electronics

MDPI

Article

A Trusted Internet of Things Access Scheme for Cloud Edge Collaboration

Wenlong Zhu *, Changli Zhou and Linmei Jiang

College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China; zcl@hqu.edu.cn (C.Z.); clough@hqu.edu.cn (L.J.)
* Correspondence: zhuwl@hqu.edu.cn

Abstract: With the rapid popularization of current Internet of Things (IoT) technology and 5G networks, as well as the continuous updating of new service lifestyles and businesses, the era of big data processing for the IoT has arrived. However, centralizing all data for processing in the cloud can lead to issues such as communication latency and privacy breaches. To solve these problems, edge computing, as a new network architecture close to terminal data sources and supporting low latency services, has gradually emerged. In this context, cloud edge collaborative computing has become an important network architecture. With the changing security requirements and communication methods of cloud edge collaborative network architecture, traditional authentication key agreement protocols are no longer applicable. Therefore, a new IoT authentication and key agreement protocol needs to be designed to solve this problem. This study proposes an IoT accessible solution for cloud edge collaboration. This scheme adopts a chaotic mapping algorithm to achieve efficient authentication. It ensures the anonymity and untraceability of users. Following this, we conducted strict security verification using BAN logic and Scyther tools. Through experimental comparative analysis, the research results show that the protocol performs better than other schemes while ensuring security. This indicates that the protocol can achieve efficient authentication and key negotiation in cloud edge collaborative network architecture, providing a secure and reliable solution for the accessibility of the IoT.

Keywords: Internet of Things; protocol; chaotic mapping algorithm; cloud edge collaboration

check for updates

Citation: Zhu, W.; Zhou, C.; Jiang, L. A Trusted Internet of Things Access Scheme for Cloud Edge Collaboration. *Electronics* **2024**, *13*, 1026. https://doi.org/10.3390/electronics13061026

Academic Editor: Fernando De la Prieta Pintado

Received: 22 January 2024
Revised: 26 February 2024
Accepted: 6 March 2024
Published: 8 March 2024

Copyright: © 2024 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) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The Internet of Things (IoT) industry has witnessed significant growth in recent years due to the rapid advancements in new generation information technology. These technological advancements have played a crucial role in fostering the development and expansion of the IoT industry [1].

The proliferation of IoT devices connecting to the Internet has led to a substantial surge in network data and processes. The updates of various information technologies have driven the upgrading of devices; image recognition, vehicle networks, industrial internet, and other time sensitive and computationally intensive businesses continue to emerge [2,3]. Currently available devices have limited computing power due to their hardware and size, which cannot guarantee the normal operation of such services. This can lead to increased network load and affect user experience. Cloud computing technology has successfully solved these problems by bringing together a large number of computing devices and establishing a highly capable computing center, providing cloud computing and storage services to all devices and users [4,5]. The device can offload the tasks to be calculated by the cloud computing center; the cloud server uses its powerful computing power to support terminal device calculations and then transmits the results back to the terminal device through the network. However, with the increasing number of IoT devices, there are more and more computing tasks requiring central support. Cloud computing centers are far away

from devices [6] and the transmission distance of tasks in the network is long, with high latency, making it difficult to meet user needs. Due to its deployment methods, distance, energy consumption, and other drawbacks, the use of cloud computing in industry is limited. Edge computing overcomes the disadvantage of cloud computing (that it is far away from terminal devices), and edge computing has become a research hotspot [7]. Cloud computing centrally uploads all tasks to the cloud for processing, resulting in increased transmission latency of data in the network. Edge computing, meanwhile, establishes nodes near terminals on the network edge side to provide services. Edge computing alleviates the computing pressure and network transmission pressure of cloud computing [8]. In the cloud edge collaborative computing system, the cloud service center deploys edge nodes, integrates intelligent applications and application service layer hardware, is responsible for high complexity and global data services, and can meet the basic needs of users for diverse scenarios. Intelligent edge nodes provide relatively lightweight services, utilizing outstanding advantages such as edge location and low latency to increase the speed of network service responses [9]. High-cost computing and storage are generally not performed on edge nodes, resulting in relatively low equipment costs. As data generators, end users submit diversified data to upper-level service providers, which helps support service providers in data analysis and decision-making. The full perception of these key links will generate sensitive, heterogeneous, and massive amounts of data in real-time, widely distributed among diverse end users and massive service providers. Accurate multivariate analysis of user data helps to estimate user needs from multiple perspectives, allocate resources reasonably, and improve and enhance the satisfaction index of end customers [10].

At present, the development of cloud computing is mature, but there are still security risks. Recently, edge computing has been introduced. Edge computing is not a method to replace the cloud center, but rather to expand the capabilities of the cloud center [11]. As such, edge computing also has the security problems of cloud computing. Edge distribution is more extensive, the geographical location is more remote, and it has characteristics such as being massive, heterogeneous, and dispersed, making the communication process more complex and therefore more susceptible to malicious attacks. In cloud edge collaborative computing architecture, new cryptographic techniques need to be considered to ensure secure interaction [7]. In the cloud edge collaborative architecture, large amounts data are transmitted through public channels, causing issues such as privacy leakage and illegal access. Authentication processes can achieve negotiation and authentication of common session keys between two or more communication entities; identity and message authentication can be achieved between cloud edges, while generating session keys required for security requirements such as message integrity and message confidentiality [12]. Due to the importance of authentication, the academic community has never stopped analyzing and designing security protocols. Due to different application environments, security requirements, and design methods, the corresponding authentication protocols are also different, and new attack methods are constantly emerging [13]. Most resource constrained terminals are unable to perform many complex and powerful calculations due to resource and capability limitations [14–21]. So, as the network architecture shifts from cloud-to-cloud edge, the corresponding authentication mode will also change accordingly. Therefore, it is necessary to design new authentication protocols to ensure the information exchange of terminals in the cloud edge collaborative architecture. Therefore, this article proposes a trustworthy IoT access scheme for cloud edge collaboration. The contributions are summarized below:

(1) This article proposes a scheme based on two factor security authentication and key negotiation. By applying chaotic mapping technology, this scheme can securely protect communication between devices, cloud servers, and edge devices at lower computational and communication costs. This scheme protects the transmission of confidential messages and improves communication security by negotiating session keys.

(2) To ensure security, this article adopts the BAN logic reduction method and the Scyther verification tool. After security verification, the results show that the protocol exhibits a high level of security. Through security analysis and performance comparison, the experimental results demonstrate that the scheme achieves enhanced security strength while simultaneously improving the efficiency of authentication.

2. Related Work

Privacy protection in the context of the Internet of Things (IoT) encompasses identity authentication, key negotiation, and data privacy. To enhance security mechanisms, Zhang et al. [22] introduced a trust-based secure multi-cloud collaboration framework for the cloud fog assisted Internet of Things. This framework prioritized the interests of cloud users by providing highly credible composite services and ensured the security of cloud service providers through efficient multi-cloud user authentication and collaborative access control. It supported role and trust-based service composition, offering precise trust levels for multi-cloud service composition. Additionally, the framework proposed an efficient and secure multi-cloud collaboration solution to safeguard data objects and other assets maintained by composite services. It underwent a security analysis and demonstrated applicability through case analysis and performance evaluation.

Qi et al. [23] proposed a cloud-centric three-factor authentication and key protocol that integrated passwords, biometrics, and smart cards. This protocol enabled secure access to both the cloud and autonomous vehicles while preserving user biometric privacy. It achieved three-factor authentication and negotiated two session keys: one for safe remote control of autonomous vehicles and another for secure cloud data access, ensuring high security guarantees even in the face of compromised short-term security parameters. The protocol was validated for security characteristics and evaluated for efficiency, demonstrating strong security with reasonable computational and communication costs.

Addressing the computing and storage challenges brought about by the rapid growth of IoT devices on gateways, Wang et al. [24] designed an authentication scheme suitable for such scenarios. This scheme aimed to protect against security threats while maintaining low computational costs for gateways. The article initially defined the adversary model and evaluation criteria for cloud-assisted IoT systems, outlining the security requirements for practical identity verification schemes. It then conducted a case study using advanced identity verification schemes, revealing challenges and subtleties in their design. Finally, the article proposed a secure and efficient authentication scheme for remote control and real-time data access in cloud-assisted IoT systems. Leveraging cloud capabilities, it reduced the computational burden on gateways and was particularly well-suited for IoT applications with a large number of devices. The scheme underwent security analysis and compared well with relevant schemes in terms of security and performance, demonstrating its ability to meet security requirements with minimal computational and storage costs on the gateway node.

