electronics

MDPI

Article

A Secure and Efficient Authentication Scheme for Large-Scale IoT Devices Based on Zero-Knowledge Proof

Ziyi Su [1], Shiwei Wang [2], Hongliu Cai [3], Jiaxuan Huang [4], Yourong Chen [1,*], Xudong Zhang [1] and Muhammad Alam [5]

[1] College of Information Science and Technology, Zhejiang Shuren University, Hangzhou 310015, China; ziyisu@zjsru.edu.cn (Z.S.)
[2] College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China
[3] Department of Critical Care Medicine, The First Affiliated Hospital, Zhejiang University School of Medicine, Hangzhou 310003, China
[4] School of Computer and Artificial Intelligence, Changzhou University, Changzhou 213164, China
[5] School of Engineering, London South Bank University, London SE1 6EN, UK
* Correspondence: chenyr@zjsru.edu.cn

Abstract: Current authentication schemes based on zero-knowledge proof (ZKP) still face issues such as high computation costs, low efficiency, and security assurance difficulty. Therefore, we propose a secure and efficient authentication scheme (SEAS) for large-scale IoT devices based on ZKP. In the initialization phase, the trusted authority creates prerequisites for device traceability and system security. Then, we propose a new registration method to ensure device anonymity. In the identity tracing and revocation phase, we revoke the real identity of abnormal devices by decrypting and updating group public keys, avoiding their access and reducing revocation costs. In the authentication phase, we check the arithmetic relationship between blind certificates, proofs, and other random data. We propose a new anonymous batch authentication method to effectively reduce computation costs, enhance authentication efficiency, and guarantee device authentication security. Security analysis and experimental results show that an SEAS can ensure security and effectively reduce verification time and energy costs. Its security and performance exceed existing schemes.

Keywords: authentication; zero-knowledge proof; IoT device; privacy protection

Citation: Su, Z.; Wang, S.; Cai, H.; Huang, J.; Chen, Y.; Zhang, X.; Alam, M. A Secure and Efficient Authentication Scheme for Large-Scale IoT Devices Based on Zero-Knowledge Proof. *Electronics* **2024**, *13*, 3735. https://doi.org/10.3390/electronics13183735

Academic Editor: Andreas Mauthe

Received: 14 July 2024
Revised: 7 September 2024
Accepted: 9 September 2024
Published: 20 September 2024

1. Introduction

In recent years, the rapid advancement of Internet of Things (IoT) technology has resulted in a significant increase the number of IoT devices. As a consequence, there has been an exponential growth in compute-intensive and latency-sensitive applications, such as smart healthcare and the Internet of Vehicles [1]. These applications generate a massive volume of data that need to be processed and analyzed in real time to derive meaningful insights [2]. Data have become a critical productive element in today's society. By analyzing and processing IoT data, the quality of network applications and services can be improved, promoting rapid societal development [3]. However, as the number of devices in the IoT increases and malicious devices exist, the demands for data security and response time are continuously rising. Thus, achieving secure and efficient data and information exchange in the IoT environment poses a considerable challenge [4]. As the first line of defense for IoT system security, authentication—combined with other security mechanisms—ensures that only legitimate users or devices can access the system. It has become a research hotspot in IoT security and privacy issues [5]. However, when faced with large-scale and frequently authenticated intelligent devices in the IoT environment, as well as various malicious devices, traditional complex authentication schemes become challenging to apply to fast authentication and IoT scenarios that are full of various malicious attacks. Therefore, a secure and efficient authentication scheme is needed to ensure the security and efficiency

of legitimate devices during the authentication process and to prevent illegal access by malicious devices.

To protect users' privacy from illegal collection, current authentication schemes mainly use methods such as symmetric encryption, pseudonym certificates, group signatures, and zero-knowledge proof to achieve user anonymity. Among these, authentication methods based on symmetric encryption require a trusted key distribution center to manage preshared keys, facing the risk of insider attacks [6,7]. Authentication methods based on pseudonym certificates cannot guarantee the unlinkability of authentication. They need to frequently update pseudonym certificates to improve security, resulting in high computation costs [8,9]. Authentication methods based on group signatures require a group administrator for the distribution of group keys, also facing the risk of insider attacks [10,11]. Authentication methods based on zero-knowledge proof do not require third-party participation and do not require complex encryption algorithms [12,13]. Compared with other anonymous authentication schemes, this method is more secure and efficient, solving the problems of privacy leakage and computational complexity in the authentication process of IoT devices. However, the current mainstream authentication schemes based on zero-knowledge proof still have the following two problems, making them difficult to apply to current IoT systems: (1) Large computation cost and low efficiency: To ensure features such as anonymity and unlinkability, existing authentication schemes need to perform operations such as elliptic curve encryption and bilinear pairing, consuming a lot of computing resources of the device. This results in a large operation time for proof verification, low device authentication efficiency, and the inability to meet the real-time needs of IoT devices. (2) Difficulties in security assurance: In the IoT, there are illegal devices that impersonate legal devices to access the network and abnormal devices that behave maliciously after being controlled by physical or network methods. Existing authentication schemes are susceptible to insider attacks by illegal devices, with risks such as privacy leaks, and they are easily subjected to the malicious behaviors of abnormal devices (such as abnormal traffic and irregular reporting frequency), leading to system failures, security threats, and other risks.

Therefore, to address these issues, this paper proposes a secure and efficient authentication scheme (SEAS) for large-scale IoT devices based on zero-knowledge proof. The specific contributions of this paper are as follows:

(1) To satisfy the anonymity and traceability of devices, we improved the initialization method for identity authentication. This method uses the exponential properties of the Diffie–Hellman protocol to generate the public and private keys of the trusted authority (TA) and introduces a group public key, thereby creating preconditions for device traceability and system security. Additionally, we issued certificates to ensure device anonymity.

(2) To address the malicious behavior by abnormal devices and the high cost of revocation, we proposed a new identity tracing and revocation mechanism. In this mechanism, a device encrypts its public key using the public key of an authority and sends the ciphertext to the authority. After decrypting the ciphertext, the authority uses the public key to query relevant identity information in the database to trace the true identity of the abnormal device. The abnormal device is then revoked by updating the group public key, thereby preventing access by the abnormal device and reducing the cost of revocation. Therefore, it cannot lead to privacy leakage.

(3) To address the issues of large authentication calculation, low efficiency, and privacy leakage, we proposed a secure and efficient authentication method. To ensure device unlinkability, we proposed a certificate blinding equation that blinds the real certificates of the devices. By checking the arithmetic relationship of blind certificates, proofs, and other data, we proposed a new proof generation equation, verification equation, and batch verification equation. Based on these equations, we proposed a new anonymous batch authentication protocol, thus reducing computation costs, improving authentication efficiency, and ensuring device authentication security.

2. Related Work

The application of authentication technology extends across various domains, including intelligent transportation, smart healthcare, and smart home systems. Symmetric encryption schemes offer the advantage of reduced computation cost. Therefore, some researchers have explored authentication schemes based on encryption, as shown in Table 1. For instance, Zeng et al. [14] used symmetric encryption, a dual message mechanism, and a separate processing mechanism to achieve two-way authentication. However, their method requires a trusted party to manage preshared keys and also needs to ensure the security of key transmission over public channels. Maurya et al. [15] proposed a secure and efficient anonymous batch authentication scheme with conditional privacy (EABAS-CP) for the Internet of Vehicles (IoV) environment based on elliptic curve cryptography. This scheme generates pseudonyms and unique signatures using distributed parameters, enabling secure communication without the need for a secure channel or trusted authority, and it supports batch signature verification to improve verification efficiency. To solve the transmission problem of the key, some researchers have focused on studying authentication schemes based on pseudonym certificates. For instance, Qi et al. [16] used pseudonyms to enhance privacy protection for vehicle users. Nevertheless, the frequent need for updating pseudonym certificates in order to maintain efficient anonymity has resulted in additional costs. Wang et al. [17] proposed an anonymous identity authentication scheme based on consortium blockchain technology for vehicular ad hoc networks (VANETs). This scheme protects vehicle privacy by using pseudonyms to prevent location tracking and leverages a decentralized database to enhance system security and reliability, all while also improving efficiency in the revocation process. Qureshi et al. [18] proposed a blockchain-based conditional privacy preservation and authentication mechanism for IoV networks. This scheme ensures the anonymity of vehicle nodes and data control, guaranteeing the anonymity, traceability, and unlinkability of data sharing, all while enhancing the security and reliability of the consensus algorithm through a reputation voting system. In order to address this challenge and provide efficient anonymity without escalating update costs, researchers have turned their attention to the study of authentication schemes based on group signatures. These schemes aim to provide secure and reliable authentication while minimizing the cost associated with certificate updates. For instance, Jiang et al. [19] introduced an authentication scheme employing zero-knowledge proof, which hinges on the Diffie–Hellman problem principles. This scheme uses regional trust institutions as group administrators and vehicles as group members, and it employs a pseudonym mechanism and an identity mechanism based on group signatures, which reduces the costs of pseudonym certificate storage and verification. Gong et al. [20] introduced a threshold group signature scheme based on elliptic curves, specifically tailored for IoT applications. This innovative scheme diminishes the dependence on the group manager by collaboratively creating private keys for both group members and the manager. Additionally, it ensures the anonymity of group members by implementing a secondary anonymization process. However, the above methods [19,20] are still under the risk of insider attacks.

