

Article

A Cross-Domain Authentication Key Agreement Protocol for Edge Computing

Zhaobo Wang, Wen Feng, Yifeng Yin *  and Zhiyong Jing

School of Computer Science and Technology, Zhengzhou University of Light Industry, Zhengzhou 450001, China; wangzhaobo7508@163.com (Z.W.); fengwen@zzuli.edu.cn (W.F.); jingzhiyong@zzuli.edu.cn (Z.J.)

* Correspondence: yinyifeng@zzuli.edu.cn

Abstract

With the rapid development of edge computing in the Industrial Internet, data sharing schemes among edge users require reliable cross-domain authentication and key agreement mechanisms to guarantee the security and reliability of inter-device communication. To tackle the deficiencies of existing group key agreement schemes, including dependence on trusted third parties, high computational overhead, and the difficulty of achieving both privacy preservation and attack resistance, this paper presents a cross-domain authenticated key agreement protocol designed for edge computing environments. This protocol supports anonymous identity authentication between cross-domain users, and innovatively constructs a multi-dimensional virtual iterative cyberspace model to generate massive secure keys via the collaborative iteration of multi-user key sequences. The proposed protocol is decentralized, lightweight, and resistant to replay attacks and man-in-the-middle attacks, while satisfying forward and backward secrecy. Security analysis and performance comparison experiments illustrate that the protocol significantly reduces computational and communication overhead, matches the resource-constrained characteristics of edge devices, and can be widely deployed in large-scale data encryption and sharing scenarios under edge computing environments.

Keywords: edge computing; cross-domain authentication; group key agreement protocol; data sharing; multi-dimensional virtual iterative

1. Introduction

With the deep integration of the industrial internet and digital technologies, edge computing has emerged as a core supporting infrastructure for transforming traditional industries towards intelligence, thanks to its distributed deployment, localized data processing, and low-latency response capabilities [1,2]. Unlike cloud computing, which centralizes computational and storage resources in remote data centers, edge computing deploys edge nodes at the network edge close to terminal devices (such as industrial sensors, intelligent controllers, and IoT terminals). This enables real-time sensing, local decision-making, and rapid control of industrial production processes [3]. For example, in smart manufacturing workshops, edge nodes can process massive real-time data generated by production lines locally, reducing the transmission delay of control commands to milliseconds; in smart grids, edge computing realizes localized analysis of power consumption data, improving the efficiency of grid scheduling and fault response [4]. By offloading a large number of computationally intensive tasks from cloud centers to edge nodes, edge computing not only alleviates the network bandwidth pressure and computational load of cloud centers but also effectively addresses the problems of high latency and poor reliability in long-distance



Academic Editor: Aryya Gangopadhyay

Received: 27 January 2026

Revised: 18 February 2026

Accepted: 21 February 2026

Published: 25 February 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.

data transmission, thereby significantly enhancing the overall operational efficiency and stability of industrial internet systems [5].

However, the widespread application of edge computing in the industrial internet has also brought new security challenges, especially in cross-domain data sharing scenarios [6]. In practical industrial environments, edge computing systems often span multiple independent server domains—for instance, a large-scale manufacturing enterprise may have production bases in different regions, each with its own edge computing cluster (forming independent domains); or different departments (such as production, logistics, and quality inspection) within an enterprise may adopt edge nodes from different vendors, resulting in heterogeneous and isolated authentication domains [7]. To realize collaborative work and data interconnection among these cross-domain edge users (e.g., real-time sharing of production data between edge devices in different regional bases, or joint analysis of quality inspection data across departments), it is necessary to establish secure and reliable communication channels. This relies on two core technologies: cross-domain identity authentication and group key agreement [8]. Cross-domain identity authentication ensures that only legitimate users in authorized domains can access shared resources, while group key agreement enables multiple cross-domain users to negotiate a common secure key for encrypting transmission data, preventing information leakage or tampering during data sharing [9].

The design of cross-domain authenticated group key agreement protocols for edge computing environments faces unique constraints and requirements. On one hand, edge devices in industrial scenarios (such as embedded sensors and low-power controllers) often have limited computational resources, storage capacity, and energy supply, requiring the protocol to be lightweight with low computational and communication overhead [10]. On the other hand, industrial data sharing has high requirements for real-time performance, so the key agreement process must be efficient to avoid delaying production scheduling or control decisions [11]. A traditional trusted third-party server centrally generates a shared session key for group members. The third party must be fully trusted, strong identity authentication is required for members, and the key is distributed in an encrypted channel. One-time session keys are adopted, with support for key update and member revocation. This approach is more applicable to small-scale, closed, low-dynamic and highly trusted scenarios like confidential meetings in enterprise intranets, and is not suitable for large-scale groups on public networks [12–14]. Edge computing environments are highly dynamic—edge users may frequently join or exit the group (e.g., temporary access of mobile inspection devices to the production network, or offline maintenance of edge nodes), requiring the protocol to support dynamic group management while ensuring forward secrecy (the compromise of subsequent keys does not affect the security of previously generated keys) and backward secrecy (the addition of new users does not allow them to obtain previously generated keys). Moreover, industrial data often involves core business secrets, so the protocol must resist common malicious attacks such as replay attacks, man-in-the-middle attacks, and eavesdropping attacks, while protecting user privacy (e.g., anonymous authentication to avoid the leakage of user identity information) [15]. Therefore, it is crucial to study high-security and lightweight group key agreement schemes for user cross-domain communication scenarios in edge computing environments.