3. Preliminaries

3.1. System Model

As shown in Figure 1, the third-party authentication model for cloud edge collaborative communication has three communication entities: IoT devices, edge servers, and cloud service center. When an edge server wants to enter the cloud service center for communication, it must first send a registration request to the cloud service center so that the edge server can obtain resource allocation and service scheduling from the cloud center. IoT devices need to flow and share the collected data and register with the cloud service center before they can proceed with subsequent communication. After registration is completed, authentication interaction is required before data communication. IoT devices first send authentication messages to edge servers, and then the edge servers perform relevant authentication while sending authentication requests to the cloud service center. The cloud service center receives the authentication messages sent by the edge servers

and determines the validity and correctness of the messages. If both IoT devices and edge servers are authenticated, the three communication entities negotiate a session key that is only known to the three parties, which is used to ensure the security and privacy of subsequent data transmission. The cloud service center mainly includes cloud server clusters, cloud computing resource pools, data storage, and security control parts. These elements work together and provide registration functionality and powerful support for IoT devices and edge servers. IoT devices are responsible for collecting raw data from the entire system, encrypting the collected data, and sending the encrypted data to edge servers. The edge server layer plays a crucial intermediate role in data processing and analysis, and is responsible for receiving, processing, and forwarding information from IoT devices while providing data support for cloud service centers.

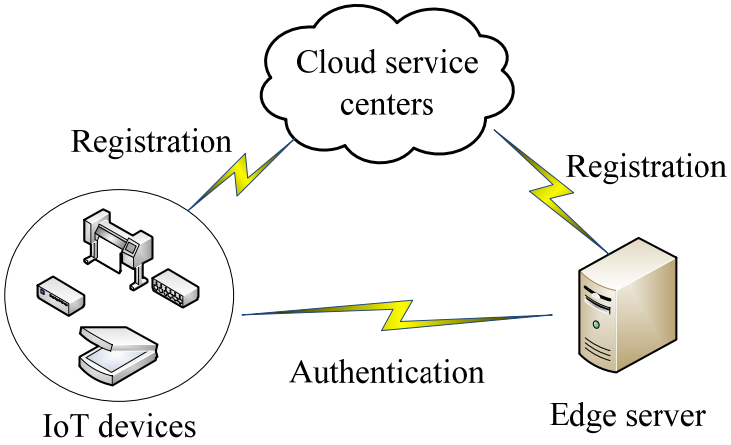


Figure 1. System model.

3.2. Threat Model

This article refers to the threat model of the scheme. The cloud service center is honest and curious, that is, it will honestly execute the protocol content, but it may attempt to invade the privacy data of users for profit or curiosity. The cloud service center may steal user privacy, while malicious external attackers may attempt to obtain privacy data through channel eavesdropping, forgery, and message modification [25–36]. The opponents above are all probability polynomial time. Based on the above threat model, the following security requirements should be ensured [18,29,37].

(1) Confidentiality: Communication between entities in the system should be protected, even if external attackers eavesdrop on the channel to obtain messages, they cannot recover the information in the ciphertext.
(2) Privacy protection: User privacy data should be protected, and other entities in the system cannot access user privacy data,
(3) Integrity: For all messages sent in the system, the receiver can verify whether the sender of the message is a legitimate user and can detect whether the message has been maliciously tampered with and replaced.

3.3. Cloud Edge Collaboration

Cloud computing is an information processing method that has a large amount of computing resources, organizes, and calls various information resources through the network, and can perform large-scale computing [12,13,38]. This technology concentrates dispersed resources and forms a powerful computing resource pool through distributed technology, resource virtualization technology, and other means. Users can apply for resources in a dynamic, on-demand, and measurable manner, and can use multiple types of terminals to obtain the required resource services. As the years have advanced, the interconnection of all things has become the mainstream direction of development, and mobile application services have also developed rapidly. Applications with high traffic and

data intensive characteristics of large connections have become increasingly mainstream. Traditional cloud computing technology, due to its distance from users, finds it difficult to process the massive amount of data generated by user side devices and terminals in real-time, resulting in longer data transmission delays and reduced service quality. Therefore, edge computing technology, which is closer to the data generation side, has gradually attracted the attention of researchers. Edge computing is an open platform with storage, computing, network, and other resources near the edge of the data source. It can provide edge intelligent services nearby, and has technical advantages such as timely service response, more intelligent applications, and strong privacy protection. Edge computing syncs computing and communication capabilities and uses edge devices close to the source of data generation to widely connect terminal devices, providing users with efficient computing services. Cloud edge collaboration means that the application scenario of edge computing requires the edge and cloud center to work together. In the cloud edge collaboration environment, data processing is determined according to the characteristics and needs of services [22]. Tasks with low computational resource requirements, sensitive feedback latency, and strong data privacy are mainly processed by the edge, while tasks with high computational resource consumption and global perception analysis are mainly processed by cloud centers. This is not a matter of separation, but rather of leveraging each other's strengths to achieve the best service results in handling the same service.

3.4. Chebyshev Chaotic Mapping

First, it is necessary to define the n-dimensional Chebyshev polynomial as $T_n(x)$, where n is a natural number and $x \in [-1, 1]$, satisfying:

$$T_n(x) = \cos(n \cdot \arccos(x))$$

Therefore, according to the trigonometric transformation, the recursive relationship of Chebyshev polynomials can be obtained as follows [8]:

$$\begin{gathered} T_0(x) = 1 \\ T_1(x) = x \\ T_2(x) = 2x^2 - 1 \\ \cdots\cdots \\ T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \end{gathered}$$

In addition, Chebyshev polynomials also possess semigroup properties. For $r, s \in N, x \in [-1, 1]$, they satisfy:

$$T_r(T_s(x)) = T_{rs}(x) = T_s(T_r(x))$$

In 2008, to compensate for the security flaws of Chebyshev polynomials, Zhang et al. [39] proposed an enhanced Chebyshev polynomial, which extended the domain of x from $[-1, 1]$ to $[-\infty, +\infty]$, and the expression became:

$$T_n(x) = (2xT_{n-1}(x) - T_{n-2}(x))(mod\ N)$$

where $n \geq 2$ and N is a large prime number, which still satisfies the semigroup property. The identity authentication protocol research in this article adopts an enhanced version of Chebyshev polynomials for protocol design.

Difficult problems refer to problems that cannot be calculated within probabilistic polynomial time algorithms. When designing cryptographic protocols, the security of the protocol is usually limited to a basic mathematical problem, with the insolvability of a specific known problem as the cornerstone of protocol security. Based on the Chebyshev polynomial, there are the following difficult assumptions:

(1) Discrete Logarithmic Problem (DLP): Given the parameters x and y, it is difficult to find a positive integer n that satisfies $\mathrm{T_n(x) = y}$.
(2) Computational Diffie-Hellman Problem (CDHP): Given the parameters x, $\mathrm{T_r(x)}$ and $\mathrm{T_s(x)}$, it is difficult to calculate $\mathrm{T_{rs}(x) = T_r(T_s(x)) = T_s(T_r(x))}$.

4. Proposed Protocol

In this paper, Figure 2 is used to represent the protocol authentication process.