Zero-knowledge proof methods safeguard the prover's identity key by preventing the verifier from obtaining any information about it during the authentication process. This protection enables the application of both interactive and non-interactive zero-knowledge proof methods in authentication scenarios. Among them, some scholars have focused on studying authentication based on interactive zero-knowledge proof. For instance, Wang et al. [21] enhanced a three-round authentication scheme. They introduced a random challenge value into the communication exchanged between the verifier and the prover, effectively reducing the deception probability in single-round interaction from 2/3 to 1/2. However, this scheme needs to repeat authentication multiple times to achieve the desired security. Han et al. [22] developed a scheme that improves upon the Feige–Fiat–Shamir (FFS) protocol. They integrated a zero-to-one inversion and a two-to-one verification method, effectively addressing the challenge of weak resistance to guessing attacks. Xi et al. [23] introduced a secure and efficient anonymous authentication scheme (ZAMA). This

scheme employs elliptic curve cryptography (ECC) and Fujisaki–Okamoto (FO) commitment to achieve robust authentication. However, this scheme requires the maintenance of a revocation list. It mandates verifying if the current user appears on this list during each authentication process, which lowers the efficiency of authentication and escalates the need for additional storage. Boubakri et al. [24] constructed a protocol for cyber physical systems using the arithmetic relationships in Chebyshev chaotic mapping, polynomial order, and polynomial output. However, due to non-random data interactions in each authentication process, the protocol does not meet the criteria for unlinkability. Zhang et al. [25] proposed an authentication scheme (EEAS) based on Chebyshev chaotic mapping, that is, using a binary power algorithm based on square matrices and combining hash, XOR, and other lightweight encryption primitives. Its design aims to expedite authentication and key negotiation, thereby reducing both computation and communication costs in the authentication process. Wang et al. [26] developed a lightweight authentication method for embedded devices, substituting traditional cryptographic operations with Chebyshev polynomial operations and amalgamating all intermediate data into a single polynomial. However, refs. [21–26] required multiple rounds of communication, making their works unsuitable for batch verification and leading to lower authentication efficiency.

As a result, numerous researchers concentrate on authentication methods based on non-interactive zero-knowledge proof. For example, Ashutosh et al. [27] improved the traditional interactive Schnorr protocol, taking the commitment hash value computed by the prover as the verifier's challenge value and quickly verifying device identity through modular exponentiation. Liu et al. [28] used the Schnorr protocol for non-key-escrow registration of devices, and they used the proof of the equality of two discrete logarithms for device-to-device authentication, thereby protecting the authentication privacy of industrial IoT devices. Andola et al. [29] implemented drone anonymous authentication on the blockchain using bilinear mapping and ring signature algorithms. However, this scheme involves complex computations due to the use of bilinear mapping and ring signatures. Liu et al. [30] proposed a privacy protection authentication scheme for smart homes (UGPA). This scheme authenticates all authentication factors directly through the gateway to resist secret leakage attacks and uses Chebyshev chaotic mapping, hash functions, XOR, and other lightweight operations to reduce computation and communication costs. Jiang et al. [31] proposed an anonymous authentication scheme based on TRUG-PBFT master–slave chains and lattice-based zero-knowledge proofs. By optimizing the PBFT consensus process and reducing the number of consensus nodes, this scheme improves authentication efficiency, enables anonymous vehicle authentication, and effectively reduces computational overhead. However, refs. [27–31] still sufferred from large computational requirements for authentication and low efficiency, and they were unable to detect, trace, and revoke the identity of abnormal devices. In addition, all of the above schemes assume that the verifier is trustworthy, making the authentication process vulnerable to insider attacks.

Table 1. Summary of authentication schemes.

Literature	Category	Purpose	Security	Methods	Disadvantages
[14]	Non-zero-knowledge proof	Strengthen the security of industrial plants against cyber threats	Based on secret key secure distribution	Symmetric encryption; dual-message mechanism; separate processing mechanism	Dependent on trusted third-party
[16]		Implement efficient authentication in IoT	Based on the credibility of the certificate issuing authority	Pseudonym certificates	Frequent updating of pseudonym certificates
[19]		Defend against blockchain consensus attacks	Relies on Diffie–Hellman problem	Group signature	Dependent on trusted group administrators
[20]		Implement bidirectional authentication	Relies on elliptic curve discrete logarithm problem	Elliptic curve threshold group signature	

Table 1. *Cont.*

Literature	Category	Purpose	Security	Methods	Disadvantages
[21]	Interactive zero-knowledge proof	Minimize potential for successful deception by adversaries	Relies on matrix padding problem	Hash; equation derivation	Multiple rounds of communication; unsuitable for batch verification; low authentication efficiency
[22]		Implement lightweight authentication against guess attacks	Relies on quadratic residue problem	FFS protocol	
[23]		Implement efficient two-way anonymous authentication	Relies on discrete logarithm problem	ECC; FO commitment	
[24]		Implement secure authentication for cyber–physical systems	Relies on chaotic map-based discrete logarithm problem	Chebyshev polynomial; modular exponentiation	
[25]		Implement lightweight authentication for smart grid	Relies on discrete logarithm problem	Chebyshev polynomial; hash; XOR	
[26]		Implement lightweight authentication for embedded devices	Relies on chaotic map-based discrete logarithm problem	Chebyshev polynomial; hash	
[27]	Non-interactive zero-knowledge proof	Ensure confidentiality and anonymity	Relies on discrete logarithm problem	Schnorr protocol	Large authentication calculation and low efficiency; unable to detect, track, and revoke abnormal device identities
[28]		Solve the distrust problem of cross-domain authentication	Relies on discrete logarithm problem	Schnorr protocol	
[29]		Implement distributed authentication	Relies on discrete logarithm problem	Bilinear mapping; ring signature	
[30]		Defend against transient secret leakage attacks	Relies on chaotic map-based discrete ogarithm problem	Schnorr protocol; Chebyshev polynomial	
[31]		Implement identity authentication in V2X networks	Relies on inhomogeneous small integer solution problem	TRUG-PBFT; lattice-based zero-knowledge proof scheme	

3. Preliminary

3.1. Zero-Knowledge Proof

The zero-knowledge proof involves a protocol between two entities: the prover P and the verifier V. In this process, P actively demonstrates its knowledge of a secret (like a private key) to V while ensuring that no valuable information is disclosed [32]. A zero-knowledge proof generally consists of the following stages:

(1) Commit: The prover makes a commitment to the proposition, which waits to be challenged and verified by the verifier.

(2) Challenge: The verifier chooses a random number to challenge the proposed commitment.

(3) Response: The prover combines the random numbers received with the given promise (the promise cannot be modified) and returns a response to the challenge.

(4) Verify: The verifier verifies that the response to the challenge is correct, and if it is incorrect, the proof fails.

The prover and verifier repeat the above steps until the probability of believability reaches the condition accepted by the verifier and the proof succeeds.

Zero-knowledge proofs have the following characteristics:

(1) Completeness: If both the prover and the verifier are honest and comply with each step of the proof process to carry out the correct calculations, the proof will definitely succeed, and the verifier will be able to accept the prover.

(2) Soundness: No one can fake the prover to make the proof successful.

(3) Zero-Knowledge: After the proof process, the verifier only learns that "the prover has the knowledge", but not any information about the knowledge itself.

There is a discrete logarithm problem in zero-knowledge proofs, which is briefly described here: Given a finite cyclic group G of order p, a generator g of G, and an element $y \in G$, it is challenging to find an integer x, where $2 \leq x \leq p-1$, such that $g^x \bmod p = y$.

3.2. Security Goal

Referencing the security requirements of existing authentication schemes [23], our authentication scheme needs to satisfy the following security goals. An authentication scheme is usually considered secure if it satisfies the following five properties:

(1) Anonymity: The real identity of the device is kept confidential during the authentication process with the authentication center (AC). The AC is a trusted entity responsible for issuing, managing, and validating digital certificates in a network system. It provides digital certificates to various communication entities (devices, users, etc.) through a public key infrastructure (PKI) to ensure the authenticity, integrity, and confidentiality of communications.

(2) Unlinkability: Even if the authentication center is maliciously attacked or the wireless channel is controlled, malicious devices cannot identify or link the identity of normal devices.

(3) Traceability: If malicious behavior occurs when a device accesses a cloud service, the trusted authority can trace and revoke the real identity of the abnormal device.

(4) Forward Security: Even if a malicious device collects all the information of the current session, it cannot obtain any useful information about the previous session.

(5) Attack Resistance: The authentication scheme can resist common IoT attacks such as replay attacks, modification attacks, insider attacks, and man-in-the-middle attacks. Attackers cannot pose a threat to devices by replaying past legitimate data, tampering with data, illegally collecting data, or intercepting the communication information between the node and the authentication center. Additionally, they will not be able to obtain the user's private information from the intercepted data. Furthermore, tampered authentication data will not be able to pass the identity verification.

4. Proposed Protocol

The IoT device authentication communication framework used in this paper is shown in Figure 1, which is composed of IoT devices, a TA, an AC, and a cloud service platform. The IoT devices include normal and malicious devices. All IoT devices, as the prover, can initiate identity authentication requests with the AC. The AC, as the verifier, can collaborate with the TA to complete the identity authentication of IoT devices. Only the IoT devices that pass the verification can communicate with the cloud service platform and access cloud service resources, where normal devices are legitimate devices that have passed authentication and are working correctly. Malicious devices include illegal devices that impersonate legitimate devices to access the network, as well as abnormal devices that exhibit malicious behavior after being controlled by physical or network methods. The AC is responsible for authenticating the identities of IoT devices attempting to access cloud services. It can resist illegal devices but cannot know the real identities of the devices. The TA, acting as a trusted third party, is responsible for initializing the entire system and supports the detection, tracking, and revocation of abnormal device identities to resist abnormal devices. The cloud service platform provides various cloud services for authenticated legitimate devices.

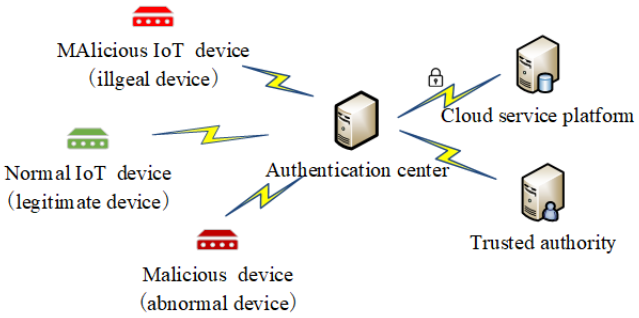

Figure 1. IoT device authentication communication framework.