Zhu et al. [16] designed a biometric anonymous authentication scheme for MEC, combining fuzzy extractors, lightweight cryptography and dynamic pseudonyms to achieve mutual authentication and key agreement, which resists impersonation and replay attacks. Yet the scheme has flaws: poor scalability due to pseudonym management, biometric template compromise risks from improper extractor configuration, single-point failure caused by trusted authority reliance, and lack of multi-factor authentication flexibility. Chen et al. [17] proposed a provable secure group key establishment scheme for fog com-

puting, enabling secure group key agreement between fog nodes and terminal devices. The scheme is proven secure against collusion, forgery and other common attacks, and is adapted to the distributed architecture of fog computing to enhance group communication security. However, it has high computing overhead in dynamic group member update, poor adaptability to resource-constrained terminals, and lacks cross-domain group key synchronization mechanisms for fog nodes. Meng et al. [18] proposed an anonymous authenticated group key agreement scheme for transfer learning edge services, enabling secure group communication with identity anonymity. The scheme is proven secure and adapted to edge service scenarios, yet suffers from high computing overhead in dynamic group updates, poor adaptability to resource-constrained devices, and lack of cross-domain expansion capability. Chen et al. [19] proposed FADIC, a fine-grained attribute-based dynamic group key agreement scheme for IoT, enabling attribute-based access control and secure dynamic group key updates. Proven secure, it enhances IoT group communication security, yet suffers from high computational overhead on resource-constrained devices and lacks cross-domain collaboration adaptability. Deng et al. [20] proposed a blockchain-based batch authentication and symmetric group key agreement scheme for MEC environments, realizing efficient batch user authentication and secure group key negotiation. The scheme leverages blockchain's tamper-proof feature to enhance trust, yet suffers from high blockchain consensus overhead and poor adaptability to dynamic MEC node changes. Zheng [21] presents a lightweight mutual authentication and key agreement protocol based on Physical Unclonable Functions (PUFs) to enhance terminal security and energy efficiency. However, the scheme's stability and universality for large-scale deployment need further verification due to its reliance on PUF physical characteristics, and its resilience against advanced attacks should be strengthened. Rajkumar et al. [22] proposed a Dickson polynomial-based group key agreement authentication scheme for VANETs, achieving conditional privacy preservation and traceability via distributed key computation (CRT) and pseudonym tracing. It reduces computation/communication overhead, yet lacks adaptability in high-density vehicle scenarios, weak resistance to quantum attacks, and poor dynamic group key synchronization. Luo [23] proposed a certificateless authenticated GKA protocol for cross-domain 5G network slicing that supports dynamic group user management. This one-round communication protocol allows group users from different network domains with varying cryptographic parameters to share a common session key. It provides proven key agreement security, mutual authentication, perfect forward secrecy, user anonymity, and partial private key transmission security, while resisting impersonation, replay, and known session key attacks. Mandal [24] presented a pairing-free certificateless authenticated group key agreement protocol to ensure secure communication in resource-constrained mobile networks. Performance analysis indicates that compared to most existing schemes, the proposed protocol offers superior efficiency in terms of security analysis, computational, and communication overhead, even as the number of mobile nodes in group communication gradually increases. Sutrala [25] proposed a conditional privacy-preserving and batch verification-based authentication scheme for the Internet of Vehicles (IoV). By adopting certificateless aggregate signature to implement efficient batch authentication, the scheme preserves vehicle identity anonymity while enabling the traceability of malicious behaviors, and effectively reduces the verification overhead under highly concurrent communication. Tan [26] employed blockchain to achieve decentralized pairing-free authentication, and devised a variable-threshold pseudonym management scheme that balances privacy preservation and traceability of malicious entities. Their self-balancing tree-based key agreement adapts well to the dynamic topology of VANETs, and the proposed scheme is secure and outperforms existing approaches in performance.

The main contributions of this paper are as follows:

1. Design a lightweight cross-domain authentication mechanism that uses elliptic curve cryptography and hash operations to reduce computational overhead, while hiding user identity information during authentication to achieve anonymous access.
2. Propose a multidimensional virtual iterative network space model, which generates a large number of secure keys through the collaborative iteration of multiple users' key sequences, realizing efficient group key agreement and supporting dynamic user management.
3. Achieve decentralization in the key agreement process—the edge server only participates in the initial public/private key distribution and does not involve subsequent key generation, avoiding single points of failure and enhancing system robustness.
4. The protocol has strong security, which can resist replay attacks, man-in-the-middle attacks, eavesdropping attacks, and other common malicious attacks, while having forward secrecy and backward secrecy.

2. Preliminary Design

To address issues such as constrained edge user resources and heterogeneous and overlapping authentication domains in industrial edge computing environments, this paper designs a cross-domain authentication group key agreement protocol for edge computing environments based on the multidimensional virtual iterative group key agreement model [27] (Our findings from another original research paper). The proposed solution enables cross-domain authentication and key agreement for edge users, ensuring that only authorized users within the group can obtain encryption/decryption keys to securely access data within the resource pool. Through security and performance analysis, the group key agreement proposal presented in this paper features cross-domain authentication while also exhibiting characteristics such as decentralization, lightweight computation, and resistance to eavesdropping attacks. The performance overhead of this scheme has also been improved compared to other approaches.

With the rapid development of edge computing in the industrial internet, security issues such as resource sharing across multiple domains have gradually emerged. To ensure the security of cross-domain communication for edge users, a cross-domain authentication group key agreement protocol is designed, as shown in Figure 1.

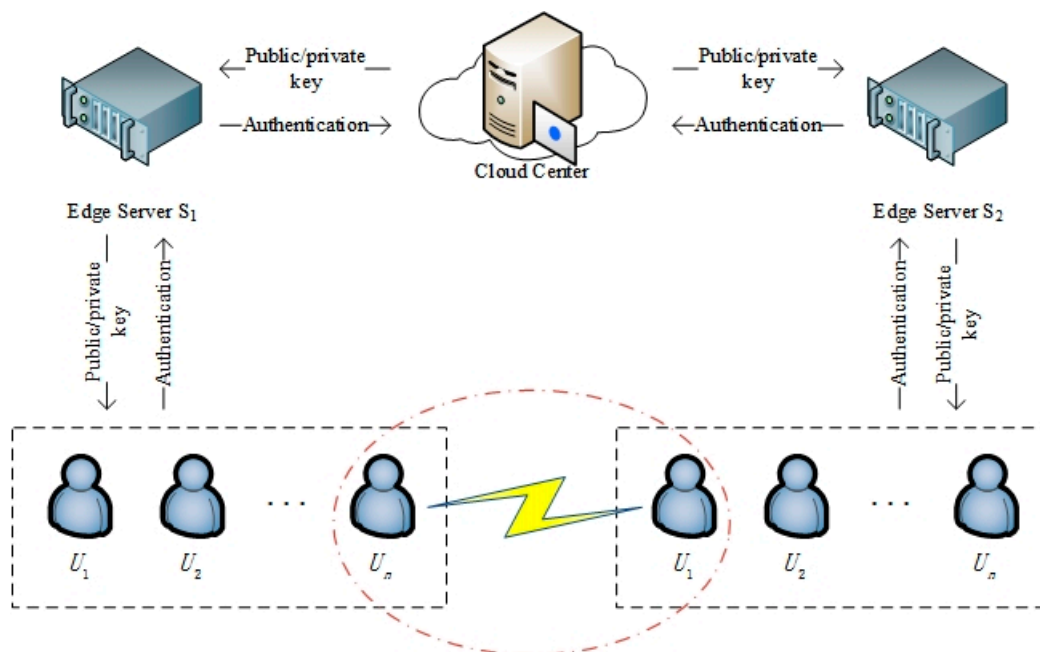


Figure 1. Scheme Framework Diagram.