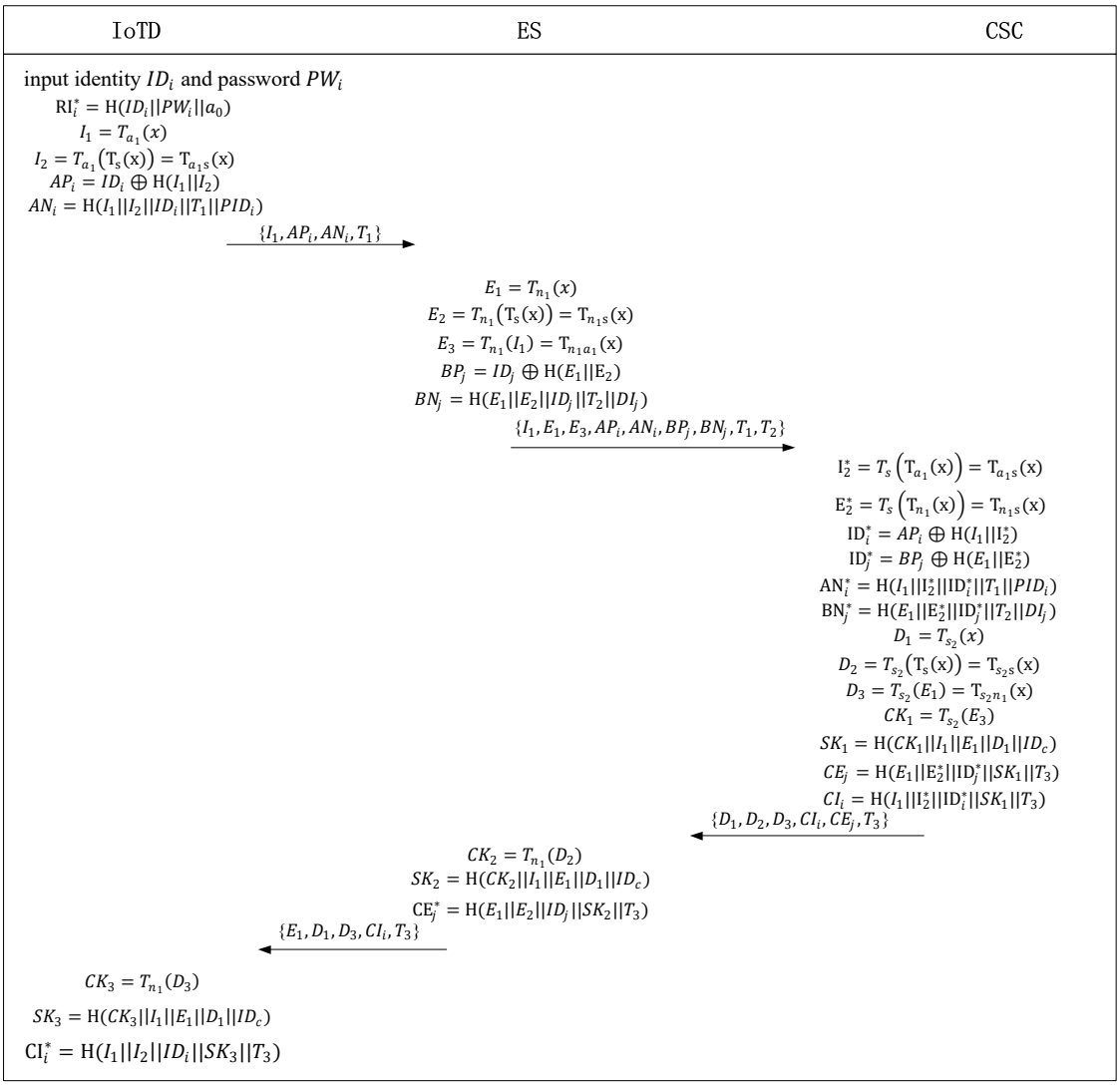


Figure 2. Authentication process.

4.1. Initialization

The Cloud Service Center (CSC) generates a random number x and a large prime number p, then selects a random number s as the private key. Next, CSC selects the one-way secure hash function $\mathrm{H}(\cdot)$ and computes the public key $\mathrm{T_s(x)}$, and publicly disclose the parameters $\{\mathrm{x}, \mathrm{T_s(x)}, \mathrm{p}, \mathrm{H}(\cdot)\}$.

4.2. Register

1. IoT device registration

(1) IoT device $IoTD_i$ first selects a random number a_0, then inputs the identity identifier ID_i, calculates $PID_i = \mathrm{H}(ID_i||a_0)$, and sends the calculation result $\{PID_i\}$ to the cloud server CSC through a secure channel.

(2) After receiving $\{PID_i\}$, CSC in the cloud server center selects the random number b_0 and identity ID_c, calculates $CI_i = \mathrm{H}(PID_i||b_0||\mathrm{s})$, $SI_i = ID_c \oplus PID_i$, and initializes Count to 0. CSC stores $\{PID_i, b_0, \mathrm{Count}\}$ in the registry and issues a smart card SC_i. CSC stores $\{CI_i, SI_i\}$ in SC_i and finally sends SC_i to $IoTD_i$

(3) After receiving the smart card SC_i, $IoTD_i$ inputs the password PW_i, calculates $IA_i = CI_i \oplus PID_i$, $RI_i = \mathrm{H}(ID_i||PW_i||a_0)$ and then updates the information $\{IA_i, RI_i, SI_i, a_0\}$ to the smart card SC_i.

2. Edge Server Registration

(1) The edge server ES_j selects a random number n_0, calculates $EI_j = \mathrm{H}(ID_j||n_0)$, and sends it to CSC.

(2) After receiving $\{EI_j\}$, CSC selects the random number s_1 to calculate $DI_j = \mathrm{H}(EI_j||s_1||\mathrm{s})$ and $SI_j = EI_j \oplus ID_c$, and returns them to ES_j. Finally, CSC stores $\{EI_j, s_1\}$.

(3) After receiving $\{EI_j, s_1\}$, ES_j calculates $BT_j = DI_j \oplus EI_j$ and stores $\{BT_j, SI_j\}$.

4.3. Authentication and Key Negotiation

(1) $IoTD_i$ can be inserted into smart card SC_i, input identity ID_i and password PW_i, calculate $\mathrm{RI}_i^* = \mathrm{H}(ID_i||PW_i||a_0)$, and then determine whether RI_i^* is equal to RI_i. This can determine whether the identity and password of $IoTD_i$ are correct. If $\mathrm{RI}_i^* = RI_i$ is true, $IoTD_i$ login is successful. Therefore, two-factor authentication ensures the user's secure login. $IoTD_i$ chooses randomly a_1 and the current timestamp T_1, uses chaotic mapping to calculate $I_1 = T_{a_1}(x)$, $I_2 = T_{a_1}(\mathrm{T_s(x)}) = \mathrm{T}_{a_1 \mathrm{s}}(\mathrm{x})$, $AP_i = ID_i \oplus \mathrm{H}(I_1||I_2)$, $AN_i = \mathrm{H}(I_1||I_2||ID_i||T_1||PID_i)$, and finally sends $\{I_1, AP_i, AN_i, T_1\}$ to the edge server ES_j.

(2) When ES_j receives $\{I_1, AP_i, AN_i, T_1\}$ from $IoTD_i$, ES_j checks T_1. If not established, ES_j will immediately terminate communication. If it is true, ES_j chooses randomly n_1 and T_2, then uses chaotic mapping to calculate $E_1 = T_{n_1}(x)$, $E_2 = T_{n_1}(\mathrm{T_s(x)}) = \mathrm{T}_{n_1 \mathrm{s}}(\mathrm{x})$, $E_3 = T_{n_1}(I_1) = \mathrm{T}_{n_1 a_1}(\mathrm{x})$, $BP_j = ID_j \oplus \mathrm{H}(E_1||\mathrm{E}_2)$, $BN_j = \mathrm{H}(E_1||E_2||ID_j||T_2||DI_j)$, and finally ES_j sends $\{I_1, E_1, E_3, AP_i, AN_i, BP_j, BN_j, T_1, T_2\}$ to the cloud server CSC.

(3) When CSC receives the message $\{A_1, E_1, E_3, AP_i, AN_i, BP_i, BN_i, T_1, T_2\}$ sent by ES_j, it checks T_2. If not established, CSC will immediately stop communication. If it is true, CSC uses chaotic mapping to calculate $\mathrm{I}_2^* = T_s(\mathrm{T}_{a_1}(\mathrm{x})) = \mathrm{T}_{a_1 \mathrm{s}}(\mathrm{x})$, $\mathrm{E}_2^* = T_s(\mathrm{T}_{n_1}(\mathrm{x})) = \mathrm{T}_{n_1 \mathrm{s}}(\mathrm{x})$, $\mathrm{ID}_i^* = AP_i \oplus \mathrm{H}(I_1||\mathrm{I}_2^*)$, $\mathrm{ID}_j^* = BP_j \oplus \mathrm{H}(E_1||\mathrm{E}_2^*)$, $\mathrm{AN}_i^* = \mathrm{H}(I_1||\mathrm{I}_2^*||\mathrm{ID}_i^*||T_1||PID_i)$, $\mathrm{BN}_j^* = \mathrm{H}(E_1||\mathrm{E}_2^*||\mathrm{ID}_j^*||T_2||DI_j)$, and then determines whether $AN_i = \mathrm{AN}_i^*$ and $BN_j = \mathrm{BN}_j^*$ are true. If the above equation results are judged to be unequal, CSC will immediately terminate communication. If the above equations are equal, CSC selects the random number s_2 and timestamp T_3, then calculates $D_1 = T_{s_2}(x), D_2 = T_{s_2}(\mathrm{T_s(x)}) = \mathrm{T}_{s_2 \mathrm{s}}(\mathrm{x}), D_3 = T_{s_2}(E_1) = \mathrm{T}_{s_2 n_1}(\mathrm{x}), CK_1 = T_{s_2}(E_3)$, session key $SK_1 = \mathrm{H}(CK_1||I_1||E_1||D_1||ID_c)$, $CE_j = \mathrm{H}\left(E_1\middle||\mathrm{E}_2^*\middle||\mathrm{ID}_j^*\middle||SK_1\middle||T_3\right), CI_i = \mathrm{H}\left(I_1\middle||\mathrm{I}_2^*\middle||\mathrm{ID}_i^*\middle||SK_1\middle||T_3\right)$, and finally updates the counter Count = Count + 1 in the CSC storage table while sending $\{D_1, D_2, D_3, CI_i, CE_j, T_3\}$ to the edge server ES_j.

