An Efficient Three-Party Authentication and Key Agreement Protocol for Privacy-Preserving of IoT Devices in Mobile Edge Computing

Sai Ji[1,2], Yang Yuan[1], Jian Shen[1*], Chin-Feng Lai[3], Bofan Chen[4]

[1] Nanjing University of Information Science & Technology, China
[2] Suqian University, China
[3] Department of Engineering Science, National Cheng Kung University, Taiwan
[4] Beijing Institute of Technology, China
jisai@nuist.edu.cn, yy_nuist@126.com, s_shenjian@126.com, cinfon@ieee.org, 1120180495@bit.edu.cn

Abstract

The advancement of 5G communication technology and Internet of Things (IoT) technology has promoted the rapid development of Mobile Edge Computing (MEC). In mobile edge, all IoT devices adopt wireless communication technology. Therefore, it is particularly important to ensure the data security and the privacy of the sender in the process of data transmission. At present, a lot of researchers have proposed a large number of schemes for the authentication of the user in MEC. However, there is no effective and lightweight solution for authentication among users, edge devices and cloud server. In this paper, an efficient three-party authentication and key agreement protocol without using bilinear pairings is designed. The proposed protocol realized authentication among users, edge devices and cloud server, and at the same time, three parties conduct key agreement to obtain a common session key. The security analysis shows that our protocol is secure and meets the security attributes such as session-key security, forward secrecy. The experiment shows that the computation cost is low in this protocol.

Keywords: Authentication, Key agreement, Three-party, IoT devices, MEC

1 Introduction

In recent years, with the rapid development of the Vehicular Ad-Hoc Network (VANETs) [1-2] and the Internet of Things (IoT) [3-4], a variety of smart devices have emerged, such as smart home, intelligent transportation, which undoubtedly make people's life more convenient. However, most of the emerging applications on the market are complex application that generate large amounts of data. This will inevitably bring some problem such as limited resources and equipment. There are approximately 50 billion IoT devices in the world by 2020. The amount of data will grow exponentially every day. With the ever-increasing user requirements for network performance, such as network service quality and service request delay, it is difficult to use mobile device terminals with limited resources to meet them.

In order to deal with the above challenges, Mobile Edge Computing (MEC) [5-6] is proposed as a new paradigm. It can distribute computing ability and services in the cloud server to the edge of the network with geographical advantages, and provide real-time data analysis and intelligent processing nearby. At the same time, MEC can avoid the core network congestion effectively and reduce the service response delay. Therefore, it becomes the focus of academic and industrial research gradually. At present, researchers have carried out a large number of researches in the field of edge computing. The universally accepted edge computing schemes include micro-cloud computing, fog computing [7] and moving edge computing.

With the popularity of the IoT and the increasing power of mobile applications, the computing demands on user devices have reached unprecedented levels. In particular, the rapid development of 5G communication technology [8-9] and cloud computing technology [10] accelerate this process. Data generated by IoT devices can be sent to edge devices for processing through wireless communication technology [11]. If the processing capacity of edge devices is exceeded, data processing can be carried out through the cloud server. On the one hand, the communication delay of the system should be controlled within milliseconds. On the other hand, the safety of data transmission should be ensured. A typical architecture of MEC is illustrated in Figure 1. The structure can be roughly divided into three layers: IoT device layer, edge device layer and cloud server layer. The IoT devices can be mobile phones, cars, traffic lights, cameras and other devices. IoT devices can communicate with edge devices through edge networks (such as WiFi, 5g networks, etc.). Edge devices can communicate with cloud servers through the core network (such as IP, MPLS, etc.). Now, we face a lot of security challenges in the implementation of this process. Therefore, it is very important to design an efficient and secure three-party communication protocol.

*Corresponding Author: Jian Shen; E-mail: s_shenjian@126.com
DOI: 10.53106/160792642022052303002

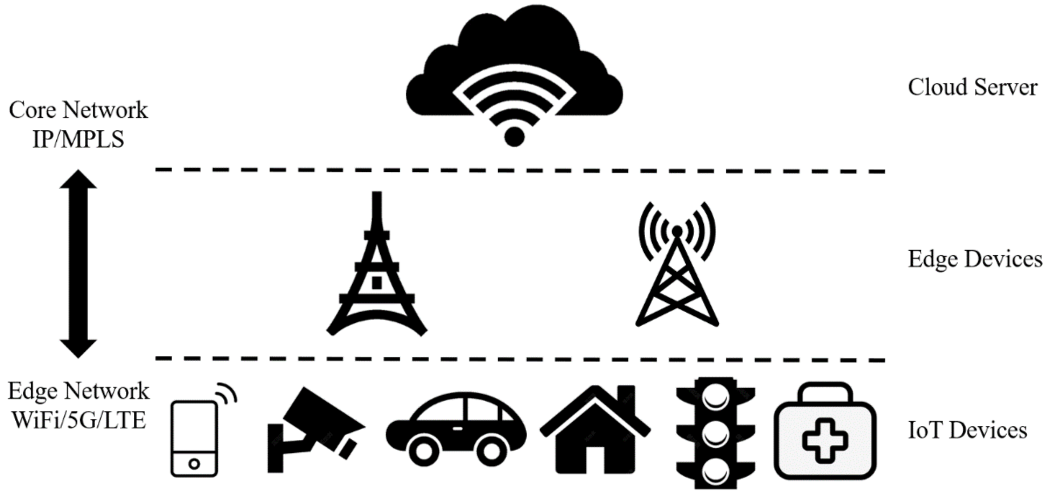

Figure 1. A typical architecture of MEC

1.1 Motivation

As we all know, security is one of the most important issues in MEC. Due to the use of wireless network communication mode, MEC is vulnerable to various attacks by adversary. For example, malicious users may intercept the information, or even modify the information, which brings great security risk to users. Some malicious users may send wrong messages to other user by impersonating other legitimate users or edge devices, which may lead serious safety hazards. Therefore, how to ensure the security of communication among IoT devices, edge devices and cloud server is a problem worth to be discussed.

Privacy-preserving is another important issue needs to be addressed in MEC. When a user communicates with edge devices or cloud server, it is necessary to ensure its private information that is not disclosed (such as real identity). Some researchers have proposed conditional privacy-preserving scheme [12-13] based on this. Under the premise of ensuring user privacy, legal authentication of cloud servers, edge devices and users is worth studying.

Our contributions: In this paper, a privacy-preserving authentication and key agreement protocol among users, edge devices and cloud server is proposed. The main contributions are described as follows.

- **An efficient three-party authentication and key agreement protocol is proposed.** In this paper, a three-party authentication and key agreement protocol for users, edge devices and cloud server is designed. In addition, bilinear pairings is not adopted in this protocol, which greatly reduces the computational and communication overhead.
- **Conditional privacy-preserving is realized by the proposed protocol.** The real identity of the user is anonymized by the owner with a random number, which cannot be analyzed by other users and edge devices based on existing information. If the user is malicious, the trusted authority cloud server can track the real identity of the user according to the master key and its anonymous identity.
- **The protocol is secure under the security model.** We have made a formal security proof for the protocol, and the security proof shows that the protocol is secure under the DDH and CDH assumption.

1.2 Organization

The reminder of this paper is organized as follows. Section 2 analyzes the current research status. Section 3 describes the preliminaries in this paper. Section 4 describes the problem statement of this paper, including system model, security model and security definition. Section 5 presents three-party authentication and key agreement protocol in detail. Section 6 proves the security and analyzes the security attributes of the scheme. Section 7 evaluates the performance of the scheme through experiment result. Section 8 concludes the work of this paper.

2 Related Works

Nowadays, a lot of researchers have put forward their own scheme for authentication.

Cui et al. [14] proposed a privacy-preserving authentication scheme using cuckoo filter. In this scheme, the hash value of the signature that has been authenticated for the first time is stored in cuckoo filter. When the signature needs to be authenticated again, the hash value of the signature only needs to be compared with the hash value of the signature in Cuckoo filter. However, this scheme requires a large data structure to store the signature. In order to improve message filtering efficiency and reduce data storage overhead. Zhou et al. [15] proposed a lightweight multi-key privacy-preserving scheme for location-based services that includes a message filtering algorithm. In this scheme, Road Side Units (RSUs) evaluate the redundancy factor of each user's message, so as to filter the duplicate message in the system according to the redundancy factor before information authentication. This way greatly reduced computation cost and communication cost. Moreover, the message is encrypted multiple times with different key, which increases the security of message and protects the privacy of user.

