An Effectively Applicable to Resource Constrained Devices and Semi-Trusted Servers Authenticated Key Agreement Scheme

Dong Xie, Jinghua Yang, Bin Wu, Weixin Bian, Fulong Chen, and Taochun Wang

***Abstract*— In a mobile edge computing environment, the computing tasks of resource-constrained IoT devices are often offloaded to mobile edge computing servers for processing. In order to ensure the security of the task offloading process, both parties need to perform mutual authentication and negotiate a session key first. The security defenses in the existing authentication schemes are often only aimed at external attackers, while ignoring the possible malicious behaviors of semi-trusted servers. Furthermore, they cannot effectively take into account the device-side lightweight and security, as well as the load problem of a single registry. In this paper, we propose a new anonymous authentication key agreement scheme that fully considers the resource constraints of terminal devices and the security risks of semi-trusted servers. In the scheme, we use the method of generating pairing information during registration to avoid the server-side directly contacting the user's private information, and support trusted third parties not to participate in the authentication process. In addition, by setting up authentication servers to outsource computing tasks, the device-side can avoid blindly selecting a computing server for task offloading, achieve accurate task assignment and efficient execution of authentication. We use Real-Or-Random model and BAN logic to demonstrate the security of the proposed scheme, and use the ProVerif tool to verify its authenticated reachability and confidentiality. Compared with other schemes with the same structure, this scheme is superior to similar schemes, and has higher security on the basis of ensuring the least amount of computation on the device-side.**

***Index Terms*— Resource-constrained IoT devices, semi-trusted servers, anonymous authentication key agreement, BAN logic, provable security.**

I. Introduction

INTERNET of Things (IoT) terminal equipment (TE) are usually resource-constrained in terms of processor and memory capacity, so they usually needs to offload computing tasks of massive data [1], [2]. Mobile Edge Computing (MEC) perfectly meets the needs of TE [3]. Deploying the MEC server close to users and devices, TE offloads computing tasks to the MEC server for processing, which not only improves the processing speed, but also reduces the transmission delay [4]. During the offloading of tasks, there is no shortage of private information and confidential data of users or enterprises. However, the transmission of IoT data generally uses public channel, important and sensitive information transmitted in such channel are easily leaked [5], [6], [7]. Additionally, IoT devices are often primary targets for hackers owing to their lax security features [8], [9]. At the same time, legitimate MEC servers may also become attackers to defraud TE's data [10]. Therefore, there is an urgent need to ensure the legitimacy of the identities of task offloading participants through mutual authentication mechanisms, as well as to ensure the secure transmission of data through the use of temporary session key.

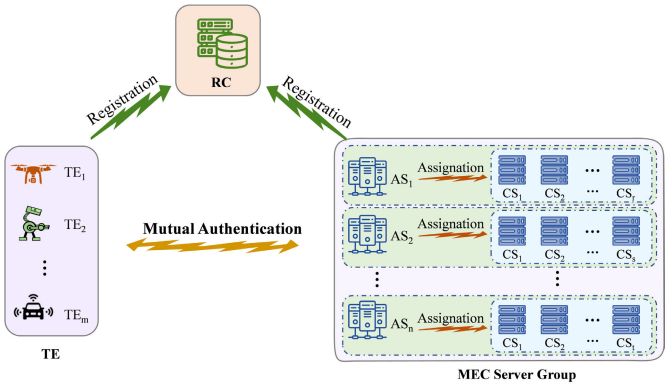


Fig. 1. The authentication framework between TE and MEC.

A. Motivation

1) Gap Analysis: Authentication key agreement (AKA) schemes with a trusted third party can achieve mutual authentication and negotiate a secure session key between the authentication parties (see Fig. 1). If this trusted third party needs to participate in many steps of authentication process, it will result in a huge communication and computing burden. Therefore, it is a consensus that the trusted third party only participate in the offline registration step in the field of AKA schemes. This mode can be easily implemented when the trusted third party distributes public-private key pair to each user in advance [11], [12], [13], [14]. Evidently, this authentication mode of using public-private key pairs is

Manuscript received 12 October 2022; revised 27 June 2023 and 7 October 2023; accepted 30 January 2024. Date of publication 5 February 2024; date of current version 14 February 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 61801004 and Grant 61972438, in part by the Natural Science Foundation of Anhui Province of China under Grant 2108085MF219 and Grant 2108085MF206, and in part by the Key Research and Development Projects in Anhui Province under Grant 202004a05020002 and Grant 2022a05020049. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mika Ylianttila. *(Corresponding author: Dong Xie.)*

The authors are with the School of Computer and Information, Anhui Normal University, Wuhu 241002, China (e-mail: xiedong@ahnu.edu.cn).

Digital Object Identifier 10.1109/TIFS.2024.3362589

not suitable for resource-constrained IoT devices. In recent years, cryptographic researchers have attempted to design efficient and secure AKA schemes based solely on symmetric encryption and hash functions [15], [16], [17], [18]. In these schemes, the terminal equipment need to provide their own private information to the servers for proving their identity. By encrypting these private information, they can be prevented from being obtained by external attackers. But the designers have *ignored* the credibility of internal participants, just like in real life, we do not know what purpose the service providers will use our private data for. For server that can directly obtain private information and long-term secret values, it can also pose a threat to users just like an adversary. Therefore, the security requirements of AKA schemes should not only target external attackers, but also consider the trustworthiness of internal participants.

In the authentication process of traditional AKA schemes, the user directly initiates an authentication request to the server, and the target server responds to the user's request. This authentication framework seems to be normal theoretically. However, in practical applications, users who need to offload computing tasks *cannot* know whether the target server they have selected is busy, or whether the target server is good at processing computing tasks. Blindly selecting the target server will inevitably reduce the efficiency and the response speed.

2) Problem Statement: Although the complex operation of public key cryptography is avoided in the existing non-public key AKA schemes, lightweight solutions do not necessarily mean that security needs to be reduced. As far as we know, existing solutions do not achieve a good balance between security and lightweight. In several AKA schemes without a public key, the trusted registration center (RC) must be participated in the authentication process. Obviously, this approach greatly increases the burden on the registration center, and the frequent use of secure channels also increases the risk of the system being compromised. However, in the existing non-public key AKA schemes that supports offline registration, the pairing issue for authentication has not been solved. In addition, there are many security issues in the system, such as device privacy being stolen, server keys being leaked, and system parameters being cracked.

B. Objective and Methodology

The goal of this paper is to propose a new applicable AKA scheme for resource constrained devices and semi-trusted servers. To achieve this goal, the methods used are as follows.

To alleviate the burden of RC and avoid frequent use of secure channels, we have established a distributed RC architecture (See Fig. 2). To prevent the semi-trusted server from stealing the user's private information, RC generates a matching list of the corresponding server for each TE during the registration phase.

For the problem of accurately selecting the best target server, we divide the servers into authentication server (AS) and computing server (CS) according to their functions. For authentication, the TE first sends an authentication message to the AS. After the AS completes the verification, it selects an idle CS that meets the requirements in the server group, and assigns the CS to the TE, so as to achieve server load balancing (See Fig. 1). Because the number of AS is much smaller than that of CS, another advantage of this approach is that it can effectively reduce the length of the pairing message list pre-generated by the RC for the user.

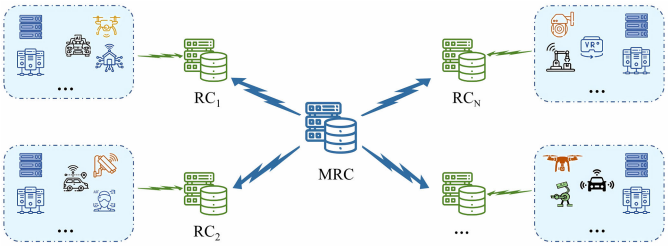


Fig. 2. The architecture of distributed RC.

To improve the authentication efficiency of resource-constrained TE devices, we use a secure and fast hash function instead of complex ECC operation. In addition, we try to simplify the authentication process as much as possible, i.e., deleting all the non-essential operations outside the process.

C. Contribution

The main contributions of this work can be summarized as follows.

- *Distributed Registration Center*. For the workload and security risk of a single sign-on, we adopt a multi-registry distributed approach to reduce the burden and risk of the registration process.
- *Server Function Division*. To improve the authentication efficiency and task processing speed, we classify servers by function. The authentication server AC is responsible for authenticating the user's identity, and the computing server CS is responsible for completing key negotiation and task calculation. Servers authenticate each other through certificates issued by RC.
- *Secure Pairing*. For the threat brought by the semi-trusted server in the non-public key authentication schemes, we use the offline registration center to generate pairing information to prevent the server from stealing the user's sensitive information.
- *Lightweight Formal Authentication.* We reduce the complex calculation on the TE side by using hash and Xor operations to meet the lightweight requirements of resource-constrained devices. For the server side, we place the operations between servers (e.g., certificate authentication, entity joining, and entity exiting the server group) outside the formal authentication.
- *Security Analysis*. For the theoretical security proof, we prove the security of the temporary session key using Real-Or-Random (ROR) and the reachability of the authentication process using Burrows-Abadi-Needham (BAN) logic. Also, we use the ProVerif verification tool to verify the security of the protocol. Last, we collate general standards and security requirements from related works, and conduct informal security analysis of the proposed scheme.

TABLE I
COMPARISON OF RELATED WORKS

Properties	[19]	[20]	[21]	[22]	[17]	[18]	[23]	[24]	[25]	[26]	[27]	[28]	[29]	[30]	[31]	[32]	[33]
P1	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y
P2	N	N	N	Y	N	N	Y	Y	Y	N	Y	N	N	Y	Y	N	Y
P3	Y	Y	Y	Y	Y	N	N	Y	Y	Y	N	Y	Y	Y	Y	Y	Y
P4	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
P5	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
P6	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
P7	N	N	N	N	N	N	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
P8	Y	Y	Y	Y	N	N	Y	N	Y	N	N	Y	Y	Y	Y	Y	Y
P9	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
P10	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
P11	Y	Y	Y	N	Y	Y	N	Y	Y	Y	Y	N	Y	Y	Y	Y	Y
P12	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
P13	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
P14	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
P15	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
P16	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y
P17	N	N	N	N	N	N	Y	N	N	N	N	Y	Y	Y	Y	Y	Y
P18	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	Y	Y	Y	N	Y	N
P19	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

P1: User anonymity; P2: Conditions privacy protection; P3: Untraceability; P4: Use trusted registration center; P5: Offline registration center; P6: Only use Hash and Xor operations; P7: Lightweight authentication; P8: Only one interaction; P9: Server function division; P10: Security of TE's secret value; P11: Security of server's key; P12: Resistant to TE impersonation attack; P13: Resistant to server spoofing attack; P14: Resistant semi-trusted server attack; P15: Resistant to man-in-the-middle attack; P16: Resistant to replay attack; P17: Resistant to TE data theft attack; P18: Resistant to insider attack; P19: Resistant to session key attack.

