# A Provably Secure and Efficient Identity-Based Anonymous Authentication Scheme for Mobile Edge Computing

Xiaoying Jia, Debiao He, Neeraj Kumar, *Senior Member, IEEE*,
and Kim-Kwang Raymond Choo, *Senior Member, IEEE*

***Abstract*—Mobile edge computing (MEC) allows one to overcome a number of limitations inherent in cloud computing, although achieving the broad range of security requirements in MEC settings remains challenging. In this paper, we focus on achieving mutual authentication with anonymity and un-traceability, as this is crucial in ensuring data security and user privacy. Specifically, we design an identity-based anonymous authenticated key agreement protocol for the MEC environment. The proposed protocol achieves mutual authentication in only a single message exchange round, as well as assures both user anonymity and un-traceability. We then evaluate the security and performance of the protocol, and demonstrate that it achieves the required security properties and outperforms prior approaches in terms of communicational and computational costs.**

***Index Terms*—Authentication, mobile edge computing (MEC), mobile server, un-traceability, user anonymity.**

## I. Introduction

THE cloud computing paradigm is maturing, partly evidenced by its current adoption by organizations and government agencies and the extension to incorporate Internet of Things (IoT) devices, edge devices, and fog devices. For example, in a cloud-based IoT setting, massive data from a range of systems and devices (e.g., sensors) are collected, stored, processed, and analyzed. There are, however, a number of challenges associated with a cloud-based IoT deployment. For example, due to the centralized nature of cloud servers and their physical location, it can be challenging for remote cloud servers to respond to real-time requirements, such as in applications requiring low latency, high mobility, and location awareness (e.g., augmented reality, vehicular networks, and adversarial settings such as battlefields).

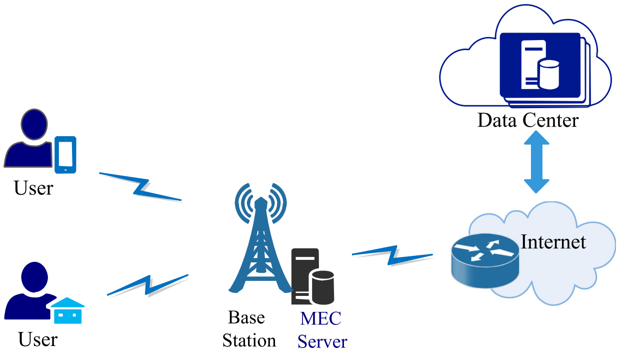

Fig. 1. MEC architecture.

Mobile edge computing (MEC) is a potential solution to address the aforementioned limitations, by bringing the computational activity/capabilities closer to the requesting devices/users. An early example of the MEC is the platform offered by Nokia Siemens Networks and IBM (NYSE: IBM), which is designed to run applications on a mobile base station (BS). In an MEC setting, MEC servers with computing and storage capabilities are deployed at the edge of the network (e.g., radio access networks–RAN), as depicted in Fig. 1. This allows applications to be executed at locations closer to the service requester, in comparison to the remote cloud server. The MEC servers also provide user content caching so that media-rich content can be delivered to users directly from the BS; thereby, reducing latency and improving user experience. In such a setting, significant volume data can be filtered and pre-processed before transmitting to the cloud server. In other words, this allows the offloading of computation and storage tasks from remote cloud servers and thus, reducing communication congestion. The MEC has been touted as a viable approach to meet the strict low-latency requirement in 5G networks [1].

For an MEC ecosystem to be established, security and privacy are two critical challenges that must be taken into account by network operators. In traditional cloud computing platform, data centers are relatively centralized. This is conducive to achiev-

Manuscript received November 4, 2018; revised January 3, 2019; accepted January 26, 2019. This work was supported in part by the National Natural Science Foundation of China under Grants 61802445, 61572379, and 61572370; in part by the Guangxi Key Laboratory of Trusted Software under Grant kx201709; and in part by the Fundamental Research Funds for the Central Universities under Grant CZY18034. The work of Kim-Kwang Raymond Choo was supported by the Cloud Technology Endowed Professorship. *(Corresponding author: Debiao He.)*

X. Jia is with the School of Mathematics and Statistics, South-Central University for Nationalities, Wuhan 430073, China, and also with the Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China (e-mail: xiaoyin.jia@mail.scuec.edu.cn).