Lee et al. [16] proposed a three-party mutual authentication and key agreement protocol which based on Diffie-Hellman key exchange. Lv et al. [17] designed a novel three-party authenticated key exchange protocol, which is more efficient with less computational overhead. At the same time, the use of one-time key solves the key escrow problem. However, the security level of the above two is not high, for example, neither of them achieves user anonymity and traceability. The three-party mutual authentication protocol can be applied to a variety of different scenarios. Chiou et al. [18] applied the three-party mutual authentication protocol into the medical environment and it satisfied more security requirements than above two. Jia et al. [19] applied the three-party mutual authentication protocol to the IoT healthcare system. Ma et al. [20] improved this protocol and applied it to VANETs, realizing mutual authentication among vehicles, fog nodes and cloud server, and reducing computational overhead.

Bagga et al. [21] proposed a new mutual authentication and key agreement protocol, which adopts two levels of authentication and key agreement, and realized the dynamic addition of vehicles and RSUs. Wazid et al. [22] applied user authentication and key agreement into the in Internet of Drones Deployment, which can be resistant against various known attacks through the formal security verification using the widely accepted AVISPA tool.

In order to support anonymity and traceability of vehicles at the same time, some researchers have adopted a conditional privacy-preserving mechanism. Mukherjee et al. [23] take advantage of lattice-based cryptography to design a conditional privacy-preserving authentication scheme. The scheme does not have fast computing power, but can also resist the quantum attack due to the hard lattice problem. Zhou et al. [24] proposed a roaming authentication scheme with conditional privacy-preserving function that is a cross-domain vehicle authentication scheme. Ali et al. [25] designed an identity-based conditional privacy-preserving scheme using bilinear pairing, which is mainly used to solve the authentication problem in V2V and V2I in VANETs. Tzeng et al. [26] proposed an identity-based privacy-preserving scheme, which support batch authentication and conditional privacy-preserving at the same time.

Since the user privacy key is managed uniformly by a trusted authority (TA) in the above scheme, there is a key escrow problem in these scheme. Zhang et al. [27] proposed a distributed aggregated privacy-preserving authentication scheme. One-time privacy key is adopted for communication between vehicles and low-level TA. However, the use of one-time private key will generate a lot of computational and communication overhead. Zhong et al. [28] proposed a privacy-preserving scheme based on certificateless aggregate signature, which not only solves the key escrow problem, but also reduces the computational and communication cost.

Group authentication is an important research field in VANETs. Han et al. [29] proposed an efficient self-certified and deniable group key agreement protocol. The protocol adopts deny negotiation to establish group communication to reduce the transmission of group keys, so as to achieve the effect of reducing vehicle authentication step. Jiang et al. [30] proposed an anonymous authentication scheme based on group signature. This scheme improves the efficiency of anonymous authentication of vehicles by adding region trust authority. Hasrouny et al. [31] designed a trust model based on group leader. The scheme evaluates the trust value of each vehicle, and selects a node with higher credibility as the group leader to manage the communication between vehicles.

3 Preliminaries

In this section, some of the preliminaries covered in this paper is described, including message authentication code and some complexity assumptions.

3.1 Message Authentication Code (MAC)

A message authentication code scheme consists of three algorithms: $MAC.KeyGen$, $MAC.Mac$ and $MAC.Verify$. Firstly, the $MAC.KeyGen$ algorithm evenly selects secret key from the key space. Secondly, input the string m under the algorithms $MAC.Mac$ and output the tag τ. Thirdly, when the receiver receives the tag τ, inputs the information (m, τ) and runs $MAC.Verify$. If the output is 1, it means the verification is passed, otherwise, the verification fails.

3.2 Complexity Assumptions

(*Computational Diffie-Hellman (CDH) Assumption*). Let G represent a finite cyclic group of order n. The CDH problem is computing gab based on the given elements (g, g^a, g^b), where g and (a, b) are represent the generator of G and the random number in Z_p^*, respectively. The probability of solving the CDH problem for any algorithm in probabilistic polynomial time (PPT) is negligible.

(*Decisional Diffie-Hellman (DDH) Assumption*). let G represent a finite cyclic group of order n. The DDH problem is distinguishing g^c and g^{ab} based on the given elements (g, g^a, g^b, g^c), where g and (a, b, c) are represent the generator of G and the random number in Z_p^*, respectively. The probability of solving the DDH problem for any algorithm in PPT is negligible.

4 Problem Statement

In this section, we describe the system model, security model and security definition of the protocol. The specific description is as follows.

4.1 System Model

In this section, the architecture of system model of this paper is shown in Figure 2. The system model consists of three entities: the user, the edge device and the cloud server.

- **User:** Each user can be a smart furniture, mobile phone, vehicle and other IoT devices, which has less computing power. Users communicate with edge devices via the wireless communication technology. The user registers with the cloud server to obtain private key by using its identity.
- **Edge device:** The edge device is deployed at the roadside that has certain computing power, which is used for communication between the user and the cloud server. Edge device can negotiate a key with user and cloud server to generate a common session key. The edge device is a semi-trust participant.
- **Cloud server:** In system setup phase, the cloud server is used for generate the master private key and security parameter, which has strong computing power and storage capacity. When users and edge devices need to be

registered, the cloud server uses the master private key and their real identity to complete the calculation of the private key and sends it to them through a secure channel.

The cloud server is a trust participant.

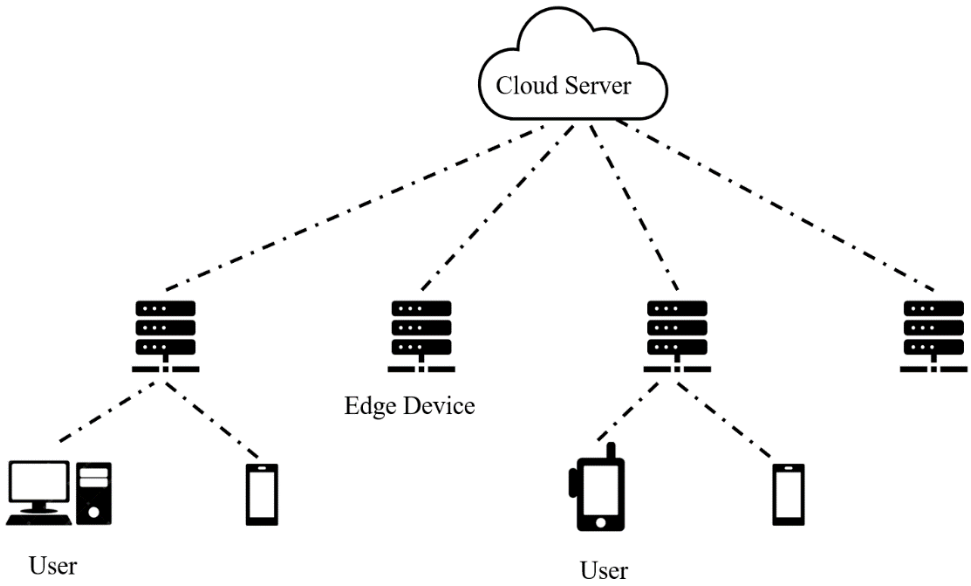

Figure 2. The architecture of system model

4.2 Security Model

We formulated a series of games between challenger C and adversary A to define our security model. Assume that participant $\prod_i \in \{U, ED, CS\}$ represents the i-th instance and Λ represents the entire protocol. The A can ask the C oracle queries, and the C can respond.

- $Send(\prod_i, m)$: If A asks the query for the message m, the C executes the specific steps of the protocol and returns the result.
- $Execute(\prod_{U_i}, \prod_{ED_j}, \prod_{CS})$: This oracle query models a passive eavesdropping attack. During the execution of the protocol, the A can obtain the communication message between $\prod_{U_i}$, $\prod_{ED_j}$ and $\prod_{CS}$ through passive eavesdropping attack.
- $Reveal(\prod_i)$: This oracle query models the leakage of the session key. During the A's query process, if the instance $\prod_i$ accepts, then outputs the session key, otherwise outputs $\perp$.
- $Test(\prod_i)$: This oracle query models the semantic security of the session key Key. When the A sends a $Test()$ query to the C, the C adopts a coin toss b. If $b = 1$, the C sends the session key of instance $\prod_i$ to the A. If $b = 0$, the C chooses a random number equals to the length of the session key and sends it to the A. After receiving the information, the A will guess the value of b. If the result of the guess is always correct, the session key of the protocol is broken.

