

Article

A Lightweight Security Authentication Scheme Based on Multiple Factors in the Industrial Internet

Hanjun Gao ¹, Haozhong Shi ¹, Shuai Jian ² and Gang Shen ^{2,3,4,*} 

¹ China Nuclear Power Operation Technology Corporation, Wuhan 430223, China; gaohj@cnnp.com (H.G.); shihz@cnnp.com (H.S.)

² School of Computer Science and Artificial Intelligence, Hubei University of Technology, Wuhan 430068, China; jianshuai@hbut.edu.cn

³ Hubei Provincial Key Laboratory of Green Intelligent Computing Power Network, Wuhan 430068, China

⁴ Hubei Provincial Engineering Research Center for Digital & Intelligent Manufacturing Technologies and Applications, Wuhan 430068, China

* Correspondence: shengang@hbut.edu.cn

Abstract

The authentication mechanism can ensure the security of data in the industrial Internet. However, existing authentication schemes mainly target traditional network entities and lack specific designs for AI service security, leaving models at risk of integrity breaches and malicious tampering. To address this, this paper proposes a novel three-party authentication scheme for the AI service environment. This scheme establishes secure two-way authentication among users, gateway nodes, and AI service providers, while ensuring user anonymity and forward security, and providing comprehensive resistance to diverse attacks such as insider attacks and DDoS. By organically combining biometric verification with lightweight cryptographic operations, our scheme maintains operational efficiency while enhancing security. A detailed analysis shows that its computational overhead is approximately 0.065 milliseconds and the communication overhead is 2304 bits, significantly outperforming existing schemes.

Keywords: three-party authentication; AI service security; industrial Internet

1. Introduction

Ensuring the legitimacy of data sources, the confidentiality and integrity of transmission, and the non-repudiation of the interaction process has become the cornerstone to guarantee the security and trustworthiness of the entire industrial Internet system, as well as the key to its development [1]. Against this backdrop, research on lightweight authentication and key negotiation protocols has shown a diversified development trend. In terms of device-layer authentication, Kumar et al. [2] proposed a lightweight mechanism based on hash functions and XOR operations, which achieved secure authentication between devices and servers through a sophisticated structural design. The advantage of this scheme lies not only in its low computational overhead but also in its successful passing of multiple formal verifications such as BAN logic, AVISPA, and ProVerif, demonstrating its ability to resist eavesdropping, man-in-the-middle attacks, and replay attacks. However, with the increasing complexity of industrial Internet application scenarios, especially in environments that require cross-domain access and multi-level permission management, this relatively simple two-party device-server authentication model has limitations. In this work, an AI service refers to a software service that provides inference or prediction capabilities through



Academic Editor: Andreas Mauthe

Received: 17 June 2026

Revised: 4 August 2026

Accepted: 5 August 2026

Published: 6 August 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

pre-trained machine learning models deployed on network-accessible nodes within industrial environments. AI service security requires that only authenticated and authorized entities may access the service, the executing model has not been tampered with or replaced, and the identity and trustworthiness of the service node can be cryptographically verified before sensitive industrial data is submitted.

To address this issue, researchers have started to explore more complex multi-factor authentication systems. Han et al. [3] conducted an in-depth analysis of the multi-factor authentication protocol proposed in [4], and found serious problems such as internal attack vulnerabilities and lack of forward security. Based on this, they propose an enhanced protocol that significantly improves security strength while maintaining the lightweight characteristics by improving key derivation mechanisms and temporary information processing methods.

In recent years, research on industrial Internet security protocols has started to deeply integrate with artificial intelligence technologies. Zhang et al. [5] explored a machine learning-based behavioral authentication mechanism, identifying potential threats by analyzing the communication patterns of devices. Meanwhile, Wang et al. [6] proposed a scheme that combines physical unclonable functions with lightweight cryptographic protocols, providing hardware-level security protection for resource-constrained devices. Chen et al. [7] studied the application of post-quantum cryptography in industrial Internet authentication, offering ideas for addressing future quantum-computing threats. Geng et al. [8] innovatively introduced blockchain technology into identity authentication and data interaction. Leveraging its decentralization and tamper-proof characteristics, they designed a phased key-generation authentication method based on Schnorr non-interactive zero-knowledge proof, providing new ideas for cross-domain collaboration scenarios. These studies have broadened the design ideas for industrial Internet authentication protocols, indicating that future protocol designs will pay more attention to collaborative innovation with technologies such as AI and blockchain.

1.1. Our Motivation

Despite significant progress in the lightweight and security aspects of industrial Internet authentication protocols in existing research, several key issues still urgently need to be addressed. Firstly, most protocols lack the adaptability to dynamic and heterogeneous network environments, making it difficult to maintain stable security performance under complex on-site working conditions. Secondly, existing solutions mostly focus on the authentication process itself and fail to deeply collaborate with the security requirements of upper-layer artificial intelligence applications, such as model security and data trustworthiness traceability.

Therefore, we propose a security scheme that adopts lightweight cryptographic operations and a multi-factor authentication mechanism. While ensuring the high efficiency of the protocol, we conduct a rigorous proof and analysis of its security based on formal methods. This scheme provides an authentication and key negotiation solution that balances security and practicality for accessing AI services in the industrial Internet environment.

1.2. Our Contribution

1. We propose a multi-factor authentication and key negotiation protocol specifically designed for the AI environment in the industrial Internet. This protocol skillfully combines symmetric encryption, hash operations, and XOR operations, significantly reducing computational and communication overheads while ensuring strong security. The protocol adopts a triple-authentication mechanism (smart card, password,

- biometrics) to ensure that only authorized users can access AI services, providing an ideal security solution for resource-constrained industrial environments.
2. We conduct a rigorous formal security analysis of the protocol through the real-or-random (ROR) model, establishing a reliable theoretical security foundation. At the same time, through a detailed security discussion, we verify the protocol's actual protection capabilities against various types of attacks, providing dual-layer security guarantees for industrial applications.
 3. We integrate the model fingerprint verification and dynamic verification mechanism into the industrial Internet authentication protocol to ensure the security and credibility of AI services in the intelligent analysis process.

1.3. Organization

The structure of this paper is arranged as follows: Section 2 reviews relevant work. Subsequently, Section 3 elaborates on the system and threat models, and Section 4 details the design of the scheme. Subsequently, in Section 5, we conduct a security analysis, and in Section 6, we assess it in terms of experimental performance. Finally, Section 7 concludes the paper.

2. Related Work

Research on security protocols in the context of the industrial Internet has evolved from traditional wireless sensor network security to specialized and scenario-specific solutions. Early studies were mostly based on relatively simple network models. The lightweight authentication scheme proposed in [9] laid the foundation for subsequent research. However, this scheme was soon found to have security flaws by Tai et al. [10]. With the increasing complexity of the industrial Internet architecture, authentication schemes in multi-gateway environments have become a research hotspot. A series of studies [11–13] have gradually refined the authentication mechanism in multi-gateway environments. However, there still remains the challenge of a trade-off between efficiency and security.

In the exploration of improving the security level, researchers have attempted various technical approaches. Public-key cryptography schemes, such as the elliptic-curve-based design by Li et al. [14], provide strong security guarantees but face the problem of excessive resource consumption. The introduction of blockchain technology, as in the scheme by Shen et al. [15], solves the cross-domain trust problem, yet its real-time performance can barely meet the requirements of industrial control. Biometric authentication, as studied by Das et al. [16], enhances the reliability of user authentication but increases the system complexity. These schemes jointly reveal the core contradiction in the design of industrial Internet security protocols: the balance between security strength and resource constraints.

Meanwhile, research on lightweight protocols based on symmetric cryptography has also been continuously deepened. The inter-machine communication authentication mechanism proposed by Esfahani et al. [17] demonstrates good practicality, and the multi-factor authentication scheme by Rafique et al. [4] represents the latest progress in this direction. However, as pointed out by Han et al. [3], lightweight does not equal security, and the design of protocols requires more rigorous formal verification. The work of Kumar et al. [2] sets an example in this regard. Their scheme provides reliable security guarantees for device-layer communication through strict formal proof.