(4) When ES_j receives the message $\{D_1, D_2, D_3, CI_i, CE_j, T_3\}$ returned by CSC, it first checks whether T_3 meets the requirements. Then ES_j uses chaotic mapping to calculate $CK_2 = T_{n_1}(D_2)$, session key $SK_2 = \mathrm{H}(CK_2||I_1||E_1||D_1||ID_c)$, $\mathrm{CE}_j^* = \mathrm{H}(E_1||E_2||ID_j||SK_2||T_3)$, and determines whether CE_j^* is equal to CE_j. If equal, ES_j verifies the identity of CSC and the session key is equal. Then ES_j returns $\{E_1, D_1, D_3, CI_i, T_3\}$ to $IoTD_i$.

(5) When $IoTD_i$ receives messages $\{E_1, D_1, D_3, CI_i, T_3\}$ sent by ES_j, it first checks T_3. If T_3 is within the time threshold range, ES_j uses chaotic mapping to calculate $CK_3 = T_{n_1}(D_3)$, session key $SK_3 = \mathrm{H}(CK_3||I_1||E_1||D_1||ID_c), \mathrm{CI}_i^* = \mathrm{H}(I_1||I_2||ID_i||SK_3||T_3)$ and determines whether CI_i^* is equal to CI_i. If they are not equal, $IoTD_i$ immediately

terminates communication. If they are the same, $IoTD_i$ believes that ES_j and CSC are true and that the session key is equal, and then $IoTD_i$ can communicate with CSC and ES_j through the session key.

4.4. Password Modification

When $IoTD_i$ wants to change its password, it first enters the identity ID_i and password PW_i, then calculates $\mathrm{RI}_i^* = \mathrm{H}(ID_i||PW_i||a_0)$ and checks whether it is equal to RI_i. If it is equal, it proves that $IoTD_i$ is legal. Then, it can choose a new password PW_i^{new} and update the content in smart card SC_i, replacing RI_i with $\mathrm{RI}_i^{new} = \mathrm{H}(ID_i||PW_i^{new}||a_0)$.

5. Security Evaluation

5.1. Security Proof

This article uses BAN logic analysis to prove the correctness of the proposed scheme.

(1) Protocol idealization

$\mathrm{Mes}_1 : \mathrm{IoTD}_i \rightarrow CSC :< I_1, AP_i, AN_i, T_1 >_{I_2}$
$\mathrm{Mes}_2 : ES_j \rightarrow \mathrm{CSC} :< E_1, E_3, BP_j, BN_j, T_2 >_{E_2}$
$\mathrm{Mes}_3 : \mathrm{CSC} \rightarrow ES_j :< D_1, D_2, D_3, CI_i, CE_j, T_3 >_{CK_1}$
$\mathrm{Mes}_4 : ES_j \rightarrow \mathrm{IoTD}_i :< E_1, D_1, D_3, CI_i, T_3 >_{CK_2}$

(2) Protocol goal

$\mathrm{G}_1 : \mathrm{IoTD_i} \mid \equiv \mathrm{IoTD}_i \overset{SK}{\leftrightarrow} \mathrm{ES_j}$
$\mathrm{G}_2 : \mathrm{ES}_j \mid \equiv \mathrm{ES}_j \overset{SK}{\leftrightarrow} \mathrm{IoTD}_i$
$\mathrm{G}_3 : \mathrm{IoTD}_i \mid \equiv \mathrm{ES}_j \mid \equiv \mathrm{ES}_j \overset{SK}{\leftrightarrow} \mathrm{IoTD}_i$
$\mathrm{G}_4 : \mathrm{ES}_j \mid \equiv \mathrm{IoTD}_i \mid \equiv \mathrm{ES}_j \overset{SK}{\leftrightarrow} \mathrm{IoTD}_i$

(3) Initial hypothesis

$\mathrm{A}_1 : CSC \mid \equiv \mathrm{CSC} \overset{I_2}{\leftrightarrow} \mathrm{IoTD_i}$
$\mathrm{A}_2 : CSC \mid \equiv \#(T_1)$
$\mathrm{A}_3 : CSC \mid \equiv \mathrm{IoTD_i} \Rightarrow < I_1, AP_i, AN_i, T_1 >$
$\mathrm{A}_4 : CSC \mid \equiv \mathrm{CSC} \overset{E_2}{\leftrightarrow} \mathrm{ES_j}$
$\mathrm{A}_5 : CSC \mid \equiv \#(T_2)$
$\mathrm{A}_6 : CSC \mid \equiv \mathrm{ES_j} \Rightarrow < E_1, E_3, BP_j, BN_j, T_2 >$
$\mathrm{A}_7 : \mathrm{ES_j} \mid \equiv \mathrm{CSC} \overset{CK_1}{\leftrightarrow} \mathrm{ES_j}$
$\mathrm{A}_8 : \mathrm{ES_j} \mid \equiv \#(T_3)$
$\mathrm{A}_9 : \mathrm{ES_j} \mid \equiv CSC \Rightarrow < D_1, D_2, D_3, CI_i, CE_j, T_3 >$
$\mathrm{A}_{10} : \mathrm{ES}_i \mid \equiv \mathrm{IoTD_i} \Rightarrow \mathrm{IoTD_i} \overset{SK}{\leftrightarrow} \mathrm{ES}_i$
$\mathrm{A}_{11} : \mathrm{IoTD_i} \mid \equiv \mathrm{IoTD_i} \overset{CK_2}{\leftrightarrow} ES_j$
$\mathrm{A}_{12} : \mathrm{IoTD_i} \mid \equiv \#(E_1)$
$\mathrm{A}_{13} : \mathrm{IoTD_i} \mid \equiv \mathrm{ES}_j \Rightarrow < E_1, D_1, D_3, CI_i, T_3 >$
$\mathrm{A}_{14} : \mathrm{IoTD_i} \mid \equiv \mathrm{ES_j} \Rightarrow \mathrm{IoTD_i} \overset{SK}{\leftrightarrow} \mathrm{ES}_i$

(4) Proof of protocol

According to Mes_1, it gets:
$\mathrm{R}_1 : CSC \lhd < I_1, AP_i, AN_i, T_1 >_{I_2}$
From R_1, A_1 and message meaning rule, we get:
$\mathrm{R}_2 : CSC| \equiv \mathrm{IoTD}_i| \sim < I_1, AP_i, AN_i, T_1 >$
According to R_2, A_2 and the random number validation rule, we obtain:
$\mathrm{R}_3 : CSC| \equiv \mathrm{IoTD}_i| \equiv < I_1, AP_i, AN_i, T_1 >$
According to R_3, A_3 and the arbitration rules, we get:
$\mathrm{R}_4 : CSC \mid \equiv < I_1, AP_i, AN_i, T_1 >$
According to Mes_2, it gets:
$\mathrm{R}_5 : CSC \lhd < E_1, E_3, BP_j, BN_j, T_2 >_{E_2}$
From R_5, A_4 and message meaning rule, we get:

$R_6 : CSC|\equiv ES_j| \sim< E_1, E_3, BP_j, BN_j, T_2 >$
According to R_6, A_5 and the random number validation rule, we obtain:
$R_7 : CSC|\equiv ES_j| \equiv< E_1, E_3, BP_j, BN_j, T_2 >$
According to R_7, A_6 and the arbitration rules, we get:
$R_8 : CSC \mid \equiv< E_1, E_3, BP_j, BN_j, T_2 >$
According to Mes_3, it gets:
$R_9 : ES_j \lhd< D_1, D_2, D_3, CI_i, CE_j, T_3 >_{CK_1}$
From R_9, A_7 and message meaning rule, we get:
$R_{10} : ES_j|\equiv CSC| \sim< D_1, D_2, D_3, CI_i, CE_j, T_3 >$
According to R_{10}, A_8 and the random number validation rule, we obtain:
$R_{11} : ES_j|\equiv CSC| \equiv< D_1, D_2, D_3, CI_i, CE_j, T_3 >$
According to R_{11}, A_9 and the arbitration rules, we get:
$R_{12} : ES_j \mid \equiv< D_1, D_2, D_3, CI_i, CE_j, T_3 >$
According to R_{12} and $SK_2 = H(CK_2||I_1||E_1||D_1||ID_c)$, we get
$R_{13} : ES_j \mid \equiv IoTD_i| \equiv ES_j \overset{SK}{\leftrightarrow} IoTD_i$
According to R_{13}, A_{10}, and the arbitration rules, we get:
$R_{14} : ES_j \mid \equiv IoTD_i \overset{SK}{\leftrightarrow} ES_j$
According to Mes_4, it gets:
$R_{15} : IoTD_i \lhd< E_1, D_1, D_3, CI_i, T_3 >_{CK_2}$
From R_{15}, A_{11} and message meaning rule, we can get:
$R_{16} : IoTD_i|\equiv ES_j| \sim< E_1, D_1, D_3, CI_i, T_3 >$
According to R_{16}, A_{12} and the random number validation rule, we obtain:
$R_{17} : IoTD_i|\equiv ES_j| \equiv< E_1, D_1, D_3, CI_i, T_3 >$
According to R_{17}, A_{13} and the arbitration rules, we get:
$R_{18} : IoTD_i \mid \equiv< E_1, D_1, D_3, CI_i, T_3 >$
According to R_{10} and A_{14}, $SK_3 = H(CK_3||I_1||E_1||D_1||ID_c)$, we get:
$R_{19} : IoTD_i \mid \equiv ES_j \mid \equiv IoTD_i \overset{SK}{\leftrightarrow} ES_j$
According to R_{19}, A_{15} and the arbitration rules, we get:
$R_{20} : IoTD_i \mid \equiv IoTD_i \overset{SK}{\leftrightarrow} ES_j$

