the ensemble classifiers, which perform better than the individual supervised classifiers. Performance measures highlight stacking over voting, with improved accuracy, recall, and F1-Score. In particular, the stacking model performs exceptionally well in terms of recall. The stacking method has an F1 score of 0.986 and scores of 0.9864 for accuracy, 0.9864 for precision, 0.9864 for recall, and 0.9864 for both. Compared with the previously used models, these enhancements constitute a major advance forward. Moreover, as shown in Table 8, stacking shows improvement in accuracy, precision, and F1-score compared with the models referenced in [28,42,53]. Our objective for the future is to develop an innovative multi-classification approach that can detect and classify anomalies and intrusions in IoT network traffic.

Author Contributions: Conceptualization, Y.A. and M.I.; methodology, Y.A.; software, Y.A.; validation, Y.A. and M.I.; formal analysis, Y.A. and M.I.; investigation, Y.A. and M.I.; writing—original draft preparation, Y.A.; writing—review and editing, M.I.; visualization, Y.A.; supervision, M.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available upon request.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Tyagi, H.; Kumar, R. Attack and Anomaly Detection in IoT Networks Using Supervised Machine Learning Approaches. *Rev. D'Intelligence Artif.* **2021**, *35*, 11–21. [CrossRef]
2. Thamilarasu, G.; Chawla, S. Towards deep-learning-driven intrusion detection for the internet of things. *Sensors* **2019**, *19*, 1977. [CrossRef] [PubMed]
3. Tama, B.A.; Rhee, K.H. Attack classification analysis of IoT network via deep learning approach. *Res. Briefs Inf. Commun. Technol. Evol.(ReBICTE)* **2017**, *3*, 1–9.
4. Abbood, Z.A.; Khaleel, I.; Aggarwal, K. Challenges and future directions for intrusion detection systems based on AutoML. *Mesopotamian J. CyberSecurity* **2021**, *2021*, 16–21.
5. Hephzipah, J.J.; Vallem, R.R.; Sheela, M.S.; Dhanalakshmi, G. An efficient cyber security system based on flow-based anomaly detection using Artificial neural network. *Mesopotamian J. Cybersecur.* **2023**, *2023*, 48–56. [CrossRef]
6. Luo, C.; Tan, Z.; Min, G.; Gan, J.; Shi, W.; Tian, Z. A novel web attack detection system for internet of things via ensemble classification. *IEEE Trans. Ind. Inform.* **2020**, *17*, 5810–5818. [CrossRef]
7. Lin, M.S.; Chiu, C.Y.; Lee, Y.J.; Pao, H.K. Malicious URL filtering—A big data application. In Proceedings of the 2013 IEEE International Conference on Big Data, IEEE, Silicon Valley, CA, USA, 6–9 October 2013; pp. 589–596.
8. Haji, S.H.; Ameen, S.Y. Attack and anomaly detection in iot networks using machine learning techniques: A review. *Asian J. Res. Comput. Sci.* **2021**, *9*, 30–46. [CrossRef]
9. Da Costa, K.A.; Papa, J.P.; Lisboa, C.O.; Munoz, R.; de Albuquerque, V.H.C. Internet of Things: A survey on machine learning-based intrusion detection approaches. *Comput. Netw.* **2019**, *151*, 147–157. [CrossRef]
10. Hussain, F.; Hussain, R.; Hassan, S.A.; Hossain, E. Machine learning in IoT security: Current solutions and future challenges. *IEEE Commun. Surv. Tutorials* **2020**, *22*, 1686–1721. [CrossRef]
11. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J. Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity* **2019**, *2*, 1–22.
12. Hasan, M.; Islam, M.M.; Zarif, M.I.I.; Hashem, M. Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. *Internet Things* **2019**, *7*, 100059. [CrossRef]
13. Liu, X.; Liu, Y.; Liu, A.; Yang, L.T. Defending ON–OFF attacks using light probing messages in smart sensors for industrial communication systems. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3801–3811. [CrossRef]
14. Khraisat, A.; Alazab, A. A critical review of intrusion detection systems in the internet of things: Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. *Cybersecurity* **2021**, *4*, 1–27. [CrossRef]
15. De Souza, C.A.; Westphall, C.B.; Machado, R.B. Two-step ensemble approach for intrusion detection and identification in IoT and fog computing environments. *Comput. Electr. Eng.* **2022**, *98*, 107694. [CrossRef]
16. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
17. Rani, D.; Gill, N.S.; Gulia, P.; Chatterjee, J.M. An Ensemble-Based Multiclass Classifier for Intrusion Detection Using Internet of Things. *Comput. Intell. Neurosci.* **2022**, *2022*, 1668676. [CrossRef] [PubMed]
18. Moustafa, N. A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets. *Sustain. Cities Soc.* **2021**, *72*, 102994. [CrossRef]
19. Booij, T.M.; Chiscop, I.; Meeuwissen, E.; Moustafa, N.; den Hartog, F.T. ToN_IoT: The role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets. *IEEE Internet Things J.* **2021**, *9*, 485–496. [CrossRef]
20. Alsaedi, A.; Moustafa, N.; Tari, Z.; Mahmood, A.; Anwar, A. TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems. *IEEE Access* **2020**, *8*, 165130–165150. [CrossRef]
21. Moustafa, N.; Keshky, M.; Debiez, E.; Janicke, H. Federated TON_IoT Windows datasets for evaluating AI-based security applications. In Proceedings of the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE, Guangzhou, China, 29 December 2020–1 January 2021; pp. 848–855.
22. Moustafa, N.; Ahmed, M.; Ahmed, S. Data analytics-enabled intrusion detection: Evaluations of ToN_IoT linux datasets. In Proceedings of the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE, Guangzhou, China, 29 December 2020–1 January 2021; pp. 727–735.