Notably, Chen et al. [7] explored the application of post-quantum cryptography in industrial Internet authentication, providing ideas for addressing future quantum-computing threats. Li et al. [18] studied a distributed authentication architecture based on edge computing, effectively reducing the burden on the central server. In addition, the resource-efficient

authentication protocol proposed by Tanveer et al. [19] further reduced the computational overhead while maintaining security by optimizing the sequence of cryptographic operations. A device authentication system based on hashing and quantum encryption is proposed by Xu et al. [20]. This scheme enhanced privacy protection by using one-dimensional single-step quantum walks and achieved device authentication without prior knowledge through sequence randomization. The authors of [21] designed a lightweight anonymous authentication scheme that uses XOR and hashing operations, shifting the computational load to resource-rich nodes. Soleymani et al. [22] achieved data confidentiality, user anonymity, and mutual authentication based on an authenticated key-exchange protocol, relying on a trusted third party to achieve secure key sharing. Zargar et al. [23] proposed a lightweight two-way authentication protocol combining blockchain and three-factor authentication. It adopted elliptic curve cryptography and SHA-3 hashing, and underwent formal and informal security verification through AVISPA and BAN logic, demonstrating good security and efficiency. Cao et al. [24] provided a critical security analysis of a multi-factor authenticated key agreement scheme for IIoT, highlighting common vulnerabilities in such protocols. Their findings indicate the need for more robust designs like the one proposed in this work.

These studies provide important references for us to construct security protocols for artificial intelligence applications in the industrial Internet.

Beyond the authentication protocols discussed above, several complementary research areas address AI service security from different angles. Model watermarking and fingerprinting techniques embed ownership information into neural network weights, while trusted execution environments such as Intel SGX and ARM TrustZone provide hardware-enforced isolation for secure model serving. Zero trust architecture mandates continuous verification of every access request, and federated learning introduces authentication challenges in distributed training settings. Our work complements these approaches by focusing on lightweight authentication-layer integration of model integrity verification for resource-constrained IIoT devices; TEE-enhanced runtime attestation and comprehensive AI attack defenses are deferred to future work.

3. System Model and Adversary Model

This section covers the system model, threat model, and security objectives.

3.1. System Model

As shown in Figure 1, this system model consists of users, a gateway node (GWN), and AI service (AIS) nodes. They collaborate with each other to accomplish identity authentication and service access in a secure and efficient manner.

1. Users are untrusted operators in the industrial environment. They complete local identity verification through a triple-authentication mechanism of smart cards, passwords, and biometric features. Subsequently, they send requests to the gateway to access specific AI services. Eventually, they establish a secure session channel with the AI service nodes to obtain intelligent analysis results.
2. The GWN is responsible for managing the registered identities of all users and AI services. It enforces strict authentication and access-control policies, and generates and distributes key materials. Meanwhile, it protects users' privacy by maintaining and updating their temporary identities.
3. As a trusted service provider uniformly deployed by the system, the AI service node, upon verifying the legitimacy of requests and demonstrating the integrity of its own model, undertakes professional AI analysis tasks for users. It ensures the security and reliability of all subsequent data communications through the negotiated session key.

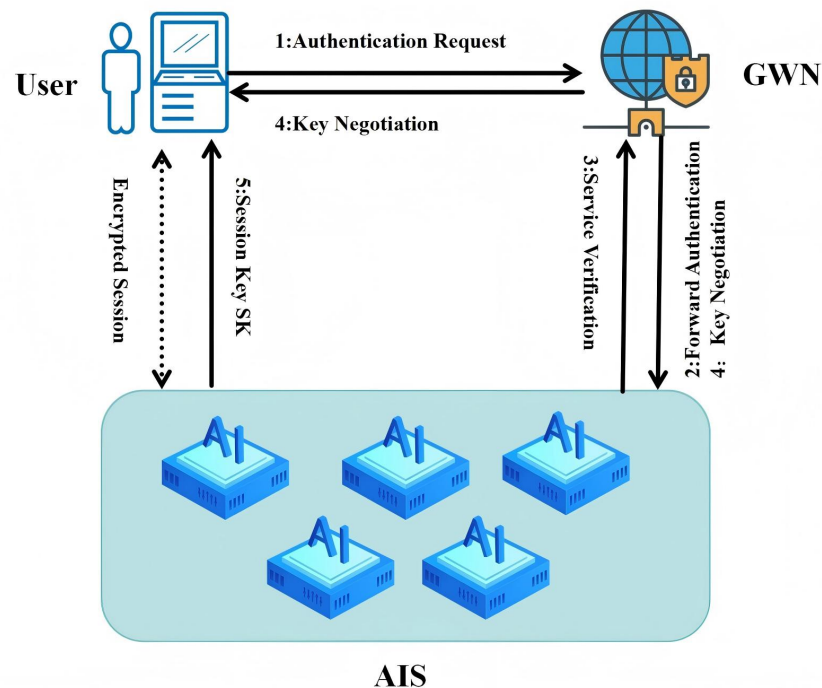


Figure 1. System model diagram.

3.2. Adversary Model

We assume that adversary $\mathcal{A}$ is a legitimate internal user who can send and receive messages. Their capabilities include: controlling the public channel (able to eavesdrop, intercept, and tamper with messages), impersonating any entity, obtaining participants' long-term private keys and random session numbers, attempting to obtain AI model parameters, launching model reverse-engineering attacks, and attempting data poisoning attacks.

3.3. Security Requirements

1. **Mutual Authentication:** Mutual identity authentication is required among users, the GWN, and AIS.
2. **Identity Confidentiality:** This protects users' real-identity information to prevent its leakage or tracking during communication.
3. **Session Key Security:** The protocol must ensure the negotiation of a unique and fresh session key and guarantee that the keys calculated by all parties are consistent.
4. **Forward Secrecy:** Even if the long-term key is compromised, the security of historical session keys should not be endangered.
5. **Resistance to Multiple Attacks:** The proposed scheme should be able to resist threats such as replay attacks, insider attacks, and smart-card-loss attacks.
6. **AI Services Security:** This ensures the integrity of the AI model, the trustworthiness of the service, and the reliability of analytical decisions.

4. Our Proposed Scheme

This section introduces the four phases of the proposed scheme in sequence. For better understanding, the core symbols involved in the scheme and their semantic explanations have been summarized in Table 1.

Table 1. Main symbols and descriptions

Symbol	Description
ID_{U_i}	Unique identity of user U_i
ID_{AIS}	Unique identity of AIS
ID_{GWN}	Identity of the gateway node
P_{U_i}	Password of user U_i
BF_{U_i}	Biometric feature of user U_i
BK_{U_i}	Biometric key generated from BF_{U_i}
sk_{GWN}	Secret key of gateway
sk_{U_i}	Secret key of user U_i
SK_{AIS-U_i}	Session key between AIS and user
PID_{U_i}	Pseudonym identity of user
m_{U_i}	Random nonce generated by user U_i
m_{GWN}	Random nonce generated by GWN
a_1, a_2, a_2^*	Random secrets selected by GWN for user registration
x	Temporary parameter, $x = B_{U_i} \oplus h(ID_{U_i} \parallel BK_{U_i})$
k_{AIS}	Secret key issued to AIS for key negotiation
s_{AIS}, f_{AIS}	Secret parameter pair for AIS identity verification
s_j, f_j	Secret parameters of the j -th AIS node: $s_j \in_R \mathbb{Z}_q^*$, $f_j = h(s_j \parallel r_j \parallel ID_{AIS_j} \parallel k_{AIS_j})$, with auxiliary secret r_j known only to GWN
λ_j	Weight factor, $\lambda_j = h(ID_{AIS_j} \parallel k_{AIS_j})$
θ_1, θ_2	Aggregated verification parameters, $\theta_1 = \sum \lambda_j s_j$, $\theta_2 = \sum \lambda_j f_j$
S	Verification result, returned as θ_1 after successful check
n	Total number of registered AIS nodes
FP_{model}	Model fingerprint
$Proof$	Model integrity proof
K_m	GWN-to-AIS verifier, $K_m = h(m_{GWN} \parallel T_2)$
$T_i (i = 1, \dots, 4)$	Timestamps
T_5	Timestamp for identity update confirmation
τ_{U_i}	Public reproduction parameter for biometric fuzzy extractor
$Enc_K(\cdot)$	Symmetric encryption
$Dec_K(\cdot)$	Symmetric decryption
$h(\cdot)$	One-way collision-resistant hash function
$Gen(\cdot), Rep(\cdot)$	Fuzzy extractor generation and reproduction functions
q	Large prime defining the finite field $\mathbb{Z}_q^*$

Cryptographic Primitive Instantiation. For concrete deployment, the abstract cryptographic operations in the protocol are instantiated with the following well-established, standardized primitives:

- Symmetric Encryption: $Enc_K(\cdot)/Dec_K(\cdot)$ is instantiated with AES-128 in Galois/Counter Mode (AES-128-GCM) as specified in NIST SP 800-38D. GCM provides authenticated encryption with associated data (AEAD), ensuring both confidentiality and integrity, which is necessary for resilience against active network adversaries.
- Hash Function: $h(\cdot)$ is instantiated with SHA-256 as specified in FIPS 180-4.