4.3 Security Definitions

Some security definitions are proposed for our protocol to provide strong security guarantees.

4.3.1 Definition

(Authentication Key Agreement (AKA) - security). Through the C's $Test()$query, the A will guess the value of b according to the query result and output the guess result b'. If $b' = b$, it means that the semantic security of the session key of the protocol has been compromised. Let $Success(A)^{Key}$ represent the event where the A guessed the value b correctly and won the game. The advantage that the A can break the semantic security of the session key by guessing the value b correctly can be defined as

$$Adv_A^{Key} = |\, 2\Pr[Succ(A)^{Key}] - 1\,|.$$

Assuming that the advantage Adv_A^{Key} is negligible for any PPT adversary, then the protocol Λ can be said to be AKA-secure.

4.3.2 Definition

(Privacy preservation). If the A can obtain the ciphertext of the communication between the user, the edge device and the cloud server through $Send()$ query. Then when the A cannot obtain the user's private data through the calculation of the ciphertext, it is considered that the protocol Λ has achieved privacy perservation.

5 Our Proposed Scheme

5.1 An Overview

In the system, both cloud server and edge devices have an ability to analyze data generated by IoT device. During the use of an IoT device, edge devices will analyze the data generated by IoT devices and give corresponding instructions. When the data processing is beyond the processing capacity of the edge device or the data is too sensitive to be processed in the edge device, the cloud server has to process the data. Therefore, it is necessary to negotiate a common session key between the IoT device, the edge device, and the cloud server for subsequent data transfers. If the transmitted information is sensitive and the edge device does not have the right to view the message, the IoT device and the cloud server should negotiate a session key. In order to ensure the security of data transmission among IoT device, edge devices and cloud server. We have proposed an efficient three-party authentication and key agreement protocol. The protocol is mainly divided into the following parts: user registration, edge device registration, authentication and key agreement, user revocation and edge device revocation. The various notations and their descriptions listed in Table 1 are used in the paper for describing the phases.

5.2 System Setup

Take the security parameter κ as input, the trust participant cloud server (CS) generates a multiplicative cyclic group G of q order with a generator g. Next, CS chooses a random number x as the master private key, sets the system public key $g_{Pub} = g^x$. And then chooses five hash functions H_1, H_2, H_3, H_4 and H_5, where H_1:$\{0,1\}^* \rightarrow Z_q^*$, H_2:$G \rightarrow \{0,1\}^*$, $H_3 : G \times \{0,1\}^* \times \{0,1\}^* \times G \rightarrow Z_q^*$, $H_4 : G \rightarrow G$, H_5:$\{0,1\}^* \times G \times G \rightarrow Z_q^*$. The public parameter is defined as $PP = \{g,\ g_{Pub},\ q,\ G,\ H_1,\ H_2,\ H_3,\ H_4,\ H_5\}$ and published publicly by CS.

Table 1. Notations in our protocol

Symbol	Description
q	A large prime number
G	A multiplicative cyclic group with q order
g	A generator of G
H_1, H_2, H_3, H_4, H_5	Cryptographic hash functions
x	The master private key of system
g_{pub}	The public key of system
SK_{U_i},SK_{ED_j}	The private key of user and edge device
ID_{U_i}	The real identity of user
ID_{ED_j}	The real identity of edge device
AID_i	The anonymous identity of user
AID_{i1}	Anonymous identity part 1 of the i -th user
AID_{i2}	Anonymous identity part 2 of the i −th user
T_{U_i} T_{ED_j} T_{CS} T_{edj}	Timestamp
r, w_1, w_2, w_3, w_4	Random number selected from Z_q^*
Key_1	Session key of user, edge device and cloud server
Key_2	Session key of user and cloud server

5.3 User Registration

During the user U_i registration phase. The U_i wants to join the system, it is necessary to ask the CS for permission. Then the CS decides whether to assign a private key to the U_i according to the validity of the user's information. The U_i sends a registration request to the CS with a real identity ID_{U_i}. After the CS is verified, the master private key x and the hash function H_1 are used to compute

$$SK_{U_i} = g^{\frac{1}{x+H_1(ID_{U_i})}}.$$

Then the CS stores the record $\{ID_{U_i},\ SK_{U_i}\}$ to the database and sends SK_{U_i} to the U_i through a secure channel.

5.4 Edge Device Registration

The Edge device ED_j register with the CS. The CS decides whether to assign a private key according to the validity of its identity. The ED_j sends a registration request to the CS with an identity ID_{ED_j}. After the CS is verified, the master private key x and the hash function H_1 are used to compute

$$SK_{ED_j} = g^{\frac{1}{x+H_1(ID_{ED_j})}}.$$

Then the CS stores the record $\{ID_{ED_j},\ SK_{ED_j}\}$ to the database and sends SK_{ED_j} to the ED_j through a secure channel.

5.5 Authentication and Key Agreement

In this phase, U_i, ED_j, and CS authenticate each other and negotiate two session keys to ensure the subsequent transfer of information. The process of authentication and key agreement is shown in Figure 3.

The U_i anonymizes its identity to prevent the disclosure of its real identity. The U_i chooses a random number r and computes anonymous identity $AID_i = (AID_{i1},\ AID_{i2})$, where $AID_{i1} = g^r$, $AID_{i2} = ID_{U_i} \oplus H_2({g_{pub}}^r)$.

Then the U_i chooses a random number $w_1 \in Z_q^*$, which is secure in this protocol. Then computes $g_{U_i} = g^{w_1}$; $\varphi_{U_i} = (SK_{U_i})^{w_1}$ and $\pi_{U_i} = H_3(g_{U_i},\ ID_{U_i},\ T_{U_i},\ \varphi_{U_i})$. Let T_{U_i} represent the current timestamp. The U_i sends $\{\varphi_{U_i}, \pi_{U_i}, AID_i,\ T_{U_i}\}$ to ED_j. After received the message, the ED_j first checks the freshness of T_{U_i}. If it is not fresh, the message will be rejected by ED_j. Otherwise, the ED_j chooses a random number $w_2 \in Z_q^*$ and computes $g_{ED_j} = g^{w_2}$; $\varphi_{UE} = (\varphi_{U_i})^{w_2}$, $\varphi_{ED_j} = (SK_{ED_j})^{w_2}$, $\pi_{ED_j} = H_3(g_{ED_j},\ ID_{ED_j},\ T_{ED_j},\ \varphi_{ED_j})$. Let T_{ED_j} represent the current timestamp. ED_j sends $\{\varphi_{U_i},\ \varphi_{ED_j}, \varphi_{UE}, \pi_{U_i}, \pi_{ED_j}, AID_i,\ ID_{ED_j},\ T_{U_i},\ T_{ED_j}\}$ to the CS.