23. Moustafa, N. New generations of internet of things datasets for cybersecurity applications based machine learning: TON_IoT datasets. In Proceedings of the eResearch Australasia Conference, Brisbane, Australia, 21–25 October 2019; pp. 21–25.
24. Moustafa, N. A systemic IoT–fog–cloud architecture for big-data analytics and cyber security systems: A review of fog computing. In *Secure Edge Computing*; CRC Press: Boca Raton, FL, USA, 2021; pp. 41–50.
25. Ashraf, J.; Keshk, M.; Moustafa, N.; Abdel-Basset, M.; Khurshid, H.; Bakhshi, A.D.; Mostafa, R.R. IoTBoT-IDS: A novel statistical learning-enabled botnet detection framework for protecting networks of smart cities. *Sustain. Cities Soc.* **2021**, *72*, 103041. [CrossRef]
26. Khan, M.A.; Khan Khattk, M.A.; Latif, S.; Shah, A.A.; Ur Rehman, M.; Boulila, W.; Driss, M.; Ahmad, J. Voting classifier-based intrusion detection for iot networks. In *Advances on Smart and Soft Computing: Proceedings of ICACIn 2021*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 313–328.
27. Batool, S.; Saqib, N.A.; Khattack, M.K.; Hassan, A. Identification of remote IoT users using sensor data analytics. In Proceedings of the Advances in Information and Communication: Proceedings of the 2019 Future of Information and Communication Conference (FICC), San Francisco, CA, USA, 14–15 March 2019; Springer: Berlin/Heidelberg, Germany, 2020; Volume 1, pp. 328–337.
28. Abbas, A.; Khan, M.A.; Latif, S.; Ajaz, M.; Shah, A.A.; Ahmad, J. A new ensemble-based intrusion detection system for internet of things. *Arab. J. Sci. Eng.* **2021**, *47*, 1805–1819. [CrossRef]
29. Kumari, A.; Mehta, A.K. A hybrid intrusion detection system based on decision tree and support vector machine. In Proceedings of the 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA), IEEE, Greater Noida, India, 30–31 October 2020; pp. 396–400.
30. Tomer, V.; Sharma, S. Detecting iot attacks using an ensemble machine learning model. *Future Internet* **2022**, *14*, 102. [CrossRef]
31. Fitni, Q.R.S.; Ramli, K. Implementation of ensemble learning and feature selection for performance improvements in anomaly-based intrusion detection systems. In Proceedings of the 2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT), IEEE, Bali, Indonesia, 7–8 July 2020; pp. 118–124.
32. Guo, G. A Machine learning framework for intrusion detection system in IoT networks using an ensemble feature selection method. In Proceedings of the 2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), IEEE, Vancouver, BC, Canada, 27–30 October 2021; pp. 0593–0599.
33. Churcher, A.; Ullah, R.; Ahmad, J.; Ur Rehman, S.; Masood, F.; Gogate, M.; Alqahtani, F.; Nour, B.; Buchanan, W.J. An experimental analysis of attack classification using machine learning in IoT networks. *Sensors* **2021**, *21*, 446. [CrossRef] [PubMed]
34. Abu Al-Haija, Q.; Al-Badawi, A. Attack-Aware IoT network traffic routing leveraging ensemble learning. *Sensors* **2021**, *22*, 241. [CrossRef] [PubMed]