In summary, all goals have been achieved. Therefore, the correctness of this scheme has been verified.

5.2. Scyther Verification

Protocol security directly determines the security of communication networks. Formal tools have made significant contributions in determining the security of verification protocols, accurately and quickly proving security and identifying security vulnerabilities in protocols. The Scyther formal tool [8] has outstanding performance in protocol security analysis due to its concise formal language and more practical and sound functions. This formal tool is based on the Athena algorithm and utilizes various techniques such as mathematical theorems and logical analysis to analyze multi-agent protocols. This tool uses Security Protocol Description Language (SPDL) to describe the protocol and models the protocol's security properties through a set of declarations (Alive, Weakage, Niagree, Nisync, and Commit), which are mainly used to detect replay attacks, man in the middle attacks, key forward and backward attacks, and other types of attacks.

Therefore, this article will use Scyther formal tools to verify the security of the protocol proposed in this article. The verification results are shown in Figure 3. The verification results of security attributes such as Alive, Weakagreee, Niagreee, and Nisynch are all "OK", with no attack path, indicating that attackers cannot launch attacks on the protocol, indicating that the authentication protocol can effectively resist multiple attacks such as man in the middle attack and key forward attack. So, the proposed solution in this article has passed formal tool security verification of the protocol, and the negotiated session key has security.

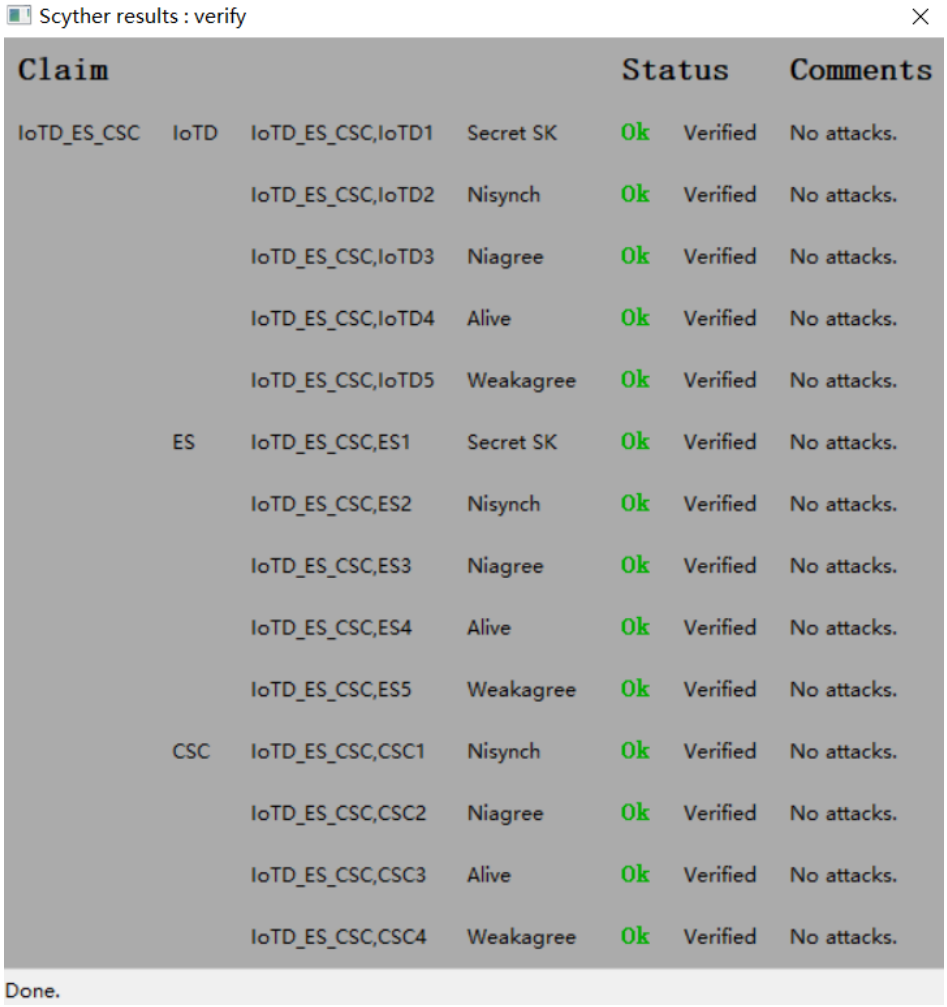


Figure 3. Scyther result.

5.3. Security Analysis

(1) Mutual authentication

The IoT device $IoTD_i$ sends the message request $\{I_1, AP_i, AN_i, T_1\}$ to the edge server ES_j for authentication. ES_j sends the new authentication request $\{I_1, E_1, E_3, AP_i, AN_i, BP_j, BN_j, T_1, T_2\}$ to the cloud server CSC. CSC first checks the timeliness of the message, and then verifies the identity of $IoTD_i$ and ES_j by comparing whether $AN_i = \mathrm{AN}_i^*$ and $BN_j = \mathrm{BN}_j^*$ are valid. ES_j receives messages $\{D_1, D_2, D_3, CI_i, CE_j, T_3\}$ from CSC to determine if CE_j^* is equal to CE_j. If equal, ES_j verifies the identity of CSC. $IoTD_i$ receives messages $\{E_1, D_1, D_3, CI_i, T_3\}$ from ES_j to determine if CI_i^* is equal to CI_i. $IoTD_i$ verified the identities of ES_j and CSC. Therefore, our protocol achieves mutual identity authentication between entities.

(2) Anonymity

During the registration phase of the protocol, ID_i is processed by $PID_i = \mathrm{H}(ID_i||a_0)$ and then sent to CSC. Moreover, the processed ID_i is registered in CSC, and the ID_i of IoT devices is not stored in CSP's database. Therefore, there is no problem of obtaining user identity by cracking CSC's database content during the registration phase, which can ensure the anonymity of users during the registration phase. Then, in the authentication stage, ID_i is encrypted by $AP_i = ID_i \oplus \mathrm{H}(I_1||I_2)$ and transmitted through the public channel after computation. However, even if the information transmitted through the public channel is intercepted, ID_i cannot be calculated. In order to obtain ID_i, it is necessary to calculate I_2, and obtaining I_2 requires solving the CDHP problem, so I_2 cannot be

calculated, that is, the value of ID_i cannot be obtained. In summary, the protocol can ensure user anonymity.

(3) Unlinkability

In the protocol, $IoTD_i$, ED_j, and CSC first set timestamps during the authentication negotiation process, and the time of each stage is changing dynamically, so the time of the message is the latest. Then, the identity of $IoTD_i$ is anonymous, and the encryption parameters are different each time, so the calculated authentication value is always changing. In the calculated authentication value, new random numbers are added during each authentication process to ensure that each authentication is a changing value. So, in summary, the protocol satisfies untraceability.

(4) Replay attack

In the protocol, on the one hand, adding timestamps to the first stage of each entity authentication can ensure the freshness of the messages sent by the three parties and ensure the effectiveness of the transmitted messages to resist replay attacks. On the other hand, authentication values are calculated using random values. As each session progresses, the authentication value varies. This increases the difficulty for attackers to replay each time.