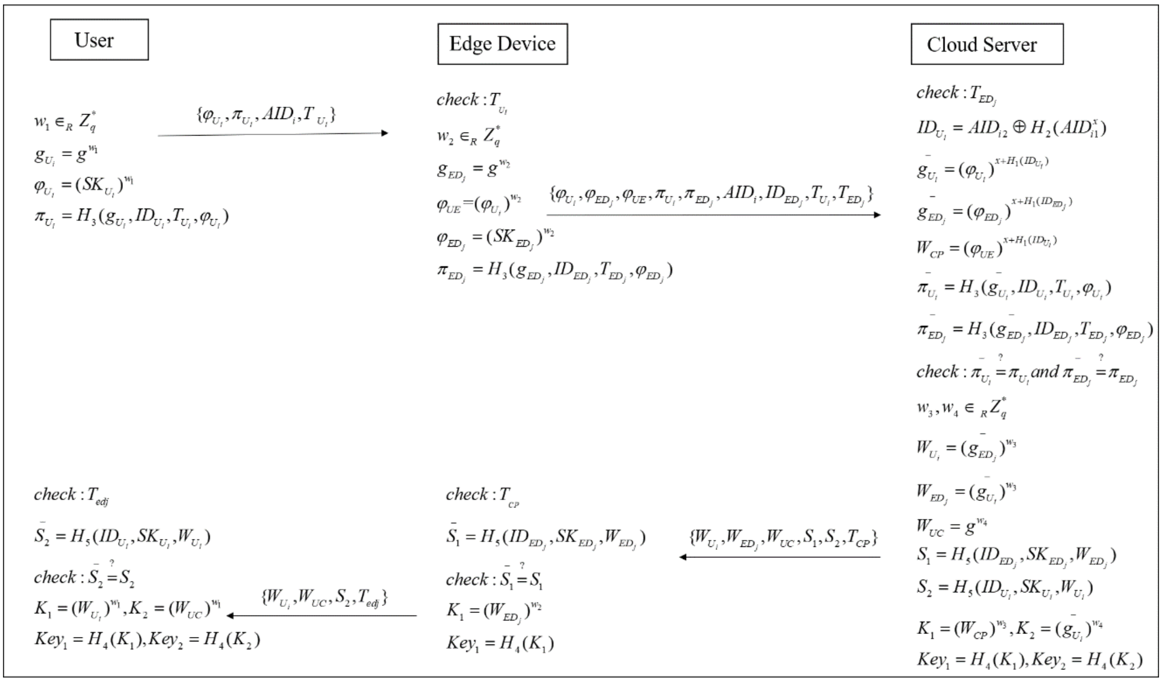

Figure 3. The architecture of system model

When received the message, the CS first checks the freshness of T_{U_i} and T_{ED_j}. If they are not fresh, the message will be rejected by the CS. Otherwise, the CS computes $\bar{\pi}_{U_i}$ and $\bar{\pi}_{ED_j}$. The calculation process is shown as follows. The CS uses master private key x and hash function H_2 to calculates $ID_{U_i} = AID_{i2} \oplus H_2(AID_{i1}{}^{x})$. Then the CS calculates $\bar{g}_{U_i} = (\varphi_{U_i})^{x+H_1(ID_{U_i})}$, $\bar{g}_{ED_j} = (\varphi_{ED_j})^{x+H_1(ID_{ED_j})}$, $W_{CS} = (\varphi_{UE})^{x+H_1(ID_{U_i})}$, $\bar{\pi}_{U_i} = H_3(\bar{g}_{U_i}, ID_{U_i}, T_{U_i}, \varphi_{U_i})$, $\bar{\pi}_{ED_j} = H_3(\bar{g}_{ED_j}, ID_{ED_j}, T_{ED_j}, \varphi_{ED_j})$. Next, the CS checks if both of $\bar{\pi}_{U_i} = \pi_{U_i}$ and $\bar{\pi}_{ED_j} = \pi_{ED_j}$ are true. If one of the two equations is not equal, the CS stops the process. Otherwise, the CS calculates the private key of user and edge device. Then the CS chooses two random numbers $w_3, w_4 \in Z_q^*$ and computes $W_{U_i} = (g^{w_2})^{w_3} = g^{w_2 w_3}$, $W_{ED_j} = (g^{w_1})^{w_3} = g^{w_1 w_3}$, $W_{UC} = g^{w_4}$, $S_1 = H_5(ID_{ED_j}, SK_{ED_j}, W_{ED_j})$, $S_2 = H_5(ID_{U_i}, SK_{U_i}, W_{U_i})$, $K_1 = (W_{CS})^{w_3}$, $K_2 = (\bar{g}_{U_i})^{w_4}$, $Key_1 = H_4(K_1)$, $Key_2 = H_4(K_2)$. Let T_{CS} represent the current timestamp. Among them, Key_1 is the common session key of the U_i, the ED_j and the CS, and Key_2 is the session key of both the U_i and the CS. The CS sends $\{W_{U_i}, W_{ED_j}, W_{UC}, S_1, S_2, T_{CS}\}$ to ED_j.

The ED_j first verifies the freshness of the timestamp T_{CS} after receiving the message. Then the ED_j uses its private key to calculates $\bar{S}_1 = H_5(ID_{ED_j}, SK_{ED_j}, W_{ED_j})$ and checks whether $\bar{S}_1 = S_1$ holds. If not, the ED_j aborts the request. Otherwise, the ED_j calculates $K_1 = (W_{ED_j})^{w_2}$, $Key_1 = H_4(K_1)$. Next, the ED_j generates a timestamp T_{edj} and sends $\{W_{U_i}, W_{UC}, S_2, T_{edj}\}$ to the U_i.

The U_i checks the freshness of the timestamp T_{edj}. Then the U_i uses its private key to calculates $\bar{S}_2 = H_5(ID_{U_i}, SK_{U_i}, W_{U_i})$ and checks whether $\bar{S}_2 = S_2$ holds. If not, the U_i aborts the request. Otherwise, the U_i calculates $K_1 = (W_{U_i})^{w_1}$, $Key_1 = H_4(K_1)$, $K_2 = (W_{UC})^{w_1}$, $Key_2 = H_4(K_2)$.

5.6 User Revocation

If one user is found to be malicious, the CS can reveal the real identity of the user through its anonymous identity, and then revoke its identity. After the user is revoked, the user cannot enjoy the privilege of data analyzing by edge devices and CS. The CS runs $MAC.Mac_K(m)$ to generate a tag τ, where $K = SK_{U_i}$, $m = (ID_{U_i}, SK_{U_i}, revoke)$ and then sends τ to the user. In the end, the record (ID_{U_i}, SK_{U_i}) will be deleted by CS from the database. The output result of user operation algorithm $MAC.Verify(\tau, m)$, if it is 1, it means that the user has been revoked.

5.7 Edge Device Revocation

In the edge computing system, once the edge device is damaged or compromised, the CS needs to revoke it. During the revocation process, the CS will delete the record (ID_{ED_j}, SK_{ED_j}) from the database. Since the private key SK_{ED_j} is deleted from the database, the edge device cannot be legally authenticated.

6. Security Analysis

In this section, we first analyze the correctness of the protocol. Then, we give a formal security proof for the security of the protocol. Finally, we analyzed the security attributes of the protocol.

6.1 Correctness

The correctness of the three-party authentication. The parameter φ_{U_i} generated by user is $\varphi_{U_i} = (SK_{U_i})^{w_1} = g^{\frac{w_1}{x+H_1(ID_{U_i})}}$, and the parameters φ_{ED_j} generated by edge device is $\varphi_{ED_j} = (SK_{ED_j})^{w_2} = g^{\frac{w_2}{x+H_1(ID_{ED_j})}}$. The proof of authentication process is shown as follows.

$$\begin{aligned}
\bar{g}_{U_i} &= (\varphi_{U_i})^{x+H_1(ID_{U_i})} \\
&= (g^{\frac{w_1}{x+H_1(ID_{U_i})}})^{x+H_1(ID_{U_i})} \\
&= g^{w_1} \\
&= g_{U_i},
\end{aligned}$$

$$\begin{aligned}
\bar{\pi}_{U_i} &= H_3(\bar{g}_{U_i}, ID_{U_i}, T_{U_i}, \varphi_{U_i}) \\
&= H_3(g_{U_i}, ID_{U_i}, T_{U_i}, \varphi_{U_i}) \\
&= \pi_{U_i},
\end{aligned}$$

$$\begin{aligned}
\bar{g}_{ED_j} &= (\varphi_{ED_j})^{x+H_1(ID_{ED_j})} \\
&= (g^{\frac{w_2}{x+H_1(ID_{ED_j})}})^{x+H_1(ID_{ED_j})} \\
&= g^{w_2} \\
&= g_{ED_j},
\end{aligned}$$

$$\begin{aligned}
\bar{\pi}_{ED_j} &= H_3(\bar{g}_{ED_j}, ID_{ED_j}, T_{ED_j}, \varphi_{ED_j}) \\
&= H_3(g_{ED_j}, ID_{ED_j}, T_{ED_j}, \varphi_{ED_j}) \\
&= \pi_{ED_j}.
\end{aligned}$$

The correctness of three-party key agreement. The session key generated by CS, edge device and user are $H_4(W_{CS}^{w_3})$, $H_4((\bar{g}_{U_i})^{w_4})$, $H_4(W_{ED_j}^{w_2})$, $H_4(W_{U_i}^{w_1})$ and $H_4(W_{UC}^{w_1})$. The parameters W_{CS}, W_{ED_j}, W_{U_i}, $H_4((\bar{g}_{U_i})^{w_4})$ and $H_4(W_{UC}^{w_1})$ are calculated as follows.

$$\begin{aligned}
W_{CS} &= (\varphi_{UE})^{x+H_1(ID_{U_i})} \\
&= (g^{\frac{w_1 w_2}{x+H_1(ID_{U_i})}})^{x+H_1(ID_{U_i})} \\
&= g^{w_1 w_2},
\end{aligned}$$

$$W_{ED_j} = (\bar{g}_{U_i})^{w_3} = g^{w_2 w_3},$$

$$W_{U_i} = (\bar{g}_{ED_j})^{w_3} = g^{w_1 w_3},$$

$$(\bar{g}_{U_i})^{w_4} = (W_{UC})^{w_1} == g^{w_1 w_4}.$$

Since $W_{CS}{}^{w_3} = W_{ED_j}{}^{w_2} = W_{U_i}{}^{w_1} = g^{w_1 w_2 w_3}$ and $(\bar{g}_{U_i})^{w_4} = (W_{UC})^{w_1} = g^{w_1 w_4}$, then we have the common session key $\mathrm{Key}_1 = H_4(W_{CS}{}^{w_3}) = H_4(W_{ED_j}{}^{w_2}) = H_4(W_{U_i}{}^{w_1})$ and $\mathrm{Key}_2 = H_4((\bar{g}_{U_i})^{w_4}) = H_4(W_{UC}{}^{w_1})$. Therefore, the correctness of our protocol has been proven.