D. He is with the Key Laboratory of Aerospace Information Security and Trusted Computing Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China (e-mail: hedebiao@163.com).

N. Kumar is with the Department of Computer Science and Engineering, Thapar University, Patiala 147003, India (e-mail: neeraj.kumar@thapar.edu).

K.-K. R. Choo is with the Department of Information Systems and Cyber Security and Department of Electrical and Computer Engineering, The University of Texas at San Antonio, San Antonio, TX 78249 USA (e-mail: raymond.choo@fulbrightmail.org).

Digital Object Identifier 10.1109/JSYST.2019.2896064

ing security and unified management. In the MEC deployment, however, MEC servers may be deployed by different service providers at the edge of the network; hence, increasing the risk of being compromised. Due to the open nature of the wireless networks, information transmitted over the communication channel may be eavesdropped, tampered, intercepted, or replayed by malicious attackers or unauthorized users; consequently, resulting in user privacy leakage.

Mutual authentication is one cryptographic mean to verify the identity of the communicating entities prior to further interaction, without sending sensitive user identity information required for authentication over the insecure channel [2]–[4]. This can be achieved using anonymous authenticated key agreement (AAKA) protocol, which prevents the disclosure of user private information while ensuring the authenticity of their identities, as well as producing a common session key to facilitate subsequent communication.

To design AAKA protocols in the MEC environment, there are several challenges that must be addressed due to MEC network-specific characteristics. For example, the authentication scheme must be sufficiently lightweight for deployment on mobile devices. In existing literature, lightweight authentication schemes were usually realized with symmetric cryptographic tools (e.g., message authentication code and symmetric encryption). However, as Wang and Wang [5] and Ding *et al.* [6] pointed out, symmetric cryptography is not adequate in ensuring user anonymity. In addition, traditional public key infrastructure (PKI) is also not suitable for MEC deployment due to its complex public key certificate management. Identity-based cryptography appears to be a desirable option, since the user's public key is based on his/her identity information (email address, social security, or driver license number, etc.). Another challenge in the MEC environment is that mobile users need to switch between multiple MEC servers frequently, for example when traveling between counties, cities, states, and/or countries. It is not only inconvenient but also insecure to transmit personal information to each MEC server for registration and login. Single-sign-on (SSO) approaches allow user's access to multiple servers without the need to re-register with different username and password for each server. Therefore, a user-friendly AAKA protocol for MEC should support SSO functionality. Moreover, due to the instability of the wireless networks, it is desirable that users and MEC servers can authenticate each other without involving an online trusted third party.

Our key contribution in this paper is the proposed identity-based AAKA protocol suitable for MEC environment, which is designed to achieve both user anonymity and un-traceability. The new protocol also achieves mutual authentication and secure key agreement with only one round message exchange. In our approach, the MEC servers do not hold private information of any mobile user and a mobile user can log on multiple MEC servers with only one registration with a trusted registration center (RC). Moreover, the protocol is carefully designed so that there is no need for a trusted third party during the authentication process.

In Section II, we will present related literature on the architecture, edge computing related security and privacy issues, and existing AAKA protocols. In Section III, we introduce relevant mathematical preliminaries. The network framework, security definition, and security model are described in Section IV. Section V describes the proposed AAKA protocol, whose security is analyzed in Section VI. Specifically, we prove the security of the protocol under the defined security model (presented in Section IV), as well as demonstrating how the new protocol meets all desirable security requirements and is resilient to various attacks. A comparison of security properties between the proposed scheme and schemes in [7] and [8] is also presented. We also evaluate the performance of the protocol, in terms of communication and computational costs of our protocol and compare it with those of [7] and [8]. Section VIII concludes the paper.

## II. Review of Literature

An MEC [9], [10] is closely related to fog computing (introduced by Cisco System in 2013) [11]–[13]. Both paradigms and mobile cloud computing (MCC) share the same goal; namely, to extend cloud services to the edge of the network and improve the quality of service in mobile networks, such as in IoT applications. A comparative summary of MEC, edge computing, and fog computing, in terms of security threats, challenges, and promising solutions is presented by Roman *et al.* [14]. Readers interested in security and privacy issues relating to fog computing, MCC, and MEC are also referred to [15]–[17] and [18] and [19], respectively. A commonality in these literature is the need for secure authentication mechanism. Here, we review a number of identity-based authentication protocols proposed for edge computing environment or mobile networks.