Based on the above system model and security goals, we defined the security parameters, as shown in Table 2, and proposed the SEAS scheme. The specifics are as follows:

Table 2. Notation used in the proposed scheme.

Notation	Description	Notation	Description
ID	the device's identity	g, p	random number in cyclic group
P	the device's public key	c_1, c_2, c_3	the device's actual certificate
Z_p	the finite field	z_1, z_2	TA's private key
C_1, C_2, C_3	the device's blind certificate	T_1, T_2, T_3	the device's public key ciphertext
Y_1	group public key	D_1, D_2, D_3	the device's proof
Y_2, Y_3, Y_4	TA's public key	$Tstamp$	timestamp
x	AC's secret parameters	m	the device's request message
y, r	the device's secret parameters	$\|\|$	concatenation operation
k	the device's private key	$H()$	hash function

4.1. Scheme Initialization

This section primarily introduces the initialization phase of the scheme, including the TA's initialization, the AC's initialization, and the IoT devices' registration. These steps were only performed once.

(1) TA Initialization: To avoid the additional cost of a revocation list, we consider revoking devices by updating the TA's group public key. Therefore, after generating the public parameters $\{g, p\}$ by the TA, a random number $x \in Z_p$ is generated, and the group public key $Y_1 = g^x \bmod p$ is calculated to revoke the identity of abnormal devices. Then, in order to ensure that the identity of abnormal devices can be traced by the TA in the anonymous authentication process, we use the exponential property of the secure and efficient Diffie–Hellman protocol to construct the public–private key pair. That is, the TA selects a random number $\{y, z_1, z_2\} \in Z_p$ and calculates its public key $\{Y_2, Y_3, Y_4\}$ through Equation (1):

$$Y_2 = g^{z_2} \bmod p, Y_3 = g^{z_1} \bmod p, Y_4 = g^{z_1 \times z_2} \bmod p \quad (1)$$

(2) AC initialization: The AC obtains the public parameters $\{g, p\}$ and the secret parameter x generated by the TA, which are used to check whether a device's identity has been revoked.

(3) IoT device registration: Most registration methods assume that the AC is trustworthy. The TA needs to distribute a commitment or public key to the AC, which validates the device's proof. However, a semi-trusted AC can link the commitment to a specific device, leading to privacy leakage. Considering that the TA issues a certificate for each device—and the AC can verify identity by checking the arithmetic relationship among the certificate, proof, secret key, and group public key provided by the device—the computation and storage of device commitment can be avoided. Therefore, to achieve this goal, we improve the device registration method. As shown in Algorithm 1, the specific process is as follows:

Step1: The new device sends its ID to the TA.

Step2: The device selects a random number $k \in Z_p$ as the device's private key and calculates the device's public key $P = g^k \bmod p$ send to TA.

Step3: The TA selects random numbers $r, y \in Z_p$, generates the device's certificate $\{c_1, c_2, c_3\}$ through Equation (2), and sends the random number r, secret parameter y, private key k, public key P, and certificate over a secure channel to the device.

$$c_1 = g^r \bmod p, c_2 = c_1^y \bmod p, c_3 = c_1^x (P)^{rxy} \bmod p \quad (2)$$

Step4: The TA stores the device's public key P and ID in a local database.

Algorithm 1 Node Registration

Input: ID, k
Output: $r, y, P, \{c_1, c_2, c_3\}$
1: Send ID to TA
2: Select the random number $k \in Z_p$
3: Calculate $P = g^k \bmod p$
4: Select the random number $r \in Z_p$
5: Calculate $c_1 = g^r \bmod p, c_2 = c_1^y \bmod p, c_3 = c_1^x (P)^{rxy} \bmod p$ for $\{c_1, c_2, c_3\}$
6: Receive $r, y, P, \{c_1, c_2, c_3\}$ from TA
7: Save P and ID

4.2. SEAS Authentication Scheme

(1) Authentication: Firstly, to ensure that the device's proof is unlinkable, we propose a certificate blinding equation. This equation blinds the real certificate to protect the device's privacy.

$$C_1 = c_1^{r_2} \bmod p, C_2 = c_2^{r_2} \bmod p, C_3 = c_3^{r_1 r_2} \bmod p \tag{3}$$

where $\{C_1, C_2, C_3\}$ are used to generate and verify the device's proof. r_1 and r_2 are random numbers in Z_p.

Next, to ensure that the TA can trace the identity of abnormal devices when necessary, the device must use the TA's public key $\{Y_2, Y_3, Y_4\}$, encrypting its own public key P using Equation (4).

$$T_1 = {Y_2}^{\alpha} \bmod p, T_2 = {Y_3}^{\beta} \bmod p, T_3 = P \times ({Y_4}^{\alpha+\beta} \bmod p) \tag{4}$$

where α and β are random numbers in Z_p. The TA uses its private key $\{z_1, z_2\}$ and obtains the device's public key P through Equation (5). Then, it checks if P is stored in the local database to ensure that the device can be successfully traced.

$$P = T_3 / (T_1^{z_1} \times T_2^{z_2}) \tag{5}$$

Then, to reduce the computational amount and validate the arithmetic relations among the blind certificate, proof, group public key, and the message, we consider that the verification equation should meet the following conditions: (1) It only requires one modular exponentiation. (2) It can check if the public key in the blind certificate matches the private key in the proof, as well as whether the correct group public key has been provided in the proof generation. (3) It assures that the message m has not been tampered with, and the signatory of the message m is indeed the owner of the blind certificate. Therefore, to meet these requirements, we proposed the following proof generation equation. Due to space limitations, the design process has been omitted.

$$\begin{cases} D_1 = C_3^{rp} \times Y_1^{-r \times r_2 \times rm \times y} \times g^{H(m||Tstamp)} \bmod p \\ D_2 = r_1^{-1} \times D_1 + rp \\ D_3 = k \times D_1 + rm \end{cases} \tag{6}$$

where rp and rm are random numbers in Z_p. Then, based on the proof provided by the device, we proposed the following proof verification equation. We verified the identity of the device by comparing whether the values of $\widetilde{D_1}$ and D_1 are equal:

$$\widetilde{D_1} = C_3^{D_2} \times C_2^{-x \times D_3} \times C_1^{-x \times D_1} \times g^{H(m||Tstamp)} \bmod p \tag{7}$$

After that, to further improve the verification efficiency, we considered integrating multiple proofs into a single polynomial with homomorphic properties and proposed the following batch verification equation to verify multiple authentication messages at once.

$$\prod_{i=1}^{n} \widetilde{D_{1,i}} = (\prod_{i=1}^{n} C_{3,i}^{D_{2,i}} C_{2,i}^{-xD_{3,i}} C_{1,i}^{-xD_{1,i}}) g^{\sum_{i=1}^{n} H(m_i||Tstamp_i)} \bmod p \tag{8}$$

Finally, based on the proposed equation mentioned above, we presented an anonymous batch authentication protocol, aiming to achieve efficient device authentication and to withstand illegal devices and semi-trusted ACs. As shown in Algorithm 2, the specific process is as follows:

Algorithm 2 Node Identity Authentication

Input: $r_1, r_2, \alpha, \beta, m, Tstamp$
Output: $C_1, C_2, C_3, D_1, D_2, D_3, T_1, T_2, T_3, verificationresult$
1: Set $r_1, r_2, \alpha, \beta \in Z_p$
2: Set $m, Tstamp$
3: Calculate $C_1 = c_1^{r_2} \bmod p, C_2 = c_2^{r_2} \bmod p, C_3 = c_3^{r_1 r_2} \bmod p$
$D_1 = C_3^{r_4} \times Y_1^{-r \times r_2 \times r_3 \times y} \times g^{H(m||Tstamp)} \bmod p$
$D_2 = r_1^{-1} \times D_1 + r_4, D_3 = k \times D_1 + r_3$
$T_1 = {Y_2}^{\alpha} \bmod p, T_2 = {Y_3}^{\beta} \bmod p, T_3 = P \times ({Y_4}^{\alpha+\beta} \bmod p)$
4: Set $C_1, C_2, C_3, D_1, D_2, D_3, T_1, T_2, T_3, m, Tstamp$ to AC
5: **if** verify single signature **then**
6: Calculate $\prod_{i=1}^{n} \widetilde{D_{1,i}} = (\prod_{i=1}^{n} C_{3,i}^{D_{2,i}} \times C_{2,i}^{-x \times D_{3,i}} \times C_{1,i}^{-x \times D_{1,i}}) \times g^{\sum_{i=1}^{n} H(m_i||Tstamp_i)} \bmod p =^{?} \prod_{i=1}^{n} D_{1,i}$
7: Then $\prod_{i=1}^{n} D_{1,i} =^{?} \prod_{i=1}^{n} \widetilde{D_{1,i}}$
8: **end if**
9: **while** verify fail **do**
10: Divide n signatures into two subsets
11: Then verify $n/2$ signatures
12: End verify all signatures
13: **end while**

Step1: The device selects random numbers $r_1, r_2 \in Z_p$ and combines them with a certificate $\{c_1, c_2, c_3\}$ to calculate blind certificate $\{C_1, C_2, C_3\}$ using Equation (3). Then, the device selects random numbers $rp, rm \in Z_p$ and combines them with a current timestamp Tstamp, as well as secret parameters y and r, to generate the device's proof $\{D_1, D_2, D_3\}$ using Equation (6).

Step2: The device utilizes the TA's public key $\{Y_2, Y_3, Y_4\}$ to encrypt its own public key P into ciphertext $\{T_1, T_2, T_3\}$ using Equation (4). It combines the blind certificate, proof, public key ciphertext, timestamp, and message m to create a digital signature package, and sends it to the AC.