6.2 Formal Security Proof

In this subsection, the protocol Λ will be proven to be AKA secure under the security model.

Theorem 6.1 The Key_1 in our protocol is AKA-secure under the attack of any PPT A.

Proof. If A can break the protocol with a non-negligible probability ε, then C can be constructed to solve the DDH assumption with a probability

$$\varepsilon' \geq \frac{1}{q_{Se}}\left(\varepsilon - \frac{\sum_{i=1}^{5} q_{h_i}{}^2 + (q_{Se} + q_{Ex})^2}{2q}\right).$$

Given information (g, g^a, g^b, g^c), the C's task is to judge whether equation $g^{ab} \stackrel{?}{=} g^c$ is true or not based on the given information. The C chooses a number x randomly and set the system public key $P_{Pub} = g^x$. The public parameter $PP = \{g, g_{Pub}, q, G, H_1, H_2, H_3, H_4, H_5\}$ is sent to A by C. Then the C assigns identities ID_{U_i} and ID_{ED_j} to the user and the edge device respectively. Next, the C calculates the private key for the user and the edge device respectively according to identity and the random value x. The C receives the results of the A's query and responds to it.

- *Send query*. The C generates and maintains a list L to record the results of the A's query. The A can ask the Send query as below, and the C will respond.
- *Send(Π_{U_i})*: When receiving a query from A, the C chooses two number w_1 and r randomly, computes $AID_i = (AID_{i1}, AID_{i2})$, $AID_{i1} = g^r$, $AID_{i2} = ID_{U_i} \oplus H_2(g_{pub}{}^r)$, $g_{U_i} = g^{w_1}$, $\varphi_{U_i} = (SK_{U_i})^{w_1}$, $\pi_{U_i} = H_3(g_{U_i}, ID_{U_i}, T_{U_i}, \varphi_{U_i})$, and send the message $M_1 = \{AID_i, \varphi_{U_i}, \pi_{U_i}, T_{U_i}\}$ to A.
- *Send(Π_{ED_j}, M_1)*: When receiving this query from A, the C chooses a number w_2 randomly, computes $g_{ED_j} = g^{w_2}$, $\varphi_{UE} = (\varphi_{U_i})^{w_2}$, $\varphi_{ED_j} = (SK_{ED_j})^{w_2}$, $\pi_{ED_j} = H_3(g_{ED_j}, ID_{ED_j}, T_{ED_j}, \varphi_{ED_j})$, and send the message $M_2 = \{\varphi_{U_i}, \varphi_{ED_j}, \varphi_{UE}, \pi_{U_i}, \pi_{ED_j}, AID_i, ID_{ED_j}, T_{U_i}, T_{ED_j}\}$ to A.
- *Send(Π_{CS}, M_2)*: When received a query from the A, the C uses the above query results to make a judgment on the correctness of π_{U_i} and π_{ED_j}. If both of them are correct, the C chooses two numbers w_3 and w_4 randomly, computes $W_{U_i} = (g_{ED_j})^{w_3}$, $W_{ED_j} = (g_{U_i})^{w_3}$, $S_1 = H_5(ID_{ED_j}, SK_{ED_j}, W_{ED_j})$, $S_2 = H_5(ID_{U_i}, SK_{U_i}, W_{U_i})$, $K_1 = (W_{CP})^{w_3}$, $Key_1 = H_4(K_1)$, and send $M_3 = \{W_{U_i}, W_{ED_j}, S_1, S_2, T_{CP}\}$ to A. Otherwise, the C rejects the query of the A and output $\perp$.
- *Send(Π_{U_i}, M_3)*: When received a query from the A, the C make a judgment on the correctness of S_1. If it is correct, the C computes $K_1 = (W_{ED_j})^{w_2}$, $Key_1 = H_4(K_1)$, and send $M_4 = \{W_{U_i}, S_2, T_{edj}\}$ to A. Otherwise, the C rejects the query of the A and output $\perp$.

- *Send(*Π_{U_i}, M_4*)*: When received a query from the A, the C make a judgment on the correctness of S_2. If it is correct, the C computes $K_1 = (W_{U_i})^{w_1}$, $Key_1 = H_4(K_1)$. Otherwise, the C rejects the query of the A and output $\perp$. Then the message $\{M_1, M_2, M_3, M_4\}$ is added to the list L.
- *Execute(*Π_{U_i}, Π_{ED_j}, Π_{CS}*)*: When received this query, the C takes out the message $\{M_1, M_2, M_3, M_4\}$ from the list L and return it to A.
- *Reveal(*Π_i*)*: When the A asks this query, if the instance Π_i agrees, the C will send the session key Key_1 to the A. Otherwise, C outputs $\perp$.
- *Test(*Π_i*)*: When the A sends a *Test(*Π_i*)* query to the C, the C adopts a coin toss b. If $b = 1$, the C sends the session key Key of instance Π_i to the A. If $b = 0$, the C chooses a random number equal to the length of the session key and sends it to the A.

The proof includes four games (G_0, G_1, G_2, G_3), where ε_i indicates that the A guessed the value b correctly in the i -th game.

Game G_0: This game simulates an ordinary attack by an A. Thus,

$$\varepsilon = | 2\Pr[\varepsilon_0] - 1 |. \quad \textbf{(1)}$$

Game G_1: In this game, A can ask *Hash* query and *Execute* query to the C. When receiving query from the A, the C searches the list L and sends the results to the A. Otherwise, the C selects a random number and return to A. Since the C simulates a real attack, the advantage of the A in this game is indistinguishable from G_0. Thus, we have

$$\Pr[\varepsilon_1] = \Pr[\varepsilon_0]. \quad \textbf{(2)}$$

Game G_2: In this game, the A can ask various queries to the C as in G_1. The difference from G_1 is that when the hash query collides or the transcripts query collides, the C will terminate the query. According to the birthday paradox, the probability of hash collision or transcripts collision is at most ${q_{h_i}}^2/2q$ and $(q_{Se} + q_{Ex})^2/2q$ respectively. Thus, we can conduct

$$| \Pr[\varepsilon_2] - \Pr[\varepsilon_1] | \le \frac{\sum_{i=1}^{5} {q_{h_i}}^2 + (q_{Se} + q_{Ex})^2}{2q} \quad \textbf{(3)}$$

Game G_3: This game simulates *Send* query. When receiving a query from A, the C responds with the following.

- When receiving a query from A, the C computes $g_{U_i} = g^a$ and $\varphi_{U_i} = (g^a)^{\frac{1}{x+H_1(ID_{U_i})}}$, then assigns values to AID_i, π_{U_i} and T_{U_i} respectively. Finally, the message $M_1 = \{AID_i, g_{U_i}, \varphi_{U_i}, \pi_{U_i}, T_{U_i}\}$ is sent to A.
- When receiving this query from A, the C computes $g_{ED_j} = g^b$, $\varphi_{UE} = (g^c)^{\frac{1}{x+H_1(ID_{U_i})}}$ and $\varphi_{ED_j} = (g^b)^{\frac{1}{x+H_1(ID_{ED_j})}}$, then sets values to π_{ED_j} and T_{ED_j} respectively. Finally, the message $M_2 = \{\varphi_{U_i}, \varphi_{ED_j}, \varphi_{UE}, \pi_{U_i}, \pi_{ED_j}, AID_i, ID_{ED_j}, T_{U_i}, T_{ED_j}\}$ is sent to A.
- When received a query from the A, the C chooses a random number w_3, computes $W_{CP} = g^c$ and $W_{U_i} = g^{bw_3}$, $W_{ED_j} = g^{aw_3}$. Then calculates S_1 and S_2. The C sets the values of $K = g^{cw_3}$, and computes $Key = H_4(K)$. The C stores w_3 in the list L. Finally, the C sends the message $M_3 = \{W_{U_i}, W_{ED_j}, S_1, S_2, T_{CP}\}$ to A.
- When received a query from the A, the C searches the list L for w_3. The C sets $K = g^{cw_3}$ and T_{edj}, calculates $Key = H_4(K)$. Finally, the C returns the message $M_4 = \{W_{U_i}, S_2, T_{edj}\}$ to A.
- When received a query from the A, the C searches the list L for w_3. The C sets $K = g^{cw_3}$, calculates $Key = H_4(K)$.