This agreement involves three entities: the cloud center, edge servers, and edge users.

Cloud Center: The cloud computing center is a large-scale cluster of computing and storage resources that provides various cloud services to edge devices, such as virtual machines, containers, storage, and computing. Edge devices can interact with the cloud computing center over the network to obtain required services based on their needs. The cloud center provides identity registration and authentication functions for edge servers. This protocol involves three entities: the cloud center, edge servers, and edge users.

Edge Server: Stores data frequently accessed by edge devices and promptly provides relevant services to them. Each server is responsible for centrally managing a subset of edge devices. Additionally, the server assigns a serial number to each edge user it possesses. The server prepares a public/private key pair for the edge user for further use. Finally, the network publicly publishes the public key and securely transmits the private key to the new edge user.

Edge users: Edge users implement the generation and processing of source data. Secure communication channels can be established between edge users, who negotiate session keys to ensure the security of data transmission between devices. This enables the provision of required services to users.

Its primary implementation consists of initialization operations, the user and remote server authentication phase, and the user-to-user signature and key agreement phase. The relevant steps are as follows:

- (1) Edge user A in domain S_1 initiates an access request to the server in domain S_2 where user B resides;
- (2) The S_2 server authenticates User A. Upon successful authentication, it establishes a direct connection between User A and User B, who then join the workstation and wait. User B performs the same authentication process with User C and establishes a connection, continuing until all users have joined the workstation.
- (3) Users in the workstation publish their parameter information;
- (4) All users perform key agreement at their workstations. Based on the order of joining the workstation, each user only needs to send a signed message once to their next user, which is then signed and confirmed by their next neighbor;
- (5) After all users confirm their signatures, construct a multidimensional virtual iterative cyberspace. Users iterate through multiple security subsystems to obtain cryptographic keys. By repeating the arrangement steps, users can generate numerous distinct secure encryption keys.

3. Design of an Authenticated Group Key Agreement Protocol

Assume there are multiple edge servers in the edge cloud, each hosting multiple edge device users. Each edge server manages its own set of edge users, forming a domain.

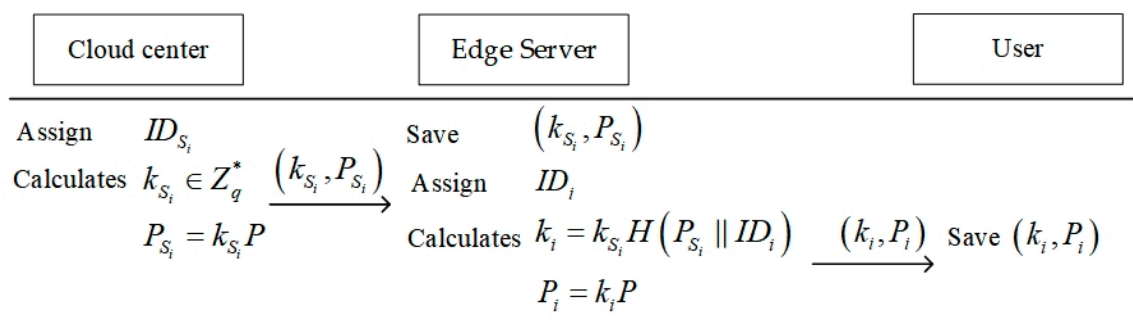
Assume that user A from domain S_1 , user B from domain S_2 , user C from domain S_3 , and user D from domain S_4 engage in key agreement. User A sends an access request to S_2 via server S_1 , and S_2 authenticates User A. If authentication succeeds, server S_2 authorizes a direct connection between User A and User B. The user then joins the workstation and waits. User B authenticates with User C and establishes a connection through the same procedure, continuing until all users have joined the workstation. Then create a token ring and establish a multidimensional permutation network for group key agreement. Table 1 lists the symbols used in this section and their explanations.

Table 1. Lists the symbols and their descriptions used in this section.

Symbol	Meaning
G	A cyclic additive group of order q
p, q	A large prime number
P	Generators of G
ID_i	Server i or user i Identity Identifier
k_{S_1}	Private key of Server S_1
P_{S_1}	Public key of Server S_1
k_i	Private key of User i
P_i	Public key of User i
$H(.)$	A hash function employing the SHA-256 encryption algorithm
r_i	User i randomly selects random number
$K_i[n]$	Key Control Array
Token	Token Ring
$VIF(x, y)$	Virtual iterative function containing parameters x and y
$F(.)$	The virtual iteration function for the i -th user without added information
IFS^x	Password iteration transformation
S_{nm}	Iterated function system with x iterations
	$n \times n$ secure subsystems shared by users

3.1. Initialization Operation

Assume there are four users in the entire network: A, B, C, and D. When negotiating security group keys, A, B, C, and D proceed in a specific order (For example, based on the order in which they entered the group or their value to the group). Initialization Flow as shown in Figure 2.

**Figure 2.** Initialization Flow.

Before the four users perform authentication key agreement, parameters must be initialized. The initialization process includes:

- (1) The cloud center provides elliptic curve operation parameters to edge servers, specifically selecting an appropriate elliptic curve group G . Two large prime numbers p and q are randomly selected. The cyclic additive group of order q on a non-singular elliptic curve E has generators P . A non-singular elliptic curve E is defined by the equation $y^2 \equiv x^3 + ax + b \pmod{q}$, where $a, b \in F_q$.
- (2) The cloud center provides edge servers with identity identifiers $ID_{S_1}, ID_{S_2}, ID_{S_3}$, and ID_{S_4} . Edge servers S_1, S_2, S_3 , and S_4 assign identity identifiers $ID_{1,A}, ID_{2,B}, ID_{3,C}$, and $ID_{4,D}$ to four users respectively.
- (3) The cloud center selects $k_{S_1} \in Z_q^*$ as the private key for server S_1 , and then calculates $P_{S_1} = k_{S_1}P$ as the public key. Similarly, the private keys for edge servers S_2, S_3 and S_4 are k_{S_2}, k_{S_3} , and k_{S_4} respectively, while their public keys are P_{S_2}, P_{S_3} , and P_{S_4} respectively.
- (4) Edge Server S_1 computes $k_A = k_{S_1} H(P_{S_1} || ID_{1,A})$ and $P_A = k_A P$, and sends the message (k_A, P_A) to User A via secure channels. Similarly, the key pairs for users B, C, and D are $(k_B, P_B), (k_C, P_C)$, and (k_D, P_D) respectively.