(5) Man in the middle attack

Assuming the attacker could intercept the messages $\{I_1, AP_i, AN_i, T_1\}$ transmitted by $IoTD_i$, it cannot obtain the secret value I_2 calculated by the chaotic mapping algorithm in $IoTD_i$. Therefore, even if the attacker tampers with the transmitted messages $\{I_1, AP_i, AN_i, T_1\}$ and sends them to CSC, authentication cannot be obtained. Similarly, attackers can intercept and modify messages $\{I_1, E_1, E_3, AP_i, AN_i, BP_j, BN_j, T_1, T_2\}$, but they still cannot obtain the secret value E_2 calculated by the chaotic mapping algorithm in ES_j. This means that CSC cannot complete the verification.

(6) Impersonation attack

Assuming that the private key s of CSC is leaked, if the attacker can use s to calculate I_2 and E_2, they can impersonate IoT devices and edge servers. However, even if they know I_2 and E_2, they cannot calculate the correct $AN_i = \mathrm{H}(I_1||I_2||ID_i||T_1||PID_i)$ and $BN_j = \mathrm{H}(E_1||E_2||ID_j||T_2||DI_j)$ because they do not know PID_i and DI_j, so they cannot obtain the correct authentication value. So, it is impossible to disguise $IoTD_i$ and ES_j.

(7) Password guessing attacks

Attackers can intercept messages $\{I_1, AP_i, AN_i, T_1\}$, $\{I_1, E_1, E_3, AP_i, AN_i, BP_j, BN_j, T_1, T_2\}$, $\{D_1, D_2, D_3, CI_i, CE_j, T_3\}$ and $\{E_1, D_1, D_3, CI_i, T_3\}$ transmitted by three role login authentication in the public channel, but cannot make password guesses based on these intercepted messages. Because these messages transmitted through public channels do not mention PW_i, smart cards can steal and obtain stored information. However, the identity ID_i and password PW_i of $IoTD_i$ are unknown, and in the first stage of device login, the correct ID_i and PW_i must be entered simultaneously for authentication. So, the protocol can resist password guessing attacks.

(8) Verification Table Loss Attack

In the proposed protocol, the parameter table maintained by $IoTD_i$ is $\{IA_i, RI_i, SI_i, a_0\}$, and the parameter table maintained by ES_j is $\{BT_j, SI_j\}$. The parameters in these two tables are not critical secret parameters, so even if an attacker obtains these parameters, they cannot use them to enhance the advantage of the attack and threaten the security of the system.

(9) Privileged internal attack

In the proposed protocol, the user only sends the PID_i to the server, without sending any information related to the password. Then, among all the parameters stored in the SC_i, it is found that PW_i is not presented in plain text, but uses a one-way function, so PW_i is

for security protection. Therefore, the proposed protocol has the ability to resist privileged internal attack.

(10) IoT device compromise

When an IoT device is compromised, the information stored on the device may be obtained by attackers. However, the attacker cannot obtain the user password, and it is difficult for the attacker to log in successfully and authenticate.

(11) Forward security

Assuming that the main key s of CSC is leaked and that the attacker has intercepted all communication content on the channel, they will want to crack the previous session key. At this point, the attacker needs to calculate $SK_3 = \mathrm{H}(CK_3||I_1||E_1||D_1||ID_c)$; however, the secret value of CK_3 is calculated using a chaotic mapping algorithm. Therefore, when an attacker calculates CK_3, they will face the CDHP difficulty problem based on Chebyshev polynomials, which is almost impossible to calculate successfully.

5.4. Security Comparison

Here, the main comparison was made between the relevant solutions and the proposed solutions in this article, and the security features and related functions were compared. The summary of the comparison results is shown in Table 1. In the table, we use ✓ to mark support for this feature or function and use × to indicate that this feature or feature is not supported.

Table 1. Functionality comparison.

Feature	[23]	[40]	[18]	[29]	[37]	[24]	[41]	Our
Mutual authentication	✓	✓	✓	✓	✓	✓	✓	✓
Anonymity	✓	✓	✓	✓	✓	✓	✓	✓
Unlinkability	×	✓	✓	✓	✓	✓	✓	✓
Replay attack	✓	✓	×	✓	✓	✓	✓	✓
Man in the middle attack	✓	✓	✓	✓	✓	✓	✓	✓
Counterfeit attack	✓	×	✓	✓	✓	✓	✓	✓
Session key negotiation	✓	✓	✓	✓	✓	✓	✓	✓
Resist offline dictionary attack	×	×	×	×	×	×	×	✓
Message integrity	✓	✓	✓	✓	✓	✓	✓	✓
Verification Table Loss Attack	✓	✓	✓	✓	✓	✓	✓	✓
IoT device compromise	✓	✓	✓	✓	✓	✓	✓	✓
Privileged internal attack	✓	✓	✓	✓	✓	✓	✓	✓
Forward security	✓	✓	✓	✓	✓	✓	✓	✓

6. Performance Analysis

6.1. Computation Overhead

For the cryptographic operations involved in the proposed scheme and other related schemes, this paper conducted network simulation experiments on a hardware platform (shown in Table 2). We used the PBC library and C oracle to develop cryptographic algorithms. The development tool we used is Visual Studio 2019. Through this experiment, the execution time of various cryptographic operations can be obtained. T_H, T_{ECC}, T_{CM}, $T_{D/E}$, and T_F are used to represent the time required to perform one-way hash functions, elliptic curve point multiplication, chaotic mapping algorithms, symmetric encryption, and fuzzy extraction, respectively. As shown in Table 3, these times are calculated. In addition, due to the very low overhead of XOR operation compared to other cryptographic operations, most existing schemes do not consider the time of XOR operation, so it can be ignored.

Table 2. Experimental Platform.

Entity	$IoTD_i$	ES_j	CSC
Platform	Smart phone	Laptop	Desktop
Processor	Hisilicon Kirin 980	Intel Core i7 processor	Intel Core i7 processor
Memory	4 GB	16 GB	32 GB
Operating system	Android 11	Windows 10	Windows 10

Table 3. Execution time.

Operation	$IoTD_i$	ES_j	CSC
T_{CM}	0.93	0.62	0.54
$T_{D/E}$	0.85	0.51	0.46
T_{ECC}	2.16	1.75	1.32
T_F	2.16	1.75	1.32
T_H	0.0098	0.0062	0.0031

The comparison results of the computational overhead between the proposed scheme and other related authentication schemes are shown in Table 4. In the authentication process between multiple participating entities, this scheme used a total of 13 chaotic mapping operations and 13 hash function operations, so the computational overhead of this scheme is $13T_{CM} + 16T_H$. Compared to the literature, the scheme proposed by Jiang et al. [23] used 10 ECC operations, 4 symmetric encryption operations, and 25 hash function operations, resulting in a total computational overhead of $10T_{ECC} + 25T_H + 4T_{D/E}$. In other relevant comparative literature, the scheme proposed by Zhao et al. [40] used 22 hash function operations, 8 ECC operations, and 1 fuzzy extraction operation. Therefore, the total time overhead is approximately $8T_{ECC} + T_F + 22T_H$. The scheme proposed by Wang et al. [24] involves 21 hash function operations and 6 ECC operations. Therefore, the total time overhead is approximately $6T_{ECC} + 21T_H$. The scheme proposed by Xu et al. [41] involves 17 hash function operations and 9 ECC operations. Therefore, the total time overhead is approximately $9T_{ECC} + 17T_H$. From Figure 4, it can be seen that the proposed scheme in this article is lower in the total computation overhead. Based on the security analysis proposed above, this scheme achieves lower computational overhead while ensuring higher level security and more functional features.

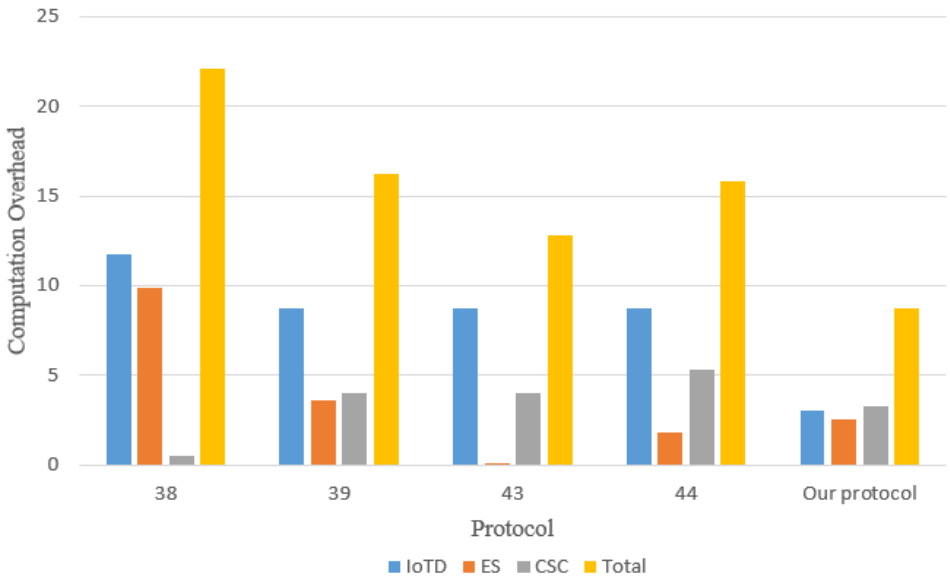


Figure 4. Computation overhead.