If exists an A who can distinguish G_3 from G_2 successfully, then the C can use A as a subroutine to break the DDH difficulty assumption. I.e. if $g^{cw_3} = g^{abw_3}$, then $g^c = g^{ab}$.

The probability that the C chooses an instance is $1/q_{Se}$, therefore,

$$| \Pr[\varepsilon_3] - \Pr[\varepsilon_2] | \le \varepsilon' q_{Se}, \quad \textbf{(4)}$$

where ε' represents the advantage of C in breaking the DDH assumption.

From (1) to (5), we can conduct

$$\varepsilon \le \frac{\sum_{i=1}^{5} {q_{h_i}}^2 + (q_{Se} + q_{Ex})^2}{2q} + \varepsilon' q_{Se}.$$

Therefore, we have

$$\varepsilon' \ge \frac{1}{q_{Se}} \left(\varepsilon - \frac{\sum_{i=1}^{5} {q_{h_i}}^2 + (q_{Se} + q_{Ex})^2}{2q}\right).$$

Thus, the security of the protocol has been proven.

Theorem 6.2. The Key_2 in our protocol is AKA-secure under the attack of any PPT A.

Proof. In our security model, A can obtain communication message among users, edge devices and cloud servers through eavesdropping. The A can obtain the message $W_{UC} = g^{w_4}$ through *Execute* query. Then the A can obtain the user's private key through *Reveal* query. A can compute the message $g_{U_i} = g^{w_1}$. Due to the hardness of CDH assumption and w_1, w_4 are private, the A cannot calculate $g^{w_1 w_4}$ according t0 g^{w_1} and g^{w_4}.

6.3 Analysis of Security Requirement

In this subsection, we analyze the security requirements that the protocol meets.

(1) **Mutual authentication**: The cloud server can authenticate user and edge device respectively from the authentication request and response corresponding result. All the authentication messages are embedded in the authentication request. The cloud server authenticates

user and edge devices by checking whether the equations $\bar{\pi}_{U_i} = \pi_{U_i}$ and $\bar{\pi}_{ED_j} = \pi_{ED_j}$ are hold. At the same time, the user and edge device authenticate the cloud server by checking whether the equations $\bar{S}_2 = S_2$ and $\bar{S}_1 = S_1$ are true. Therefore, our protocol can support mutual authentication.

(2) **Anonymity**: The real identity of the user is anonymized by choosing random number r. If one adversary wants to extract the real identity from the anonymous identity, it can calculate the real identity of the user from $ID = AID_{i2} \oplus H_2(g_{pub}{}^r)$ .The adversary need to computes $g_{pub}{}^r = AID_{i1}{}^x = g^{xr}$. The random number r and the master private key x are held by the user and the cloud server respectively, so that the adversary cannot calculate the real identity according to the anonymous identity of the user.

(3) **Traceability**: The cloud server is the legal holder of the master private key x, and it can calculate the real identity of the vehicle according to the master private key from the anonymous identity.

(4) **Known-key security**: The session key Key contains random number w_1, w_2, w_3 and w_4 that are randomly selected by user, edge device and cloud server. So the different session key are independent of each other among protocol executions. Hence, our protocol support known-key security.

(5) **Session key security**: In our protocol, the session key Key_1 and Key_2 are calculated by formula $Key_1 = H_4(g^{w_1 w_2 w_3})$ and $Key_2 = H_4(g^{w_1 w_4})$, where the parameters w_1, w_2, w_3 and w_4 are random numbers. Therefore the adversary cannot calculate the session key according to the existing information.

(6) **Forward secrecy**: In our protocol, w_1, w_2, w_3 and w_4 are selected randomly in Z_q^*. Therefore, the disclosure of the user's private key will not lead to the disclosure of the session key, which ensures the communication security of the system.

(7) **Resistant against various kinds of attacks**: Our protocol is resistant to man-in-the-middle attack and replay attack as follows.

- **Man-in-the-middle attacks**: In our protocol, supposing that the adversary obtains the real identity of user and edge device successfully. The adversary chooses two random number w_1, w_2 and calculates $g_{U_i} = g^{w_1}$, $g_{ED_j} = g^{w_2}$. However, the adversary can not calculates $\varphi_{U_i}$and φ_{ED_j}without the private key of user and edge devices. Thus, the legitimate authentication information can not be forged.
- **Replay attacks**: The authentication information contains the timestamp, so the participants can resist the replay attacks according to the freshness of the timestamp.
- **Stolen verifier table attacks**: In our protocol, neither the user nor the edge devices need to generate a verifier table to store authentication message, they only need to store the private key and session key of their own. The adversary can not obtain the authentication message among the user, edge devices and cloud server by using stolen verifier table attack.

7. Performance Analysis

In this section, we analyze the performance of the protocol from the following two aspects: computation cost and communication cost.

7.1. Computation Cost

In this subsection, we analyze the computation cost of our protocol. For convenience, we define some execution time notations as follows.

- T_{exp}: The execution time of exponential operations in multiplicative cyclic group.
- T_{htp}: The execution time of a hash-to-point operation.
- T_h: The execution time of an ordinary hash operation.
- T_{sm}: The execution time of a scalar multiplication operation in additive cyclic group.
- T_{bp} : The execution time of a bilinear pairing operation $e(P,Q)$, where P and Q belong to additive cyclic group.

We compared the computation cost of our protocol with Jia's scheme [19] and Ma's scheme [20]. The execution time of the cryptographic operations in our protocol is completed with the PBC library. Our hardware consists of an Intel(R) Core (TM) i5-9500 CPU with 3.00 GHz clock frequency, 8G memory and runs Window 10 operation system. The execution time of the basic cryptographic operations are listed in Table 2.

The total computation cost of the participants (user U_i, edge device ED_j and cloud server CS) of our scheme, Jia's scheme [19] and Ma's scheme [20] are shown in Table 3.

For our scheme, U_i needs perform six exponential operations, three ordinary hash operations and two hash-to-point operations. ED_j needs perform four exponential operations, two ordinary hash operations and one hash-to-point operation. CS needs perform eleven exponential operations, nine ordinary hash operations and two hash-to-point operations. Therefore, the total execution time of U_i, ED_j and CS is 11.3689ms, 7.5665ms and 20.7795ms.

Table 2. The execution time of cryptographic operations

Cryptographic operation	Execution time (ms)
T_{exp}	1.882
T_{htp}	0.0383
T_h	0.0001
T_{sm}	8.006
T_{bp}	16.064

For the scheme of [19], U_i needs perform six ordinary hash operations, two scalar multiplication operations and one bilinear pairing operation. FN_j needs perform four ordinary hash operations, two scalar multiplication operations and one bilinear pairing operation. CS needs perform eleven ordinary hash operations, three scalar multiplication operations and one bilinear pairing operation. The total execution time of U_i, FN_j and CS is 32.0766ms, 32.0764ms and 40.0831ms.

For the scheme of [20], U_i needs perform four ordinary hash operations and three scalar multiplication operations. Fog node FN_j needs perform four ordinary hash operations and four scalar multiplication operations. CS needs perform eleven ordinary hash operations and ten scalar multiplication operations. Therefore, the total execution time of U_i, FN_j and CS is 24.0184ms, 32.0244ms and 80.0611ms.