- (5) All users only disclose their identity credentials and public keys within the same domain, and edge servers only disclose their identity credentials and public keys within the same cloud.

3.2. User and Foreign Server Authentication Phase

User and Foreign Server Authentication Phase as shown in Figure 3.

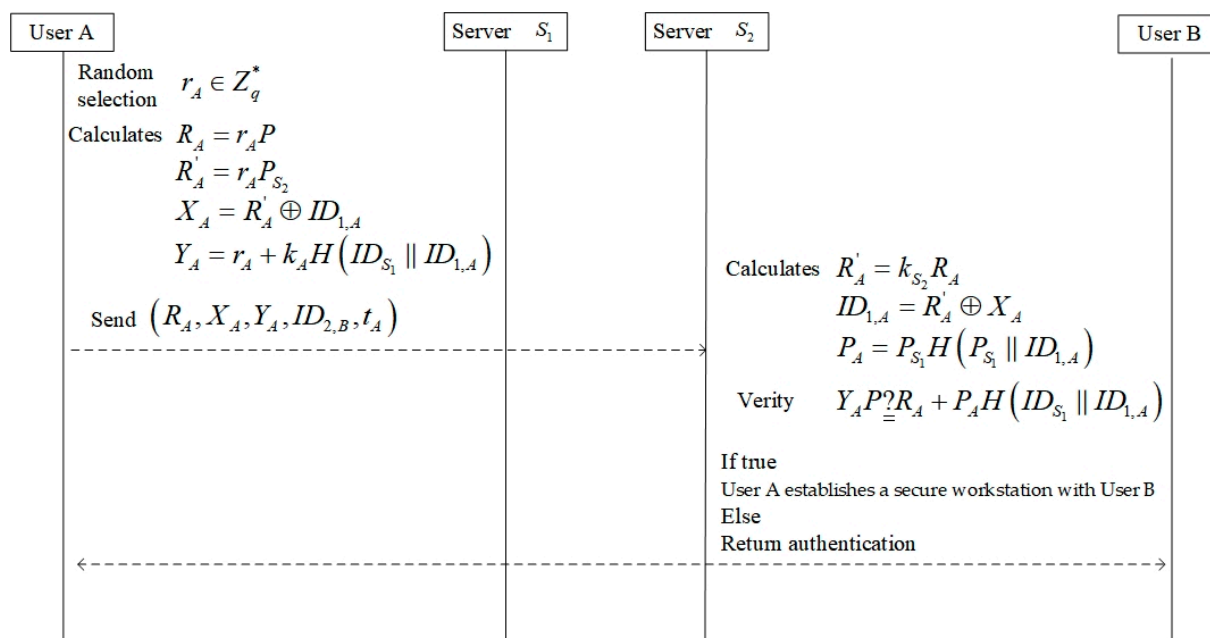


Figure 3. Authentication Phase Flow.

- (1) User A randomly selects $r_A \in Z_q^*$, calculates $R_A = r_A P$, $R'_A = r_A P_{S_2}$, $X_A = R'_A \oplus ID_{1,A}$, $Y_A = r_A + k_A H(ID_{S_1} || ID_{1,A})$, generates a timestamp t_A , and computes the initial key sequence $K_A[n] = H(k_A \oplus (R_A || ID_{1,A}))$ for personal storage. Then send $(R_A, X_A, Y_A, ID_{2,B}, t_A)$ from the local server S_1 to the remote server S_2 .
- (2) When foreign server S_2 receives message $(R_A, X_A, Y_A, ID_{2,B}, t_A)$, the server learns that a user in domain S_1 wishes to communicate with user B. The server first checks the timestamp t_A freshness. After calculating $R'_A = k_{S_2} R_A$ and $ID_{1,A} = R'_A \oplus X_A$, the identity identifier $ID_{1,A}$ for User A can be obtained. The server can compute User A's public key is $P_A = P_{S_1} H(P_{S_1} || ID_{1,A})$, and verify $Y_A P_A \stackrel{?}{=} R_A + P_A H(ID_{S_1} || ID_{1,A})$. If the conditions are met, the server will approve User A's authentication request. If satisfied, server S_2 agrees to user A's authentication request.

User A establishes a secure workstation with User B. Participating users publish their identity information within the network and obtain the identity information of other users besides themselves. User B authenticates User C using the same method, and User C authenticates User D using the same method, adding them to the workstation.

3.3. Group Key Agreement Phase

This stage requires constructing a multidimensional key space. Using a lightweight cryptographic algorithm based on stream ciphers, a multidimensional virtual iterative is constructed to generate symmetric keys. Group key agreement Phase as shown in Figure 4.

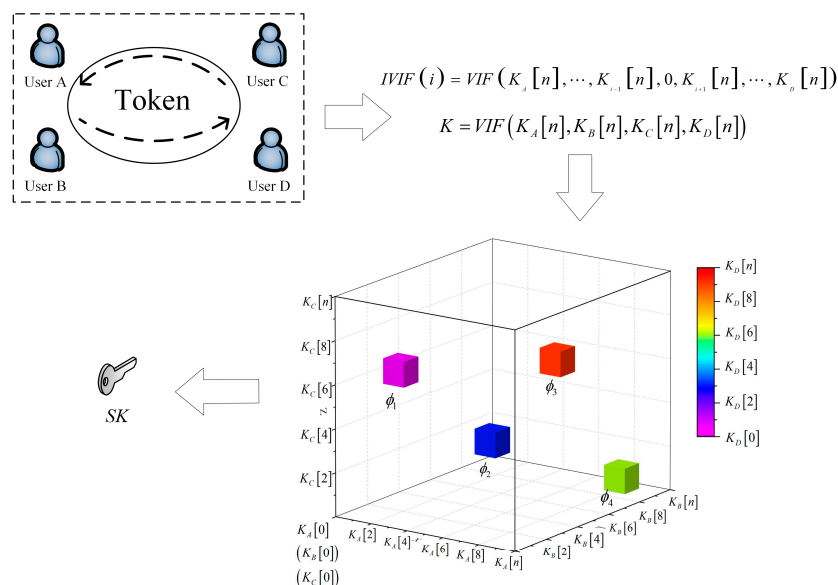


Figure 4. Group key agreement Phase Flow.