- Fuzzy Extractor: $Gen(\cdot)/Rep(\cdot)$ is instantiated with a standard fuzzy extractor construction.

These instantiations are provided as concrete deployment recommendations for reproducibility; the protocol's security arguments are parametric in the underlying primitives and hold for any secure instantiation satisfying the stated security definitions (IND-CPA + INT-CTXT for encryption, collision resistance for hashing, and information-theoretic security for the fuzzy extractor).

4.1. Registration Phase

1. User Registration:

- User U_i selects his unique identity ID_{U_i} and password P_{U_i} , and enters the biometric feature BF_{U_i} .
- U_i processes the biometric feature BF_{U_i} using a fuzzy extractor to generate the biometric key BK_{U_i} and the helper data τ_{U_i} : $(BK_{U_i}, \tau_{U_i}) = Gen(BF_{U_i})$.
- U_i calculates the temporary password $TP_{U_i} = h(ID_{U_i} \parallel P_{U_i} \parallel BK_{U_i}) \oplus x$. Then U_i sends $\{ID_{U_i}, TP_{U_i}\}$ and a registration request to the GWN through a secure channel.
- Upon receiving U_i 's registration request, the GWN randomly selects $a_1, a_2, sk_{GWN} \in \mathbb{Z}_q^*$, designates sk_{GWN} as its own secret key, and then computes:

$$sk_{U_i} = h(ID_{U_i} \parallel sk_{GWN} \parallel a_1), \quad (1)$$

$$A_{U_i} = h(ID_{U_i} \parallel P_{U_i} \parallel BK_{U_i}) \oplus sk_{U_i}, \quad (2)$$

$$V_{U_i} = h(ID_{U_i} \parallel A_{U_i} \parallel BK_{U_i} \parallel P_{U_i}). \quad (3)$$

The GWN stores $\{A_{U_i}, V_{U_i}, \tau_{U_i}, Gen(\cdot), Rep(\cdot), h(\cdot)\}$ in the smart card and issues it to U_i through a secure channel.

- The GWN stores (a_1, a_2) and (PID_{U_i}, A_{U_i}) in its own database, and writes $\{PID_{U_i}, A_{U_i}, V_{U_i}, \tau_{U_i}\}$ into the smart card SC_{U_i} .
- SC_{U_i} is delivered to U_i securely. The final stored values are:
Smart Card: $SC_{U_i}: \{PID_{U_i}, A_{U_i}, V_{U_i}, \tau_{U_i}\}$.
GWN Database: $\{a_1, a_2, PID_{U_i}, A_{U_i}\}$ per user.
AIS: $\{k_{AIS}, s_{AIS}, f_{AIS}, ID_{AIS}, FP_{model}\}$.

2. AIS Registration:

- The AIS generates its unique identity ID_{AIS} and prepares its service description information SD (such as service type, functional scope, performance indicators, etc.), and sends a registration request message $\{ID_{AIS}, SD\}$ to the GWN via a secure channel.
- Upon receiving the request, the GWN first verifies the legitimacy of the AIS and the compliance of its service description. After successful verification, the GWN generates a set of exclusive authentication parameters for the AIS, including the parameter k_{AIS} for key negotiation, and the secret parameter pair (s_{AIS}, f_{AIS}) for identity authentication.
- Specifically, for each registered AIS node j , the GWN selects two independent random numbers $s_j, r_j \in \mathbb{Z}_q^*$, and computes the cryptographic commitment:

$$f_j = h(s_j \parallel r_j \parallel ID_{AIS_j} \parallel k_{AIS_j}). \quad (4)$$

The weight factor λ_j for the j -th AIS node is computed as:

$$\lambda_j = h(ID_{AIS_j} \parallel k_{AIS_j}), \quad (5)$$

and normalized such that $\lambda_j \in \mathbb{Z}_q^*$. The auxiliary secret r_j is retained exclusively by the GWN and never disclosed to any AIS.

- (d) The GWN calculates the hash value of the core model of this AIS as its unique fingerprint, that is, $FP_{model} = h(\text{model parameters} \parallel \text{model architecture})$, and binds this fingerprint to the identity information of the AIS. We adopt a cryptographic hash of model parameters and architecture as the fingerprint for three reasons: it is lightweight, requiring only a single pass over model weights; its collision resistance guarantees that a different model produces a different fingerprint; and it is deterministic and reproducible. Model integrity during training is outside the scope of this protocol; we assume models are trained in a secure environment. When a model is legitimately updated, the AIS must re-register with the GWN to refresh the fingerprint by repeating the registration phase.
- (e) The GWN issues the generated authentication parameters $\{k_{AIS}, s_{AIS}, f_{AIS}\}$ to the AIS via a secure channel. Meanwhile, the GWN fully records the registration information $\{ID_{AIS}, FP_{model}, k_{AIS}\}$ of this AIS in its secure private database, thus completing the entire registration process. This procedure ensures that every AIS undergoes strict identity authentication and capability review before entering the system, and lays a trust foundation for subsequent three-party authentication.

4.2. Login Phase

1. U_i inserts SC_{U_i} , and inputs ID_{U_i} , $P_{U_i}^*$, and $BF_{U_i}^*$. The smart card computes:

$$BK_{U_i}^* = \text{Rep}(BF_{U_i}^*, \tau_{U_i}), \quad (6)$$

$$sk_{U_i}^* = A_{U_i} \oplus h(ID_{U_i} \parallel P_{U_i}^* \parallel BK_{U_i}^*), \quad (7)$$

$$V_{U_i}^* = h(ID_{U_i} \parallel A_{U_i} \parallel BK_{U_i}^* \parallel P_{U_i}^*). \quad (8)$$

2. The smart card verifies $V_{U_i}^* \stackrel{?}{=} V_{U_i}$. If equal, login succeeds and $sk_{U_i}^*$ is recovered; otherwise, the session is terminated. An incorrect password or biometric produces a wrong $sk_{U_i}^*$, causing all subsequent cryptographic operations to fail.

4.3. Authentication and Key Negotiation Phase

1. User Authentication Request and GWN Verification:

- (a) U_i selects the target AIS ID_{AIS} , picks a random number $m_{U_i} \in \mathbb{Z}_q^*$, and generates the current timestamp T_1 . Then, U_i calculates the encrypted message $M_1 = \text{Enc}_{sk_{U_i}}(m_{U_i})$ and the integrity verification value $M_2 = h(PID_{U_i} \parallel M_1 \parallel ID_{GWN} \parallel m_{U_i} \parallel T_1)$. Subsequently, an authentication request $\{PID_{U_i}, M_1, M_2, T_1, ID_{AIS}\}$ is sent to the GWN.
- (b) Upon receiving the request, the GWN first verifies the freshness of T_1 . Then, it retrieves PID_{U_i} through E_{U_i} and calculates $sk_{U_i} = h(ID_{U_i} \parallel sk_{GWN} \parallel a_1)$ to decrypt M_1 and obtain $m_{U_i}^* = \text{Dec}_{sk_{U_i}}(M_1)$. Finally, it verifies the integrity of M_2 to ensure that the message has not been tampered with. The GWN calculates $M_2^* = h(PID_{U_i} \parallel M_1 \parallel ID_{GWN} \parallel m_{U_i}^* \parallel T_1)$, and then verifies whether $M_2 \stackrel{?}{=} M_2^*$.