In general, the proposed AKA scheme does not weaken its security while ensuring the efficiency of the authentication process. In addition, the security risk brought by the semi-trusted server can also be resisted. Table I shows the comparison between the proposed scheme and many existing related works in terms of common security properties.

D. Organization

The remaining parts of this paper is organized as follows. In Section II, we classify and analyze the existing related literature. We introduce the system model in Section III. We describe the detail of the proposed scheme in Section IV. In Section V, we conduct formal and informal security analyses of the proposed scheme. Section VI shows the performance evaluation, and the paper is concluded in Section VII.

II. RELATE WORKS

There are mainly two types of encryption methods used in the current popular AKA schemes. One is to use elliptic curve cryptography (ECC) based public key encryption algorithms, and the other is to use lightweight hash function, Xor operation and symmetric encryption.

Among public key encryption algorithms, ECC can use shorter keys to achieve higher security. However, for the lightweight authentication of resource-constrained IoT terminal devices, the amount of computation involved in ECC is still large. In this regard, researchers try to use hash operation for authentication. Garg et al. [19] proposed an authentication scheme for smart meters and neighborhood network gateways. In the scheme, ECC based public key encryption algorithm and hash operation are combined for reducing the computation overhead. It can be viewed as a transition scheme for lightweight authentication. Ma et al. [20] designed an AKA scheme applied to fog computing. Zhang et al. [21] designed a many-to-many AKA scheme that can be authenticated simultaneously between multiple vehicles and service providers. Sutrala et al. [22] proposed a mutual AKA scheme for devices in industrial cyber-physical systems. The common feature of the above-mentioned AKA schemes is that the user side needs to perform ECC point multiplication, whose computational complexity is much greater than that of hash operations.

To further reduce the authentication overhead, researchers try to do not use ECC and design more lightweight authentication schemes. In 2020, Umar et al. [17] proposed a lightweight AKA scheme between vehicles and roadside units, which requires a trusted third party to perform most of operations during the authentication process. This can place a significant burden on the trusted third party. In addition, the scheme lacks user anonymity, and cannot resist impersonation attacks and replay attacks. In the same year, Vinoth et al. [18] proposed an AKA scheme for industrial IoT. They used secret sharing and Chinese remainder theorem to construct a group key between data providers, and then used such key to negotiate the session key. Like Umar et al.'s scheme, it also does not support offline registration and lacks user anonymity, and the communication process is cumbersome. Zhang et al. [23] proposed an AKA scheme for Internet of Drones. There is a security problem with secret values of users in the scheme. Li et al. [24] proposed a lightweight AKA scheme for wearable devices. There is no trusted third party to provide system security, and the scheme has several security risks, such as impersonation attacks and insider attacks. In this scenario, Guo et al. [25] also proposed a scheme. With the help of the cloud server, the wearable device and the user, also the user and the cloud server are mutually authenticated. In 2021, Li et al. [26] proposed a lightweight AKA scheme for Internet of Medical Things. The scheme does not consider internal attacks initiated by the corrupted sensor node, and each user is also not anonymous to sensor nodes. In 2022, Sureshkumar et al. [27] proposed

a lightweight AKA scheme between vehicles and charging stations in Internet of Vehicles scenario, assuming that the service provider is a trusted third party. All of the above solutions require a trusted third party to participate in the authentication process. It not only increases the burden on trusted third parties, but also reduces the response speed and the flexibility of authentication. It is very unfriendly to mobile and resource-constrained TEs.

In 2020, Zhao et al. [28] designed an AKA scheme between users and servers in a multi-server environment. The scheme not only ensures the communication security, but also ensures the physical security of the smart card and biometric information. However, there is a problem in the design of the private key of the server, and internal users can easily obtain the private key of the corresponding server. Zhang et al. [29] designed an AKA scheme between users and medical servers, which protects users' biometric information through a dynamic privacy protection mechanism. However, it is not secure to directly provide private biometric information for the server as authentication credentials. Both Shen et al. [30] and Bansal et al. [31] design AKA schemes between vehicles and smart grids. The scheme links the vehicle to the grid through an intermediary called the aggregator. Wang et al. [32] pointed out that Kumar et al.'s scheme [33] may encounter trace attacks, session-specific temporary information attacks, and key compromise impersonation attacks. The authentication process of these schemes does not require the participation of a trusted third party. But users need to transmit secret parameters or private information to prove their identity to the server. For a comparison of these relevant works described above, see Table I.

III. Preliminaries

A. System Model

The system model of the proposed scheme is shown in Fig. 1. There are three types of participants, i.e., RC, TE and MEC server group.

1) RC: The trusted third party is responsible for the offline registration. Note that the third party interacts with all other entities through secure channels. We built a distributed RC architecture as shown in Fig. 2. Among them, all the sub-registration centers RC_i are authorized by the main registration center (MRC). Sub-registration centers are deployed in different places. Each RC has the same functions and powers as the MRC, so as to achieve load balancing of the registration process.

2) TE: It includes mobiles and resource-constrained IoT devices, such as small drones, high-definition cameras, and wearable devices. They need to complete data acquisition and analysis independently.

3) MEC Server Group: We divide the servers in each group into AS and CS.

AS mainly has the following functions.

- Receive and authenticate the newly added CS with a certificate.
- Complete the authentication and the negotiation process of the TE by verifying the login message.
- Through TE's offload description, decide and designate an appropriate CS to interact with the TE.

The main functions of CS are as follows.

- Authenticate the AS and supervise the behavior of the AS.
- Complete mutual authentication and session key negotiation with the assigned TE.
- Process the computing tasks offloaded by the TE and give rensponse.

B. Threat Model

We assume that an adversary $\mathcal{A}$ in the scheme has the same capabilities as the adversary in the widely adopted Dolev-Yao (DY) threat model. Device users and servers are considered insecure and any entity transmits information on public channels. An adversary can eavesdrop, intercept, modify, and delete messages transmitted between any two entities. It is even possible to inject fake messages to deceive legitimate entities. Furthermore, the adversary can impersonate the behavior of the communication entities TE_i, AS_j and CS_k.

The RC is the most important part of the AKA protocol. It is considered to be completely trusted and cannot be compromised by the adversary, and any information stored on the RC will not be leaked to the adversary.

C. Security Requirements

- *Mutual Authentication.* It realizes the mutual identity authentication between TE and the server, and ensure the legitimacy of the authenticated parties.
- *Key Agreement.* The session keys can be securely established between the two authenticated parties.
- *Key Security.* The key negotiated by the authenticated parties is secure and can not be leaked.
- *User Anonymity.* The real identity of TE will not be disclosed during the authentication process.
- *Resist Various Attacks.* The designed protocol is resistant to various attacks from internal and external sources.

IV. Proposed Scheme

In this section, the details of the proposed identity-based AKA scheme are explained. It is composed of four phases, including setup phase, TE and servers registration, authentication preparation, TE login and mutual authentication key agreement. For simplicity, Table II provides the list of notations used in the this paper.

A. Setup Phase

RC needs to perform the following initialization operations when starting the protocol.

- Choose two long-term secret values P and Q.
- Choose a collision-resistant one-way hash function $h()$.
- Calculate the values of $h(Q)$ and $h(P||Q)$.

After, RC publishes hash function $h()$, as well as keeps P, Q, $h(Q)$ and $h(P||Q)$ secretly, and safely pass these secret values to other distributed RCs.

TABLE II
NOTATIONS

Notations	Description
RC	Registration Center
TE	Terminal Equipment
AS	Authentication Server
CS	Computing Server
P, Q	Two long-term secret values of RC
ID_X	Identity information of entity X
SK_X	Private key of entity X
PK_X	Public key of entity X
CT_X	Certificate issued by RC for entity X
AU_X	Authentication message generated by entity X
AS_j	Internal serial code assigned to AS
PW_{TE}	Password of TE
R_{TE}, x_1, x_2, y	Random value (256 bits)
SV_1, SV_2	Secret Value
ARG_1, ARG_2	Temporary Parameter
$M_i (i = 1, 2, 3, 4, 5, 6)$	Temporary message
T_1, T_2, T_3	Timestamp
Δt	Upper limit of the timestamp difference
$\oplus$	Bitwise exclusive-or operation
$\|\|$	Concatenation operation
$E(sk, m)$	Encrypt message m with secret key sk
$D(sk, c)$	Decrypt ciphertext c with secret key sk
$\mathcal{A}$	Adversary

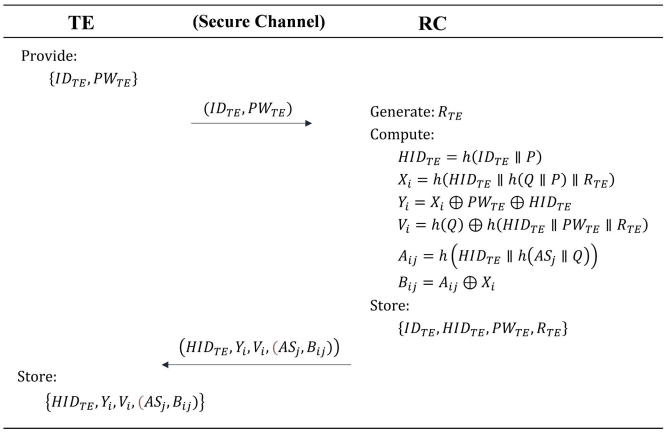


Fig. 3. TE registration.

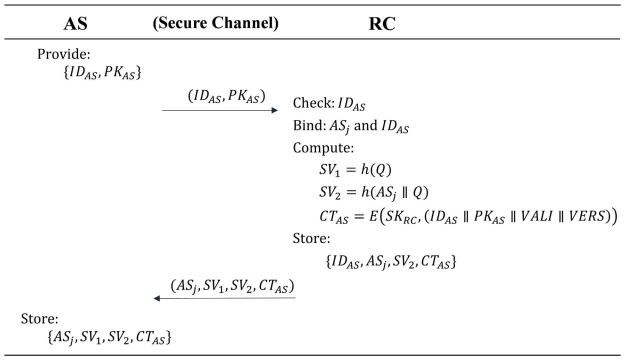


Fig. 4. AS registration.

B. Registration Phase

All authentication entities in the system need to be registered in the RC.