Step 1: All users on the workstation begin creating a token ring. Initiator A serves as the primary node, with the remaining users added to the token ring sequentially based on their order of joining.

Step 2: When an initiator begins a negotiation request at a given moment, the initiator first obtains the right to use the token and uploads containing their own key sequence $VIF(K_i[n])$.

Step 3: After Initiator A completes their operation, they release the token ring. The token ring is passed to the next user, User B, who performs the same operation. Following the sequence, when the token ring completes a full cycle and returns to initiator A, the token ring ceases waiting, and all users complete message transmission.

Step 4: All users upload $K_i[n]$ to $VIF(\cdot)$ in the order of the token ring. Upon successful upload, they obtain the incomplete virtual iteration function as shown in the formula. Consequently, all users also acquire the incomplete virtual iteration functions of all other users except themselves. As shown in Equation (1):

$$IVIF(i) = VIF(K_A[n], \dots, K_{i-1}[n], 0, K_{i+1}[n], \dots, K_D[n]) \quad (1)$$

Step 5: Each user uploads their own $K_i[n]$ to $VIF(\cdot)$, completing the operation to replace the zero function with actual parameters. At this point, all users can obtain the complete virtual iteration function. As shown in Equation (2):

$$VIF(K_A[n], K_B[n], K_C[n], K_D[n]) \quad (2)$$

At this point, the virtual iterative function triggers the self-compilation property of the function. In accordance with the object-oriented overloading theory, the actual parameters control the function's functionality. The n^4 candidate security subsystem is distributed with equal probability in the multi-dimensional permutation network space, forming a multi-dimensional iterative permutation network shared by a fixed set of four users. The key space shared by four users is uniformly distributed across the network. K denotes the shared multi-dimensional iterative permutation network. As shown in Equation (3):

$$K = VIF(K_A[n], K_B[n], K_C[n], K_D[n]) \quad (3)$$

Within the key space, the key sequences of four users are collectively mapped as actual parameters into four-dimensional space. Each subspace corresponds to a secure subsystem, with a set of four-dimensional coordinates defining that subsystem.

Step 6: In a four-dimensional permutation network, each user randomly selects x elements from the key sequence. The i -th coordinate group selected by four users represents a secure subsystem within the permutation network, denoted as ϕ_i . Thus, x security subsystems can be rapidly permuted from the key space, which are denoted in sequence as $\phi_1, \phi_2, \phi_3, \phi_4$.

Step 7: Using a self-compiler, the $n \times n$ security subsystems obtained in Step 1 are iteratively generated into an iterative function system IFS^x in a specific order, where x is denoted as 4, as shown in Equation (4).

$$F_1 = (\phi_1)F_2 = (F_1, \phi_2)F_3 = (F_2, \phi_3)SK \rightarrow IFS^4 = F_4 = (F_3, \phi_4) \quad (4)$$

Step 8: Finally, the iterative result can serve as the unique secure encryption key(SK), which provides long-period key sequences for various symmetric encryption algorithms and thus completes the key agreement. Since the session key(SK) is unique, all users can decrypt the ciphertext to obtain the information transmitted by other users.

4. Security Analysis

Theorem 1. *Under the hardness assumption of the Elliptic Curve Discrete Logarithm Problem (ECDLP), the user's identity information is encrypted and transmitted over a secure channel, and no adversarial entity can decrypt it to obtain the user's real identity.*

Proof. The user's identity is hidden by the result of elliptic curve operations, and its extraction relies on the server's private key and the hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP). For user A, the identity $ID_{1,A}$ satisfies $X_A = R'_A \oplus ID_{1,A}$, where $R'_A = r_A P_{S_2} = k_{S_2} R_A$. Here, k_{S_2} is the private key of server S_2 , and $P_{S_2} = k_{S_2} P$ is its public key. If an adversary intends to derive $ID_{1,A}$ from X_A , it must first obtain R'_A . Acquiring R'_A requires either stealing the private key k_{S_2} , or computing the random number r_A from $R_A = r_A P$. Both problems are reduced to ECDLP: given elliptic curve points P and $Q = aP$, no probabilistic polynomial-time adversary can compute a . Therefore, the adversary cannot obtain R'_A , and thus cannot extract $ID_{1,A}$.

Elliptic curve operations guarantee that authentication parameters are unforgeable, and adversaries cannot reversely derive user identity information. The authentication parameter Y_A is generated by the user's private key k_A and elliptic curve-related hash operations, where $k_A = k_{S_1} H(P_{S_1} \parallel ID_{1,A})$ and $P_{S_1} = k_{S_1} P$. Here, k_{S_1} is the private key of the local server and remains confidential at all times. Therefore, adversaries can neither forge Y_A nor reversely derive any information related to the user's identity through elliptic curve operations. $\square$

Theorem 2. *The mutual authentication of this protocol is implemented based on the cyclic additive group G of elliptic curve (where q is a large prime, and P is the generator) and hash operations. If an adversary intends to compromise the authentication security, it must forge elliptic curve-related authentication parameters, while the hardness of ECDLP and the confidentiality of private keys form a dual protection against such attacks.*

Proof. Server's authentication of user A: User A sends an authentication message containing $R_A = r_A P$ and $Y_A = r_A + k_A H(ID_{S_1} \parallel ID_{1,A})$. The server validates the legitimacy of user A by verifying whether $Y_A P \stackrel{?}{=} R_A + P_A H(ID_{S_1} \parallel ID_{1,A})$ holds, where $P_A = k_A P$ and

k_A is the private key of user A. Computing Y_A requires the knowledge of k_A , so Y_A cannot be forged by any adversary. Alternatively, deriving r_A from $R_A = r_AP$ is also required. Both operations rely on the Elliptic Curve Discrete Logarithm Problem (ECDLP), which is computationally infeasible in polynomial time.