Yang and Chang [20] designed an identity-based AKA scheme for mobile devices, based on elliptic curve cryptosystem (ECC). However, Yoon and Yoo [21] demonstrated that the scheme is not secure against impersonation attack and does not provide perfect forward secrecy. Cao *et al.* [22] proposed a pairing-free identity-based AKA protocol with minimal message exchanges. However, similar to the scheme of Yang and Chang, Cao *et al.*'s scheme does not support user anonymity and un-traceability.

Tsai and Lo [7] proposed another identity-based authentication scheme for distributed MCC services. In their setting, mobile users and service providers register with a trusted third party (i.e., smart card generator–SCG), who produces long-term secret keys for each mobile user and service provider. Their protocol utilizes time consuming bilinear pairings, but the computation of bilinear mapping is executed by service providers, which usually have relatively powerful computing ability. The protocol is claimed to be privacy-aware. However, it was later pointed out by Jiang *et al.* [23] that it does not withstand the service provider impersonation attack and fails to achieve mutual authentication. Other design flaws are also pointed out. Jiang *et al.*, however, did not present any mitigation solution, although potential solutions were presented in [8], [24], and[25].

Yang *et al.* [26] proposed an efficient handover authentication scheme with user anonymity and un-traceability for MCC. In their protocol, a user is assigned a number of pseudo-IDs, as well as a family of secret keys corresponding to each pseudo-ID. This key pre-distribution process is executed by an Access Service Network-Gateway (ASN-GW). Their scheme is based on the elliptic curve cryptography and no bilinear pairing is required. However, the ASN-GW needs to generate a large number of pseudo-IDs for each registered user, and any mobile user has to store many pseudo-IDs and the corresponding secret keys. It is, hence, impractical for mobile devices with limited storage capacity.

Ibrahim [27] proposed an edge-fog authentication scheme for fog computing environment, which allows any fog user and fog node to authenticate each other. In their protocol, the secure channel between the registration authority and mobile users is established using PKI, while the communication between mobile users and fog nodes is protected using symmetric encryption. In their setting, fog nodes must store pre-generated secret keys of all fog users in its domain. This again is not practical, and does not scale well. In addition, the protocol does not guarantee anonymity and un-traceability.

He *et al.* [28] presented an anonymous mobile user authentication protocol for multiserver environment. They used self-certified public key cryptography in their scheme, which is essentially identity-based cryptography. Also recently in 2017, Xiong *et al.* [29] presented a privacy-aware authentication scheme for MCC services.

In general, it remains a challenging task to design secure and efficient privacy-preserving authentication protocols to be deployed at the edge of the network, as demonstrated by Wang *et al.* [30], [31].

## III. Mathematical Preliminaries

Compared with traditional public key schemes whose security based on number-theory assumptions, ECC offers better performance and achieves the same security level with shorter key size. For example, an ECC-based cryptographic scheme with a key length of 160 b can achieve the same security level with a 1024-b RSA. In addition, identity-based cryptosystem is often implemented by ECC. Here, is a brief review of ECC and bilinear map, as well as the complexity assumptions used in the proposed protocol.

### *A. Elliptic Curve Cryptosystem and Bilinear Pairings*

Let $p, q$ be large prime numbers. An elliptic curve $E/F_p$ is defined by the equation $y^2 = x^3 + ax + b \pmod{p}$ while $a, b, x, y \in F_p$. The set of all the points on the curve together with a "point at infinity" $O$ form an additive group under the point addition operation. Let $G$ be a subgroup of order $q$, and $P$ is a generator of $G$. Scalar multiplication is defined as $nP = P + P + \cdots + P$ ($n$ times), where $n \in Z_q$.

Let $G_T$ be a multiplicative cyclic group of the same order $q$. The map $e : G \times G \rightarrow G_T$ is said to be an admissible bilinear map if the following conditions hold.

1) *Bilinearity:* $e(aP, bQ) = e(P, Q)^{ab}$ for all $a, b \in Z_q$ and $P, Q \in G$.
2) *Non-degeneracy:* There exists a $P \in G$, such that $e(P, P) \neq 1_{G_T}$.
3) *Computability:* For all $P, Q \in G$, $e(P, Q)$ can be efficiently computed.