2. GWN Forwarding and AIS Verification:

- (a) After the GWN passes the verification, it selects $m_{\text{GWN}} \in \mathbb{Z}_q^*$, generates T_2 , and constructs:

$$M_3 = \text{Enc}_{k_{\text{AIS}}}(m_{\text{GWN}}, ID_{U_i}, ID_{\text{GWN}}, h(m_{\text{GWN}} \oplus sk_{U_i})), \quad (9)$$

$$M_4 = h(ID_{U_i} \parallel ID_{\text{GWN}} \parallel M_3 \parallel h(m_{\text{GWN}} \oplus sk_{U_i}) \parallel T_2). \quad (10)$$

The GWN computes the hash-based verifier $K_m = h(M_4 \parallel m_{\text{GWN}})$ and sends $\{M_3, M_4, K_m, T_2\}$ to the AIS.

- (b) After the AIS verifies the freshness of T_2 , it decrypts M_3 using k_{AIS} to obtain $(m_{\text{GWN}}^*, ID_{U_i}, ID_{\text{GWN}}, h(m_{\text{GWN}} \oplus sk_{U_i})) = \text{Dec}_{k_{\text{AIS}}}(M_3)$. It then verifies $M_4 \stackrel{?}{=} M_4$ and additionally checks $K_m \stackrel{?}{=} h(M_4 \parallel m_{\text{GWN}}^*)$ to confirm that the message originates from the legitimate GWN.

3. AIS Response and GWN Verification:

- (a) Upon successful verification by the AIS, it generates the current timestamp T_3 , calculates the model integrity proof $\text{Proof} = h(\text{FP}_{\text{model}} \parallel T_3)$, and constructs the response message $M_5 = \text{Enc}_{k_{\text{AIS}}}(s_{\text{AIS}}, f_{\text{AIS}}, ID_{\text{AIS}}, \text{Proof}, T_3)$, which is then sent to the GWN.
- (b) The GWN decrypts M_5 to obtain $(s_{\text{AIS}}, f_{\text{AIS}}, ID_{\text{AIS}}, \text{Proof}, T_3)$ and verifies the freshness of T_3 . It authenticates the AIS through the following steps:
- Model fingerprint verification: The GWN retrieves the stored fingerprint $\text{FP}_{\text{model}}^{\text{stored}}$ for this AIS and verifies $\text{Proof} \stackrel{?}{=} h(\text{FP}_{\text{model}}^{\text{stored}} \parallel T_3)$. This confirms that the AIS reports the same model fingerprint that was initially registered.
 - Parameter aggregation: The GWN computes the aggregated verification parameters using the secret pairs of all n registered AIS nodes:

$$\theta_1 = \sum_{j=1}^n \lambda_j s_j, \quad \theta_2 = \sum_{j=1}^n \lambda_j f_j. \quad (11)$$

- Identity verification: For each registered AIS node j , the GWN independently verifies the hash commitment using the stored auxiliary secret r_j :

$$f_j \stackrel{?}{=} h(s_j \parallel r_j \parallel ID_{\text{AIS}_j} \parallel k_{\text{AIS}_j}). \quad (12)$$

Since r_j is known only to the GWN, any AIS attempting to forge parameters for another node cannot compute a valid f'_j without knowledge of the corresponding r_j . The hash-based construction eliminates the linear algebraic vulnerability identified by the reviewer while preserving defense-in-depth. If all verifications succeed, the GWN sets $S = \theta_1$ and proceeds.

4. Key Negotiation and Confirmation:

- (a) The GWN selects a random number $a_2^* \in \mathbb{Z}_q^*$, calculates $M_6 = h(S \parallel m_{\text{GWN}} \parallel \text{Proof})$, then generates a new temporary identity $PID_{U_i}^* = \text{Enc}_{sk_{\text{GWN}}}(ID_{U_i} \oplus a_2^*)$ and a timestamp T_4 .
- (b) Then, the GWN sends the key materials $\{M_6, T_4\}$ to the AIS and $\{M_7, M_8, M_9, T_4\}$ to U_i . Here, $M_7 = \text{Enc}_{sk_{U_i}}(m_{\text{GWN}}, m_{U_i}, M_6)$, $M_8 = h(ID_{U_i} \parallel h(m_{\text{GWN}} \oplus sk_{U_i}) \parallel T_4) \oplus PID_{U_i}^*$, and $M_9 = h(M_6 \parallel M_7 \parallel m_{U_i})$.
- (c) The AIS calculates the session key $SK_{\text{AIS}-U_i} = h(ID_{U_i} \parallel ID_{\text{GWN}} \parallel ID_{\text{AIS}} \parallel m_{\text{GWN}} \parallel M_6 \parallel h(m_{\text{GWN}} \oplus sk_{U_i}) \parallel \text{Proof} \parallel \text{"SK-AIS-Ui"})$ and sends the confirmation value $M_{10} = h(SK_{\text{AIS}-U_i} \parallel ID_{\text{GWN}} \parallel ID_{\text{AIS}} \parallel ID_{U_i})$ to U_i .

- (d) After verifying the timestamp, U_i decrypts M_7 to obtain $(m_{GWN}^*, m_{U_i}^*, M_6^*) = Dec_{sk_{U_i}}(M_7)$. After verifying the integrity of $M_9^* = h(M_6^* || M_7 || m_{U_i}^*)$, U_i calculates $SK_{AIS-U_i}^* = h(ID_{U_i} || ID_{GWN} || ID_{AIS} || m_{GWN}^* || M_6^* || h(m_{GWN}^* \oplus sk_{U_i}) || Proof || "SK-AIS-Ui")$ and completes the key consistency verification by comparing $M_{10}^* = h(SK_{AIS-U_i}^* || ID_{GWN} || ID_{AIS} || ID_{U_i})$.

Proof Delivery Chain. *Proof* is transmitted from the AIS to the GWN encrypted within M_5 . The GWN then embeds it into $M_6 = h(S || m_{GWN} || Proof)$, which is encapsulated in M_7 and sent to U_i . U_i decrypts M_7 to obtain M_6^* , and *Proof* is implicitly bound into the session key SK_{AIS-U_i} through M_6^* , providing implicit verification of AIS model integrity without requiring *Proof* to be transmitted in cleartext.

5. Identity Update and Confirmation:

- (a) U_i calculates the new temporary identity $PID_{U_i}^* = h(ID_{U_i} || h(m_{GWN} \oplus sk_{U_i}) || T_4) \oplus M_8$, encrypts it and sends $M_{11} = Enc_{sk_{U_i}}(PID_{U_i}^*)$ to the GWN for confirmation.
- (b) After successfully verifying M_{11} , the GWN updates the U_i identity information in the database, completing the secure update of the temporary identity and ensuring the privacy protection of the U_i identity.

Through strict message integrity verification, two-way identity authentication, and the temporary identity update mechanism, the entire authentication and key negotiation process provides comprehensive security guarantees for AIS access in the industrial Internet.

4.4. Secure Communication Phase

- U_i and the AIS establish a secure communication channel using the negotiated session key SK_{AIS-U_i} .
- All business data are encrypted and transmitted using $SK_{AIS-U_i}: U_i \rightleftharpoons AIS : Enc_{SK_{AIS-U_i}}(\text{business data})$.
- After the session ends, SK_{AIS-U_i} is discarded to ensure forward secrecy.

Discussion on Model Integrity Assurance. We note that the model fingerprint mechanism $Proof = h(FP_{\text{model}} || T_3)$ provides static integrity verification—it ensures that the AIS presents the same model fingerprint that was registered with the GWN. However, in the current design, a compromised AIS could retain the original fingerprint while executing a tampered model, since the verification relies solely on the AIS's own computation of the hash.

To achieve stronger runtime integrity guarantees, future work may integrate hardware-assisted trusted execution environments (TEEs) such as Intel SGX or ARM TrustZone, or adopt remote attestation protocols that enable the GWN to cryptographically verify the actual code and model executing within the AIS enclave. We therefore qualify our claim: the proposed mechanism ensures static model fingerprint verification, which is a necessary first step, while runtime model integrity attestation with hardware root of trust is deferred to future extensions. We emphasize that this mechanism provides static verification at authentication time—it confirms that the fingerprint matches the registered value, but does not constitute continuous runtime model attestation. Readers should not interpret this as real-time integrity monitoring.