35. Namasudra, S.; Devi, D.; Choudhary, S.; Patan, R.; Kallam, S. Security, Privacy, Trust, and Anonymity. In *Advances of DNA Computing in Cryptography*; Chapman and Hall/CRC: New York, NY, USA, 2018; p. 137.
36. Pavithran, P.; Mathew, S.; Namasudra, S.; Singh, A. Enhancing randomness of the ciphertext generated by DNA-based cryptosystem and finite state machine. *Clust. Comput.* **2023**, *26*, 1035–1051. [CrossRef]
37. Verma, R.; Kumari, A.; Anand, A.; Yadavalli, V. Revisiting shift cipher technique for amplified data security. *J. Comput. Cogn. Eng.* **2022**. [CrossRef]
38. Gutub, A. Boosting image watermarking authenticity spreading secrecy from counting-based secret-sharing. *CAAI Trans. Intell. Technol.* **2022**. [CrossRef]
39. Namasudra, S.; Roy, P.; Balusamy, B.; Vijayakumar, P. Data accessing based on the popularity value for cloud computing. In Proceedings of the 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), IEEE, Coimbatore, India, 17–18 March 2017; pp. 1–6.
40. Chen, Z. Research on internet security situation awareness prediction technology based on improved RBF neural network algorithm. *J. Comput. Cogn. Eng.* **2022**, *1*, 103–108.
41. Wani, A.; Khaliq, R. SDN-based intrusion detection system for IoT using deep learning classifier (IDSIoT-SDL). *CAAI Trans. Intell. Technol.* **2021**, *6*, 281–290. [CrossRef]
42. Naz, N.; Khan, M.A.; Alsuhibany, S.A.; Diyan, M.; Tan, Z.; Khan, M.A.; Ahmad, J. Ensemble learning-based IDS for sensors telemetry data in IoT networks. *Math. Biosci. Eng.* **2022**, *19*, 10550–10580. [CrossRef]
43. Alajanbi, M.; Ismail, M.A.; Hasan, R.A.; Sulaiman, J. Intrusion Detection: A Review. *Mesopotamian J. CyberSecurity* **2021**, *2021*, 1–4.
44. Abu Al-Haija, Q.; Al-Dala'ien, M. ELBA-IoT: An ensemble learning model for botnet attack detection in IoT networks. *J. Sens. Actuator Netw.* **2022**, *11*, 18. [CrossRef]
45. Abu Al-Haija, Q. Top-Down Machine Learning-Based Architecture for Cyberattacks Identification and Classification in IoT Communication Networks. *Front. Big Data* **2022**, *4*, 121. [CrossRef] [PubMed]
46. Kumar, P.; Gupta, G.P.; Tripathi, R. An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IoMT networks. *Comput. Commun.* **2021**, *166*, 110–124. [CrossRef]
47. Sarhan, M.; Layeghy, S.; Moustafa, N.; Portmann, M. Netflow datasets for machine learning-based network intrusion detection systems. In Proceedings of the Big Data Technologies and Applications: 10th EAI International Conference, BDTA 2020, and 13th EAI International Conference on Wireless Internet, WiCON 2020, Proceedings 10, Virtual Event, 11 December 2020; Springer: Berlin/Heidelberg, Germany, 2021; pp. 117–135.
48. Tsogbaatar, E.; Bhuyan, M.H.; Taenaka, Y.; Fall, D.; Gonchigsumlaa, K.; Elmroth, E.; Kadobayashi, Y. DeL-IoT: A deep ensemble learning approach to uncover anomalies in IoT. *Internet Things* **2021**, *14*, 100391. [CrossRef]