1) TE Registration: The TE provides the RC with its own identity ID_{TE} and password PW_{TE}. After the RC receives the registration request from the TE, it starts to perform the registration operation. First, an R_{TE} is randomly selected for the TE, and then $HID_{TE} = h(ID_{TE}||P)$ is calculated to hide the real identity of the TE. Then RC computes $X_i = h(HID_{TE}||h(Q||P)||R_{TE})$, $Y_i = X_i \oplus PW_{TE} \oplus HID_{TE}$, $V_i = h(Q) \oplus h(HID_{TE}||PW_{TE}||R_{TE})$, where X_i prepares to protect A_{ij}, Y_i is used to deliver X_i, and V_i is used to deliver the secret value $h(Q)$. Finally, calculates $A_{ij} = h(HID_{TE}||h(AS_j||Q))$ and $B_{ij} = A_{ij} \oplus X_i$, where A_{ij} is the paired message list between the TE and each area authentication server AS_j, and B_{ij} is used to transmit A_{ij}. RC returns $\{HID_{TE}, Y_i, V_i, (AS_j, B_{ij})\}$ to TE after the computation is completed. The above process is shown in Fig. 3.

2) AS Registration: The AS provides its own identity ID_{AS} and public key PK_{AS} to the RC. After the RC receives the registration request, it first strictly verifies the identity information of the AS. Then RC assigns an authentication server code AS_j to it. Then RC computes $SV_1 = h(Q)$, $SV_2 = h(AS_j||Q)$, and generates a certificate $CT_{AS} = E(SK_{RC}, (ID_{AS}||PK_{AS}||VALI||VERS))$, where SK_{RC} is the private key of the RC, $VALI$ is the validity period of the certificate, and $VERS$ is the version number. Finally the RC returns the message $\{AS_j, SV_1, SV_2, CT_{AS}\}$ to the registrant. The aforementioned process is described in Fig. 4.

3) CS Registration: The CS provides its own identity ID_{CS} and public key PK_{CS} to the RC. After the RC receives the registration request, it first strictly checks the identity information. Then it generates a certificate $CT_{CS} = E(SK_{RC}, (ID_{CS}||PK_{CS}||VALI||VERS))$. Note that the certificate format of CS is the same as that of AS certificate. Finally the RC returns the CS certificate CT_{CS} to the registrant. The process is shown in Fig. 5.

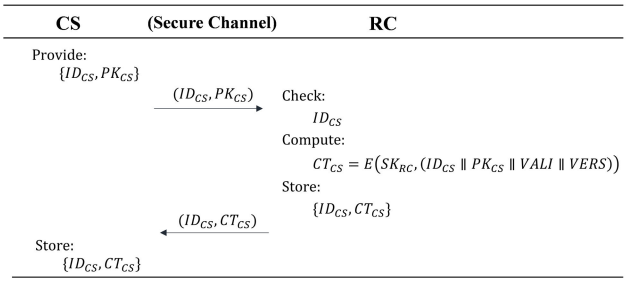


Fig. 5. CS registration.

C. Authentication Preparation Phase

Note that the computing server CS is only responsible for completing the session key negotiation with the TE assigned by the AS. Therefore, it will not directly touch TE's private information. And it can opt in and out of its own accord. If the CS chooses to join, it needs to provide the certificate issued by RC to AS. Then AS checks the certificate. The purpose of doing this is to verify the identity of CS and obtain its public key. Similarly, the CS can also authenticate the AS. They then negotiate a symmetric key SK_{jk} for subsequent communication (See Fig. 6). To prevent attackers from cracking the key, it needs to be changed periodically. Assuming that CS does not want to participate in the TE authentication process for other reasons, it only needs to apply to the AS for revocation.

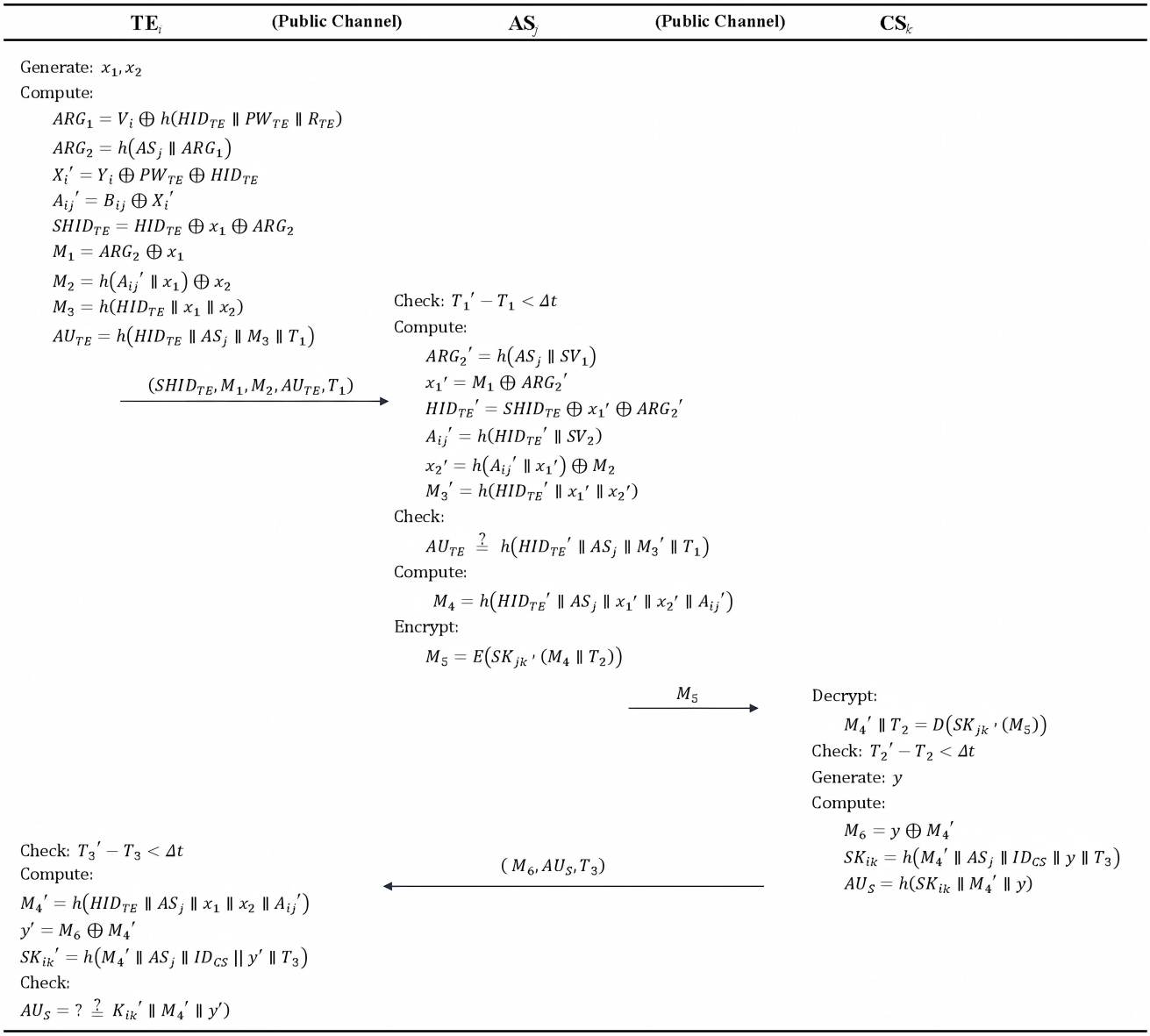


Fig. 6. The proposed authentication key agreement scheme.

AS agrees and it's done. It should be noted that, the above process is the preprocessing step of the proposed scheme.

D. Authentication Key Agreement Phase

In this phase, TE and servers (AS and CS) authenticate each other and negotiate a common temporary session key for further communication. From the registration process of TE, it can be seen that TE knows which authentication server RC has chosen for it. Thus, each TE knows its matched AS_j. We use the subscripts i, j, and k to represent the device labels of TE, AS, and CS, respectively. For convenience, we will not rewrite the subscripts when describing entities. The login and mutual authentication with key agreement phases are summarized in Fig. 6.

1) TE Login: TE is the initiator of the authentication process, and the login steps are as follows:

Step 1: TE selects two random numbers x_1 and x_2 as secret values for this authentication.

Step 2: TE first computes $ARG_1 = V_i \oplus h(HID_{TE}||PW_{TE}||R_{TE})$. Taking ARG_1 and AS_j, it obtains $ARG_2 = h(AS_j||ARG_1)$.

Step 3: TE calculates $X_i{}' = Y_i \oplus PW_{TE} \oplus HID_{TE}$. Then it obtains the pairing message $A_{ij}{}'$ between the TE and the target server AS_j from B_{ij}, $A_{ij}{}' = B_{ij} \oplus X_i{}'$.

Step 4: TE calculates $SHID_{TE} = HID_{TE} \oplus x_1 \oplus ARG_2$, which is used to hide the real value of HID_{TE}.

Step 5: TE computes $M_1 = ARG_2 \oplus x_1$, which is used to hide the secret value x_1. Then, it calculates $M_2 = h(A_{ij}{}'||x_1) \oplus x_2$ for hiding the secret value x_1. Last, it calculates $M_3 = h(HID_{TE}||x_1||x_2)$, $AU_{TE} = h(HID_{TE}||AS_j||M_3||T_1)$.

Step 6: TE sends $(SHID_{TE}, M_1, M_2, AU_{TE}, T_1)$ to AS via a public channel, where T_1 is the current timestamp.

2) Authentication and Key Agreement: After receiving the login message from the TE, the target AS and CS will perform the following steps to authenticate the TE.

Step 1: The AS validates timestamp T_1 by checking $T_1{}' - T_1 < \Delta t$, where Δt is the upper limit of tolerable delay.

If it does not hold, authentication is aborted. Otherwise, the next step proceeds.

Step 2: The AS calculates $ARG_2' = h(AS_j||SV_1), x_1' = M_1 \oplus ARG_2'$, and $HID_{TE}' = SHID_{TE} \oplus x_1' \oplus ARG_2'$.

Step 3: The target AS computes $A_{ij}' = h(HID_{TE}'||SV_2)$ with its unique key SV_2. Then the server can calculate $x_2' = h(A_{ij}||x_1') \oplus M_2$ to obtain another secret value x_2' of TE.

Step 4: After obtaining HID_{TE}' and the secret values x_1' and x_2' from the TE, the AS can verify whether $AU_{TE} = h(HID_{TE}'||AS_j||M_3'||T_1)$ holds, where $M_3' = h(HID_{TE}'||x_1'||x_2')$. If the equation holds, the authentication proceeds.

Step 5: The AS packs the obtained secret parameters to calculate $M_4 = h(HID_{TE}'||AS_j||x_1'||x_2'||A_{ij}')$, and decides the target CS which the computing task can be assigned.