Step3: The AC securely transmits the received public key ciphertext to the TA through a secure channel. The TA uses Equation (5) to extract the device's public key P and checks if P is stored in the local database. Then, the TA returns the result to the AC. If P is stored in the database, the TA caches the public key ciphertext for future anomaly-tracing purposes.

Step4: If the AC receives a digital signature from an individual device, it combines the secret parameter x and calculates the verification data $\widetilde{D_1}$ using Equation (7). Then, the AC checks if the equation $\widetilde{D_1} =^{?} D_1$ is valid and whether the device's public key is stored in the database. If both conditions are met, the device's authentication is successful. Otherwise, the device's authentication fails. Lastly, the AC provides feedback on the authentication result to the current device.

Step5: If the AC receives more than n digital signatures from devices within a short period, it uses Equation (8) to calculate the verification data $\prod_{i=1}^{n} \widetilde{D_{1,i}}$ for the n devices. The AC checks if the equation $\prod_{i=1}^{n} D_{1,i} \stackrel{?}{=} \prod_{i=1}^{n} \widetilde{D_{1,i}}$ is valid and verifies if the public keys of all n devices are stored in the database. If both conditions are met, then the authentication for all devices is successful. Otherwise, the AC divides the n digital signatures into two subsets and performs batch verification again. This process continues until all legitimate digital signatures successfully pass the verification. Finally, the AC provides feedback on the authentication results to the respective n devices, thereby preventing access from illegal devices.

(2) Identity Traceability and Revocation Mechanism: In the field of detecting malicious behavior in devices, the TA employs convolutional neural networks with attention mechanisms to extract fine-grained features from historical observational time series data. Then, it uses long short-term memory (LSTM) modules to predict time series, thus discovering abnormal devices in the IoT [33]. Regarding the identity tracing and revocation of abnormal devices, we considered revoking the identity of abnormal devices by updating the group public key and not sharing it with these devices. This approach avoids the additional cost of revocation lists rather than adding the ID of abnormal devices to these lists. As shown in Algorithm 3, the detailed process is as follows:

Step1: The TA uses a convolutional neural network with an attention mechanism and LSTM to detect malicious behavior in devices in real time.

Step2: If the TA detects abnormal information from a device, it extracts the device's public key ciphertext from the anomaly. Combining this with the private key $\{z_1, z_2\}$, it calculates the abnormal device's public key P using Equation (5). Then, the TA searches the local database to find the real identity ID corresponding to the public key P.

Step3: The TA randomly selects a secret parameter $x' \in Z_p$ and regularly sends the secret parameter and the abnormal device's ID to the AC through a secure channel.

Step4: The AC generates a new group public key ${Y_1}'$ based on the secret parameter x', and it forwards this group public key to all legitimate IoT devices via a secure channel. Then, the AC cancels the forwarding to the abnormal device.

The abnormal device only knows the old group public key Y_1 and cannot provide the new correct group public key ${Y_1}'$ to successfully authenticate its identity, thereby defending against access from the abnormal device.

(3) Update of Certificates: This section mainly introduces the certificate update of the scheme. We have improved the method of certificate updating. The specific operation process is as follows:

Step1: The device sends its ID' to the TA.

Step2: The device selects a random number $k' \in Z_p$ as the device's private key and calculates the device's public key $P' = g^{k'} \bmod p$ send to the TA.

Step3: The TA selects random numbers $r', y' \in Z_p$, generates the device's certificate $\{{c_1}', {c_2}', {c_3}'\}$ through Equation (9), and sends the random number r', secret parameter y', private key k', public key P', and certificate over a secure channel to the device.

$$ {c_1}' = g^{r'} \bmod p, \quad {c_2}' = {c_1}^{y'} \bmod p, \quad c_3 = \left({c_1}^{x} (P')^{rxy}\right)' \bmod p \tag{9} $$

Step4: The TA stores the device's public key P' and ID' in a local database.

Algorithm 3 Detect Malicious Behavior

Input: none
Output: none
Detects the malicious behavior
if detect malicious behavior **then**
  Extract $\{T_1, T_2, T_3\}$
  Calculate $P = T_3/(T_1^{z_1} \times T_2^{z_2})$
  Search ID for abnormal node
  Select $x' \in Z_p$
end if
Send $\{x', ID\}$ to AC
Upgrade x'
Calculate ${Y_1}' = g^{x'} \bmod p$
Send ${Y_1}'$ to legal node Send ID to TA

5. Security Analysis

5.1. Zero-Knowledge Proof Analysis

First, we analyze the three important properties of the zero-knowledge proof that the SEAS satisfies to prove that the zero-knowledge proof protocol in the SEAS scheme is secure.

5.1.1. Completeness

Theorem 1. *In the SEAS, legitimate devices can generate a valid proof and successfully verify its identity based on data such as certificates, private keys, and secret parameters.*

Proof. Since a legitimate device knows data, such as certificate $\{c_1, c_2, c_3\}$, private key k, and secret parameter y and r, it can calculate the correct proof $\{D_1, D_2, D_3\}$ during each authentication process and meet the following conditions.

$$\begin{aligned}
&D_1 = C_3^{rp} \times {Y_1}^{-yrr_2 \times rm} \times g^{H(m||Tstamp)} \bmod p \\
&= (c_3^{r_1 r_2})^{rp} \times g^{-xyrr_2 \times rm} \times g^{H(m||Tstamp)} \bmod p \\
&= {c_3}^{r_1 r_2 \times rp} \times {c_1}^{-xyr_2 \times rm} \times g^{H(m||Tstamp)} \bmod p \\
&= (c_1^x \times P^{rxy})^{r_1 r_2 \times rp} \times {c_1}^{-xyr_2 \times rm} \times g^{H(m||Tstamp)} \bmod p \\
&= (c_1^{x+kxy})^{r_1 r_2 \times rp - xyr_2 \times rm} \times g^{H(m||Tstamp)} \bmod p \\
&= c_1^{xr_2 D_1 + xr_1 r_2 \times rp + kxyr_2 D_1 + kxyr_1 r_2 \times rp - r_2 xykD_1 - r_2 xy \times rm - xD_1 r_2} \times g^{H(m||Tstamp)} \bmod p \\
&= c_1^{(xr_1 r_2 + kxyr_1 r_2)(r_1^{-1} \times D_1 + rp) - r_2 xykD_1 - r_2 xy \times rm - xD_1 r_2} \times g^{H(m||Tstamp)} \bmod p \\
&= c_1^{(x+kxy) r_1 r_2 (r_1^{-1} D_1 + rp) - r_2 xy(kD_1 + rm) - xD_1 r_2} \times g^{H(m||Tstamp)} \bmod p \\
&= (c_1^x \times P^{rxy})^{r_1 r_2 (r_1^{-1} D_1 + rp)} \times {c_1}^{-r_2 xy(kD_1 + rm)} \times {c_1}^{-xD_1 r_2} \times g^{H(m||Tstamp)} \bmod p \\
&= (c_3^{r_1 r_2})^{r_1^{-1} D_1 + rp} \times (c_2^{r_2})^{-x(kD_1 + rm)} \times (c_1^{r_2})^{-xD_1} \times g^{H(m||Tstamp)} \bmod p \\
&= C_3^{D_2} \times C_2^{-xD_3} \times C_1^{-xD_1} \times g^{H(m||Tstamp)} \bmod p = \widetilde{D_1}
\end{aligned} \tag{10}$$

Equation (10) calculates that D_1 and $\widetilde{D_1}$ are equal in the batch validation mode and that the validation equations still holds, so the SEAS satisfies completeness. □

5.1.2. Soundness

Theorem 2. *In the SEAS, the probability of malicious devices passing authentication without data, such as certificates or private keys, is negligible.*

Proof. We assume that if malicious devices can pass the verification equation $\widetilde{D_1} = D_1$, then the devices must have a forged certificate $\{c_1, c_2, c_3\}$, and this certificate must have the form $c_2 = c_1^y \bmod p$ and $c_3 = c_1^{x+kxy} \bmod p$. Let $c_1 = g^\gamma \bmod p$, $c_2 = g^\eta \bmod p$, and

$c_3 = c_1^{x+kxy} \bmod p$. According to the discrete logarithm problem, $\eta/\gamma = y$ and $\eta/\gamma = y$ cannot be calculated. Therefore, the SEAS satisfies soundness. □

5.1.3. Zero-Knowledge

Theorem 3. *In the SEAS, the AC cannot obtain any valuable information from the digital signature provided by the device.*

Proof. The blind certificate that the AC receives from the device is independent of the real certificate. This is because we choose two large-range random numbers r_1 and r_2, and we calculate the blind certificate through Equation (3). We can find that C_1, C_2, and C_3 are random in each authentication process. Then, we prove its zero-knowledge property by constructing a simulator *sim*. The simulator *sim* chooses random numbers ${r_1}'$, ${r_2}'$, and ${r_3}'$ to calculate ${C_1}' = g^{{r_1}'} \bmod p$, ${C_2}' = g^{{r_2}'} \bmod p$, and ${C_3}' = g^{{r_3}'} \bmod p$. Since the probability distribution of the real blind certificate $\{C_1, C_2, C_3\}$ is the same as that of the simulated blind certificate $\{{C_1}', {C_2}', {C_3}'\}$, the blind certificate is correctly simulated. Similarly, we can prove that the distribution of the public key ciphertext and the proof in the digital signature are also correctly simulated. The AC cannot distinguish whether the digital signature is real or simulated. Therefore, the SEAS satisfies the zero-knowledge proof. □

5.2. Informal Security Analysis

The SEAS can effectively satisfy the security goals. The specific analysis is as follows.