Authentication of server S_2 by user A: The server computes $R'_A = k_{S_2}R_A$ with its private key k_{S_2} and recovers $ID_{1,A}$ to complete authentication. User A can confirm that the server holds the legitimate private key k_{S_2} by checking whether the server can correctly return the authentication result. Since an adversary cannot acquire k_{S_2} , it cannot simulate the server to compute R'_A correctly and thus cannot pass the verification of user A. $\square$

Theorem 3. *The protocol adopts an anti-replay mechanism combining timestamps, elliptic curve random numbers, and multi-dimensional virtual iterative functions. Any adversary with polynomial computational power cannot launch an attack by replaying previous authentication messages, and the protocol thus achieves security against replay attacks.*

Proof. Timestamps block the replay of expired messages. Each authentication message carries a timestamp t (e.g., t_A sent by user A), and the receiver verifies the timestamp against a valid time window. When an attacker replays previous messages, the timestamp will inevitably fall outside the valid time window and fail the basic verification, thereby directly thwarting simple replay attacks.

Elliptic curve random numbers prevent message replay within the valid time window. Messages in each session are bound to a unique random number r_A , and the elliptic curve scalar multiplication result $R_A = r_AP$ is calculated. Owing to the intractability of the Elliptic Curve Discrete Logarithm Problem (ECDLP), an adversary cannot recover r_A from R_A . Even if replay is performed within the valid time window, the receiver can detect it through the uniqueness of R_A .

During the key agreement phase, the virtual iterative function(VIF) must be updated for each instance of information sharing by the user. The VIF is generated from the user key-controlled array $K_i[n]$, which depends on the random number r randomly selected in each interaction. The update of r with each interaction directly induces an irreversible change in both the computational logic and output of the VIF. As the core component, the virtual iterative function varies dynamically, such that the old key from the n -th session cannot adapt to the new VIF in the $n + 1$ -th session. $\square$

Theorem 4. *Can resist man-in-the-middle attacks, and an adversary cannot impersonate legitimate entities or tamper with messages.*

Proof. During the authentication phase between the user and the foreign server, it is difficult for an adversary to obtain the private key k_A and random number r_A of user A. Therefore, the adversary can hardly construct $Y_A = r_A + k_A H(ID_{S_1} \parallel ID_{1,A})$.

During the key agreement phase, an adversary may obtain $K_i[n]$ in the transmission channel. However, both the virtual iterative function(VIF) and the construction of the key space require the security subsystem S_{nn} , which is not exchanged in the transmission channel. Thus, the adversary cannot obtain the security subsystem S_{nn} and consequently cannot participate in the iteration phase to acquire the session key(SK). $\square$

Theorem 5. *Via directional transmission, collaborative generation of the key space, and encrypted transmission mechanisms, the proposed protocol can resist eavesdropping attacks. Any adversary with polynomial computational power is unable to obtain the user's plaintext data or session key by eavesdropping on the public channel, thus guaranteeing communication confidentiality.*

Proof. Token Ring directional transmission avoids the leakage of the key sequence. In the proposed protocol, the user key sequence $K_i[n]$ is only transmitted directionally within workstations in accordance with the Token Ring order, instead of being broadcast to the entire public channel. Even if an adversary eavesdrops on the channel, it cannot obtain the complete single-user key sequence and thus finds it difficult to acquire the basic parameters for key generation.

The multi-dimensional virtual iterative network requires multi-user collaboration, and eavesdropping on a single node is ineffective. The session key is generated through the multi-dimensional virtual iterative network constructed by multi-user collaboration, which requires all users to upload their own key sequences to complete the iteration. If an adversary only eavesdrops on the transmission information of a single or partial users, it cannot obtain the complete virtual iterative function and key space, nor can it participate in the key generation process. $\square$

5. Performance Analysis

This subsection utilizes the Crypto++ library to measure the time overhead of cryptographic operations. The experiments were conducted on a machine with an Ubuntu 11.10 operating system, an Intel Core 2 Duo 1.86 GHz processor, and 2 GB of RAM. In this subsection, several existing schemes are compared and analyzed. The time consumption for information transmission is neglected in the analysis. The notations and data used in the performance analysis are shown in Table 2. Table 3 compares the time cost of the proposed protocol with that of existing protocols, where n is the number of users participating in the key agreement.

Table 2. Symbols and Their Definitions Used in Performance Analysis.

Operation	Symbol	Times (ms)
SHA-256 hash algorithm	T_h	0.067
generate pseudorandom Numbers	T_{rand}	0.045
modular inversion	T_{inv}	0.124
bilinear pairings	T_{bp}	4.514
Multiplication on Elliptic Curves	T_{pm}	0.612
Addition on Elliptic Curves	T_{pa}	0.125
AES-256 symmetric encryption/decryption	$T_{Ek/Dk}$	0.161
Multiplication of two numbers	T_{mul}	0.015

Table 3. Time Consumption Comparison.

Scheme	Total Time Consumption
Luo	$(10n + 1)T_{pm} + (4n + 1)T_{pa} + 7nT_h$
Mandal	$(10n + 5)T_{pm} + (9n + 38)T_h + nT_{pa}$
Naresh	$(4n + 2)T_{pm} + (4n + 13)T_{mul} + 4nT_{inv}$
Our protocol	$3nT_{pm} + 4nT_h + nT_{rand} + nT_{Ek/Dk}$

Luo [23] proposed a cross-domain certificateless authenticated group key agreement protocol that supports dynamic group user management. This scheme employs a single round of communication, allowing users from different network domains to reach a consensus on a shared session key for data sharing. While the scheme demonstrates favorable security performance, it involves a large number of elliptic curve scalar multiplication and elliptic curve scalar addition operations. Data indicates that its computational overhead is comparable to that of the Mandal scheme, yet it still incurs higher computational costs compared with the protocol proposed in this paper.

Mandal [24] proposed a certificateless public-key cryptography group key agreement scheme, which eliminates the computational and communication overhead incurred by certificate management and key escrow issues. The signcryption technique adopted in the scheme ensures the confidentiality of the session key. It supports the dynamic feature of adding or removing any user during the communication process, and can achieve forward and backward secrecy. However, in practical implementation, it involves an excessive number of elliptic curve scalar multiplication and hash operations, resulting in excessively high computational overhead. Meanwhile, performance analysis demonstrates that with the gradual increase in the number of users participating in group communication, compared with Mandal's scheme, the scheme proposed in this paper exhibits superior efficiency in terms of security analysis, computational overhead and communication overhead.