4.5. Lifecycle Management Phases

Smart Card Revocation and Re-issuance. If U_i reports the smart card as lost or stolen, the GWN revokes the old credentials and initiates a fresh registration as described

in Section 4.1 with new random parameters a'_1, a'_2 , ensuring that the lost card cannot be used for authentication.

AIS Revocation. When an AIS node is decommissioned or its credentials are suspected of being compromised, the following steps take place:

1. The GWN removes the AIS entry $\{ID_{AIS}, FP_{model}, k_{AIS}\}$ from its database and revokes the corresponding (s_{AIS}, f_{AIS}) pair.
2. The aggregated verification parameters θ_1, θ_2 are recomputed without the revoked node's contribution. This requires $O(n)$ hash operations for n remaining AIS nodes, which is acceptable for periodic administrative operations.

Desynchronization Recovery. If M_{11} (the identity update confirmation) is lost due to network failure, U_i retains the old $PID_{U_i}^{old}$ while the GWN has already updated to $PID_{U_i}^{new}$. To recover M_{11} , the following steps are taken:

1. In the next authentication attempt, U_i includes both $PID_{U_i}^{old}$ and a flag indicating a pending update.
2. The GWN checks its update log: if the flag matches a pending update record, the GWN accepts $PID_{U_i}^{old}$ and re-initiates the identity update procedure.
3. This ensures liveness without requiring out-of-band recovery channels.

Discussion on the GWN as a Single Point of Compromise. We acknowledge that the GWN serves as a central trust anchor. Its compromise would allow the adversary to impersonate any AIS and issue fraudulent credentials. Mitigations include: deploying the GWN in a hardware security module (HSM); thresholdizing the GWN's master key across multiple independent nodes using Shamir secret sharing; and (3) integrating with a distributed trust framework for detection of anomalous GWN behavior. These enhancements are orthogonal to the core protocol design and represent directions for system-level hardening.

5. Security Proof and Analysis

In this section, the proposed scheme is evaluated from two aspects through formal security analysis and informal security discussion. The analysis results show that our scheme has sufficient security and robustness.

5.1. Security Proof

We conducted a formal analysis of the protocol using the RoR model. The security proof of the session key follows the analysis method of the security definition of indistinguishability under chosen-plaintext attack (IND-CPA) presented in the literature [4]. The detailed formal proof process is as follows.

Theorem 1. Let the advantage of the adversary $\mathcal{A}$ be ϵ , and the numbers of its hash queries, send queries, and AI queries be q_1, q_2 , and q_3 respectively. Let L be the bit-length of the biometric key BK_{U_i} , $|H|$ be the size of the output space of the hash function, and $|D|$ be the size of the password dictionary. Then the advantage of $\mathcal{A}$ in breaking the semantic security of the protocol P_s satisfies:

$$\epsilon_{\mathcal{A}}^{P_s}(\lambda) \leq 2\left(\frac{q_1^2}{|H|} + \frac{1}{q} \cdot \epsilon_{\mathcal{A}}^{IND-CPA}(\lambda)\right) + \frac{1}{2^{L-1}}\left(\frac{q_2}{|D|} + q_3\right). \quad (13)$$

In Formula (13), $\epsilon_{\mathcal{A}}^{IND-CPA}(\lambda)$ is the advantage of $\mathcal{A}$ in breaking the IND-CPA security of the symmetric encryption schemes $Enc_{sk_{U_i}}$ and Enc_{K_g} , and q is the size of the space of the random numbers m_{GWN} and m_{U_i} .

Proof. The proof of this scheme is completed through seven rounds of games ($Game_1$ to $Game_7$), and the process is similar to that in [4]. Let S_i denote that the adversary guesses the result correctly in $Game_i$. The following will analyze each round of the game one by one.

$Game_1$: This game simulates the actual scenario where $\mathcal{A}$ launches an attack on the real-world protocol P_s . The adversary makes a one-bit guess to attempt to distinguish the real session key, and its winning advantage is

$$\epsilon_{\mathcal{A}}^{P_s}(\lambda) = |2\Pr[S_1] - 1|. \quad (14)$$

$Game_2$: In this game, $\mathcal{A}$ can eavesdrop on all communication messages during login and authentication phases of the participants. Since all the transmitted messages are protected by hash functions or symmetric encryption, $\mathcal{A}$ cannot obtain any secret information through passive eavesdropping. Therefore,

$$\Pr[S_1] = \Pr[S_2]. \quad (15)$$

$Game_3$: In this game, $\mathcal{A}$ uses AIS for in-depth analysis of collected messages via AI-queries. It tries to identify patterns or weaknesses in the scheme, such as analyzing timestamp patterns, detecting the randomness of random numbers, or looking for hash value collisions. Because the scheme adopts robust cryptographic primitives (the timestamps meet $|T_i - T'_i| \leq \Delta T$ to guard against replay attacks, the random numbers have sufficient entropy, and the hash functions are collision-resistant and pseudo-random), AIS cannot extract useful information from the encrypted and hashed messages. Consequently,

$$|\Pr[S_2] - \Pr[S_3]| \leq \frac{q_3}{2^L}. \quad (16)$$

$Game_4$: In this game, $\mathcal{A}$ performs send queries and hash oracle queries, aiming to send forged messages and detect hash collisions. The messages incorporate random numbers m_{U_i} , m_{GWN} , the secret key sk_{U_i} , and timestamps to guarantee the randomness and timeliness of the messages. Based on the birthday paradox, the upper bound of the probability of hash collisions is

$$|\Pr[S_3] - \Pr[S_4]| \leq \frac{q_1^2}{2|H|}. \quad (17)$$

$Game_5$: In this game, $\mathcal{A}$ obtains the data in SC_{U_i} through queries. As a multi-factor authentication scheme, $\mathcal{A}$ also has to guess the password P_{U_i} and the biometric key BK_{U_i} . The scheme uses a fuzzy extractor to generate BK_{U_i} from biometric feature BF_{U_i} . The probability of a successful guess is bounded by

$$|\Pr[S_4] - \Pr[S_5]| \leq \frac{q_2}{2^L|D|}. \quad (18)$$

$Game_6$: In this game, $\mathcal{A}$ can steal the long-term master key sk_{GWN} of the GWN and $\{a_1, a_2, E_{U_i}, PID_{U_i}\}$ in the security database through queries. To obtain ID_{U_i} from $PID_{U_i} = Enc_{sk_{GWN}}(ID_{U_i} \oplus a_2)$, besides sk_{GWN} , the important parameter a_2 is also needed. And to calculate the user session key $sk_{U_i} = h(ID_{U_i} || sk_{GWN} || a_1)$, a_1 is required. Although $\mathcal{A}$ has obtained the database, it cannot associate E_{U_i} with a specific user, and PID_{U_i} is updated each time. Thus,

$$|\Pr[S_5] - \Pr[S_6]| \leq \frac{q_1^2}{2|H|}. \quad (19)$$

$Game_7$: In this game, $\mathcal{A}$ can obtain the AIS node data $\{k_{AIS}, s_{AIS}, f_{AIS}, ID_{AIS}\}$ through queries. To forge the correct message $M_5 = Enc_{K_s}(s_{AIS}, f_{AIS}, ID_{AIS}, Proof)$, it requires

$K_g = m_{GWN} \oplus a_1$, where m_{GWN} is a temporary random number and a_1 is a secret parameter of the GWN. The advantage of this game transformation is constrained by the IND-CPA security property of the encryption scheme Enc_{K_g} . Consequently,

$$|Pr[S_6] - Pr[S_7]| \leq \frac{1}{q} \cdot \epsilon_{\mathcal{A}}^{IND-CPA}(\lambda). \quad (20)$$