5.2.1. Resisting Illegal Devices and Semi-Trusted AC

For illegal devices and the semi-trusted AC, the SEAS scheme can satisfy the following security goals, proving that the SEAS scheme can ensure the security of legal devices. The scheme is as follows:

(1) Anonymity

Theorem 4. *In the SEAS, illegal devices cannot obtain the real identity of legitimate devices by eavesdropping on the wireless channel or colluding with the AC.*

Proof. During the authentication phase, the device calculates a blind certificate, public key ciphertext, and proof that can be verified by the AC without exposing the real identity ID. Meanwhile, illegal devices cannot infer the real identity ID of legitimate devices. This is because the real identity ID of the legitimate device is never involved in the computation of all data. Therefore, the SEAS satisfies anonymity. □

(2) Unlinkability

Theorem 5. *In the SEAS, illegal devices and the AC cannot determine whether two digital signatures come from the same device.*

Proof. The SEAS uses a blind certificate to protect the device's real certificate. During the proof generation process, $\{r_1, r_2\}$, $\{\alpha, \beta\}$, and $\{rp, rm\}$ are random when calculating the blind certificate, public key ciphertext, and proof, respectively. Therefore, the digital signature sent by the device to the AC is different each time. In the proof verification process, the $\widetilde{D_1}$ value calculated by the AC is different each time. Illegal devices and the AC cannot determine whether two random digital signatures come from the same device. Therefore, the SEAS satisfies unlinkability. □

(3) Forward Secrecy

Theorem 6. *In the SEAS, illegal devices cannot obtain information from the previous session by observing the current session of a legitimate device.*

Proof. During the authentication process, the random numbers of each session are different, ensuring that each secret is the latest in the current session. Illegal devices cannot infer the random number of the last session from the random number of the current session, so it is difficult to obtain any previous information. Therefore, the SEAS satisfies forward secrecy. □

(4) Replay Attacks Resistance

Theorem 7. *In the SEAS, illegal devices cannot pass authentication by replaying the legitimate information obtained from previous sessions.*

Proof. During the authentication process, we use a timestamp Tstamp to calculate the identity proof of the device, that is, $D_1 = C_3^{rp} Y_1^{-yrr_2 \times rm} g^{H(m||Tstamp)} \bmod p$. Therefore, the parameter D_1 will change with the change in Tstamp, thereby avoiding outdated proof. Therefore, the SEAS resists replay attacks. □

(5) Modification Attacks Resistance

Theorem 8. *In the SEAS, a tampered digital signature cannot pass authentication.*

Proof. As known from Theorem 2, the device must provide the correct blind certificate, public key ciphertext, and proof to pass authentication. Therefore, the tampering of this information by illegal devices cannot pass the verification equation $\widetilde{D_1} =^? D_1$. Therefore, the SEAS resists modification attacks. □

(6) Man-in-the-middle Attacks Resistance

Theorem 9. *In the SEAS scheme, an attacker cannot intercept messages between the device and the AC.*

Proof. The authentication scheme of the SEAS uses a zero-knowledge proof protocol, which satisfies the properties of zero knowledge. Even if an attacker intercepts the communication between the device and the AC, they cannot obtain the device's private information from the intercepted data. Moreover, any tampered data cannot pass the authentication. □

(7) Insider Attacks Resistance

Theorem 10. *In the SEAS, a semi-trusted AC cannot know any privacy of the device.*

Proof. If the AC can infer the device's real certificate, public key, and private key from the known blind certificate, public key ciphertext, and proof, respectively, then this contradicts the discrete logarithm problem. In addition, the SEAS uses a zero-knowledge proof protocol and simultaneously satisfies properties such as anonymity and unlinkability. Therefore, the SEAS resists insider attacks. □

5.2.2. Resisting Abnormal Devices

For abnormal devices, the SEAS scheme supports the traceability of the identity of abnormal devices, proving that the SEAS scheme can prevent abnormal devices from reaccessing, damaging, and interfering with the system, thereby improving the security and stability of the entire IoT system.

Theorem 11. *In the SEAS, only the TA can trace and revoke the real identity of abnormal devices.*

Proof. Only the TA knows the private key $\{z_1, z_2\}$, obtains the device's public key P through Equation (5), and queries the local database to find the device's real identity ID. Therefore, the device's identity can only be traced by the TA. To ensure successful

authentication, each device is required to present the correct group public key. By updating this key, the TA can effectively revoke the identity of any abnormal device. □

5.3. Formal Security Analysis

In our study, we applied a detailed evaluation of the SEAS, utilizing the extensively recognized random-or-real (ROR) model [26]. We envisioned a scenario where adversary A manipulates a rogue device to intercept all communications between the normal device D and the AC. In the SEAS, adversary A can undermine the protocol's security by executing several oracle queries within a polynomial time frame: (1) $Send(AC, m_1, m_1{}')$. In this query sequence, adversary A masquerades as the normal device D, initiates the transmission of message m_1 to the AC, and subsequently receives the response message $m_1{}'$ from the AC. (2) $Send(D, m_2, m_2{}')$. In this query sequence, adversary A takes on the role of the AC, dispatches message m_2 to the normal device D, and then collects the reply message $m_2{}'$ sent back by the normal device D. (3) $Execute(AC, D, m)$. This query sequence depicts adversary A's capability to surreptitiously monitor the communication link between the normal device D and the AC, intercepting the message m in the process. (4) $hash(M)$. This query sequence represents the adversary A, which uses the message M to perform a *hash* query, checking whether the message M exists in the *hash* list. (5) $Test(D)$. This query is instrumental in evaluating the semantic security of the device's proof. Adversary A dispatches a test query to device D. Depending on whether the coin b equals 1 or 0, D either returns the genuine proof or a random proof of identical length to adversary A, respectively. In our analysis, we engaged in a security assessment of the SEAS through a series of games $G_i(i = 0, 1, 2, 3, 4)$. During these games, adversary A actively launches a set of $Test(D)$ queries targeting device D. Upon receiving these queries, device D conducts a random flip of a coin b, which can result in either a 0 or 1. Adversary A wins the game by accurately predicting its value. We established the probability $Adv_D(A)$ of adversary A's success in compromising our scheme's security and here denote this probability as

$$Adv_D(A) = |2\Pr[Succ_A] - 1| < \varepsilon \tag{11}$$

where $\Pr[Succ_A]$ represents the probability of adversary A emerging victorious in the game, ε is an extremely small value, which is considered negligible.

Theorem 12. *In the game G_i, adversary A possesses only a negligible chance of achieving victory within a probabilistic polynomial timeframe. To undermine the SEAS's semantic security, adversary A actively conducts Send queries, Execute queries, and hash queries, with each limited to a maximum of q_s, q_e, and q_h times, respectively. Then, we have*

$$Adv_D(A) = \frac{q_h^2}{2|hash|} + \frac{(q_s + q_e)^2}{2p} + \frac{q_s^6}{|hash|^6} + Adv_{DLP}(A) \tag{12}$$

where$|hash|$ represents the size of the hash query, and $Adv_{DLP}(A)$ represents the probability of adversary A successfully solving the DLP within a polynomial time frame.

Proof. We deduce the likelihood of adversary A's success in compromising the scheme via the game $G_i(i = 0, 1, 2, 3, 4)$. $\Pr[Succ_i]$ is defined as the probability of adversary A precisely forecasting the outcome of coin b in game G_i.

(a) G_0: The real scheme in random oracles and the initial game are assumed to be identical, so we obtain

$$Adv_D(A) = |2\Pr[Succ_0] - 1| \tag{13}$$

(b) G_1: In this game, we simulate adversary A's replay attack and man-in-the-middle attack by performing $Send(AC, m_1, m_1{}')$, $Send(D, m_2, m_2{}')$, and $Execute(AC, D, m)$ queries. Despite intercepting messages $\{C_1, C_2, C_3, D_1, D_2, D_3, T_1, T_2, T_3\}$, adversary A remains incapable of deriving the secret parameter $\{y, r\}$, private key k, and actual certificate $\{c_1, c_2, c_3\}$

from these messages. Furthermore, the incorporation of timestamp $Tstamp$ prevents outdated proofs from passing verification. Consequently, by the game's end, adversary A remains unable to discern whether the parameter device D that returns through the $Test(D)$ query is an authentic proof or merely a random proof. As a result, adversary A has no extra advantage in this real attack scenario. Therefore, we have

$$\Pr[Succ_1] = \Pr[Succ_0] \tag{14}$$

(c) G_2: This game effectively eliminates two collision cases in G_1 and simulates adversary A's modification attack.

C_1: Drawing from the birthday paradox [34], we deduce that the likelihood of a collision occurring in the *hash* query output is at most $\frac{q_h^2}{2|hash|}$. C_2: We also consider the scenario where the selected random number results in a collision, which is a probability that is no greater than $\frac{(q_s+q_e)^2}{2p}$. Excluding these cases, G_1 and G_2 remain indistinguishable. Therefore, we have

$$\Pr[Succ_2] - \Pr[Succ_1] \leq \frac{q_h^2}{2|hash|} + \frac{(q_s+q_e)^2}{2p} \tag{15}$$

(d) G_3: This game prevents adversary A from directly guessing the proof $\{C_1, C_2, C_3, D_1, D_2, D_3\}$ in G_2 without having to execute a hash query. The chances of this specific scenario happening are limited to a maximum of $\frac{q_s^6}{|hash|^6}$. Therefore, we can obtain

$$\Pr[Succ_3] - \Pr[Succ_2] \leq \frac{q_s^6}{|hash|^6} \tag{16}$$

(e) G_4: The game modifies G_3 to simulate the insider attack. Within this game, adversary A carries out the $Execute(AC, D, m)$ query to capture all messages exchanged between the device D and the AC over the communication channel. With knowledge of $\{C_1, C_2, C_3, D_1, D_2, D_3\}$, adversary A must know $\{c_1, c_2, c_3\}$ and $\{k, y, r\}$ to calculate the device's proof. However, to deduce $\{c_1, c_2, c_3\}$ from $C_1 = c_1^{r_2} \bmod p$, $C_2 = c_2^{r_2} \bmod p$ and $C_3 = c_3^{r_1 r_2} \bmod p$, as well as to deduce $\{k, y, r\}$ from $D_1 = C_3^{rp} Y_1^{-rr_2 rmy} g^{H(m||Tstamp)} \bmod p$, $D_2 = r_1^{-1} D_1 + rp$, and $D_3 = kD_1 + rm$, adversary A must solve the DLP. By analyzing the discrepancies between this game and G_3, we can obtain

$$\Pr[Succ_4] - \Pr[Succ_3] \leq Adv_{DLP}(A) \tag{17}$$

In the above games, adversary A's ability to correctly predict the b value does not improve. There, we have

$$\Pr[Succ_4] = \frac{1}{2} \tag{18}$$

By analyzing Equations (13) and (14), we have

$$\frac{1}{2} Adv_D(A) = \left| \Pr[Succ_0] - \frac{1}{2} \right| = \left| \Pr[Succ_1] - \frac{1}{2} \right| \tag{19}$$