Naresh [28] proposed a blockchain-based group key agreement scheme. Owing to the adoption of blockchain technology, this scheme shows high adaptability to large-scale communication scenarios. This scheme involves numerous point multiplication and modular inversion operations, resulting in approximate time overhead compared with the protocol proposed in this paper. Nevertheless, it lacks mutual authentication among users and provides no defense against replay attacks, so its security still needs to be improved.

Experimental data indicate that, in comparison with the schemes reported in the reference literature, the proposed scheme incurs significantly lower time overhead in repeated key generation processes under the condition of the same number of users. This advantage is mainly attributed to the fact that the schemes in the reference literature need to repeatedly execute numerous steps during secure key generation, while the asymmetric encryption algorithms they adopt also entail longer execution time. In the authentication phase of the proposed scheme, elliptic curve operations are employed to authenticate users and cross-domain servers. During the key agreement process, digital signatures are utilized to achieve mutual identity authentication among users. Although these operations incur higher time overhead in contrast to symmetric encryption algorithms, they significantly enhance the security of the authentication process. Through comprehensive analysis, it is demonstrated that the protocol proposed in this paper achieves lower computational overhead compared with the existing schemes listed in Table 3 and Figure 5.

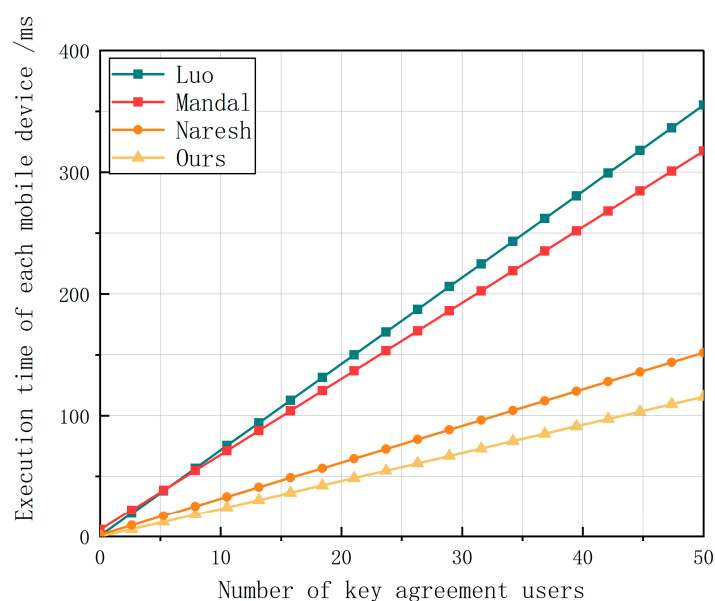


Figure 5. Comparison of Time Consumption Among Four Protocols.

6. Conclusions

This paper proposes a cross-domain authenticated group key agreement protocol for edge computing environments. The edge server distributes public/private key pairs to edge devices, which then establish anonymous authenticated connections with other users via cross-domain servers. After all users have established connections, keys are generated through the multi-dimensional virtual iterative network space. The edge server does not participate in the group key agreement process, enabling the protocol to effectively resist replay attacks, man-in-the-middle attacks, and other common malicious attacks. Furthermore, the multi-dimensional virtual iterative network space exhibits stream cipher characteristics. Comparative analysis of time overhead demonstrates that the protocol proposed in this paper has lightweight characteristics. Therefore, this protocol can be effectively applied in edge computing environments, enabling rapid key generation with high security performance, and is suitable for large-scale data transmission scenarios.

In the construction of the multi-dimensional virtual iterative network model, the number of security subsystems increases correspondingly with the number of users. However, an unbounded expansion of subspaces does not significantly enhance protocol security; instead, it leads to greater storage overhead. Future research will focus on limiting subspace growth and ensuring protocol security by adjusting the complexity of the key space, constraining the length of arguments in the multi-dimensional virtual iterative function, and addressing high-order nonlinearity issues within the key space.

Author Contributions: Conceptualization, Z.W. and Y.Y.; methodology, Z.W. and W.F.; software, W.F.; validation, Z.W. and Z.J.; formal analysis, W.F. and Y.Y.; investigation, Z.W.; resources, Z.J.; data curation, W.F.; writing—original draft preparation, Z.W.; writing—review and editing, Z.W.; visualization, Z.J.; supervision, Y.Y.; project administration, W.F.; funding acquisition, Z.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Science and Technology Project of Henan Province, grant numbers 252102210108.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: We sincerely thank the editor and reviewers for their insightful and constructive comments.

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

References

1. Xu, H.S.; Wu, J.; Pan, Q.Q.; Guan, X.P.; Guizani, M. A Survey on Digital Twin for Industrial Internet of Things: Applications, Technologies and Tools. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 2569–2598. [\[CrossRef\]](#)
2. Leng, J.W.; Zhu, X.F.; Huang, Z.Q.; Li, X.Y.; Zheng, P.; Zhou, X.L.; Mourtzis, D.; Wang, B.C.; Qi, Q.L.; Shao, H.D.; et al. Unlocking the power of industrial artificial intelligence towards Industry 5.0: Insights, pathways, and challenges. *J. Manuf. Syst.* **2024**, *73*, 349–363. [\[CrossRef\]](#)
3. Zhukabayeva, T.; Ahmad, Z.; Adamova, A.; Karabayev, N.; Abdildayeva, A. An Edge-Computing-Based Integrated Framework for Network Traffic Analysis and Intrusion Detection to Enhance Cyber-Physical System Security in Industrial IoT. *Sensors* **2025**, *25*, 2395. [\[CrossRef\]](#)
4. Hassaballah, E.G.; Keshta, H.E.; Abdel-Latif, K.M.; Ali, A.A. A novel strategy for real-time optimal scheduling of grid-tied microgrid considering load management and uncertainties. *Energy* **2024**, *299*, 131419. [\[CrossRef\]](#)
5. Elbamby, M.S.; Bennis, M.; Saad, W.; Latva-Aho, M.; Hong, C.S. Proactive edge computing in fog networks with latency and reliability guarantees. *EURASIP J. Wirel. Commun. Netw.* **2018**, *2018*, 209. [\[CrossRef\]](#)
6. Wang, F.Q.; Cui, J.; Zhang, Q.Y.; He, D.B.; Zhong, H. Blockchain-Based Secure Cross-Domain Data Sharing for Edge-Assisted Industrial Internet of Things. *IEEE Trans. Inf. Forensics Secur.* **2024**, *19*, 3892–3905. [\[CrossRef\]](#)
7. Li, X.H.; Chen, T.; Cheng, Q.F.; Ma, S.Q.; Ma, J.F. Smart Applications in Edge Computing: Overview on Authentication and Data Security. *IEEE Internet Things J.* **2021**, *8*, 4063–4080. [\[CrossRef\]](#)