Step 6: M_4 and T_2 is encrypted using the prepared symmetric key SK_{jk}, i.e., $M_5 = E(SK_{jk}, (M_4||T_2))$.

Step 7: The AS sends M_5 to the corresponding CS.

Step 8: When the designated CS receives the message, it uses the symmetric key SK_{jk} to decrypt and obtain M_4' and T_2.

Step 9: The CS validates timestamp T_2 by checking $T_2' - T_2 < \Delta t$. If satisfied, it proceeds to the next step. Otherwise, it terminates.

Step 10: The CS chooses a random number y and compute $M_6 = y \oplus M_4'$ to hide y.

Step 11: The CS generates the session key by calculating $SK_{ik} = h(M_4'||AS_j||y||T_2)$, and compute $AU_s = h(SK_{ik}||M_4'||y)$ for verifing the consistency of the session key.

Step 12: The CS returns the message $\{M_6, AU_s, T_3\}$ to TE.

Step 13: TE receives the message returned from the CS_k, and validates timestamp T_3 by checking $T_3' - T_3 < \Delta t$. If so, proceed. Otherwise, stop.

Step 14: TE calculates $M_4' = h(HID_{TE}||AS_j||x_1||x_2||A_{ij}')$ and $y' = M_6 \oplus M_4'$.

Step 15: TE computes $h(M_4'||AS_j||y'||T_2)$ to obtain the session key SK_{ik}'.

Step 16: TE calculates $h(SK_{ik}'||M_4'||y')$, and verifies whether it is eqaul to AU_s. If yes, this round of verification is completed, and the TE accepts the session key.

V. Security Analysis

In this section, we conduct formal and informal security analyse of the proposed scheme. Suppose that both the used encryption algorithm and the hash function are secure. First we prove the security of the temporary session key under the ROR model and the reachability of the authentication process using BAN logic. Then the proposed scheme is formally verified using the ProVerif tool. In addition, we also provide informal security analysis in this paper.

A. Formal Security Proof

In this subsection, we perform a formal security analysis of the protocol under the ROR model. The main purpose of this analysis is to demonstrate that our protocol has session key security. The ROR model comprises the following components.

1) Participants: There are mainly three types of participants in the authentication process, including TE, AS and CS. Each participant may run various instances, and the instances of participants are also known as oracles. Let $\Pi_{\mathrm{TE}}^{t_1}$, $\Pi_{\mathrm{AS}}^{t_2}$ and $\Pi_{\mathrm{CS}}^{t_3}$ be the t_1, t_2 and t_3 instances of TE, AS and CS, respectively. For simplicity, we sometimes use Π^t to represent instances of TE, AS, and CS.

2) Partnering: Similar to [18] and [25], the session identification (sid) also be used to define the notion of partnering. Two instances Π^{t_1} and Π^{t_2} are said to be partnered if the following three conditions hold. First, both Π^{t_1} and Π^{t_2} have accepted. Second, both Π^{t_1} and Π^{t_2} share the same sid. Last, both Π^{t_1} and Π^{t_2} are partners of each other.

3) Freshness: If the session key of the instance Π^t is not revealed to an adversary $\mathcal{A}$, Π^t can be deemed as freshness.

4) Adversary: The adversary $\mathcal{A}$ can eavesdrop, modify, fabricate and inject messages over the communication channel. The interaction between $\mathcal{A}$ and the protocol participants is done only through oracle queries. The ability of $\mathcal{A}$ is simulated by querying oracles. $\mathcal{A}$ can perform the following queries:

$Hash(m)$: When $\mathcal{A}$ submits a hash query with message m, $\mathcal{C}$ first checks whether the tuple $(m, h(m))$ exists in the hash list. If it exists, it returns $h(m)$. Otherwise, $\mathcal{C}$ randomly selects r, and set $h(m) = r$. While returning r to $\mathcal{A}$, add the tuple $(m, h(m))$ to the hash list.

$Execute(\Pi_{TE}^{t_1}, \Pi_{AS}^{t_2}, \Pi_{CS}^{t_3})$: This query models passive eavesdropping attacks. $\mathcal{A}$ performs this query to obtain all the messages interacted among honest participants TE, AS, and CS.

$Send(\Pi^t, Msg)$: This query models active attacks initiated by $\mathcal{A}$, take replay attacks for example. When $\mathcal{A}$ initiates this query, a message Msg is sent to the participant instance Π^t, and receives a response message from this instance.

$Reveal(\Pi_X^t)$: This query simulates the ability of $\mathcal{A}$ to obtain a session key. Through this query, $\mathcal{A}$ can obtain the temporary session key SK_{ik} created by the instance Π_X^t and its partner.

$CorruptTE(\Pi_{TE}^{t_1})$: This query simulates the attack of stealing TE's storage information. $\mathcal{A}$ can obtain sensitive data stored in TE.

$CorruptCS(\Pi_{CS}^{t_3})$: This query simulates the attack of stealing CS's storage data. $\mathcal{A}$ can obtain sensitive data stored in TE.

$Test(\Pi^t)$: This query models the semantic security of the session key. When $\mathcal{A}$ executes this query for the instance Π^t which has not yet established a session key, it returns the symbol $\perp$. If the session key of the instance Π^t has been established and is fresh, the $Test$ query randomly select a bit $b \in \{0, 1\}$, and returns the real session key if $b = 1$ or a random string with the same length if $b = 0$. The goal of $\mathcal{A}$ is to guess the value of the random bit b. $\mathcal{A}$ is deemed to successfully break the semantic security of the session key if $\mathcal{A}$ guesses b correctly with nonnegligible probability.

Definition 1 (Semantic security of the session key [25]): In the ROR model, the adversary $\mathcal{A}$ is required to distinguish between an instance's real session key and a random key. $\mathcal{A}$ can perform $Hash$, $Execute$, $Send$, $Reveal$, $CorruptTE$, $CorruptCS$ and $Test$ queries on the instance Π^t. At the end of the game, $\mathcal{A}$ guesses that the value of bit b in the $Test$ query is b', and he/she wins the game if $b = b'$. Let $Succ$

be an event that $\mathcal{A}$ wins the game. The advantage of $\mathcal{A}$ in breaking the session key semantic security of the proposed scheme $\mathcal{P}$ is defined by $Adv_{\mathcal{P}}^{AKE}(\mathcal{A}) = |2\dot{P}r[Succ] - 1|$. For any probabilistic polynomial time adversary $\mathcal{A}$, if there exists a negligible function ϵ that satisfies $Adv_{\mathcal{P}}^{AKE}(\mathcal{A}) \leq \epsilon$, we say that the proposed scheme $\mathcal{P}$ is semantically secure under ROR model.

Theorem 1: Suppose that there exists an adversary $\mathcal{A}$ want to brake the semantic security of the proposed scheme $\mathcal{P}$ and derive the session key between *TE* and *CS* in probabilistic polynomial time (PPT). Then $\mathcal{A}$'s advantage in this regard is

$$Adv_{\mathcal{P}}^{\mathrm{AKA}}(\mathcal{A}) \leq \frac{{q_h}^2 + 2q_s}{|H|},$$

where q_h, q_s, and $|H|$represent the number of $Hash$ queries, the number of $Send$ queries, and the range of the hash function, respectively.

Proof. We define five games $Game_i$ $(i = 0, 1, 2, 3, 4)$ to demonstrate the security of the negotiated session key in the protocol. Let $Succ_i$ represent the event that the random bit b in the game $Game_i$ is successfully guessed by $\mathcal{A}$, i.e., $b' = b$.

$Game_0$: The game simulates $\mathcal{A}$'s real attack on the true protocol $\mathcal{P}$ under the ROR model. $\mathcal{A}$ needs to guess the random bit b in the semantic security game. Its advantage is

$$Adv_{\mathcal{P}}^{\mathrm{AKA}}(\mathcal{A}) = \left|2 \cdot Pr[Succ_0] - 1\right|. \tag{1}$$

$Game_1$: This game simulates eavesdropping attacks by $\mathcal{A}$. $\mathcal{A}$ can perform multiple $Execute(\Pi_{\mathrm{TE}}^{t_1}, \Pi_{\mathrm{AS}}^{t_2}, \Pi_{\mathrm{CS}}^{t_3})$ queries. At the end of this game, $\mathcal{A}$ performs a $Test(\Pi^t)$ query to determine whether the output of $Test$ is the real session key or a random number.

Suppose $\mathcal{A}$ eavesdrops on all messages exchanged in the protocol. To get the session key SK_{ik}, $\mathcal{A}$ must calculate $SK_{ik} = h(h(HID_{\mathrm{TE}}||AS_j||x_1||x_2||A_{ij})||AS_j||ID_{\mathrm{CS}}||y||T_3)$, but several parameters (e.g., x_1, x_2, and A_{ij}) are secret to $\mathcal{A}$. Therefore, in $Game_1$, the probability of winning the indistinguishable game does not increase when $\mathcal{A}$ performs eavesdropping attacks. Thus, we have

$$Pr[Succ_1] - Pr[Succ_0] = 0. \tag{2}$$

$Game_2$: In this game, $\mathcal{A}$ achieves the real attack by adding $Send$ and $Hash$ queries based on $Game_1$. $Game_2$ is to simulate an active attack that the adversary $\mathcal{A}$ aims to forge a message and deceive the other party into thinking it is an authentication message. $\mathcal{A}$ can perform multiple $Hash$ queries to check if hash collisions occur. Because each message in the scheme contains a random number, timestamp and secret parameters, the probability of collision occurring is negligible when $\mathcal{A}$ executes a $Send$ query. For $Hash$ queries, the probability of a collision is about ${q_h}^2/2|H|$ based on the birthday paradox. Thus we have

$$\left|Pr[Succ_2] - Pr[Succ_1]\right| \leq \frac{{q_h}^2}{2|H|}. \tag{3}$$

$Game_3$: This game is to add the simulation of $CorruptTE$ query in $Game_2$. In this game, $\mathcal{A}$ performs $CorruptTE(\Pi_{\mathrm{TE}}^{t_1})$ query to obtain the information $\{HID_{TE}, Y_i, V_i, (AS_j, B_{ij})\}$ stored in *TE*. After obtaining this information, $\mathcal{A}$ tries to guess secret values such as PW_{TE}, A_{ij} and $h(AS_j||Q)$. Similar to the analysis process of [18] and [25], it is difficult for $\mathcal{A}$ to obtain the secret values (e.g., PW_{TE}) from the stored information. Let D be the used uniformly distributed password dictionary [18], [25]. Without loss of generality, we assume that $|H| < \min(|D|, |P|, |Q|)$. If the system limits the number of incorrect password inputs, then we can get

$$\left|Pr[Succ_3] - Pr[Succ_2]\right| \leq \frac{q_s}{|H|}. \tag{4}$$

$Game_4$: This game is to add the simulation of $CorruptCS$ query in $Game_3$. In this game, $\mathcal{A}$ performs $CorruptCS(\Pi_{CS}^{t_3})$ query to obtain the information stored in CS. Because CS is not responsible for authentication operations, thus $\mathcal{A}$ can not extract any valid information of the session key. Thus,

$$Pr[Succ_4] - Pr[Succ_3] = 0. \tag{5}$$

All queries are simulated and $\mathcal{A}$ only needs to guess the random bit b after executing the $Test$ query for winning the game. Evidently, the probability of winning $Game_4$ is equal to the probability of guessing the random bit value, Therefore,

$$Pr[Succ_4] = 1/2. \tag{6}$$

Combining Eqs. (2)-(5), we have

$$\begin{aligned}
&\left|Pr[Succ_4] - Pr[Succ_0]\right| \\
&\quad = \left|Pr[Succ_4] - Pr[Succ_3]\right| \\
&\qquad + \left|Pr[Succ_3] - Pr[Succ_2]\right| \\
&\qquad + \left|Pr[Succ_2] - Pr[Succ_1]\right| \\
&\qquad + \left|Pr[Succ_1] - Pr[Succ_0]\right| \\
&\quad \leq \frac{{q_h}^2 + 2q_s}{2|H|}.
\end{aligned} \tag{7}$$