### *B. Complexity Assumptions*

Let $p, q, G, G_T, P, e$ be defined as mentioned in Section III-A. The mathematical problems listed in the following are known to be hard to solve. These complexity assumptions form the basis of the security of our protocol.

1) *Discrete logarithm (DL) problem:* Given an element $Q \in G$, find $x$ such that $Q = xP$.
2) *Computational Diffie–Hellman (CDH) problem:* Given two elements $aP, bP \in G$, where $a, b \in Z_q$ are unknown elements, compute $abP$.
3) *k-Modified bilinear inverse Diffie–Hellman (k-mBIDH) Problem:* Given $a_1, a_2, \ldots, a_k \in Z_q^*$ and $sP, \tau P$, $\frac{1}{s+a_1}P, \frac{1}{s+a_2}P, \ldots, \frac{1}{s+a_k}P \in G$, where $s, \tau \in Z_q$ are unknown elements, compute $e(P, P)^{\frac{\tau}{s+a^*}}$ for $a^* \notin \{a_1, a_2, \ldots, a_k\}$.

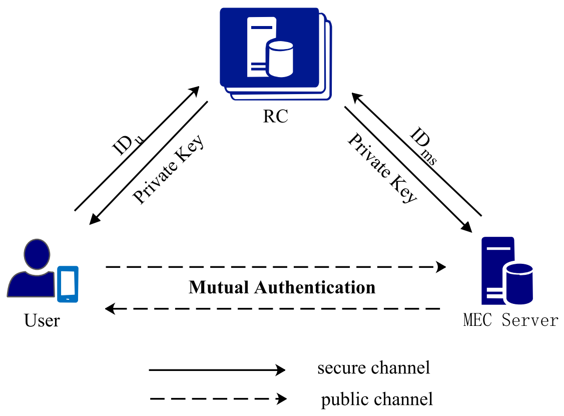

Fig. 2. Network framework.

## IV. System Framework and Security Model

### *A. Network Framework*

There are three types of entities in our proposed AAKA protocol. The trusted RC, the mobile user, and the MEC server. Every mobile user and MEC server should register with RC before they can enjoy or provide system services. The RC is only responsible for user registration, without participating in the mutual authentication process, so it can be placed on remote cloud servers. The RC issues long term secret keys for all mobile users or MEC servers according to their identities. The mutual authentication proceeds between a mobile user and any MEC server he/she wants to access, without the help of the RC. The network framework is depicted in Fig. 2.

### *B. Security Requirements*

According to the inherent characteristics of MEC environment, an AAKA protocol should satisfy the following security requirements.

1) *Mutual authentication:* Only registered mobile users and MEC servers are allowed in the MEC ecosystem and they can verify the legality of each other by executing the protocol.
2) *Session key agreement:* Successful execution of the protocol will generate a common session key shared by the mobile user and the MEC server for further communication, while any other user, including the RC, is unable to get any information about the session key.
3) *User anonymity:* The mobile user should be anonymous to everyone except for the RC and the MEC server being

accessed. Any adversary is unable to obtain user identity from the intercepted messages.

4) *Un-traceability:* Except for the specific MEC server been accessed, any adversary or system user cannot acquire the activities and behavior patterns of a mobile user from the intercepted messages.
5) *Perfect forward secrecy:* It is impossible for an adversary to learn about the session key in the previous session, even if he/she knows the long-term secret keys of both participants.
6) *SSO functionality:* To enjoy the services from multiple MEC servers, the mobile user only needs to register with the RC for once.
7) *No online RC:* It is not necessary for the RC to keep online all the time, i.e., after obtaining the private keys, the mobile user and the MEC server can achieve mutual authentication without the help of the RC.
8) *Resistance against various attacks:* The AAKA protocol should resist against regular attacks including impersonation attack, replay attack, stolen verifier attack, man-in-the-middle attack, etc.