Table 3. Comparison of computation cost

Scheme	U_i (ms)	ED_j or FN_j (ms)	CS (ms)
Our scheme	$6T_{\exp}+3T_h+2T_{htp}\approx 11.3689$	$4T_{\exp}+2T_h+T_{htp}\approx 7.5665$	$11T_{\exp}+9T_h+2T_{htp}\approx 7.5665$
Jia's scheme	$2T_{sm}+6T_h+T_{bp}\approx 32.0766$	$2T_{sm}+4T_h+T_{bp}\approx 32.0764$	$3T_{sm}+11T_h+T_{bp}\approx 40.503$
Ma's scheme	$3T_{sm}+4T_h\approx 24.0184$	$4T_{sm}+4T_h\approx 32.0244$	$10T_{sm}+11T_h\approx 80.0611$

Table 4. Comparison of communication cost

Scheme	U_i (bits)	ED_j or FN_j (bits)	CS (bits)
Our scheme	$\lvert G\rvert+2\lvert Z\rvert+\lvert T\rvert=1376$	$6\lvert G\rvert+4\lvert Z\rvert+3\lvert T\rvert=6880$	$3\lvert G\rvert+2\lvert Z\rvert+\lvert T\rvert=3424$
Jia's scheme	$\lvert G\rvert+4\lvert Z\rvert+\lvert T\rvert=1696$	$4\lvert G\rvert+6\lvert Z\rvert+3\lvert T\rvert=5152$	$\lvert G\rvert+4\lvert Z\rvert+\lvert T\rvert=1696$
Ma's scheme	$\lvert G\rvert+4\lvert Z\rvert+\lvert T\rvert=1696$	$6\lvert G\rvert+6\lvert Z\rvert+3\lvert T\rvert=7200$	$3\lvert G\rvert+4\lvert Z\rvert+\lvert T\rvert=3744$

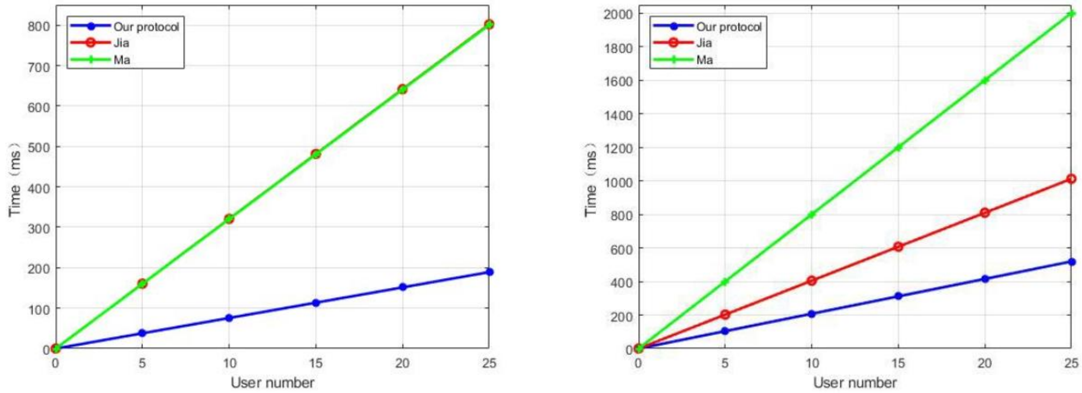

Figure 4. The computation cost of edge device and cloud server increases with the growth of users

The performance evaluation result shows that the computation cost of our scheme on the user, edge device and cloud server is less than [19] and [20]. Compared with [20], our scheme uses less 12.6495, 24.4579 and 59.2816 milliseconds on user, edge device and cloud server. And our scheme uses less 20.7077, 24.5099 and 19.7235 milliseconds than [19] on user, edge device and cloud server.

The computation cost of edge device and cloud server increases with the growth of users as shown in Figure 4. It is worth noting that the computation cost of edge devices and the cloud server increases linearly with the increase of users.

7.2. Communication Cost

In this subsection, we compare the communication cost of our protocol with [19] and [20]. For convenience, let the length values of G, Z_q^* and T_i are expressed as $|G|$, $|Z|$ and $|T|$. The size of G, Z_q^* and T_i are 1024, 160 and 32 bits. The communication cost of our scheme, Jia's scheme and [19] Ma's scheme [20] are shown in Table 4. In authentication and key agreement phase, for our scheme, user needs to transmit the information $\{\varphi_{U_i}, \pi_{U_i}, AID_i, T_{U_i}\}$ to edge device. Edge device needs to transmit the information $\{\varphi_{U_i}, \varphi_{ED_j}, \varphi_{UE}, \pi_{U_i}, \pi_{ED_j}, AID_i, ID_{ED_j}, T_{U_i}, T_{ED_j}\}$ and $\{W_{U_i}, W_{UC}, S_2, T_{edj}\}$ to cloud server and user, respectively. After received the information, cloud server needs to respond the message $\{W_{U_i}, W_{ED_j}, W_{UC}, S_1, S_2, T_{CS}\}$ to edge device. Therefore, the communication cost of user, edge device and cloud server is 1376bits, 6880bits and 3424bits, respectively.

For the scheme of [19], user needs to transmit the information $\{A, PID_i, N_i, T_u\}$ to fog node. Fog node needs to transmit the information $\{A, B, PID_i, N_i, L_j, T_u, T_f\}$ and $\{B, C, Auth_i, T_c\}$ to cloud server and user, respectively. After cloud server verified the message, it needs to respond the message $\{C, Auth_i, Auth_j, T_c\}$ to fog node. So the communication cost of user, fog node and cloud server is 1696bits, 5152bits and 1696bits, respectively.

For the scheme of [20], user needs to transmit the information $\{AID_{U_i}, T_{U_i}, R_1, \alpha\}$ to fog node. Fog node needs to transmit the information $\{AID_{U_i}, AID_{FN_j}, T_{U_i}, T_{FN_j}, R_1, R_2, \hat{R}_2, \bar{R}_2, D_{ID_j}\}$ and $\{R_2, R_3, \hat{R}_3', T_{CS}, \bar{\gamma}\}$ to cloud server and user, respectively. After cloud server verified the message, it needs to respond the message $\{R_3, \hat{R}_2, \hat{R}_2', T_{CS}, \gamma, \bar{\gamma}\}$ to fog node. So the communication cost of user, fog node and cloud server is 1696bits, 7200bits and 3744bits, respectively.

According to the analysis above, our scheme performs better than [19] and [20] on user side. On the edge device side and cloud server side, the communication overhead of our scheme is smaller than [20] and larger than [19]. Although the communication overhead of our scheme is inferior to [19] on edge device side and cloud server side, our scheme is also within the acceptable range due to the strong communication capabilities of the above two.

8. Conclusion

In this paper, an efficient and secure three-party authentication and key agreement protocol for privacy-preserving of IoT devices in MEC is put forward. The scheme realized three-party authentication and key agreement among users, edge devices and cloud server. Then, we prove the security of and analyze the security attributes of the protocol. The security analysis shows that our protocol secure and meets the security attributes such as session key security, forward secrecy. In the end, we evaluate the performance of the protocol, and the evaluation result shows that our protocol is more uperior in terms of computation cost and communication cost.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grants No. U1836115, No. 61922045, No. 61877034, the Natural Science Foundation of Jiangsu Province under Grant No. BK20181408, the CICAEET fund, and the PAPD fund.

References

[1] Z. Lu, G. Qu, Z. Liu, A Survey on Recent Advances in Vehicular Network Security, Trust, and Privacy, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 20, No. 2, pp. 760-776, February, 2019.

[2] R. Hussain, J. Lee, S. Zeadally, Trust in VANET: A Survey of Current Solutions and Future Research Opportunities, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 22, No. 5, pp. 2553-2571, May, 2021.

[3] M. N. Khan, A. Rao, S. Camtepe, Lightweight Cryptographic Protocols for Iot-Constrained Devices: A Survey, *IEEE Internet of Things Journal*, Vol. 8, No. 6, pp. 4132-4156, March, 2021.

[4] G. Mei, N. Xu, J. Qin, B. Wang, P. Qi, A Survey of Internet of Things (iot) for Geohazard Prevention: Applications, Technologies, and Challenges, *IEEE Internet of Things Journal*, Vol. 7, No. 5, pp. 4371-4386, May, 2020.

[5] A. Asheralieva, D. Niyato, Fast and Secure Computational Offloading with Lagrange Coded Mobile Edge Computing, *IEEE Transactions on Vehicular Technology*, Vol. 70, No. 5, pp. 4924-4942, May, 2021.

[6] C. Park, J. Lee, Mobile Edge Computing-Enabled Heterogeneous Networks, *IEEE Transactions on Wireless Communications*, Vol. 20, No. 2, pp. 1038-1051, February, 2021.