After all oracle simulations are completed, the coin-tossing test query is executed. $\mathcal{A}$ guesses the session key $SK_{AIS-U_i} = h(ID_{U_i} || ID_{GWN} || m_{GWN} || M_6 || h(m_{GWN} \oplus sk_{U_i}) || Proof)$. Since all secret information is protected, $\mathcal{A}$ can only guess randomly. Hence,

$$Pr[Succ_7] = \frac{1}{2}. \quad (21)$$

By combining the differences in advantages among various games, we get:

$$\begin{aligned} \epsilon_{\mathcal{A}}^{P_s}(\lambda) &= |2Pr[S_1] - 1| \\ &= |2Pr[S_1] - 2Pr[S_7]| \\ &\leq 2(|Pr[S_1] - Pr[S_2]| + |Pr[S_2] - Pr[S_3]| \\ &\quad + |Pr[S_3] - Pr[S_4]| + |Pr[S_4] - Pr[S_5]| \\ &\quad + |Pr[S_5] - Pr[S_6]| + |Pr[S_6] - Pr[S_7]|) \\ &\leq 2\left(0 + \frac{q_3}{2^L} + \frac{q_1^2}{2|H|} + \frac{q_2}{2^L|D|} + \frac{q_1^2}{2|H|} + \frac{1}{q} \cdot \epsilon_{\mathcal{A}}^{IND-CPA}(\lambda)\right) \\ &= 2\left(\frac{q_1^2}{|H|} + \frac{1}{q} \cdot \epsilon_{\mathcal{A}}^{IND-CPA}(\lambda)\right) + \frac{1}{2^{L-1}}\left(\frac{q_2}{|D|} + q_3\right). \end{aligned} \quad (22)$$

Since $\frac{2q_3}{2^L} = \frac{q_3}{2^{L-1}}$, the advantage is $2\left(\frac{q_1^2}{|H|} + \frac{1}{q} \cdot \epsilon_{\mathcal{A}}^{IND-CPA}(\lambda)\right) + \frac{1}{2^{L-1}}\left(\frac{q_2}{|D|} + q_3\right)$, which is consistent with the Theorem 1. $\square$

5.2. Security Analysis

1. **Mutual Authentication:** This scheme constructs a complete three-party two-way authentication system. The user calculates M_2 and verifies the GWN's response to confirm the authenticity of the GWN's identity. The GWN decrypts M_1 to recover the random number m_{U_i} and verifies the integrity of M_2 to authenticate the legitimacy of the user's identity. Between the GWN and AIS, by verifying the secret parameter pair (s_{AIS}, f_{AIS}) , calculating θ_1 and θ_2 , and verifying the mathematical relationship $\theta_1 \stackrel{?}{=} \theta_2$, and combining them with the model integrity proof *Proof*, the trustworthiness of the AIS is ensured. This multi-level and multi-factor authentication mechanism establishes a reliable identity verification foundation for the system.
2. **Identity Confidentiality:** This scheme adopts a dynamic temporary identity mechanism to provide comprehensive protection for user privacy. During the communication process, the encrypted PID_{U_i} is used to replace the real identity ID_{U_i} , ensuring that adversaries cannot directly obtain the user's identity information from the communication data. After each session ends, the system updates the temporary identity by calculating $PID_{U_i}^*$, effectively preventing the tracking of user behavior through identity information. In addition, all identity-related information is transmitted through secure encryption, such as M_3 . Combined with the dynamic changes in random number m_{U_i} and timestamp, the communication patterns in different sessions are made uncorrelated, providing multi-level privacy protection for user identities.
3. **Session Key Security:** This scheme ensures the security and reliability of the session key through a carefully designed composite key-generation mechanism. In each

session, fresh m_{U_i} and m_{GWN} are used, and the timestamp mechanism is incorporated to guarantee the uniqueness and timeliness of the key. Both the user and AIS calculate the session key SK_{AIS-U_i} using exactly the same parameter set, ensuring key consistency. Through the M_{10} verification mechanism, both parties can confirm the correctness of key calculation, thus establishing a secure and reliable encrypted communication channel. This design not only ensures the strength of the key but also guarantees the reliability of the key-negotiation process.

4. **Forward Secrecy:** This scheme achieves strong forward-secrecy through a key-separation architecture design. The session key SK_{AIS-U_i} is completely separated from the system's long-term key sk_{GWN} , ensuring that the calculation of the session key mainly depends on the temporary random numbers m_{GWN} and m_{U_i} rather than the long-term key. Even if an adversary manages to obtain the GWN's master key sk_{GWN} at a certain point in time, since the temporary parameters m_{GWN} and m_{U_i} in historical sessions only exist in memory during the session and are immediately destroyed after the session ends, the adversary cannot acquire these critical parameters. As a result, the adversary is unable to back-calculate any historical session keys. This design provides persistent protection for the historical communication content of the system.
5. **Resistance to Multiple Attacks:** This scheme has comprehensive anti-attack capabilities and constructs a deep-defense system through multi-level security mechanisms. The timestamp verification mechanism, combined with the use of random numbers, effectively resists replay attacks. The triple-binding mechanism of smart cards, passwords, and biometrics effectively defends against internal attacks and security threats caused by the loss of smart cards. The key materials go through multiple hash transformations to prevent KSSTI attacks. Message integrity verification can effectively prevent man-in-the-middle attacks. For DDoS attacks, the gateway verifies the freshness of the timestamp T_2 before performing expensive cryptographic operations: messages with stale or invalid timestamps are discarded early, preventing an attacker from exhausting the gateway's computational resources through replayed or fabricated requests. We note that this timestamp-based mechanism provides basic DoS mitigation through early discarding of replayed messages. It is not a comprehensive DDoS defense—full resource-exhaustion countermeasures (e.g., client puzzles, rate limiting) are deferred to future work.
6. **AI Services Security:** Aiming at the special security requirements of AISs, this scheme provides a complete security guarantee mechanism. Through the model fingerprint FP_{model} and the dynamically generated integrity proof *Proof*, it ensures that the AIS has not been tampered with during the operation. The mathematical verification based on secret parameters ($\theta_1^2 \stackrel{?}{=} \theta_2$) guarantees the legitimacy of AIS identity. The GWN conducts strict qualification reviews during the AIS registration phase and performs complete identity verification and integrity checks during each service access. By binding the session key to the integrity proof of the AIS, the reliability and credibility of the analysis and decision-making process are ensured, providing comprehensive security protection for AI applications in the industrial Internet environment.

6. Performance Analysis

In this section, we conducted a comprehensive performance evaluation of the proposed scheme.

6.1. Comparison of Security Features

We first compared the security features of our scheme with those of existing schemes. As shown in Table 2, the comparison results indicate that our scheme supports a wider

range of security requirements, especially those that are crucial for AI-oriented industrial environments. The evaluation of security features is based on the formal and informal analyses provided in each paper.

As shown in Table 2, Han et al. [3] has significantly improved upon Rafique et al. [4] by adding forward secrecy and the ability to resist insider attacks and KSSTI attacks. Kumar et al. [2] provides a solid foundation for device-layer authentication, performing especially well in resisting DDoS attacks. Esfahani et al. [17]’s scheme, as an earlier M2M authentication scheme, is lacking in some security features. However, none of the existing schemes consider the security of the AI service itself. Our scheme is the only one that provides a comprehensive security framework covering user devices, communication channels, and AISs, including model integrity verification and service credibility authentication. This makes it particularly suitable for the emerging AI-driven industrial Internet environment.

Table 2. Security feature comparison.

Security Feature	[2]	[3]	[4]	[17]	Our
Mutual Authentication	✓	✓	✓	✓	✓
Three-party Authentication	×	✓	✓	×	✓
User Anonymity/Identity Privacy	✓	✓	✓	×	✓
Session Key Agreement	✓	✓	✓	✓	✓
Forward Secrecy	✓	✓	×	✓	✓
Resistance to Replay Attacks	✓	✓	✓	✓	✓
Resistance to Insider Attacks	✓	✓	×	×	✓
Resistance to Smart-Card-Loss Attack	×	✓	✓	×	✓
Resistance to KSSTI Attack	×	✓	×	✓	✓
Resistance to Impersonation Attacks	✓	✓	✓	✓	✓
Resistance to Man-in-the-Middle Attacks	✓	✓	✓	✓	✓
Resistance to DDoS Attacks	✓	×	×	×	✓
AI Service Security	×	×	×	×	✓

6.2. Comparison of Computational Overhead

For a fair comparison, all cryptographic operation timings are measured using the MIRACL library running on a desktop computer featuring an Intel Core i5-12400F processor (2.50 GHz) and 32 GB of main memory. We only analyze the time consumed by each entity during the authentication phase, excluding the one-time overhead of the registration phase. The execution times of cryptographic operations are defined as follows:

- T_h : The running time of a one-way hash function. $T_h \approx 0.002$ ms.
- T_{sym} : The running time of symmetric encryption or decryption.
- $T_{sym} \approx 0.004$ ms. T_{fe} : The running time of fuzzy extractor operations. $T_{fe} \approx 0.005$ ms.