### C. Security Model

The formal security of an identity-based AAKA protocol can be defined through a game played between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. Let $\Gamma$ be the protocol. $P$ denotes two participants: the mobile user $U$ and the MEC server MS. $\Pi_P^i$ means the oracle machine of $i$th instance of participant $P$. $\mathcal{A}$ interacts with $\Gamma$ by issuing a series of queries to the oracle machine adaptively and acquires the information he/she needs from the responses. The oracle queries issued by $\mathcal{A}$ are listed as follows.

1) $h(m)$: When $\mathcal{A}$ submits a message $m$ to the hash oracle, the oracle first checks if $m$ has been asked before. If yes, return the same value. Otherwise, randomly choose a number $r$ and return $r$ to $\mathcal{A}$.
2) $\text{Extract}(\text{ID})$: This query models $\mathcal{A}$'s ability of corrupting a legal entity and obtaining the private key of it. When $\mathcal{A}$ queries this oracle with identity ID, the oracle returns the private key corresponding to ID.
3) $\text{Send}(P^i, \text{Msg})$: This query models ability of the adversary $\mathcal{A}$ to launch an active attack. When $\mathcal{A}$ sends a message Msg to the oracle $\Pi_P^i$, the oracle responses with the result that would be output by the real protocol. $\mathcal{A}$ can start the protocol by issuing a $\text{Send}(P^i, \text{Start})$ query.
4) $\text{Reveal}(P^i)$: When $\mathcal{A}$ makes this query, the oracle returns the session key of instance $\Pi_P^i$ if it has been successfully produced, otherwise returns a $\perp$.
5) $\text{Execute}(U^i, \text{MS}^j)$: This models the passive eavesdropping over the public channel. When $\mathcal{A}$ issues such a query, the oracle runs the protocol between the instances $\Pi_U^i$ and $\Pi_{\text{MS}}^j$ according to the routines, and returns all the messages exchanged during the process.
6) $\text{Test}(\Pi_P^i)$: The adversary $\mathcal{A}$ can submit this query for only once. The oracle chooses a random bit $b \in \{0, 1\}$. If $b = 1$, the oracle returns the real session key $\text{sk}_U^i$. Otherwise, returns a random value of the same size.

Note that we replace the traditional Corrupt oracle with $\text{Extract}(\text{ID})$. In the identity-based cryptography, the long-term private key is exactly the output of $\text{Extract}(\text{ID})$.

TABLE I
NOTATIONS IN THE PROPOSED PROTOCOL

| Notations | Description |
|---|---|
| $U$ | Mobile user |
| $MS$ | MEC server |
| $RC$ | Registration center |
| $ID_{ms}$ | Identity of MEC server |
| $ID_u$ | Identity of mobile user |
| $p, q$ | Large prime number |
| $G$ | An additive cyclic group |
| $G_T$ | A multiplicative cyclic group |
| $P$ | Generator of $G$ |
| $s, \hat{s}$ | Private key of RC |
| $P_{pub}, \hat{P}_{pub}$ | Public key of RC |
| $SID_u$ | Private key of mobile user |
| $SID_{ms}$ | Private key of MEC server |
| $h_i(i = 0, 1, \ldots, 5)$ | Secure hash function |
| $r_u, x, y$ | Random numbers in $Z_q^*$ |
| $T_u$ | Timestamp of mobile user |
| $T_{ms}$ | Timestamp of MEC server |
| $SK$ | Session key |

*Partnership:* Two instances $\Pi_U^i$ and $\Pi_{\text{MS}}^j$ are said to be *partners* if:

1) $\Pi_U^i$ and $\Pi_{\text{MS}}^j$ exchange messages directly;
2) $\Pi_U^i$ and $\Pi_{\text{MS}}^j$ agree on the same session key sk;
3) there is no other instance accepts sk except for $\Pi_U^i$ and $\Pi_{\text{MS}}^j$.

*Freshness:* An instance $\Pi_P^i$ is *fresh* if the session key sk has been accepted and Reveal and Extract oracles have never been queried on $\Pi_P^i$ and its partner.

The following two security definitions come from the aforementioned game.

After receiving the response from Test query, $\mathcal{A}$ outputs a bit $b'$ as its guess of $b$. If $b' = b$, then $\mathcal{A}$ successfully breaking the semantic security of the protocol. The advantage of $\mathcal{A}$ is denoted as $\text{Adv}_\Gamma^{AKA}(\mathcal{A}) = |2Pr[b = b'] - 1|$.