By analyzing Equations (18) and (19), we have

$$\frac{1}{2} Adv_D(A) = |\Pr[Succ_0] - \Pr[Succ_4]| \tag{20}$$

Considering Equations (15)–(17) and applying the triangular inequality, we have

$$\begin{aligned}
&|\Pr[Succ_1] - \Pr[Succ_4]| = |\Pr[Succ_1] - \Pr[Succ_2] \\
&+ \Pr[Succ_2] - \Pr[Succ_3] \\
&+ \Pr[Succ_3] - \Pr[Succ_4]| \\
&\leq |\Pr[Succ_1] - \Pr[Succ_2]| \\
&+ |\Pr[Succ_2] - \Pr[Succ_3]| \\
&+ |\Pr[Succ_3] - \Pr[Succ_4]| \\
&\leq \frac{q_h^2}{2|hash|} + \frac{(q_s+q_e)^2}{2p} + \frac{q_s^6}{|hash|^6} + Adv_{DLP}(A)
\end{aligned} \tag{21}$$

Finally, by integrating the findings from Equations (20) and (21), we derive Equation (12). This process validates the proof of Theorem 11. It effectively demonstrates that adversary A's likelihood of succeeding in the game is minuscule.

Under the ROR model, the guessing attacks, replay attacks, modification attacks, and insider attacks have low success rates. Therefore, the SEAS scheme can effectively defend against these attacks. It is relatively safe. □

5.4. Security Comparison

Based on the above analysis, we compared the security of the SEAS with existing authentication schemes [23,25,30] aimed at IoT devices. Additionally, we simulated four common types of attacks on the system using malicious devices:

1. Replay Attacks: The malicious device listens to and records messages between honest devices and the autonomous system on the wireless channel;
2. Modification Attacks: The malicious device disrupts the system's integrity, confidentiality, and availability by altering communication data;
3. Insider Attacks: The malicious device gains physical access to the device registration data stored in the database and uses these data to establish authentication communication with the application server;
4. Man-in-the-Middle Attacks: The malicious device intercepts and modifies communication data between honest devices and the autonomous system via the wireless channel.

As shown in Table 3, the SEAS, ZAMA, EEAS, and UGPA could defend against replay attacks, modification attacks, insider attacks,man-in-the-middle attacks while also performing well in terms of traceability and forward secrecy. The SEAS had stronger security compared to other authentication schemes. In the ZAMA scheme, the AC was able to know the old session key and random challenge value of the device during the reverification process, so the AC could link two random proofs to the same device. At the same time, if we assume that the attacker is an internal member of the AC, the attacker can also link two random proofs to the same device and infer private information. Therefore, ZAMA does not satisfy unlinkability and is prone to insider attacks. As the AC of the EEAS and UGPA schemes must obtain the device's ID through the XOR operation to verify identity, the AC can know the true identity of each device during the authentication process. Similarly, assuming the attacker is an internal member of the AC, the attacker can know the true identity of the device and link two random proofs to the same device. Therefore, the EEAS and UGPA do not satisfy anonymity and unlinkability and are also susceptible to insider attacks.

Table 3. Security comparison.

Authentication	Anonymity	Unlinkability	Traceability	Forward Secrecy	Replay Attacks Resistance	Modification Attacks Resistance	Insider Attacks Resistance	Man-in-the-Middle Attacks Resistance
SEAS	✓	✓	✓	✓	✓	✓	✓	✓
ZAMA	✓	×	✓	✓	✓	✓	×	✓
EEAS	×	×	✓	✓	✓	✓	×	✓
UGPA	×	×	✓	✓	✓	✓	×	✓

6. Experimental Analysis

6.1. Experiment Parameters and Performance Index Selection

This paper conducted a performance evaluation of the SEAS using an experimental environment. The environment comprised a 64-bit Win11 operating system and an AMD Ryzen 7 5800U. We used the Python 3.7 language to implement an authentication system composed of IoT devices (including normal devices, illegal devices, and abnormal devices), the AC, the TA, and cloud service platforms. In the experiment, when the finite field security size reached 1024 bits, the authentication was secure enough [22]. Therefore, combined with the above-mentioned authentication system, we chose the size of the finite field to be 1024 bits. We studied the time and energy costs of the SEAS, ZAMA, EEAS, and UGPA under malicious device attacks in the registration stage, identity verification stage, and batch verification stage to evaluate their computational efficiency. At the same time, we studied the time and energy costs of the identity traceability and revocation mechanism, as well as their impacts on the throughput of the cloud service platform. The time cost was used to evaluate the authentication efficiency, which was defined as the total time spent by the device. The energy cost was calculated based on CPU cost and was used to evaluate the computational amount, which was defined as the product of the CPU power consumption and the time cost spent by the device in the above stages [23]. The throughput was defined as the number of requests processed by the cloud service platform divided by the total processing time [35].

6.2. Experimental Result Analysis

We first compared the time and energy costs of various cryptographic elements. The bit lengths of the modular exponentiation and Chebyshev polynomial operations were both selected to be 1024 bits using the elliptic curve (EC) secp256k1, and the unified one-way hash function used was SHA1 [25]. We constructed different cryptographic elements and executed each cryptographic element 1000 times. The results are shown in Table 4.

Table 4. Cost of cryptographic elements.

Operations	Time Cost (ms)	Energy Cost (W)
Chebyshev polynomial	1.4984999	0.2364956
SHA1	0.0018001	0.0006572
Modular exponentiation	0.1312522	0.0190409
EC encryption	0.2199816	0.0368030
EC decryption	0.2300024	0.0790758

6.2.1. Registration Phase

We analyzed the impact of the number of devices on the computational overhead and energy costs in the registration phase. As shown in Figure 2, with the number of devices increasing, the computational overhead and energy costs of the SEAS, ZAMA, UGPA, and EEAS gradually increased, but the SEAS and ZAMA essentially maintained consistency and were lower than the UGPA and EEAS. The reason is that in the registration phase of each device, the SEAS and ZAMA only need to perform four modular exponentiation operations. In contrast, the UGPA needs to perform a total of four hash operations and two Chebyshev polynomial operations, The EEAS needs to perform a total of thirteen hash operations and seven Chebyshev polynomial operations.

6.2.2. Identity Verification Phase

We analyzed the impact of the number of devices on the computational overhead and energy costs in the identity verification phase. As shown in Figure 3, with the number of devices increasing, the computational overhead and energy costs of the SEAS were significantly lower than other schemes. The reason is that in the identity verification phase of each device, the AC of the SEAS only needs to perform one hash operation and six modular exponentiation operations. However, the AC of the ZAMA performs one hash operation, eight modular exponentiation operations, and one operation each for elliptic

curve encryption and decryption. The UGPA requires eighteen hash operations and nine Chebyshev polynomial operations. The EEAS involves executing six hash operations and two Chebyshev polynomial operations.

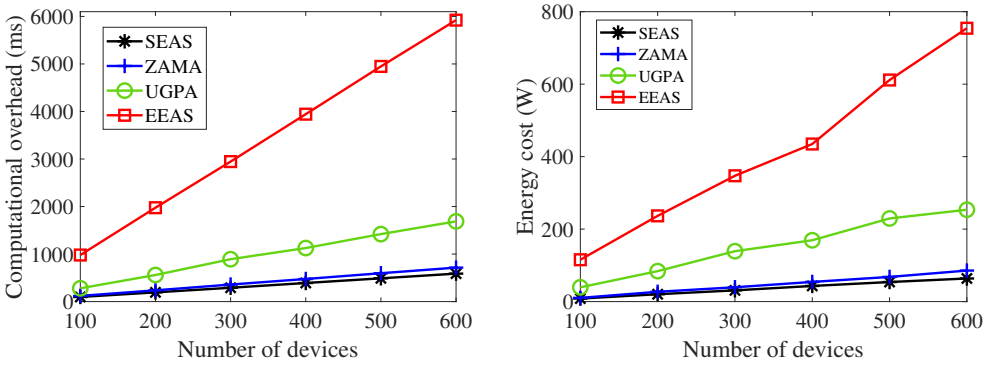

Figure 2. Computational overhead and energy costs in the registration phase.

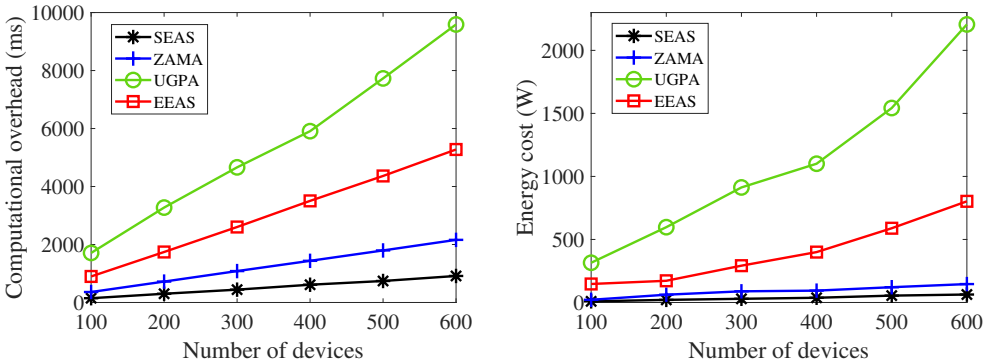

Figure 3. Computational overhead and energy costs in the identity verification phase.