Substituting Eq. (6) into Eq. (7), we get

$$\left|Pr[Succ_0] - \frac{1}{2}\right| \leq \frac{{q_h}^2 + 2q_s}{2|H|}. \tag{8}$$

Finally, substituting Eq. (8) into Eq. (1), we have

$$\begin{aligned}
Adv_{\mathcal{P}}^{\mathrm{AKA}}(A) &= \left|2 \cdot Pr[Succ_0] - 1\right| \\
&= 2\left|Pr[Succ_0] - \frac{1}{2}\right| \\
&\leq \frac{{q_h}^2 + 2q_s}{|H|}.
\end{aligned} \tag{9}$$

B. Formal Analysis Using BAN Logic

We verify the mutual authentication between TE and servers of the proposed protocol using the widely accepted BAN logic. The primary notations of the BAN logic are listed in Table III.

Rules. The five BAN logic rules used in the proof process are as follows.

TABLE III
NOTATIONS OF THE BAN LOGIC

Notations	Description
P, Q, R	Principals
C	Statement
K	Encryption key
$P\| \equiv C$	The principal P believes the statement C
$P\| \Rightarrow C$	The principal P has jurisdiction over the statement C
$P \triangleleft C$	The principal P sees the statement C
$P\| \sim C$	The principal P once said the statement C
$\#(C)$	The statement C is fresh
$\{C\}_K$	The statement C is encrypted with key K
$\langle X \rangle_Y$	The formula X combined with the formula Y
$P \overset{K}{\longleftrightarrow} Q$	The principal P and Q may use the shared key K

R1: *Message-Meaning Rule*. If P believes that K is shared with Q, and sees C combined with K, then P believes Q once said C.

$$\frac{P| \equiv (P \overset{K}{\longleftrightarrow} Q),\ P \triangleleft \{C\}_K}{P| \equiv Q \sim C}$$

R2: *Freshness Rule*. If P believes that C is fresh, then P believes the freshness of full condition.

$$\frac{P| \equiv \#(C)}{P| \equiv \#(C, M)}$$

R3: *Nonce-Verification Rule*. If P believes that C is fresh, and that Q once said C, then P believes that Q believes C.

$$\frac{P| \equiv \#(C),\ P| \equiv Q \sim (C)}{P| \equiv Q| \equiv C}$$

R4-1: *Believe Rule 1*. If P believes (C, M), then P believes C.

$$\frac{P| \equiv (C, M)}{P| \equiv (C)}$$

R4-2: *Believe Rule 2*. If P believes C and M, then P believes (C, M).

$$\frac{P| \equiv (C),\ P| \equiv (M)}{P| \equiv (C, M)}$$

R5: *Jurisdiction Rule*. If P believes that Q has jurisdiction over C, and that Q believes C, then P also believes C.

$$\frac{P| \equiv Q| \Rightarrow C,\ P| \equiv Q| \equiv C}{P| \equiv (C)}$$

Goals. We intend to satisfy the following goals.

Goal 1: $TE_i| \equiv (TE_i \overset{TE}{\longleftrightarrow} CS_k)$;

Goal 2: $CS_k| \equiv (TE_i \overset{TE}{\longleftrightarrow} CS_k)$;

Goal 3: $TE_i| \equiv CS_k| \equiv (TE_i \overset{TE}{\longleftrightarrow} CS_k)$;

Goal 4: $CS_k| \equiv TE_i| \equiv (TE_i \overset{TE}{\longleftrightarrow} CS_k)$.

Idealized form of messages. The idealized form of all transmitted messages is shown below.

Msg 1: $TE_i \rightarrow AS_j$,

$$\{HID_{TE_i}, h(Q), AS_j, A_{ij}, x_1, x_2, T_1\}_{h(AS_j||Q)}$$

Msg 2: $AS_j \rightarrow CS_k$,

$$\{HID_{TE_i}, AS_j, A_{ij}, x_1, x_2, T_2\}_{SK_{jk}}$$

Msg 3: $CS_k \rightarrow TE_i$,

$$\{HID_{TE_i}, AS_j, A_{ij}, x_1, x_2, y, T_3\}_{SK_{ik}}$$

Assumptions. The initial assumptions of the proposed protocol are as follows.

A1: $AS_j| \equiv \#(T_1)$;

A2: $CS_k| \equiv \#(T_2)$;

A3: $TE_i| \equiv \#(T_3)$; A4: $AS_j| \equiv (TE_i \overset{h(AS_j||Q)}{\longleftrightarrow} AS_j)$;

A5: $CS_k| \equiv (AS_j \overset{SK_{jk}}{\longleftrightarrow} CS_k)$;

A6: $TE_i| \equiv (CS_k \overset{SK_{ik}}{\longleftrightarrow} TE_i)$;

A7: $CS_k| \equiv TE_i| \Rightarrow (CS_k \overset{SK_{ik}}{\longleftrightarrow} TE_i)$.

The following steps proves that the proposed protocol achieves mutual authentication between TE and servers using the above hypotheses and rules.

According to Msg 1, we have S1.

S1: $AS_j \triangleleft \{(HID_{TE_i}, h(Q), AS_j, A_{ij}, x_1, x_2), T_1\}_{h(AS_j||Q)}$. Based on the rule R1, the assumption A4 and the conclusion S1, we can derive S2.

S2: $AS_j| \equiv TE_i \sim \{(HID_{TE_i}, h(Q), AS_j, A_{ij}, x_1, x_2), T_1\}$. Based on the rule R2, the assumption A1 and the conclusion S2, we can derive S3.

S3: $AS_j| \equiv \#\{(HID_{TE_i}, h(Q), AS_j, A_{ij}, x_1, x_2), T_1\}$. Based on the rule R3, the conclusion S2 and S3, we can derive S4.

S4: $AS_j| \equiv TE_i| \equiv \{(HID_{TE}, h(Q), AS_j, A_{ij}, x_1, x_2), T_1\}$. Based on the rule R4-1, and the conclusion S4, we can derive S5.

S5: $AS_j| \equiv TE_i| \equiv \{HID_{TE_i}, AS_j, A_{ij}, x_1, x_2\}$. According to Msg 2, we have S6.

S6: $CS_k \triangleleft \{HID_{TE}, AS_j, A_{ij}, x_1, x_2, T_2\}_{SK_{jk}}$. Based on the rule R1, the assumption A5 and the conclusion S6, we can derive S7.

S7: $CS_k| \equiv AS_j \sim \{(HID_{TE}, AS_j, A_{ij}, x_1, x_2), T_2\}$. Based on the rule R2, the assumption A2 and the conclusion S7, we can derive S8.

S8: $CS_k| \equiv \#\{(HID_{TE_i}, AS_j, A_{ij}, x_1, x_2), T_2\}$. Based on the rule R3, the conclusion S6 and S8, we can derive S9.

S9: $CS_k| \equiv AS_j| \equiv \{(HID_{TE_i}, AS_j, A_{ij}, x_1, x_2), T_2\}$. Based on the rule R4-1, and the conclusion S9, we can derive S10.

S10: $CS_k| \equiv AS_j| \equiv \{HID_{TE_i}, AS_j, A_{ij}, x_1, x_2\}$. Based on the conclusion S5 and S10, we can derive S11.

S11: $CS_k| \equiv TE_i| \equiv \{HID_{TE_i}, AS_j, A_{ij}, x_1, x_2\}$. Based on the conclusion S11 and the session key $SK_{ik} = h(HID_{TE}||AS_j||x_1||x_2||A_{ij}||y||T_3)$, we can derive S12.

S12: $CS_k| \equiv TE_i| \equiv (CS_k \overset{SK_{ik}}{\longleftrightarrow} TE_i)$. (Goal 4) Based on the rule R5, the assumption A7 and S12, we can derive S13.

S13: $CS_k| \equiv CS_k \overset{SK_{ik}}{\longleftrightarrow} TE_i)$. (Goal 2) According to Msg 3, we have S14.

S14: $TE_i \triangleleft \{HID_{TE_i}, AS_j, A_{ij}, x_1, x_2, y, T_3\}_{SK_{ik}}$. Based on the rule R1, the assumption A6 and the conclusion S14, we can derive S15.

S15: $TE_i| \equiv CS_k \sim \{(HID_{TE_i}, AS_j, A_{ij}, x_1, x_2, y), T_3\}$. Based on the rule R2, the assumption A3 and the conclusion S15, we can derive S16.

S16: $TE_i| \equiv \#\{(HID_{TE_i}, AS_j, A_{ij}, x_1, x_2, y), T_3\}$. Based on the rule R3, the conclusion S15 and S16, we can derive S17.