*Definition 1 (AKA-security):* An AAKA protocol is said to be $AKA$-secure if for any efficient adversary $\mathcal{A}$, its advantage $\text{Adv}_\Gamma^{AKA}(\mathcal{A})$ is negligible.

If $\mathcal{A}$ can forge the messages transmitted during the aforementioned process on behalf of a participant and the forged message is accepted by its partner, we say that $\mathcal{A}$ successfully breaks the mutual authentication of the protocol. Let $E_{u-\text{ms}}$ denote the event that $\mathcal{A}$ successfully impersonates the user $U$ and produces a legal login message, and $E_{\text{ms}-u}$ denote the event that $\mathcal{A}$ produces a legal response message. The corresponding advantage is defined as $\text{Adv}_\Gamma^{MA}(\mathcal{A}) = Pr[E_{u-\text{ms}}] + Pr[E_{\text{ms}-u}]$.

*Definition 2 (MA-security):* An AAKA protocol is said to be $MA$-secure if for any efficient adversary $\mathcal{A}$, its advantage $\text{Adv}_\Gamma^{MA}(\mathcal{A})$ in the aforementioned game is negligible.

## V. PROPOSED SCHEME

In this section, we illustrate the proposed AAKA protocol in detail. Main notations used in the protocol are listed in Table I.

### A. System Setup

In the setup phase, the RC initializes all the system parameters needed in the protocol as follows.

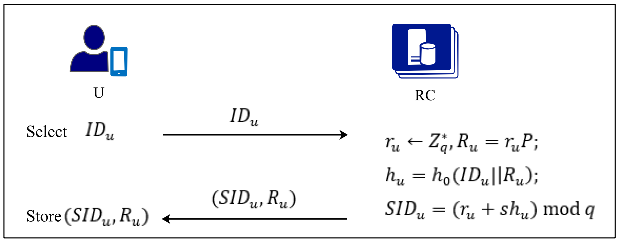

Fig. 3. Mobile user registration.

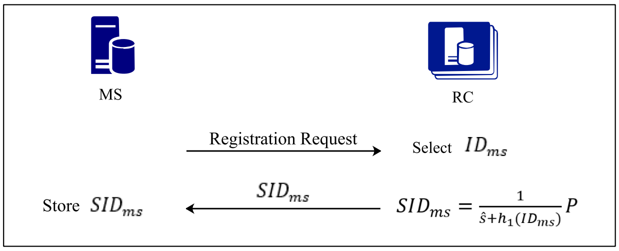

Fig. 4. MEC server registration.

1) RC selects a cyclic additive group $G$ over a nonsingular elliptic curve $E/F_p$, a multiplicative group $G_T$, both with the same order $q$. RC also chooses a bilinear map $e : G \times G \to G_T$, a generator $P$ of $G$ and computes $g = e(P, P)$.
2) RC randomly chooses $s, \hat{s} \in Z_q^*$, and computes $P_{\text{pub}} = sP, \hat{P}_{\text{pub}} = \hat{s}P$.
3) RC selects six secure hash functions $h_0 : \{0,1\}^* \times G \to Z_q^*$, $h_1 : \{0,1\}^* \to Z_q^*$, $h_2 : G_T \to \{0,1\}^* \times G \times G$, $h_3 : \{0,1\}^* \times G \times G \times \{0,1\}^* \to Z_q^*$, $h_4 : \{0,1\}^* \times \{0,1\}^* \times G \times G \times \{0,1\}^* \to Z_q^*$, $h_5 : G \times \{0,1\}^* \times \{0,1\}^* \times G \times G \to \{0,1\}^*$.
4) RC publishes the system parameters $(G, G_T, e, P, P_{\text{pub}}, \hat{P}_{\text{pub}}, g, h_0, h_1, h_2, h_3, h_4, h_5)$ and keeps $(s, \hat{s})$ as the master private key.

### B. User Registration

The mobile user $U$ submits his/her registration request to RC. RC extracts the long-term private key for $U$. $U$ and RC interact through a secure channel as follows.