[7] M. Abdel-Basset, R. Mohamed, M. Elhoseny, A. K. Bashir, A. Jolfaei, N. Kumar, Energy-Aware Marine Predators Algorithm for Task Scheduling in Iot-Based Fog Computing Applications, *IEEE Transactions on Industrial Informatics*, Vol. 17, No. 7, pp. 5068-5076, July, 2021.

[8] W. Xiang, K. Zheng, D. Niyato, L. Militano, G. Araniti, Editorial of ETT Special Issue: Emerging Topics in Device to Device Communications as Enabling Technology for 5g Systems, *Transactions on Emerging Telecommunications Technologies*, Vol. 28, No. 2, Article No. e3023, February, 2017.

[9] N. Kumar, K. Rawat, F. M. Ghannouchi, Multi-Band All-Digital Transmission for 5g NG-RAN Communication, *2020 IEEE 3rd 5G World Forum (5GWF)*, Bangalore, India, 2020, pp. 497-501.

[10] S. Hu, Y. Xiao, Design of Cloud Computing Task Offloading Algorithm Based on Dynamic Multi-Objective Evolution, *Future Generation Computer Systems*, Vol. 122, pp. 144-148, September, 2021.

[11] J. A. F. F. Dias, J. J. P. C. Rodrigues, L. Zhou, Cooperation Advances on Vehicular Communications: A Survey, *Vehicular communications*, Vol. 1, No. 1, pp. 22-32, January, 2014.

[12] D. He, S. Zeadally, B. Xu, X. Huang, An Efficient Identity-Based Conditional Privacy-Preserving Authentication Scheme for Vehicular Ad Hoc Networks, *IEEE Transactions on Information Forensics and Security*, Vol. 10, No. 12, pp. 2681-2691, December, 2015.

[13] A. K. Sutrala, P. Bagga, A. K. Das, N. Kumar, J. J. P. C. Rodrigues, P. Lorenz, On the Design of Conditional Privacy Preserving Batch Verification-Based Authentication Scheme for Internet of Vehicles Deployment, *IEEE Transactions on Vehicular Technology*, Vol. 69, No. 5, pp. 5535-5548, May, 2020.

[14] J. Cui, J. Zhang, H. Zhong, Y. Xu, SPACF: A Secure Privacy-Preserving Authentication Scheme for VANET with Cuckoo Filter, *IEEE Transactions on Vehicular Technology*, Vol. 66, No. 11, pp. 10283-10295, November, 2017.

[15] J. Zhou, Z. Cao, Z. Qin, X. Dong, K. Ren, LPPA: Lightweight Privacy-Preserving Authentication from Efficient Multi-Key Secure Outsourced Computation for Location-based Services in VANETs, *IEEE Transactions on Information Forensics and Security*, Vol. 15, pp. 420-434, June, 2019.

[16] T. Lee, J. Liu, M. Sung, S. Yang, C. Chen, Communication-Efficient Three-Party Protocols for Authentication and Key Agreement, *Computers & Mathematics with Applications*, Vol. 58, No. 4, pp. 641-648, August, 2009.

[17] C. Lv, M. Ma, H. Li, J. Ma, Y. Zhang, An Novel Three-Party Authenticated Key Exchange Protocol Using One-Time Key, *Journal of Network and Computer Applications*, Vol. 36, No. 1, pp. 498-503, January, 2013.

[18] S. Chiou, C. Lin, An Efficient Three-Party Authentication Scheme for Data Exchange in Medical Environment, *Security and Communication Networks*, Vol. 2018, pp. 9146297:1-9146297:15, January, 2018.

[19] X. Jia, D. He, N. Kumar, K. R. Choo, Authenticated Key Agreement Scheme for Fog-Driven Iot Healthcare System, *Wireless Networks*, Vol. 25, No. 8, pp. 4737-4750, November, 2019.

[20] M. Ma, D. He, H. Wang, N. Kumar, K. R. Choo, An Efficient and Provably Secure Authenticated Key Agreement Protocol for Fog-Based Vehicular Ad-Hoc Networks, *IEEE Internet of Things Journal*, Vol. 6, No. 5, pp. 8065-8075, October, 2019.

[21] P. Bagga, A. K. Das, M. Wazid, J. J. P. C. Rodrigues, K. R. Choo, Y. Park, On the Design of Mutual Authentication and Key Agreement Protocol in Internet of Vehicles-Enabled Intelligent Transportation System, *IEEE Transactions on Vehicular Technology*, Vol. 70, No. 2, pp. 1736-1751, February, 2021.

[22] M. Wazid, A. K. Das, N. Kumar, A. V. Vasilakos, J. J. P. C. Rodrigues, Design and Analysis of Secure Lightweight Remote User Authentication and Key Agreement Scheme in Internet of Drones Deployment, *IEEE Internet of Things Journal*, Vol. 6, No. 2, pp. 3572-3584, April, 2019.

[23] S. Mukherjee, D. S. Gupta, G. P. Biswas, An Efficient and Batch Verifiable Conditional Privacy-Preserving Authentication Scheme for Vanets Using Lattice, *Computing*, Vol. 101, No. 12, pp. 1763-1788, December, 2019.

[24] Y. Zhou, X. Long, L. Chen, Z. Yan, Conditional Privacy-Preserving Authentication and Key Agreement Scheme for Roaming Services in VANETs, *Journal of Information Security and Applications*, Vol. 47, pp. 295-301, Augest, 2019.

[25] I. Ali, F. Li, An Efficient Conditional Privacy-Preserving Authentication Scheme for Vehicle-to-Infrastructure Communication in VANETs, *Vehicular Communications*, Vol. 22, Article No. 100228, April, 2020.

[26] S. Tzeng, S. Horng, T. Li, X. Wang, P. Huang, M. K. Khan, Enhancing Security and Privacy for Identity-Based Batch Verification Scheme in VANETs, *IEEE Transactions on Vehicular Technology*, Vol. 66, No. 4, pp. 3235-3248, April, 2017.

[27] L. Zhang, Q. Wu, J. Domingo-Ferrer, B. Qin, C. Hu, Distributed Aggregate Privacy-Preserving Authentication in VANETs, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 18, No. 3, pp. 516-526, March, 2017.

[28] H. Zhong, S. Han, J. Cui, J. Zhang, Y. Xu, Privacy-Preserving Authentication Scheme with Full Aggregation in VANET, *Information Sciences*, Vol. 476, pp. 211-221, February, 2019.

[29] M. Han, L. Hua, S. Ma, A Self-Authentication and Deniable Efficient Group Key Agreement Protocol for VANET, *arXiv preprint arXiv: 1611.09009*, November, 2016.

[30] Y. Jiang, S. Ge, X. Shen, AAAS: An Anonymous Authentication Scheme Based on Group Signature in VANETs, *IEEE Access*, Vol. 8, pp. 98986-98998, May, 2020.

[31] H. Hasrouny, A. E. Samhat, C. Bassil, A. Laouiti, Trust Model for Secure Group Leader-Based Communications in VANET, *Wireless Networks*, Vol. 25, No. 8, pp. 4639-4661, November, 2019.

Biographies

Sai Ji received his M.S. degree from the Nanjing Aeronautics and Astronautics University (NUAA), Nanjing, China, in 2006. He works as an Associate Professor at the NUIST. His research interests are in the areas of structural health monitoring, and WSNs. Ji has published more than 20 journal/conference papers.

Yang Yuan received the BS degree, in 2019. He is currently working toward the ME degree at the Nanjing University of Information Science and Technology, Nanjing, China. His current research interests include authentication and privacy preserving in VANETs.

Jian Shen received the Ph.D. degree in computer science from Chosun University, South Korea, in 2012. Since 2012, he has been a Professor in the School of Computer and Software at NUIST. His research interests include public cryptography, cloud computing and security, and data auditing and sharing.

Chin-Feng Lai (SM'14) received the Ph.D. degree in engineering science from National Cheng Kung University, Tainan,Taiwan, in 2008. Since 2016, he has been an Associate Professor of Engineering Science, National Cheng Kung University, Tainan. His research focuses on Internet of Things, body sensor networks, e-healthcare, mobile cloud computing, etc.

Bofan Chen received the Ph.D. degree in Automation (English teaching) from the School of Automation, Beijing Institute of Technology in 2018. Her research focuses on electronic circuits, image processing, signal processing and soft direction.