49. Chakraborty, D.; Narayanan, V.; Ghosh, A. Integration of deep feature extraction and ensemble learning for outlier detection. *Pattern Recognit.* **2019**, *89*, 161–171. [CrossRef]
50. An, N.; Ding, H.; Yang, J.; Au, R.; Ang, T.F. Deep ensemble learning for Alzheimer's disease classification. *J. Biomed. Inform.* **2020**, *105*, 103411. [CrossRef]
51. Zolanvari, M.; Teixeira, M.A.; Jain, R. Effect of imbalanced datasets on security of industrial IoT using machine learning. In Proceedings of the 2018 IEEE International Conference on Intelligence and Security Informatics (ISI), IEEE, Miami, FL, USA, 9–11 November 2018; pp. 112–117.
52. Samara, M.A.; Bennis, I.; Abouaissa, A.; Lorenz, P. A survey of outlier detection techniques in IoT: Review and classification. *J. Sens. Actuator Netw.* **2022**, *11*, 4. [CrossRef]
53. Rani, D.; Kaushal, N.C. Supervised machine learning based network intrusion detection system for Internet of Things. In Proceedings of the 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), IEEE, Kharagpur, India, 1–3 July 2020; pp. 1–7.
54. Zhou, Y.; Cheng, G.; Jiang, S.; Dai, M. Building an efficient intrusion detection system based on feature selection and ensemble classifier. *Comput. Netw.* **2020**, *174*, 107247. [CrossRef]
55. Ioannou, C.; Vassiliou, V. Network attack classification in IoT using support vector machines. *J. Sens. Actuator Netw.* **2021**, *10*, 58. [CrossRef]
56. Pham, N.T.; Foo, E.; Suriadi, S.; Jeffrey, H.; Lahza, H.F.M. Improving performance of intrusion detection system using ensemble methods and feature selection. In Proceedings of the Australasian Computer Science Week Multiconference, Brisband, Australia, 29 January–2 February 2018; pp. 1–6.
57. Yang, L.; Shami, A. A lightweight concept drift detection and adaptation framework for IoT data streams. *IEEE Internet Things Mag.* **2021**, *4*, 96–101. [CrossRef]
58. Danso, P.K.; Neto, E.C.P.; Dadkhah, S.; Zohourian, A.; Molyneaux, H.; Ghorbani, A.A. Ensemble-based Intrusion Detection for Internet of Things Devices. In Proceedings of the 2022 IEEE 19th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET), IEEE, Marietta, GA, USA, 19–21 December 2022; pp. 034–039.
59. Gad, A.R.; Nashat, A.A.; Barkat, T.M. Intrusion detection system using machine learning for vehicular ad hoc networks based on ToN-IoT dataset. *IEEE Access* **2021**, *9*, 142206–142217. [CrossRef]
60. Wang, A.X.; Chukova, S.S.; Nguyen, B.P. Ensemble k-nearest neighbors based on centroid displacement. *Inf. Sci.* **2023**, *629*, 313–323. [CrossRef]
61. Alkanjr, B.; Alshammari, T. IoBT Intrusion Detection System using Machine Learning. In Proceedings of the 2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC), IEEE, Las Vegas, NV, USA, 8–11 March 2023; pp. 886–892.
62. Polat, H.; Polat, O.; Cetin, A. Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models. *Sustainability* **2020**, *12*, 1035. [CrossRef]
63. Brownlee, J. *Data Preparation for Machine Learning: Data Cleaning, Feature Selection, and Data Transforms in Python*; Machine Learning Mastery: Vermont, Australia, 2020.
64. Alalwany, E.; Mahgoub, I. Classification of Normal and Malicious Traffic Based on an Ensemble of Machine Learning for a Vehicle CAN-Network. *Sensors* **2022**, *22*, 9195. [CrossRef] [PubMed]

MDPI AG
Grosspeteranlage 5
4052 Basel
Switzerland
Tel.: +41 61 683 77 34

Sensors Editorial Office
E-mail: sensors@mdpi.com
www.mdpi.com/journal/sensors

MDPI

Academic Open
Access Publishing

mdpi.com

ISBN 978-3-7258-3838-7