Table 4. Computation overhead.

	[23]	[40]	[24]	[41]	Our Protocol
$IoTD_i$	$5T_{ECC} + 9T_H + T_{D/E}$	$3T_{ECC} + T_F + 9T_H$	$3T_{ECC} + T_F + 8T_H$	$4T_{ECC} + 7T_H$	$3T_{CM} + 5T_H$
ES_i	$5T_{ECC} + 12T_H + 2T_{D/E}$	$2T_{ECC} + 8T_H$	$9T_H$	$T_{ECC} + 4T_H$	$4T_{CM} + 4T_H$
CSC	$4T_H + T_{D/E}$	$3T_{ECC} + 5T_H$	$3T_{ECC} + 14T_H$	$4T_{ECC} + 6T_H$	$6T_{CM} + 7T_H$
Total	$10T_{ECC} + 25T_H + 4T_{D/E}$	$8T_{ECC} + T_F + 22T_H$	$6T_{ECC} + 21T_H$	$9T_{ECC} + 17T_H$	$13T_{CM} + 16T_H$

6.2. Communication Overhead

To compare the communication costs of different schemes, this article unifies the output length of each cryptographic operation. We set the length of random numbers in the scheme to 128 bits, the output length of elliptic curve point multiplication operation to 256 bits, the output length of hash algorithm to 128 bits, the output length of Chebyshev chaotic map to 160 bits, the output length of identity information to 128 bits, and the output length of timestamp to 32 bits. Based on this, the cost of different schemes can be calculated, as shown in Table 5. The proposed solution includes four interactions. We can calculate that the size of the first message $\{I_1, AP_i, AN_i, T_1\}$ is 448 bits. Similarly, we can obtain that the size of the second message $\{I_1, E_1, E_3, AP_i, AN_i, BP_j, BN_j, T_1, T_2\}$ is 1056 bits, the size of $\{D_1, D_2, D_3, CI_i, CE_j, T_3\}$ is 768 bits, and the size of $\{E_1, D_1, D_3, CI_i, T_3\}$ is 640 bits. Therefore, the total cost of our proposed solution is 2912 bits. From the comparison results in Table 5 and Figure 5, it can be seen that our scheme has lower communication overhead compared to the scheme in [24]. Although the communication overhead of the proposed scheme in this article is higher than that of [23,40,41], after systematically comparing the security, functional characteristics, authentication, and key negotiation efficiency, it can be concluded that the proposed scheme achieves higher security and functional characteristics and minimizes computational and communication overhead as much as possible.

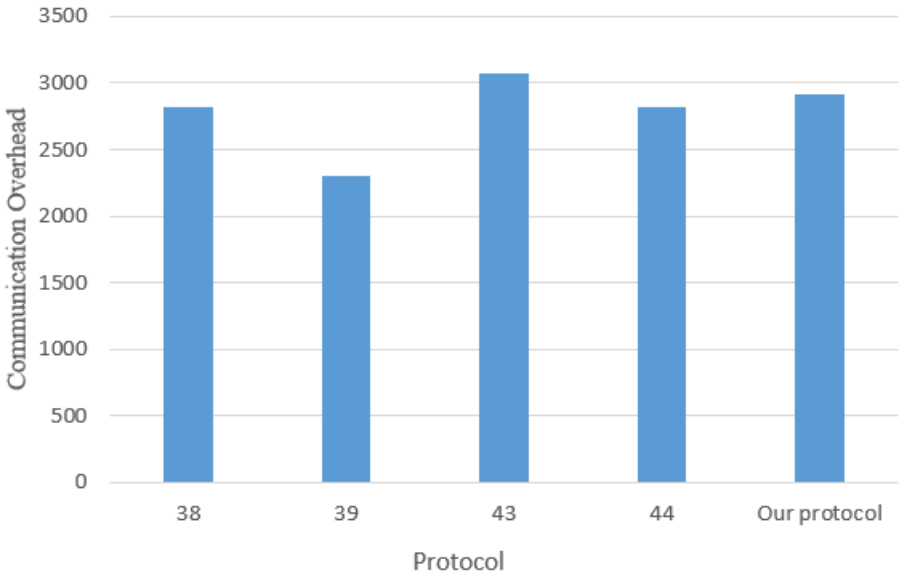


Figure 5. Communication overhead.

Table 5. Communication overhead.

Protocol	Communication Overhead (bits)
[23]	2816
[40]	2304
[24]	3072
[41]	2816
Our protocol	2912

7. Conclusions

With the rapid development of information technology and cloud computing, data plays an important role in various fields, especially in the fields of the Internet, healthcare, finance, etc. To fully leverage the value of data, the flow and sharing of data are essential. Cloud edge collaboration is the current trend. Terminal devices need to establish network connectivity with the edge or cloud center, which requires powerful computing, storage, and analysis capabilities from the edge and cloud center. Authentication is one of the convenient methods to solve the security of information transmission between communication entities. Therefore, based on the cloud edge collaboration, this paper proposes a protocol, which ensures protocol security and forward confidentiality based on chaotic mapping algorithm, and achieves the goal of lightweight authentication for terminals. Finally, the BAN logical reduction method and Scyther are used for security verification. Experimental comparative analysis shows that the proposed protocol had better overhead on the terminal than other schemes. However, the cloud edge collaboration scheme in this article mainly considers the authentication of one IoT device and has not yet considered the scenario of multiple IoT devices being authenticated simultaneously. Therefore, when multiple IoT devices are authenticated simultaneously, it may cause network congestion and reduce efficiency. In addition, this article mainly considers identity verification within the same region and has not yet considered cross cloud server scenarios that exist between different regions. In the future, advanced cryptographic techniques, machine learning algorithms, and blockchain technology will also be introduced to achieve secure and efficient cross domain authentication, while considering authentication of multiple IoT devices.

Author Contributions: Conceptualization, W.Z. and C.Z.; methodology, C.Z. and L.J.; software, W.Z. and L.J.; validation and formal analysis, C.Z. and L.J.; writing—original draft preparation, W.Z. and L.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Scientific Research Funds of Huaqiao University (11BS211).

Data Availability Statement: Data is contained within the article.

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

References

1. Saqib, M.; Moon, A.H. A systematic security assessment and review of internet of things in the context of authentication. *Comput. Secur.* **2023**, *125*, 103053. [CrossRef]
2. Miao, J.; Huang, Y.; Wang, Z.; Wu, Z.; Lv, J. Image recognition of traditional Chinese medicine based on deep learning. *Front. Bioeng. Biotechnol.* **2023**, *11*, 1199803. [CrossRef]
3. Sheik, S.A.; Muniyandi, A.P. Secure authentication schemes in cloud computing with glimpse of artificial neural networks: A review. *Cyber Secur. Appl.* **2023**, *1*, 100002. [CrossRef]
4. Li, X.; Chen, T.; Cheng, Q.; Ma, S.; Ma, J. Smart applications in edge computing: Overview on authentication and data security. *IEEE Internet Things J.* **2020**, *8*, 4063–4080. [CrossRef]
5. Shahidinejad, A.; Ghobaei-Arani, M.; Souri, A.; Shojafar, M.; Kumari, S. Light-edge: A lightweight authentication protocol for IoT devices in an edge-cloud environment. *IEEE Consum. Electron. Mag.* **2021**, *11*, 57–63. [CrossRef]
6. Mei, Q.; Xiong, H.; Chen, Y.C.; Chen, C.M. Blockchain-enabled privacy-preserving authentication mechanism for transportation cps with cloud-edge computing. *IEEE Trans. Eng. Manag.* **2022**. [CrossRef]
7. Souri, A.; Zhao, Y.; Gao, M.; Mohammadian, A.; Shen, J.; Al-Masri, E. A trust-aware and authentication-based collaborative method for resource management of cloud-edge computing in social internet of things. *IEEE Trans. Comput. Soc. Syst.* **2023**. [CrossRef]
8. Miao, J.; Wang, Z.; Miao, X.; Xing, L. A secure and efficient lightweight vehicle group authentication protocol in 5G networks. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 4079092. [CrossRef]
9. Babu, E.S.; Barthwal, A.; Kaluri, R. Sec-edge: Trusted blockchain system for enabling the identification and authentication of edge based 5G networks. *Comput. Commun.* **2023**, *199*, 10–29. [CrossRef]
10. Xu, Y.; Zhou, Y.; Yang, B.; Qiao, Z.; Wang, Z.; Xia, Z.; Zhang, M. An Efficient Identity Authentication Scheme with Provable Security and Anonymity for Mobile Edge Computing. *IEEE Syst. J.* **2022**, *17*, 1012–1023. [CrossRef]
11. Miao, J.; Wang, Z.; Xue, X.; Wang, M.; Lv, J.; Li, M. Lightweight and Secure D2D Group Communication for Wireless IoT. *Front. Phys.* **2023**, *11*, 433. [CrossRef]