We analyzed the impact of the number of devices on the computational overhead and energy costs in the batch verification phase. As shown in Figure 4, with the number of devices increasing, the computational overhead and energy costs of the batch verification phase of the SEAS were significantly lower than those verifying all devices individually. The reason is that we considered integrating the proofs of multiple devices into the proposed polynomial with homomorphic properties. By combining the verification operations of multiple devices, we could verify multiple devices simultaneously, reducing the repetitive verification computations and thereby lowering the computational overhead and energy costs.

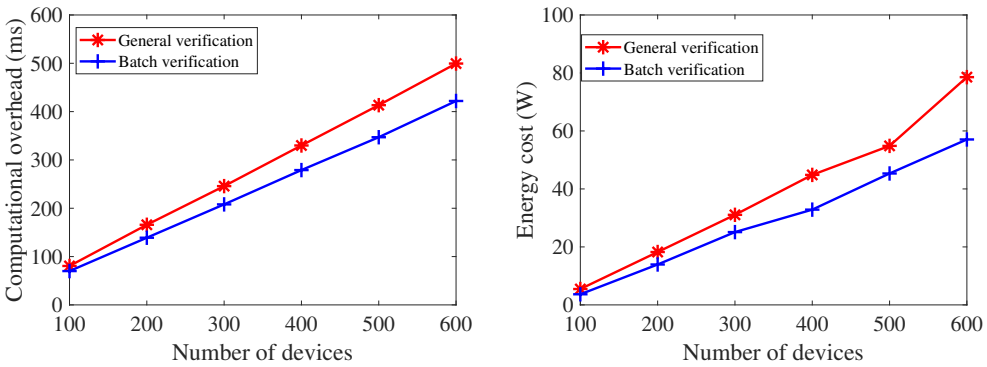

Figure 4. Computational overhead and energy costs in the batch verification phase.

6.2.3. Identity Tracing and Revocation Phase

We analyzed the impact of the proportion of abnormal devices on the throughput of the platform. As shown in Figure 5, as the proportion of abnormal devices increased, the throughput of the platform showed a downward trend. However, because the SEAS adopted an identity tracing and revocation mechanism, its platform throughput decreased relatively slowly and was significantly higher than the throughput of the platforms without this mechanism. The reason is that as the proportion of abnormal devices increases, the platform receives more and more requests and data from abnormal devices, and the platform spends more computational overhead and resources on processing device requests. Thereby, it reduces its throughput. Meanwhile, the identity tracing and revocation mechanism of the SEAS can comprehensively measure the malicious behavior of abnormal devices and can achieve the high-accuracy detection of abnormal devices, as well as tracing and revoking their real identities. Thereby, it prevents abnormal devices from successfully accessing the cloud service platform again and improves platform throughput.

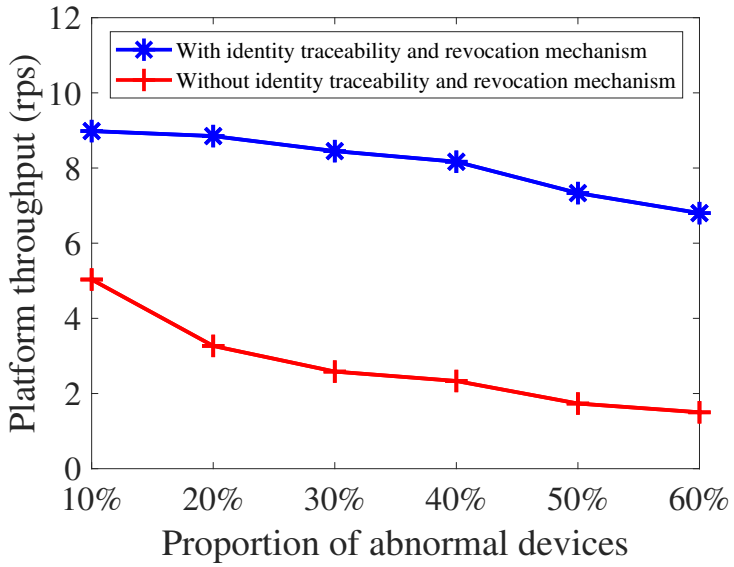

Figure 5. Impact of the proportion of abnormal devices on platform throughput.

We analyzed the impact of the number of abnormal devices on the computational overhead and energy costs of the identity tracing and revocation phase. Since the EEAS does not support the revocation function of device identities, Figure 6 only compares the computational overhead and energy costs of SEAS, ZAMA, and UGPA. As shown in Figure 6, with the number of abnormal devices increasing, the computational overhead and energy costs of the SEAS were significantly lower than other schemes. The reason is that the SEAS can trace the identity of a single abnormal device by only performing two modular exponentiations and only needs to perform one modular exponentiation to update the group public key, which can revoke all device identities at the same computational overhead. Therefore, the computational overhead and energy costs in this phase were the lowest. Although the ZAMA also only needs to perform two modular exponentiations, it needs to check whether the revoked device identity is stored in the local database and needs to be added to the revocation list, resulting in additional computational overhead and energy costs. On the other hand, the UGPA needs to perform costly Chebyshev polynomial operations to recalculate the device identity credentials and replace the original credentials stored in the local database, so its cost was the highest.

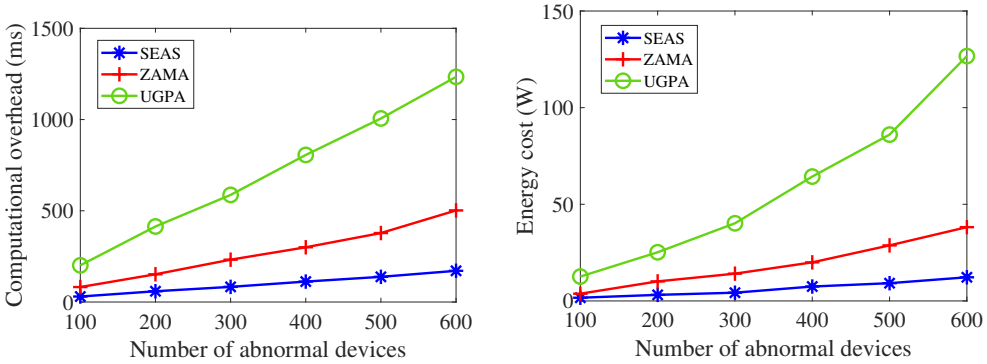

Figure 6. Computational overhead and energy costs in the revocation phase.

7. Conclusions and Discussion

7.1. Conclusions

Compared to the ZAMA, UGPA, and EEAS schemes, our SEAS scheme improved the identity authentication protocol, avoiding the internal attack of the semi-trusted AC while ensuring anonymity and unlinkability, as well as improving enhancing security. Therefore, due to the anonymity and unlinkability provided by the SEAS during the identity authentication process, attackers cannot obtain any valuable information about the device, even if they steal the authentication information. In terms of efficiency, we optimized the verification equation to complete the authentication with a single modular exponentiation operation, which make it suitable for the scenario where a large number of devices need to be verified. Moreover, in terms of privacy, the SEAS verifies the device identity by transmitting random blind certificates and proofs during each authentication process, so the attacker cannot deduce any identity information of the device from random data. Security analysis and experimental results show that SEAS can ensure the security of IoT devices and systems while effectively reducing the computational overhead and energy costs of the verification phase.

7.2. Discussion

Additionally, the SEAS scheme primarily operates on IoT devices and the AC. Namely, IoT devices perform identity registration and authentication. In the registration phase, the device generates the public and private keys and proofs, which involve modular exponentiation. Therefore, the SEAS scheme in IoT devices has the time complexity of O(log$l$), where l represents the length of the secret key, namely, the size of the finite field. In the authentication phase, the device performs blind certificate and proof generation, secret key encryption, and other operations, all of which involve modular exponentiation. Its time complexity is O(log$l$). Therefore, the time complexity of the SEAS scheme in the IoT devices is O(log$l$). The AC handles the verification of single and multiple devices. When the AC processes decryption, database lookup, and verification proofs of a single device, the time complexity is O(log$l$). When the AC verifies n devices in batches, the time complexity is O(nlog$l$). Therefore, the time complexity of the SEAS scheme is O(nlog$l$). In the ZAMA scheme, the device generates the proof, and the AC verifies the device requests through ECC encryption and modular exponentiation operations. It has the time complexity of O(nlog$l$). In the EEAS and UGPA schemes, the device and the AC generate authentication data and verify the authentication request through Chebyshev polynomials, respectively, so their time complexity is O(nl). In summary, the time complexities of the SEAS and ZAMA schemes were lower than those of the EEAS and UGPA schemes.

The zero-knowledge-proof-based authentication scheme of the SEAS holds great potential for practical applications, especially in IoT environments. With the rapid growth of IoT devices, security and efficiency have become core requirements across various application scenarios. In intelligent transportation systems (IoVs), the SEAS effectively addresses the real-time communication authentication needs between vehicles and infrastructure,

reducing security risks from potential insider attacks. The SEAS lowers computational overhead through an efficient authentication process, making it suitable for the IoV scenario that demands high real-time performance. Additionally, in data-intensive environments such as smart healthcare and smart grids, the SEAS not only ensures device authentication but also secures data transmission, providing robust security for large-scale IoT device management and enhancing system scalability and resilience. However, despite the significant security and efficiency improvements in the SEAS, there are still challenges in real-world applications. For instance, differences in the computational capabilities and network bandwidth of various IoT devices affect the scheme's execution efficiency. Therefore, device processing power and bandwidth limitations must be considered in different IoT scenarios to avoid potential bottlenecks. On highly resource-constrained devices, SEAS implementation requires careful optimization to prevent performance issues. Moreover, in complex network environments, further optimization of the scheme's communication and computational costs is necessary to adapt to dynamically changing network conditions. These challenges should be the focus of future research and practical applications. Therefore, the next goal is to improve the practicality and scalability of the scheme through integration with emerging technologies like blockchain and 5G communication.