TABLE IV
SUBPROCESS CODE OF TE

```
let processTE(HIDte: bitstring, ASj: bitstring, Rte: bitstring,
PWte: bitstring, Vi: bitstring, Yi: bitstring, Bij: bitstring) =
new x1: nonce;
new x2: nonce;
new T1: timestamp;
let ARG1=XOR(Vi,hash(con(con(HIDte,PWte),Rte))) in
let ARG2=hash(con(ASj,ARG1)) in
let Xi'=XOR(XOR(Yi,PWte),HIDte) in
let Aij'=XOR(Bij,Xi') in
let SHIDte=XOR(XOR(HIDte,nonce_to_bitstring(x1)),ARG2) in
let M1=XOR(ARG2,nonce_to_bitstring(x1)) in
let M2=XOR(hash(con(Aij',nonce_to_bitstring(x1))),
nonce_to_bitstring(x2)) in
let M3=hash(con(con(HIDte,nonce_to_bitstring(x1)),
nonce_to_bitstring(x2))) in
let AUte=hash(con(con(con(HIDte,ASj),M3),
timestamp_to_bitstring(T1))) in
event beginTEASact(TE);
out(c1, (SHIDte, M1, M2, AUte, T1, true));
in(c1, (m0:bitstring, m1:bitstring, m2:timestamp, m3:bool));
let T3=m2 in
if checkfresh(T3, m3) = true then
let M6=m0 in
let AUs=m1 in
let M4'=hash(con(con(con(con(HIDte,ASj),nonce_to_bitstring(x1)),
nonce_to_bitstring(x2)),Aij')) in
let y'=XOR(M6,M4') in
let SKik'=hash(con(con(con(M4',ASj),y'),
timestamp_to_bitstring(T3))) in
let AUs'=hash(con(con(key_to_bitstring(SKik),M4'),y')) in
if AUs'=AUs then
let SessionKeyik=SKik' in
event endCSTEact(TE).
```

S17: $TE_i| \equiv CS_k| \equiv \{(HID_{TE_i}, AS_j, A_{ij}, x_1, x_2, y), T_3\}$. Based on the rule R4-1, and the conclusion S17, we can derive S18.

S18: $TE_i| \equiv CS_k| \equiv \{HID_{TE_i}, AS_j, A_{ij}, x_1, x_2, y\}$. Based on the conclusion S18 and the session key $SK_{ik} = h(HID_{TE_i}||AS_j||x_1||x_2||A_{ij}||y||T_3)$, we can derive S19.

S19: $TE_i| \equiv CS_k| \equiv (TE_i \overset{SK_{ik}}{\longleftrightarrow} CS_k)$. (Goal 3) Based on the rule R5, the assumption A7 and S12, we can derive S20.

S20: $TE_i| \equiv (TE_i \overset{SK_{ik}}{\longleftrightarrow} CS_k)$. (Goal 1)

The above analysis indicates that the proposed scheme can reach the Goals 1-4. Thus, it is convinced that TE_i and severs achieve mutual authentication, and the session key SK_{ik} is securely negotiated between these parties.

C. Formal Security Verification Using ProVerif Tool

As an automatic verifier of cryptographic protocols, ProVerif, can detect whether the protocol has some security properties based on a symbolic approach. In ProVerif, the adversary can monitor the public channel, and can intercept, tamper, and replay all messages transmitted in the channel.

Here we select the latest version 2.00 of ProVerif [34] to verify the security of the proposed scheme. Table IV shows the code of TE, and Table V describe the codes of the AS and CS. From Table VI, it can be seen that the session key is secure.

TABLE V
SUBPROCESS CODES OF AS AND CS

```
let processAServer(ASj: bitstring, SV1: secretvalue,
SV2: secretvalue, SKjk:key) =
new T2: timestamp;
in(c1, (m0: bitstring, m1: bitstring, m2: bitstring,
m3: bitstring, m4: timestamp, m5: bool));
let T1=m4 in
if checkfresh(T1, m5) = true then
let SHIDte=m0 in
let M1=m1 in
let M2=m2 in
let AUte=m3 in
let ARG2'=hash(con(ASj,secretvalue_to_bitstring(SV1))) in
let x1'=XOR(M1,ARG2') in
let HIDte'=XOR(XOR(SHIDte,x1'),ARG2') in
let Aij'=hash(con(HIDte',secretvalue_to_bitstring(SV2))) in
let x2'=XOR(hash(con(Aij',x1')),M2) in
let M3'=hash(con(con(HIDte',x1'),x2')) in
let AUte'=hash(con(con(con(HIDte',ASj),M3'),
timestamp_to_bitstring(T1))) in
if AUte'=AUte then
event endTEASact(AS);
let M4=hash(con(con(con(HIDte',ASj),x1'),x2'),Aij')) in
let M5=encrypt(SKjk,con(M4,timestamp_to_bitstring(T2))) in
out(c2,(M5,true));
event beginASCSact(AS).
```

```
let processCServer(ASj:bitstring, SKjk:key) =
new y: nonce;
new T3: timestamp;
in(c2, (m0:bitstring, m1:bool));
let M5=m0 in
let plaintext=decrypt(SKjk,M5) in
let M4'=separate1(plaintext) in
let T2=bitstring_to_timestamp(separate1(plaintext)) in
if checkfresh(T2, m1) = true then
let M6=XOR(nonce_to_bitstring(y),M4') in
event endASCSact(CS);
let SKik=hash(con(con(con(M4',ASj),nonce_to_bitstring(y)),
timestamp_to_bitstring(T3))) in
let AUs=hash(con(con(SKik,M4'),nonce_to_bitstring(y))) in
out(c1,(M6, AUs, T3, true));
event beginCSTEact(CS).
```

TABLE VI
TEST RESULTS

```
Verification summary:
Query inj-event(endTEASact(x)) ==> inj-event(beginTEASact(x)) is true.
Query inj-event(endCSTEact(x)) ==> inj-event(beginCSTEact(x)) is true.
Query inj-event(endASCSact(x)) ==> inj-event(beginASCSact(x)) is true.
Query not attacker(HIDte[]) is true.
Query not attacker(SKjk[]) is true.
Query not attacker(SKik[]) is true.
```

D. Informal Security Analysis

In this subsection, we provide informal analysis to demonstrate the securiy of the proposed scheme.

1) User Anonymity: The scheme provides anonymity for TE identity during the mutual authentication process. In TE registration phase, RC calculates $HID_{\text{TE}} = h(ID_{\text{TE}}||P)$ for TE, where P is the secret parameter of RC. Combined with the

irreversible property of one-way hash functions, the adversary cannot calculate ID_{TE} even if he/she knows HID_{TE}. Therefore, the proposed scheme has the property of user anonymity.

2) Conditions Privacy Protection: The anonymous identity of TE is obtained by calculating $HID_{\mathrm{TE}} = h(ID_{\mathrm{TE}}||P)$ by RC, and only RC knows the corresponding relationship between HID_{TE} and ID_{TE}. In this way, RC can track the identity information corresponding to the malicious TE. Therefore, the scheme can track malicious TEs while protecting TE's anonymity, which meets the property of conditional privacy protection.

3) Untraceability: In each session, TE and CS need select a new random value x_1, x_2 and y, combined with fresh timestamp. Therefore, the generated parameters (e.g., $SHID_{\mathrm{TE}}$, M_1, and M_2) are also random. In other words, messages within each session are dynamic and random, so the adversary cannot track the participants from different sessions. Therefore, our solution satisfies non-traceability.

4) Offline Registration Center: Generally, RC is often set to be secure and trusted. If RC participates in the authentication process, the frequent registration process of entities not only increases the communication burden of authentication, but also increases the probability of RC being attacked. In the proposed scheme, RC is mainly responsible for the registration, and not participate in the authentication process.

5) Mutual Authentication: As the initiator of the authentication process, TE needs to generate the authentication login message first. Then this message is sent to the target authentication server AS. The main function of the AS is to complete the authentication of legitimate TEs. Next, TE and AS perform the mutual authentication process. After the verification is passed, AS packages the relevant information of TE and sends it to the matched CS. Note that the CS is a trusted computing server in the server group. Finally, a temporary session key is negotiated between CS and TE.

6) Resistant to Session Key Attack: The formed session key in the proposed scheme is $SK = h(M_4||AS_j||y||T_3)$, where $M_4 = h(HID_{TE}||AS_j||x_1||x_2||A_{ij})$. The session key is computed by a hash function, which includes inputs for many secret parameters, such as x_1, x_2 and y. Note that x_1 and x_2 is chosen by TE at random, and y is selected by CS. Therefore, if the adversary does not steal all the secret parameters, he/she do not be able to obtain the session private key.

7) Resistant to Impersonation Attack of TE: During the authentication process, suppose there exists an attack who wants to pretend to be a legitimate TE to log in. According to the protocol, the correct login message is $(SHID_{\mathrm{TE}}, M_1, M_2, AU_{\mathrm{TE}}, T_1)$. Obviously, the first four values are all random, and the probability of the adversary guessing correctly is negligible.

8) Resistant to Data Theft Attack of TE: Suppose the adversary hacks the terminal equipment TE and extracts $\{HID_{TE}, Y_i, V_i, (AS_j, B_{ij})\}$ from the memory of TE, where $HID_{TE} = h(ID_{TE}||P)$ and P is a random secret value of RC. The adversary cannot get ID_{TE} from HID_{TE}. Note that $B_{ij} = A_{ij} \oplus X_i$, $X_i = h(HID_{TE}||h(Q||P)||R_{TE})$ and $Y_i = X_i \oplus PW_{TE} \oplus HID_{TE}$. Due to the adversary's knowledge of B_{ij} and Y_i, a common attack method is to perform Xor operation between them and obtain $B_{ij} \oplus Y_i = A_{ij} \oplus PW_{\mathrm{TE}} \oplus HID_{TE}$. Thus, if the adversary wants to obtain A_{ij}, then it must know PW_{TE}. Therefore, the adversary can not obtain the pairing data A_{ij}.

9) Resistant to Spoofing Attack of Servers: In the scheme, we use certificates signed by RC to perform identity authentication between the two types of servers. If a fake server wants to join the server group, it must need a certificate issued by RC through a secure offline channel. Therefore, the adversary cannot impersonate the behaviour of the AS and CS servers.

10) Resistant to Man-in-the-Middle Attack: When an adversary launching a man-in-the-middle attack, he/she has the following three messages.

- TE's login message $(SHID_{TE}, M_1, M_2, AU_{\mathrm{TE}}, T_1)$, where $M_1 = ARG_2 \oplus x_1$, $SHID_{\mathrm{TE}} = HID_{\mathrm{TE}} \oplus x_1 \oplus ARG_2$, $ARG_2 = h(AS_j||ARG_1)$. The function of M_1 is to deliver x_1, and $SHID_{\mathrm{TE}}$ is to hide HID_{TE}. The adversary cannot calculate x_1 without knowing ARG_2. The function of M_2 is to transmit x_2, and it is infeasible to get x_2 from M_2. Moreover, if the adversary tampered with the message, then the authentication message $AU_{\mathrm{TE}} = h(HID_{\mathrm{TE}}||AS_j||h(HID_{\mathrm{TE}}||x_1||x_2)||T_1)$ will be invalid because it cannot pass the verification.
- AS delivers $M_5 = E(SK_{jk}, M_4)$ to the target CS, where $M_4 = h(HID_{\mathrm{TE}}'||AS_j||x_1'||x_2'||A_{ij}')$. Since the message is encrypted with a shared key between AS and CS, the adversary without the key cannot decrypt M_5. Even if the adversary successfully decrypts the message, he or she cannot obtain the secret parameters due to the irreversibility of hash functions.
- The message (M_6, AU_S, T_3) returned by CS to TE. Since $M_6 = y \oplus M_4'$, M_6 is used for transmitting y. Because M_4 cannot be obtained by the adversary, thus y does not be leaked. Moreover, if the adversary tampered with the message, then the authentication message $AU_S = h(SK_{ik}||M_4'||y')$ can not pass the verification process.