12. Jiang, X.; Dou, R.; He, Q.; Zhang, X.; Dou, W. EdgeAuth: An intelligent token-based collaborative authentication scheme. *Softw. Pract. Exp.* **2023**. [CrossRef]
13. Song, W.; Liu, M.; Baker, T.; Zhang, Q.; Tan, Y.A. A group key exchange and secure data sharing based on privacy protection for federated learning in edge-cloud collaborative computing environment. *Int. J. Netw. Manag.* **2023**, *33*, e2225. [CrossRef]
14. Miao, J.; Wang, Z.; Wu, Z.; Ning, X.; Tiwari, P. A blockchain-enabled privacy-preserving authentication management protocol for Internet of Medical Things. *Expert Syst. Appl.* **2024**, *237*, 121329. [CrossRef]
15. Amin, R.; Kumar, N.; Biswas, G.P.; Iqbal, R.; Chang, V. A light weight authentication protocol for IoT-enabled devices in distributed Cloud Computing environment. *Future Gener. Comput. Syst.* **2018**, *78*, 1005–1019. [CrossRef]
16. Wazid, M.; Bagga, P.; Das, A.K.; Shetty, S.; Rodrigues, J.J.; Park, Y. AKM-IoV: Authenticated key management protocol in fog computing-based Internet of vehicles deployment. *IEEE Internet Things J.* **2019**, *6*, 8804–8817. [CrossRef]
17. Wu, T.Y.; Guo, X.; Yang, L.; Meng, Q.; Chen, C.M. A Lightweight Authenticated Key Agreement Protocol Using Fog Nodes in Social Internet of Vehicles. *Mob. Inf. Syst.* **2021**, *2021*, 3277113. [CrossRef]
18. Ma, M.; He, D.; Wang, H.; Kumar, N.; Choo, K.K.R. An efficient and provably secure authenticated key agreement protocol for fog-based vehicular ad-hoc networks. *IEEE Internet Things J.* **2019**, *6*, 8065–8075. [CrossRef]
19. Eftekhari, S.A.; Nikooghadam, M.; Rafighi, M. Security enhanced three-party pairwise secret key agreement protocol for fog-based vehicular ad-hoc communications. *Veh. Commun.* **2021**, *28*, 100306. [CrossRef]
20. Jia, X.; He, D.; Kumar, N.; Choo, K.K.R. Authenticated key agreement scheme for fog-driven IoT healthcare system. *Wirel. Netw.* **2019**, *25*, 4737–4750. [CrossRef]
21. Thantharate, P.; Anurag, T. CYBRIA-Pioneering Federated Learning for Privacy-Aware Cybersecurity with Brilliance. In Proceedings of the 2023 IEEE 20th International Conference on Smart Communities: Improving Quality of Life using AI, Robotics and IoT (HONET), Boca Raton, FL, USA, 4–6 December 2023; pp. 56–61.
22. Zhang, J.; Li, T.; Ying, Z.; Ma, J. Trust-Based Secure Multi-Cloud Collaboration Framework in Cloud-Fog-Assisted IoT. *IEEE Trans. Cloud Comput.* **2023**, *11*, 1546–1561. [CrossRef]
23. Jiang, Q.; Zhang, N.; Ni, J.; Ma, J.; Ma, X.; Choo, K.K.R. Unified biometric privacy preserving three-factor authentication and key agreement for cloud-assisted autonomous vehicles. *IEEE Trans. Veh. Technol.* **2020**, *69*, 9390–9401. [CrossRef]
24. Wang, C.; Wang, D.; Duan, Y.; Tao, X. Secure and Lightweight User Authentication Scheme for Cloud-Assisted Internet of Things. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 2961–2976. [CrossRef]
25. Kaur, K.; Garg, S.; Kaddoum, G.; Guizani, M.; Jayakody, D.N.K. A lightweight and privacy-preserving authentication protocol for mobile edge computing. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBE-COM), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6. [CrossRef]
26. Hou, Y.; Garg, S.; Hui, L.; Jayakody, D.N.K.; Jin, R.; Hossain, M.S. A data security enhanced access control mechanism in mobile edge computing. *IEEE Access* **2020**, *8*, 136119–136130. [CrossRef]
27. Lee, J.; Kim, D.; Park, J.; Park, H. A multi-server authentication protocol achieving privacy protection and traceability for 5g mobile edge computing. In Proceedings of the 2021 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 10–12 January 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–4.
28. Sun, H.; Tan, Y.A.; Li, C.; Lei, L.; Zhang, Q.; Hu, J. An edge-cloud collaborative cross-domain identity-based authentication protocol with privacy protection. *Chin. J. Electron.* **2022**, *31*, 721–731. [CrossRef]
29. Challa, S.; Wazid, M.; Das, A.K.; Kumar, N.; Reddy, A.G.; Yoon, E.J.; Yoo, K.Y. Secure signature-based authenticated key establishment scheme for future IoT applications. *IEEE Access* **2017**, *5*, 3028–3043. [CrossRef]
30. Jia, X.; He, D.; Li, L.; Choo, K.K.R. Signature-based three-factor authenticated key exchange for internet of things applications. *Multimed. Tools Appl.* **2018**, *77*, 18355–18382. [CrossRef]
31. Wazid, M.; Das, A.K.; Odelu, V.; Kumar, N.; Conti, M.; Jo, M. Design of secure user authenticated key management protocol for generic IoT networks. *IEEE Internet Things J.* **2017**, *5*, 269–282. [CrossRef]
32. Banerjee, S.; Odelu, V.; Das, A.K.; Srinivas, J.; Kumar, N.; Chattopadhyay, S.; Choo, K.K.R. A provably secure and lightweight anonymous user authenticated session key exchange scheme for Internet of Things deployment. *IEEE Internet Things J.* **2019**, *6*, 8739–8752. [CrossRef]
33. Kumar, D.; Jain, S.; Khan, A.; Pathak, P.S. An improved lightweight anonymous user authenticated session key exchange scheme for Internet of Things. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *14*, 5067–5083. [CrossRef]
34. Sadhukhan, D.; Ray, S.; Biswas, G.P.; Khan, M.K.; Dasgupta, M. A lightweight remote user authentication scheme for IoT communication using elliptic curve cryptography. *J. Supercomput.* **2021**, *77*, 1114–1151. [CrossRef]
35. Liu, Z.; Guo, C.; Wang, B. A physically secure, lightweight three-factor and anonymous user authentication protocol for IoT. *IEEE Access* **2020**, *8*, 195914–195928. [CrossRef]
36. Miao, J.; Wang, Z.; Ning, X.; Xiao, N.; Cai, W.; Liu, R. Practical and secure multifactor authentication protocol for autonomous vehicles in 5G. *Softw. Pract. Exp.* **2022**. [CrossRef]
37. He, D.; Wang, D. Robust biometrics-based authentication scheme for multiserver environment. *IEEE Syst. J.* **2014**, *9*, 816–823. [CrossRef]
38. Fang, H.; Xiao, Z.; Wang, X.; Xu, L.; Hanzo, L. Collaborative Authentication for 6G Networks: An Edge Intelligence Based Autonomous Approach. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 2091–2103. [CrossRef]

39. Zhang, L. Cryptanalysis of the public key encryption based on multiple chaotic systems. *Chaos Solitons Fractals* **2008**, *37*, 669–674. [CrossRef]
40. Zhao, X.; Li, D.; Li, H. Practical Three-Factor Authentication Protocol Based on Elliptic Curve Cryptography for Industrial Internet of Things. *Sensors* **2022**, *22*, 7510. [CrossRef]
41. Xu, H.; Hsu, C.; Harn, L.; Cui, J.; Zhao, Z.; Zhang, Z. Three-factor anonymous authentication and key agreement based on fuzzy biological extraction for Industrial Internet of Things. *IEEE Trans. Serv. Comput.* **2023**, *16*, 3000–3013. [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.