8. Daniel, R.M.; Thomas, A.; Rajsingh, E.B.; Silas, S. A strengthened eCK secure identity based authenticated key agreement protocol based on the standard CDH assumption. *Inf. Comput.* **2023**, *294*, 105067. [[CrossRef](#)]
9. Chen, Q.N.; Wu, T.; Hu, C.N.; Chen, A.B.; Zheng, Q.H. An Identity-Based Cross-Domain Authenticated Asymmetric Group Key Agreement. *Information* **2021**, *12*, 112. [[CrossRef](#)]
10. Rakeei, M.; Moazami, F. An efficient and provably secure authenticated key agreement scheme for mobile edge computing. *Wirel. Netw.* **2022**, *28*, 2983–2999. [[CrossRef](#)]
11. Hsu, C.F.; Harn, L.; Xia, Z.; Zhao, Z.; Xu, H. Fast and Lightweight Authenticated Group Key Agreement Realizing Privacy Protection for Resource-Constrained IoMT. *Wirel. Pers. Commun.* **2023**, *129*, 2403–2417. [[CrossRef](#)]
12. Kemmoe, V.Y.; Kwon, Y.; Shin, S.; Hussain, R.; Cho, S.; Son, J. Leveraging Smart Contracts for Asynchronous Group Key Agreement in Internet of Things. In *2020 IEEE International Conference on Systems, Man, And Cybernetics (SMC)*; IEEE: Piscataway, NJ, USA, 2020; pp. 70–75.
13. Wang, Z.H.; Zhang, J.Q.; Tang, Y.W.; Cheng, H.Y. BGAS: Blockchain-based secure and efficient group key agreement scheme for dynamic vehicular platoon. *Clust. Comput.* **2025**, *28*, 59. [[CrossRef](#)]
14. Liu, G.; Li, H.; Wang, N.; Xiang, T.; Liu, Y.N. DeGKM: Decentralized Group Key Management for Content Push in Integrated Networks. *IEEE Trans. Dependable Secur. Comput.* **2024**, *21*, 4784–4800. [[CrossRef](#)]
15. Xie, D.; Yang, J.H.; Wu, B.; Bian, W.X.; Chen, F.L.; Wang, T.C. An Effectively Applicable to Resource Constrained Devices and Semi-Trusted Servers Authenticated Key Agreement Scheme. *IEEE Trans. Inf. Forensics Secur.* **2024**, *19*, 3451–3464. [[CrossRef](#)]
16. Zhu, Q.; You, L.; Hu, G.R.; Wang, S.G. Secure and Efficient Biometric-Based Anonymous Authentication Scheme for Mobile-Edge Computing. *IEEE Internet Things J.* **2024**, *11*, 33604–33623. [[CrossRef](#)]
17. Chen, W.C.; Huang, Y.T.; Wang, S.D. Provable Secure Group Key Establishment Scheme for Fog Computing. *IEEE Access* **2021**, *9*, 158682–158694. [[CrossRef](#)]
18. Meng, X.W.; Liang, W.; Xu, Z.S.; Li, K.C.; Khan, M.K.; Kui, X.Y. An Anonymous Authenticated Group Key Agreement Scheme for Transfer Learning Edge Services Systems. *ACM Trans. Sens. Netw.* **2024**, *20*, 75. [[CrossRef](#)]
19. Chen, Z.; Jin, M.; Jiang, J.; Zhang, H. FADIC: A fine-grained attribute-based dynamic group key agreement scheme for IoT. *Cybersecurity* **2025**, *8*, 91. [[CrossRef](#)]
20. Deng, Y.; Zhang, J.; Liu, J.; Li, J. Blockchain-Based Batch Authentication and Symmetric Group Key Agreement in MEC Environments. *Symmetry* **2025**, *17*, 2160. [[CrossRef](#)]
21. Barbareschi, M.; Casola, V.; Emmanuele, A.; Lombardi, D. A Lightweight PUF-Based Protocol for Dynamic and Secure Group Key Management in IoT. *IEEE Internet Things J.* **2024**, *11*, 16. [[CrossRef](#)]
22. Rajkumar, Y.; Kumar, S. A Dickson polynomial based group key agreement authentication scheme for ensuring conditional privacy preservation and traceability in VANETs. *Sci. Rep.* **2025**, *15*, 6211. [[CrossRef](#)]
23. Luo, M.; Wu, J.Y.; Li, X.J. Cross-domain certificateless authenticated group key agreement protocol for 5G network slicings. *Telecommun. Syst.* **2020**, *74*, 437–449. [[CrossRef](#)]
24. Mandal, S.; Mohanty, S.; Majhi, B. CL-AGKA: Certificateless authenticated group key agreement protocol for mobile networks. *Wirel. Netw.* **2020**, *26*, 3011–3031. [[CrossRef](#)]
25. Sutrala, A.K.; Bagga, P.; Das, A.K.; Kumar, N.; Lorenz, P. On the Design of Conditional Privacy Preserving Batch Verification-Based Authentication Scheme for Internet of Vehicles Deployment. *IEEE Trans. Veh. Technol.* **2020**, *69*, 5535–5548. [[CrossRef](#)]
26. Tan, H.; Wang, M.; Shen, J.; Vijayakumar, P.; Moh, S.; Wu, Q.M.J. Blockchain-Assisted Conditional Anonymous Authentication and Adaptive Tree-Based Group Key Agreement for VANETs. *IEEE Trans. Dependable Secur. Comput.* **2025**, *18*, 322. [[CrossRef](#)]
27. Yin, Y.F.; Wang, Z.B.; Zhou, W.Y.; Gan, Y.; Zhang, Y.H. Group key agreement protocol for edge computing in industrial internet. *Math. Biosci. Eng.* **2022**, *19*, 12730–12743. [[CrossRef](#)] [[PubMed](#)]
28. Naresh, V.S.; Allavarpu, V.; Reddi, S. Blockchain privacy-preserving smart contract centric multiple multiparty key agreement over large WANETs. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4165. [[CrossRef](#)]

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.