As can be seen from the data in Table 3, although the computational overhead of our scheme is higher than that of [2,17], this is a necessary cost for achieving three-party authentication and AI security functions. However, compared with similar three-party authentication schemes, our scheme shows significant advantages. It is 31.6% lower than [3] and 15.6% lower than [4]. This improvement mainly benefits from the optimized sequence of cryptographic operations and the efficient key management strategy, which improves computational efficiency while ensuring enhanced security.

Based on the comparison of computational overhead for n-time authentication shown in Figure 2, as the number of authentication times increases, the computational overhead

of each scheme shows a linear growth trend. In scenarios with multiple authentications, our scheme consistently maintains its computational efficiency advantage over similar three-party authentication schemes. This cumulative effect is of great significance in the long-running industrial Internet of Things environment, as it can effectively reduce the overall computational overhead of the system.

Table 3. Computation cost comparison in authentication phase.

Scheme	Entity	Hash (T_h)	Symmetric Encryption/Decryption (T_{sym})	Fuzzy Extractor (T_{fe})	Total Cost in Authentication Phase (ms)
[2]	User + Server	9	5	0	0.038
[3]	User + GWN + Sensor	21	12	1	0.095
[4]	User + GWN + Sensor	16	9	1	0.073
[17]	Device + Router	7	3	0	0.026
Ours	User + GWN + AIS	18	6	1	0.065

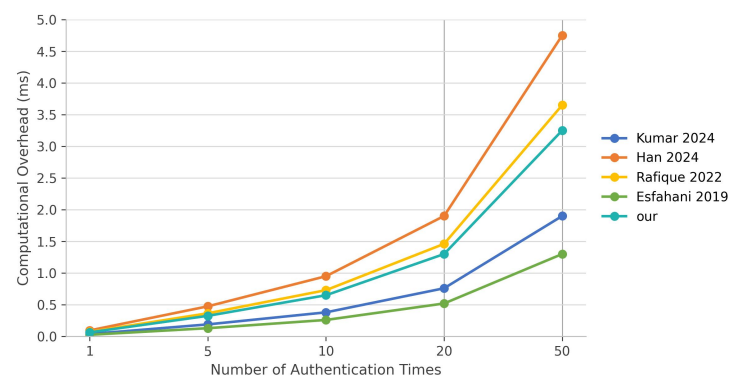


Figure 2. Computational overhead for n-time authentication [2–4,17].

6.3. Comparison of Communication Overhead

We calculated the total number of bits transmitted during the authentication and key-agreement phases. The bit-lengths of each parameter are as follows:

Symbol Explanation: T_{id} : Identity (128 bits); T_s : Timestamp (128 bits); T_n : Nonce or Secret Parameter (128 bits); T_h : Hash Output (256 bits); T_{sym} : Symmetric Ciphertext (128 bits); T_o : Other Specific Parameters (160 bits).

Table 4 shows that although the communication overhead of our scheme (2304 bits) is higher than that of the basic two-party authentication schemes [2,17], this is an inevitable requirement for supporting complete three-party authentication and the transmission of AI security parameters. However, it performs outstandingly among similar three-party authentication schemes, being 28.7% lower than [3] and 25.0% lower than [4]. This optimization stems from the streamlined message-round design and the compact parameter encoding scheme, which effectively controls the communication burden while maintaining all security functions.

According to the comparison of communication overhead for n-time authentication shown in Figure 3, our scheme demonstrates a continuous communication efficiency advantage in scenarios with multiple authentications. As the number of authentication times increases, the communication overhead advantage of our scheme over similar three-party authentication schemes shows a linearly magnifying trend. This cumulative effect is

particularly crucial for resource-constrained industrial Internet of Things devices, as it can significantly extend the network lifespan and enhance the system's scalability.

Table 4. Communication cost comparison.

Scheme	Number of Messages	Total Parameter Composition	Total Cost (bits)
[2]	3	$1T_{id} + 5T_h + 2T_s + 2T_{sym}$	1920
[3]	7	$1T_o + 6T_n + 7T_h + 4T_s$	3232
[4]	6	$1T_{id} + 5T_n + 7T_h + 4T_s$	3072
[17]	3	$1T_{id} + 4T_h + 3T_s + 1T_{sym}$	1664
Our	6	$2T_{sym} + 4T_n + 4T_h + 4T_s$	2304

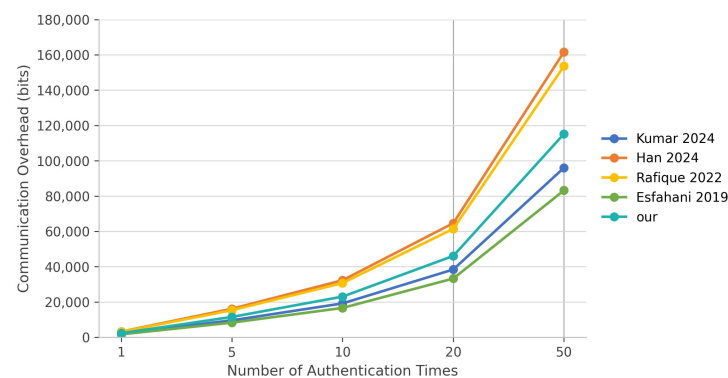


Figure 3. Communication overhead for n-time authentication [2–4,17].

6.4. Scenarios-Based Performance Evaluation

Considering the network volatility in industrial environments, we tested the authentication completion time under different network latency conditions. The results are presented in Figure 4. Even in a high-latency environment of 200 ms, our scheme can complete the authentication within 568 ms, fulfilling the real-time requirements of most industrial applications.

As can be seen from Figure 4, the schemes from [2,17], being two-party authentication schemes, perform best in network-latency environments, which is determined by their simplified protocol interaction processes. Among similar three-party authentication schemes, our scheme has the shortest authentication time, being 28.8% faster than the scheme in [3] and 20.2% faster than the scheme in [4]. As the network latency increases, the performance advantage of our scheme becomes more prominent, reflecting the effect of the optimized message-round design.

For the core innovative aspects of the scheme, we specifically tested the effectiveness of the AIS security mechanism.

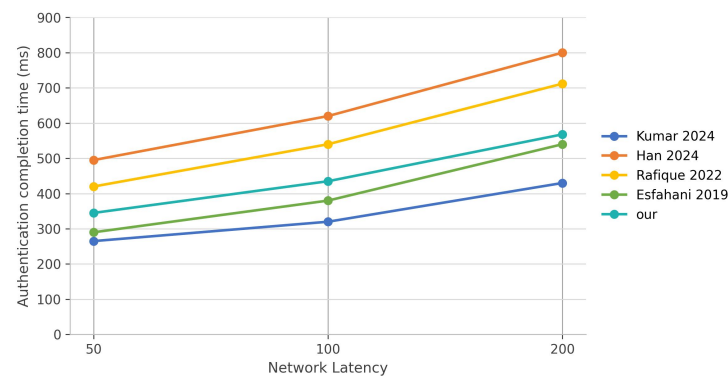


Figure 4. Impact of network latency on authentication time (ms) [2–4,17].