1) $U$ selects his/her unique identity $\text{ID}_u$ and sends it to RC with the registration request.
2) On receiving the request of $U$, RC randomly chooses $r_u \in Z_q^*$, and computes $R_u = r_u P$, $h_u = h_0(\text{ID}_u||R_u)$, $SID_u = (r_u + sh_u) \mod q$.
3) RC sends $(R_u, SID_u)$ to $U$ as his/her private key.

The registration process is demonstrated in Fig. 3.

### C. MEC Server Registration

In this phase, MEC server MS registers with RC through a secure channel. RC extracts the long-term private key for MS. Details follow.

1) MS sends its registration request to RC.
2) RC selects a unique identity $\text{ID}_{\text{ms}}$ for MS, and computes $h_{\text{ms}} = h_1(\text{ID}_{\text{ms}})$, $SID_{\text{ms}} = \frac{1}{\hat{s}+h_{\text{ms}}}P$.
3) RC sends $SID_{\text{ms}}$ to MS.

The aforementioned process is depicted in Fig. 4.

### D. Mutual Authentication

In this phase, the mobile user $U$ and the MEC server MS authenticate each other and negotiate a common session key SK for further communication.

1) $U$ randomly selects a number $x \in Z_q^*$, and calculates $g_x = g^x$, $X = xP$, $M = x(\hat{P}_{\text{pub}} + h_1(\text{ID}_{\text{ms}})P)$, $N = h_2(g_x) \oplus (\text{ID}_u||R_u||X)$, $\sigma = SID_u + xh_3(\text{ID}_u||R_u||X||T_u)$, where $T_u$ is the current timestamp. $U$ sends $(M, N, \sigma, T_u)$ to MS via a public channel.
2) On receiving the login request from $U$, MS first checks if the timestamp is fresh. If not, MS terminates the session. Otherwise, MS calculates $g'_x = e(M, SID_{\text{ms}})$, $\text{ID}'_u||R'_u||X' = h_2(g'_x) \oplus N$, $W = R'_u + h_0(\text{ID}'_U||R'_u)P_{\text{pub}}$. MS then checks if the equation $\sigma P = W + h_3(\text{ID}'_u||R'_u||X'||T_u)X'$ holds. If not, MS rejects the request and aborts the session. Otherwise, MS chooses a random number $y \in Z_q^*$, computes $Y = yP$, $t = h_4(\text{ID}'_u||\text{ID}_{\text{ms}}||X'||Y||T_{\text{ms}})$, $K_{\text{ms}-u} = y(tX + W)$, where $T_{\text{ms}}$ is the current timestamp. MS sets the session key $\text{SK}_{\text{ms}-u} = h_5(K_{\text{ms}-u}||\text{ID}'_u||\text{ID}_{\text{ms}}||X'||Y)$ and sends $(t, Y, T_{\text{ms}})$ to $U$.
Notice that we have $\text{ID}_u = \text{ID}'_u$, $R_u = R'_u$, $X = X'$ from the following two equations:

$$
\begin{aligned}
g'_x &= e(M, SID_{\text{ms}}) \\
&= e\left(x(\hat{P}_{\text{pub}} + h_1(\text{ID}_{\text{ms}})P), \frac{1}{\hat{s} + h_1(\text{ID}_{\text{ms}})}P\right) \\
&= e\left(x(\hat{s}P + h_1(\text{ID}_{\text{ms}})P), \frac{1}{\hat{s} + h_1(\text{ID}_{\text{ms}})}P\right) \\
&= e\left(x(\hat{s} + h_1(\text{ID}_{\text{ms}}))P, \frac{1}{\hat{s} + h_1(\text{ID}_{\text{ms}})}P\right) \\
&= e(P, P)^{x(\hat{s}+h_1(\text{ID}_{\text{ms}}))\cdot\frac{1}{\hat{s}+h_1(\text{ID}_{\text{ms}})}} \\
&= e(P, P)^x \\
&= g^x
\end{aligned}
\tag{1}
$$

and

$$
\begin{aligned}
\sigma P &= (SID_u + xh_3(\text{ID}_u||R_u||X||T_u))P \\
&= ((r_u + sh_u) + xh_3(\text{ID}_u||R_u||X||T_u))P \\
&= r_u P + h_0(\text{ID}_u||R_u)P_{\text{pub}} \\
&\quad + h_3(\text{ID}_u||R_u||X||T_u)X \\
&= W + h_3(\text{ID}_u||R_u||X||T_u)X.
\end{aligned}
\tag{2}
$$