11) Resistant to Replay Attack: In the authentication process, a timestamp is included and participates in the construction of the encrypted message. If the adversary replays a captured massage of TE, the session will be terminated immediately because of timestamp verification. Even if the adversary uses a new timestamp to pass the verification process of the timestamp, the forged message will not be successfully pass the verification of AU_{TE} of AS. If the adversary replays a captured massage of CS and modifies the timestamp, the forged message also can not pass the verification of AU_{S} of TE.

12) Resistant to Insider Attack: In the scheme, we assume that the server is semi-trusted. Because TE needs to provide its own information to AS for authentication. And, the AS server may launch an internal attack to steal sensitive information of TE. Our solution is that RC generates the pairing information between each AS and all the TEs within its area in the registration process. This method avoids partly avoids the risk of TE disclosing sensitive information to the server.

TABLE VII
EXECUTION TIME OF BASIC OPERATIONS (MS)

Notations	Description	TE	Server
T_h	SHA-256 hash algorithm	0.309	0.024
T_{sym}	Symmetric encryption/decryption	0.018	0.001
T_{ecm}	Elliptic curve point multiplication	2.288	0.382
T_{cheb}	Chebyshev polynomial computing	21.63	1.26

13) Resistant to Semi-Trusted Server Attack: The security discussion of the scheme from the viewpoint of semi-trusted server is as follows.

- Anonymous identity of TE. TE always use the anonymous identity HID_{TE} during authentication. Apart from a trusted registration center, no other system participant knows the true identity of TE.
- TE's private information and secret value. In the scheme, a trusted registration center distributes authentication pairing information between AS and TE in advance. This approach partly prevents the server from stealing the secret information of TE.
- RC's secret parameters. In the registration process, the response information of TE and AC hides the secret parameters P and Q of RC. But it is difficult for anyone to crack it because of the irreversible property of hash algorithms.
- Mutual authentication between servers. The servers are divided into AS and CS according to their functions. Between various servers in the same server group, they can conduct mutual verification and supervision through certificates issued by the trusted RC. Therefore, any semi-trusted server can not pretend to be the other one for doing evil.

VI. PERFORMANCE ANALYSIS

The performance of AKA schemes are often measured by comparing the computational overhead and the communication overhead. In the proposed scheme, RC does not participate in the authentication process. Therefore, we choose AKA schemes supporting offline registration for comparison. Here we compare the proposed scheme with Sutrala et al.'s protocol [22], Guo et al.'s protocol [25], Wang et al.'s protocol [32], Li et al.'s protocol [35], Zhang et al.'s protocol [36], Baghestani et al.'s protocol [37] and Chai et al.'s protocol [38].

A. Computation Overhead

For comparison, the execution time of basic operations are often used to quantify the computing overhead. Here we run the program on the personal device and the cloud server to simulate TE and servers in the protocol. We perform 100 tests and take the the average as the result. Table VII shows the execution time of some basic operations used in this paper. It is noted that the execution time of these operations still may be affected by variable factors, such as the degree of optimization of the algorithm, hardware environment, and the size of input data.

One of the main goals of the proposed scheme is to be applicable to resource-constrained devices, thus here we focus on comparing the computational overhead on the end-device side. Note that the basic operations corresponding to T_h, T_{sym}, T_{ecm}, and T_{cheb} are SHA-256 hash algorithm, symmetric encryption and decryption, elliptic curve point multiplication and Chebyshev polynomial. The elliptic curve equation is the Weierstrass standard form $y^2 = x^3+ax+b \mod p$. The prime number p is 512 $bits$, and we choose an additive elliptic curve group G over F_p. The type-1 ate pairing $e : G \times G \rightarrow G_T$ is used for comparison, where G and G_T are groups with a 160 $bits$ prime order q. The reason we overlook Xor operation in comparison is that the execution time is negligible compared to other operations. The comparison results are shown in Table VIII, and it shows that the proposed scheme has the lowest computation overhead on the TE end.

TABLE VIII
COMPARISON OF COMPUTATION OVERHEAD (MS)

Schemes	Computation cost of TE (ms)
[22]	$15T_h$+$4T_{ecm}$=13.787
[25]	$29T_h$=8.961
[32]	$9T_h$=2.781
[35]	$11T_h$+$2T_{cheb}$=46.659
[36]	$8T_h$+T_{ecm}=4.76
[37]	$5T_h$+$2T_{ecm}$=6.121
[38]	$8T_h$+$2T_{ecm}$=7.048
Ours	$8T_h$+$2T_{sym}$=2.508

TABLE IX
COMPARISON OF COMMUNICATION COST (BITS)

Schemes	Communication Cost	Length(bits)
[22]	$7\|H\|+5\|T\|+4\|G\|+1\|Z_q\|+3\|ID\|$	7072
[25]	$16\|H\|+7\|T\|+7\|ID\|$	6112
[32]	$7\|H\|$	1792
[35]	$7\|H\|+2\|cheb\|$	2304
[36]	$5\|H\|+2\|T\|$	1344
[37]	$6\|H\|+2\|G\|+\|T\|$	3616
[38]	$4\|H\|+2\|G\|+2\|T\|$	3136
Ours	$6\|H\|+2\|T\|+1\|sym\|$	2112

B. Communication Cost

Combining the previous works, the communication cost of the scheme is mainly measured by the number of bits required to exchange messages. To facilitate the comparison, we make an unified assumption, i.e., the identity $|ID|$, the random number, the output results of hash $|H|$, and the computational result of Chebyshev polynomial $|cheb|$ are all 256 bits. The timestamp $|T|$ is 32 bits, the size of the symmetric encryption $|sym|$ is 512 bits, and the size of the cyclic group $|G|$ is 1024 bits. The comparison results in Table IX show that the communication cost of the proposed scheme is better than those of the schemes [22], [25], [35], [37], [38], and slightly worse than those of the scheme [32], [36].

C. Security Features

Based on state-of-the-art similar schemes, here we also provide a comparison of security features according to some security evaluation metrics. The comparison results are shown in Table X. Compared with their schemes, the proposed scheme can meet more security requirements and features.

TABLE X
COMPARISON OF SECURITY PROPERTIES

Properties	[22]	[25]	[32]	[35]	[36]	[37]	[38]	Ours
User Anonymity	Y	Y	Y	Y	Y	N	N	Y
Conditions Privacy Protection	Y	Y	N	Y	N	N	N	Y
Untraceability	Y	Y	Y	Y	Y	Y	Y	Y
Offline Registration Center	Y	Y	Y	Y	Y	Y	Y	Y
Lightweight Authentication	N	Y	Y	Y	Y	Y	Y	Y
Server Function Division	N	N	N	N	N	Y	Y	Y
TE Secret Value Security	Y	Y	Y	Y	Y	Y	Y	Y
Server Key Security	N	Y	Y	Y	Y	Y	N	Y
Resistant to TE Impersonation Attack	Y	Y	Y	Y	N	Y	Y	Y
Resistant to Server Spoofing Attack	Y	Y	Y	Y	Y	Y	N	Y
Resistant Semi-Trusted Server Attack	N	N	N	N	N	N	N	Y
Resistant to Man-In-the-Middle Attack	Y	Y	Y	Y	Y	Y	Y	Y
Resistant to Replay Attack	Y	Y	N	N	Y	Y	Y	Y
Resistant to TE Data Theft Attack	N	N	Y	Y	Y	Y	Y	Y
Resistant to Insider Attack	Y	Y	Y	Y	N	Y	N	Y
Resistant to Session Key Attack	Y	Y	Y	Y	Y	Y	Y	Y

VII. CONCLUSION

In this paper, we propose a new and secure authentication key agreement scheme suitable for the MEC environment, which can meet the requirements of mutual authentication between resource-constrained IoT terminal devices and servers. For efficient task processing, we divide the servers into the AS and CS servers according to their functions. In addition, the scheme provides multiple registration centers to share the communication and the computation overhead for single point registration. The terminal device does not perform complex cryptographic operations, thereby reducing the computation overhead of TE. At the same time, in order to prevent the semi-trusted server from directly stealing the sensitive information of TE, RC generates a pairing AS for each TE during registration.

We use the ROR model and BAN logic to prove the security of the proposed scheme, and verify it with the ProVerif verification tool. Furthermore, we conduct security requirements and performance comparisons with some similar works. The results show that the proposed scheme has advantages in terms of efficiency and security.

To resist attacks from semi-trusted servers, and protect the privacy of terminal devices, we adopt the method of allowing the registration center to generate the pairing information in advance. This approach can not resist the denial of service attacks, and do not have the property of forward secrecy. In future, we will consider a lightweight AKA scheme with perfect forward secrecy.

REFERENCES

[1] Z. Lv, R. Lou, J. Li, A. K. Singh, and H. Song, "Big data analytics for 6G-enabled massive Internet of Things," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5350–5359, Apr. 2021.

[2] Z. Liao et al., "Distributed probabilistic offloading in edge computing for 6G-enabled massive Internet of Things," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5298–5308, Apr. 2021.

[3] M. Liyanage, P. Porambage, A. Y. Ding, and A. Kalla, "Driving forces for multi-access edge computing (MEC) IoT integration in 5G," *ICT Exp.*, vol. 7, no. 2, pp. 127–137, Jun. 2021.

[4] Z. Ning et al., "Dynamic computation offloading and server deployment for UAV-enabled multi-access edge computing," *IEEE Trans. Mobile Comput.*, vol. 22, no. 5, pp. 2628–2644, May 2023, doi: 10.1109/TMC.2021.3129785.

[5] V. Sureshkumar, R. Amin, V. R. Vijaykumar, and S. R. Sekar, "Robust secure communication protocol for smart healthcare system with FPGA implementation," *Future Gener. Comput. Syst.*, vol. 100, pp. 938–951, Nov. 2019, doi: 10.1016/j.future.2019.05.058.

[6] L. Deng, J. Shao, and Z. Hu, "Identity based two-party authenticated key agreement scheme for vehicular ad hoc networks," *Peer–Peer Netw. Appl.*, vol. 14, no. 4, pp. 2236–2247, Jul. 2021.