Model Integrity Verification Guarantee. In the verification mechanism design, the collision resistance of the hash function ensures that any modification to the model will produce a different Proof value. Specifically, since $Proof = h(FP_{model} \parallel T_3)$, any alteration to the model parameters changes the fingerprint FP_{model} , and by the collision resistance of SHA-256, finding $FP'_{model} \neq FP_{model}$ such that $h(FP'_{model} \parallel T_3) = h(FP_{model} \parallel T_3)$ is computationally infeasible. Thus, under the standard cryptographic assumption of hash collision resistance, the verification mechanism provides a theoretical guarantee that model tampering will be detected. We note that this is a design-level argument derived from the properties of the underlying hash primitive, not an empirical measurement. Comprehensive experimental evaluation of the fingerprint mechanism under diverse model modifications and real-world attack scenarios is deferred to future hardware-based experimental work.

7. Conclusions

In this article, we introduce a secure and efficient authentication scheme tailored specifically for an AIS environment. Specifically, a robust three-party authentication mechanism is proposed to ensure secure interactions among users, the GWN, and AIS providers. Through comprehensive security analysis, we have verified that this design achieves all fundamental security properties, including mutual authentication, user anonymity, forward secrecy, and resistance to various known attacks, while maintaining practical efficiency in both computation and communication.

For future work, several directions merit further investigation: deploying the proposed scheme on representative IIoT hardware platforms to measure real-world metrics including memory consumption, energy usage, authentication throughput, and latency percentiles; evaluating the model fingerprint mechanism with diverse deep learning architectures under fine-tuning, pruning, and backdoor injection attacks with statistical treatment and confidence intervals; extending the static model fingerprint to hardware-assisted runtime attestation using Intel SGX or ARM TrustZone; conducting a refined formal security proof using the standard ROR model with explicitly defined Execute, Send, Reveal, Corrupt, and Test oracles with detailed game-hopping sequences and tighter probability bounds; performing a precise communication cost analysis that accounts for GCM nonces and authentication tags with tripartite separation of user, gateway, and AIS computational overhead; modeling the complete protocol in ProVerif or Tamarin for machine-checkable security proofs; designing and evaluating resource-exhaustion defenses for the protocol's initial message exchange; mitigating the single-point-of-failure risk inherent in the trusted gateway model through threshold cryptography or distributed consensus mechanisms; adding lifecycle management support including password update and key rotation; and investigating defenses against AI-specific attacks (e.g., adversarial examples, data poisoning, and prompt injection) to complement the authentication-layer security provided by this protocol.

Author Contributions: Conceptualization, H.G.; Formal analysis, H.S.; Investigation, H.S.; Data curation, H.S.; Writing—original draft, S.J.; Writing—review & editing, G.S.; Supervision, Hanjun Gao. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: Authors Hanjun Gao and Haozhong Shi were employed by the company China Nuclear Power Operation Technology Corporation. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Serror, M.; Hack, S.; Henze, M.; Schuba, M.; Wehrle, K. Challenges and opportunities in securing the Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2021**, *17*, 2985–2996.
2. Kumar, A.; Jain, U.; Hussain, M.; Rahmani, M.K.I.; Banga, A.S. Mechanism for device authentication and session key generation in Industrial Internet of Things networks. *IEEE Access* **2024**, *12*, 108770–108785.
3. Han, Y.; Guo, H.; Liu, J.; Ehui, B.B.; Wu, Y.; Li, S. An enhanced multifactor authentication and key agreement protocol in Industrial Internet of Things. *IEEE Internet Things J.* **2024**, *11*, 16243–16254.
4. Rafique, F.; Obaidat, M.S.; Mahmood, K.; Ayub, M.F.; Ferzund, J.; Chaudhry, S.A. An efficient and provably secure certificates protocol for Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2022**, *18*, 8039–8046.
5. 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.
6. Wang, W.; Xu, H.; Alazab, M.; Gadekallu, T.R.; Han, Z.; Su, C. Blockchain-based reliable and efficient certificateless signature for IIoT devices. *IEEE Trans. Ind. Inform.* **2022**, *18*, 7059–7067.
7. Chen, Y.; Yin, F.; Hu, S.; Sun, L.; Li, Y.; Xing, B.; Chen, L.; Guo, B. ECC-based authenticated key agreement protocol for industrial control system. *IEEE Internet Things J.* **2023**, *10*, 4688–4697.
8. Geng, C.; Zhang, Y.; Xu, X.; Yao, Y.; Lu, C.; Zhi, Z. Blockchain-based identity authentication and data interaction scheme for Industrial Internet of Things. *Comput. Electr. Eng.* **2025**, *123*, 110143.
9. Turkanovic, M.; Brumen, B.; Hölbl, M. A novel user authentication and key agreement scheme for heterogeneous Ad Hoc wireless sensor networks, based on the Internet of Things notion. *Ad Hoc Netw.* **2014**, *20*, 96–112.
10. Tai, W.-L.; Chang, Y.-F.; Li, W.-H. An IoT notion-based authentication and key agreement scheme ensuring user anonymity for heterogeneous Ad Hoc wireless sensor networks. *J. Inf. Secur. Appl.* **2017**, *34*, 133–141.
11. Amin, R.; Biswas, G. A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks. *Ad Hoc Netw.* **2016**, *36*, 58–80.
12. Wu, F.; Xu, L.; Kumari, S.; Li, X.; Shen, J.; Choo, K.K.R.; Wazid, M.; Das, A.K. An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in IoT deployment. *J. Netw. Comput. Appl.* **2017**, *89*, 72–85.
13. Guo, H.; Gao, Y.; Xu, T.; Zhang, X.; Ye, J. A secure and efficient three-factor multi-gateway authentication protocol for wireless sensor networks. *Ad Hoc Netw.* **2019**, *95*, 101965.
14. Li, X.; Niu, J.; Bhuiyan, M.Z.A.; Wu, F.; Karupiah, M.; Kumari, S. A robust ECC-based provable secure authentication protocol with privacy preserving for Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3599–3609.
15. Shen, M.; Liu, H.; Zhu, L.; Xu, K.; Yu, H.; Du, X.; Guizani, M. Blockchain-assisted secure device authentication for cross-domain industrial IoT. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 942–954.
16. Das, A.K.; Wazid, M.; Kumar, N.; Vasilakos, A.V.; Rodrigues, J.J.P.C. Biometric-based privacy-preserving user authentication scheme for cloud-based Industrial Internet of Things deployment. *IEEE Internet Things J.* **2018**, *5*, 4900–4913.
17. Esfahani, A.; Mantas, G.; Matischek, R.; Saghezchi, F.B.; Rodriguez, J.; Bicaku, A.; Maksuti, S.; Tauber, M.G.; Schmittner, C.; Bastos, J. A lightweight authentication mechanism for M2M communications in industrial IoT environment. *IEEE Internet Things J.* **2019**, *6*, 288–296.
18. Li, X.; Ibrahim, M.H.; Kumari, S.; Sangaiah, A.K.; Gupta, V.; Choo, K.-K.-R. Anonymous mutual authentication and key agreement scheme for wearable sensors in wireless body area networks. *Comput. Netw.* **2017**, *129*, 429–443.
19. Tanveer, M.; Alkhayyat, A.; Khan, A.U.; Kumar, N.; Alharbi, A.G. REAP-IIoT: Resource-efficient authentication protocol for the Industrial Internet of Things. *IEEE Internet Things J.* **2022**, *9*, 24453–24465.
20. Xu, D.; Yu, K.; Ritcey, J.A. Cross-layer device authentication with quantum encryption for 5G enabled IIoT in industry 4.0. *IEEE Trans. Ind. Inform.* **2021**, *18*, 6368–6378.

21. Attir, A.; Naït-Abdesselam, F.; Faraoun, K.M. Lightweight anonymous and mutual authentication scheme for wireless body area networks. *Comput. Netw.* **2023**, *224*, 109625.
22. Soleymani, S.A.; Goudarzi, S.; Anisi, M.H.; Cruickshank, H.; Jindal, A.; Kama, N. Truth: Trust and authentication scheme in 5G-IIoT. *IEEE Trans. Ind. Inform.* **2022**, *19*, 880–889.
23. Zargar, G.R.; Barati, H.; Barati, A. An authentication mechanism based on blockchain for IoT environment. *Clust. Comput.* **2024**, *27*, 13239–13255.
24. Cao, Z.; Zhu, J.; Liu, L. Analysis of One Multifactor Authenticated Key Agreement Scheme for Industrial IoT. *Int. J. Netw. Secur.* **2024**, *26*, 605–609.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.