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

Funding: This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFC2504501 and the Key Research and Development Program of Zhejiang under Grant 2022C03147.

Data Availability Statement: Data are contained within the article.

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

References

1. Mishra, K.; Rajareddy, G.N.; Ghugar, U.; Chhabra, G.S.; Gandomi, A.H. A collaborative computation and offloading for compute-intensive and latency-sensitive dependency-aware tasks in dew-enabled vehicular fog computing: A federated deep Q-learning approach. *IEEE Trans. Netw. Serv. Manag.* **2023**, *20*, 4600–4614. [CrossRef]
2. Chen, H.; Chen, Y.; Xiong, Z.; Han, M.; He, Z.; Liu, B.; Wang, Z.; Ma, Z. Prevention method of block with-holding attack based on miners' mining behavior in blockchain. *Appl. Intell.* **2023**, *53*, 9878–9896. [CrossRef]
3. Zhang, Y.; Chen, Y.; Miao, K.; Ren, T.; Yang, C.; Han, M. A novel data-driven evaluation framework for fork after withholding attack in blockchain systems. *Sensors* **2022**, *22*, 9125. [CrossRef] [PubMed]
4. Chen, Y.; Chen, H.; Zhang, Y.; Han, M.; Siddula, M.; Cai, Z. A survey on blockchain systems: Attacks, defenses, and privacy preservation. *High-Confid. Comput.* **2022**, *2*, 100048. [CrossRef]
5. Nayak, B.P.; Hota, L.; Kumar, A.; Turuk, A.K.; Chong, P.H. Autonomous vehicles: Resource allocation, security, and data privacy. *IEEE Trans. Green Commun. Netw.* **2022**, *6*, 117–131. [CrossRef]
6. Zhao, X.; Li, D. A lightweight user authentication scheme for multi-gateway based wireless sensor networks using rabin cryptosystem. *IEEE Access* **2023**, *11*, 79874–79889. [CrossRef]
7. Zhang, Y.; He, D.; Vijayakumar, P.; Luo, M.; Huang, X. SAPFS: An efficient symmetric-key authentication key agreement scheme with perfect forward secrecy for industrial internet of things. *IEEE Internet Things J.* **2023**, *10*, 9716–9726. [CrossRef]
8. Zhuang, L.; Guo, N.; Chen, Y. TriNymAuth: Triple pseudonym authentication scheme for vanets based on cuckoo filter and paillier homomorphic encryption. *Sensors* **2023**, *23*, 1164. [CrossRef]
9. Sang, G.; Chen, J.; Liu, Y.; Wu, H.; Zhou, Y.; Jiang, S. PACM: Privacy-preserving authentication scheme with on-chain certificate management for VANETs. *IEEE Trans. Netw. Serv. Manag.* **2023**, *20*, 216–228. [CrossRef]
10. Yuan, W.; Li, X.; Li, M.; Zheng, L. DCAGS-IoT: Dynamic cross-domain authentication scheme using group signature in IoT. *Appl. Sci.* **2023**, *13*, 5847. [CrossRef]
11. Li, J.; Hou, N.; Zhang, G.; Zhang, J.; Liu, Y.; Gao, X. Efficient conditional privacy-preserving authentication scheme for safety warning system in edge-assisted internet of things. *Mathematics* **2023**, *11*, 3869. [CrossRef]
12. Chen, Z.; Jiang, Y.; Song, X.; Chen, L. A survey on zero-knowledge authentication for internet of things. *Electronics* **2023**, *12*, 1145. [CrossRef]

13. Hamila, F.; Hamad, M.; Salgado, D.C.; Steinhorst, S. Enhancing security in fiat-shamir transformation-based non-interactive zero-knowledge protocols for iot authentication. *Int. J. Inf. Secur.* **2023**, *1*, 1131–1148. [CrossRef]
14. Upadhyay, D.; Zaman, M.; Joshi, R.; Sampalli, S. An efficient key management and multi-layered security framework for scada systems. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 642–660. [CrossRef]
15. Chanchal, M.; Chaurasiya, V. Efficient anonymous batch authentication scheme with conditional privacy in the Internet of Vehicles (IoV) applications. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 9670–9683.
16. Duan, L.; Li, Y.; Liao, L. Non-interactive certificate update protocol for efficient authentication in IoT. *Future Gener. Comput. Syst. -Int. J. Escience* **2020**, *113*, 132–144. [CrossRef]
17. Wang, L.; Zheng, D.; Guo, R.; Hu, C.; Jing, C. A blockchain-based privacy-preserving authentication scheme with anonymous identity in vehicular networks. *Int. J. Netw. Secur.* **2020**, *22*, 981–990.
18. Qureshi, K.N.; Shahzad, L.; Abdelmaboud, A.; Elfadil Eisa, T.A.; Alamri, B.; Javed, I.T.; Al-Dhaqm, A.; Crespi, N. A blockchain-based efficient, secure and anonymous conditional privacy-preserving and authentication scheme for the internet of vehicles. *Appl. Sci.* **2022**, *12*, 476–492. [CrossRef]
19. Zhang, S.; Lee, J. A group signature and authentication scheme for block-chain-based mobile-edge computing. *IEEE Internet Things J.* **2020**, *7*, 4557–4565. [CrossRef]
20. Gong, B.; Zhang, X.; Cao, Y.; Li, Z.; Yang, J.; Wang, W. A threshold group signature scheme suitable for the internet of things. *Concurr. Comput.-Pract. Exp.* **2021**, *33*, e6243. [CrossRef]
21. Houzhen, W.; Xinwei, C.; Yan, G.; Huanguo, Z. 5-pass zero-knowledge identity authentication scheme based on matrix completion problem. *J. Commun.* **2021**, *42*, 79–86.
22. Han, M.; Yin, Z.; Cheng, P.; Zhang, X.; Ma, S. Zero-knowledge identity authentication for internet of vehicles: Improvement and application. *PLoS ONE* **2021**, *15*, e0239043. [CrossRef] [PubMed]
23. Xi, N.; Li, W.; Jing, L.; Ma, J. ZAMA: A zkp-based anonymous mutual authentication scheme for the iov. *IEEE Internet Things J.* **2022**, *9*, 22903–22913. [CrossRef]
24. Boubakri, W.; Abdallah, W.; Boudriga, N. ZAO-AKA: A zero knowledge proof chaotic authentication and key agreement scheme for securing smart city cyber physical system. *Wirel. Netw.* **2021**, *27*, 4199–4215. [CrossRef]
25. Zhang, L.; Zhu, Y.; Ren, W.; Wang, Y.; Choo, K.K.R.; Xiong, N.N. An energy-efficient authentication scheme based on chebyshev chaotic map for smart grid environments. *IEEE Internet Things J.* **2021**, *8*, 17120–17130. [CrossRef]
26. Wang, Z.; Huang, J.; Miao, K. Lightweight zero-knowledge authentication scheme for IoT embedded devices. *Comput. Netw.* **2023**, *236*, 110021. [CrossRef]
27. Dwivedi, A.D.; Singh, R.; Ghosh, U.; Mukkamala, R.R.; Tolba, A.; Said, O. Privacy preserving authentication system based on non-interactive zero knowledge proof suitable for Internet of Things. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *13*, 4639–4649. [CrossRef]
28. Liu, S.; Chen, L.; Yu, H.; Gao, S.; Fang, H. BP-AKAA: Blockchain-enforced privacy-preserving authentication and key agreement and access control for IIoT. *J. Inf. Secur. Appl.* **2023**, *73*, 103443. [CrossRef]
29. Andola, N.; Raghav, Yadav, V.K.; Venkatesan, S.; Verma, S. SpyChain: A lightweight blockchain for authentication and anonymous authorization in IoD. *Wirel. Pers. Commun.* **2021**, *119*, 343–362. [CrossRef]
30. Liu, W.; Wang, X.; Peng, W. NCZKP based privacy-preserving authenti-cation scheme for the untrusted gateway node smart home environment. In Proceedings of the 2020 IEEE Symposium on Computers and Communications (ISCC), Rennes, France, 7–10 July 2021; pp. 391–396.
31. Jiang, W.; Guo, Z. An anonymous authentication scheme for Internet of Vehicles based on TRUG-PBFT master-slave chains and Zero-Knowledge Proof. *IEEE Internet Things J.* **2024**, 1–15. [CrossRef]
32. Singh, R.; Dwivedi, A.D.; Srivastava, G.; Chatterjee, P.; Lin, J.C.W. A privacy-preserving internet of things smart healthcare financial system. *IEEE Internet Things J.* **2023**, *10*, 18452–18460. [CrossRef]
33. Liu, Y.; Garg, S.; Nie, J.; Zhang, Y.; Xiong, Z.; Kang, J.; Hossain, M.S. Deep anomaly detection for time-series data in industrial iot: A communication-efficient on-device federated learning approach. *IEEE Internet Things J.* **2021**, *8*, 6348–6358. [CrossRef]
34. Lyubashevsky, V.; Micciancio, D. Generalized compact knapsacks are collision resistant. *Proc. Autom. Lang. Program.* **2006**, *4052*, 144–155.
35. Chen, Y.; Zhang, Y.; Chen, H.; Han, M.; Liu, B.; Ren, T. Efficient consistency consensus algorithm of blockchain for heterogeneous nodes in the internet of vehicles. *J. Electron. Inf. Technol.* **2022**, *44*, 314–323.