[7] V. Thakur, G. Indra, N. Gupta, P. Chatterjee, O. Said, and A. Tolba, "Cryptographically secure privacy-preserving authenticated key agreement protocol for an IoT network: A step towards critical infrastructure protection," *Peer–Peer Netw. Appl.*, vol. 15, no. 1, pp. 206–220, Jan. 2022.

[8] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, "A survey on the Internet of Things (IoT) forensics: Challenges, approaches, and open issues," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 1191–1221, 2nd Quart., 2020.

[9] K. Tange, M. De Donno, X. Fafoutis, and N. Dragoni, "A systematic survey of industrial Internet of Things security: Requirements and fog computing opportunities," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2489–2520, 4th Quart., 2020.

[10] D. Xie, J. Yang, W. Bian, F. Chen, and T. Wang, "An improved identity-based anonymous authentication scheme resistant to semi-trusted server attacks," *IEEE Internet Things J.*, vol. 10, no. 1, pp. 734–746, Jan. 2023, doi: 10.1109/JIOT.2022.3203991.

[11] A. G. Reddy, A. K. Das, V. Odelu, A. Ahmad, and J. S. Shin, "A privacy preserving three-factor authenticated key agreement protocol for client–server environment," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 2, pp. 661–680, 2019.

[12] D. Mishra, D. Dharminder, P. Yadav, Y. S. Rao, P. Vijayakumar, and N. Kumar, "A provably secure dynamic ID-based authenticated key agreement framework for mobile edge computing without a trusted party," *J. Inf. Secur. Appl.*, vol. 55, Dec. 2020, Art. no. 102648, doi: 10.1016/j.jisa.2020.102648.

[13] X. Li, J. Liu, M. S. Obaidat, P. Vijayakumar, Q. Jiang, and R. Amin, "An unlinkable authenticated key agreement with collusion resistant for VANETs," *IEEE Trans. Veh. Technol.*, vol. 70, no. 8, pp. 7992–8006, Aug. 2021.

[14] K. Mahmood, M. F. Ayub, S. Z. Hassan, Z. Ghaffar, Z. Lv, and S. A. Chaudhry, "A seamless anonymous authentication protocol for mobile edge computing infrastructure," *Comput. Commun.*, vol. 186, pp. 12–21, Mar. 2022, doi: 10.1016/j.comcom.2022.01.005.

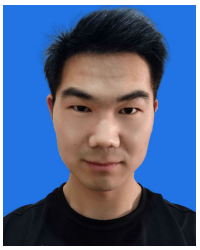
[15] P. Gope and B. Sikdar, "A privacy-aware reconfigurable authenticated key exchange scheme for secure communication in smart grids," *IEEE Trans. Smart Grid*, vol. 12, no. 6, pp. 5335–5348, Nov. 2021.

[16] A. Gupta, M. Tripathi, T. J. Shaikh, and A. Sharma, "A lightweight anonymous user authentication and key establishment scheme for wearable devices," *Comput. Netw.*, vol. 149, pp. 29–42, Feb. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128618304389

[17] M. Umar, S. H. Islam, K. Mahmood, S. Ahmed, Z. Ghaffar, and M. A. Saleem, "Provable secure identity-based anonymous and privacy-preserving inter-vehicular authentication protocol for VANETS using PUF," *IEEE Trans. Veh. Technol.*, vol. 70, no. 11, pp. 12158–12167, Nov. 2021.

[18] R. Vinoth, L. J. Deborah, P. Vijayakumar, and N. Kumar, "Secure multifactor authenticated key agreement scheme for industrial IoT," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3801–3811, Mar. 2021.
[19] S. Garg, K. Kaur, G. Kaddoum, J. J. P. C. Rodrigues, and M. Guizani, "Secure and lightweight authentication scheme for smart metering infrastructure in smart grid," *IEEE Trans. Ind. Informat.*, vol. 16, no. 5, pp. 3548–3557, May 2020.
[20] M. Ma, D. He, H. Wang, N. Kumar, and K.-K. R. Choo, "An efficient and provably secure authenticated key agreement protocol for fog-based vehicular ad-hoc networks," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8065–8075, Oct. 2019.
[21] J. Zhang, H. Zhong, J. Cui, Y. Xu, and L. Liu, "SMAKA: Secure many-to-many authentication and key agreement scheme for vehicular networks," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 1810–1824, 2021, doi: 10.1109/TIFS.2020.3044855.
[22] A. K. Sutrala, M. S. Obaidat, S. Saha, A. K. Das, M. Alazab, and Y. Park, "Authenticated key agreement scheme with user anonymity and untraceability for 5G-enabled softwarized industrial cyber-physical systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2316–2330, Mar. 2022, doi: 10.1109/TITS.2021.3056704.
[23] Y. Zhang, D. He, L. Li, and B. Chen, "A lightweight authentication and key agreement scheme for Internet of Drones," *Comput. Commun.*, vol. 154, pp. 455–464, Mar. 2020, doi: 10.1016/j.comcom.2020.02.067.
[24] Y. Li, Q. Cheng, X. Liu, and X. Li, "A secure anonymous identity-based scheme in new authentication architecture for mobile edge computing," *IEEE Syst. J.*, vol. 15, no. 1, pp. 935–946, Mar. 2021.
[25] Y. Guo, Z. Zhang, and Y. Guo, "Anonymous authenticated key agreement and group proof protocol for wearable computing," *IEEE Trans. Mobile Comput.*, vol. 21, no. 8, pp. 2718–2731, Aug. 2022, doi: 10.1109/TMC.2020.3048703.
[26] J. Li, Z. Su, D. Guo, K. R. Choo, and Y. Ji, "PSL-MAAKA: Provably secure and lightweight mutual authentication and key agreement protocol for fully public channels in Internet of Medical Things," *IEEE Internet Things J.*, vol. 8, no. 17, pp. 13183–13195, Sep. 2021.
[27] V. Sureshkumar, P. Chinnaraj, P. Saravanan, R. Amin, and J. J. P. C. Rodrigues, "Authenticated key agreement protocol for secure communication establishment in vehicle-to-grid environment with FPGA implementation," *IEEE Trans. Veh. Technol.*, vol. 71, no. 4, pp. 3470–3479, Apr. 2022.
[28] J. Zhao et al., "A secure biometrics and PUFs-based authentication scheme with key agreement for multi-server environments," *IEEE Access*, vol. 8, pp. 45292–45303, 2020, doi: 10.1109/ACCESS.2020.2975615.
[29] L. Zhang, Y. Zhang, S. Tang, and H. Luo, "Privacy protection for e-health systems by means of dynamic authentication and three-factor key agreement," *IEEE Trans. Ind. Electron.*, vol. 65, no. 3, pp. 2795–2805, Mar. 2018.
[30] J. Shen, T. Zhou, F. Wei, X. Sun, and Y. Xiang, "Privacy-preserving and lightweight key agreement protocol for V2G in the social Internet of Things," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2526–2536, Aug. 2018.
[31] G. Bansal, N. Naren, V. Chamola, B. Sikdar, N. Kumar, and M. Guizani, "Lightweight mutual authentication protocol for V2G using physical unclonable function," *IEEE Trans. Veh. Technol.*, vol. 69, no. 7, pp. 7234–7246, Jul. 2020.
[32] J. Wang et al., "An efficient hash-based authenticated key agreement scheme for multi-server architecture resilient to key compromise impersonation," *Digit. Commun. Netw.*, vol. 7, no. 1, pp. 140–150, 2021.
[33] A. Kumar and H. Om, "An improved and secure multiserver authentication scheme based on biometrics and smartcard," *Digit. Commun. Netw.*, vol. 4, no. 1, pp. 27–38, 2018.
[34] B. Blanchet, B. Smyth, V. Cheval, and M. Sylvestre, "ProVerif 2.00: Automatic cryptographic protocol verifier, user manual and tutorial," Version From, May 2018, pp. 5–16. [Online]. Available: https://www.crs811.com/wp-content/uploads/2019/01/manual.pdf and https://www.crs811.com/uploads/2019/01/manual.pdf
[35] X. Li, F. Wu, M. K. Khan, L. Xu, J. Shen, and M. Jo, "A secure chaotic map-based remote authentication scheme for telecare medicine information systems," *Future Gener. Comput. Syst.*, vol. 84, pp. 149–159, Jul. 2018, doi: 10.1016/j.future.2017.08.029.
[36] J. Li, N. Zhang, J. Ni, J. Chen, and R. Du, "Secure and lightweight authentication with key agreement for smart wearable systems," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7334–7344, Aug. 2020.
[37] S. H. Baghestani, F. Moazami, and M. Tahavori, "Lightweight authenticated key agreement for smart metering in smart grid," *IEEE Syst. J.*, vol. 16, no. 3, pp. 4983–4991, Sep. 2022.
[38] S. Chai et al., "Provably secure and lightweight authentication key agreement scheme for smart meters," *IEEE Trans. Smart Grid*, vol. 14, no. 5, pp. 3816–3827, Sep. 2023.

Dong Xie received the Ph.D. degree in cryptography from the Beijing University of Posts and Telecommunications, China, in 2017. He is currently an Associate Professor with the School of Computer and Information, Anhui Normal University. He is the coauthor of over 40 scientific articles. His research interests include cryptography, information security, and image encryption.



Jinghua Yang is currently pursuing the master's degree with the School of Computer and Information, Anhui Normal University, Wuhu, China. His research interests include authentication protocols and applied cryptography.

Bin Wu is currently pursuing the master's degree with the School of Computer and Information, Anhui Normal University. His research interests include secret sharing, image security, and compressed sensing.

Weixin Bian received the Ph.D. degree in computer science from the China University of Mining and Technology, Beijing, China, in 2018. In 2005, he joined the School of Computer and Information, Anhui Normal University, Wuhu, China, where he is currently a Professor. His research interests include information security, biometric privacy protection, image processing, machine learning, and pattern recognition.

Fulong Chen received the bachelor's degree from Anhui Normal University in 2000, the M.S. degree from China West Normal University in 2005, and the Ph.D. degree from Northwestern Polytechnical University in 2011. From 2008 to 2010, he visited Rice University, USA, and worked with Prof. Walid Taha in the field of cyber-physical systems. He is currently a Professor, a Master Instructor, and the Director of the Department of Computer Science and Technology, Anhui Normal University. His research interests are embedded computing and pervasive computing, cyber-physical systems, and high-performance computer architecture.

Taochun Wang received the bachelor's and master's degrees from Yunnan University, Kunming, China, in 2002 and 2005, respectively, and the Ph.D. degree in computer applications technology from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2016. He is currently a Professor with the School of Computer and Information, Anhui Normal University, Wuhu, China. His research interests are privacy preservation and the Internet of Things.