3) On receiving the response from MS, $U$ first checks if $T_{\text{ms}}$ is fresh. If not, $U$ terminates the session. Otherwise, $U$ continues to check if $t = h_4(\text{ID}_u||\text{ID}_{\text{ms}}||X||Y||T_{\text{ms}})$

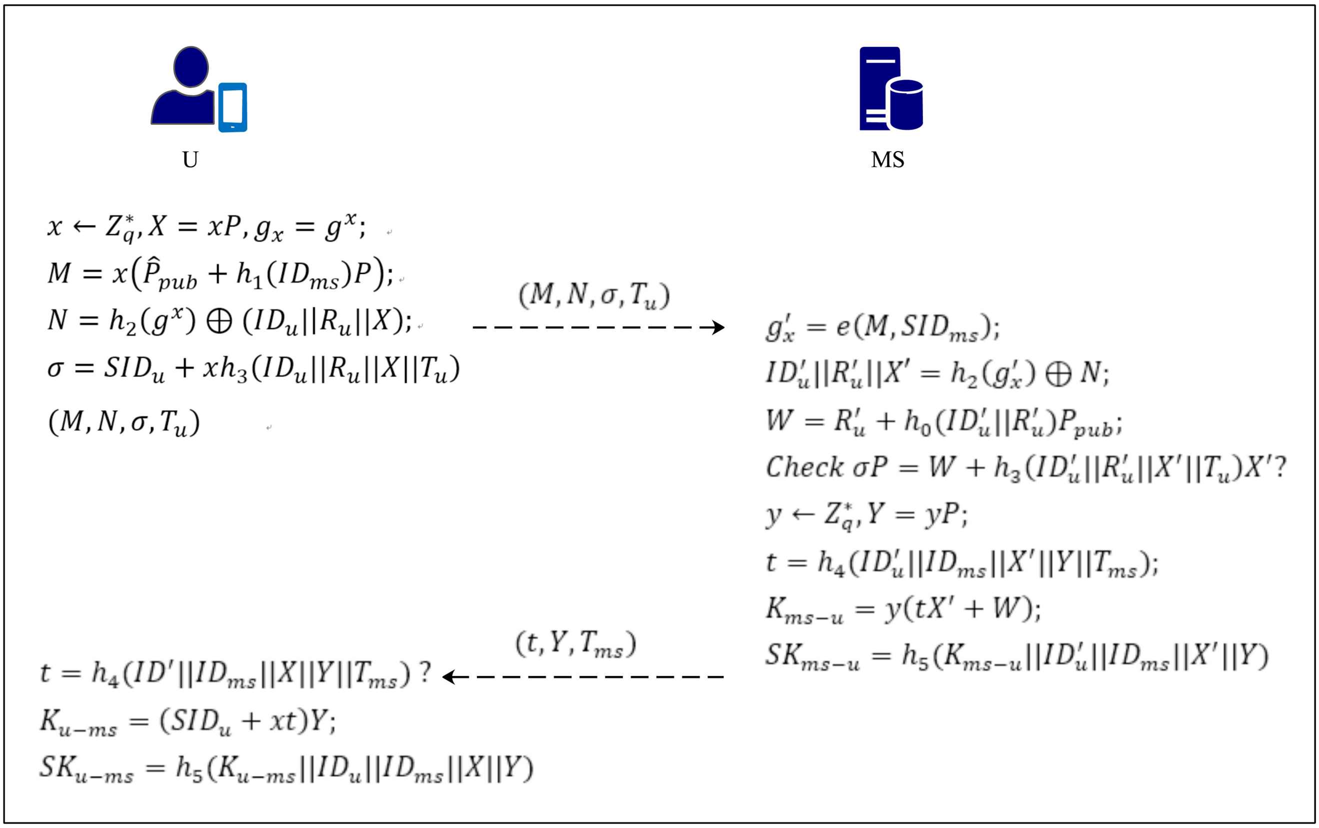
U
MS
$x \leftarrow Z_q^*, X = xP, g_x = g^x;$
$M = x(\hat{P}_{pub} + h_1(ID_{ms})P);$
$N = h_2(g^x) \oplus (ID_u||R_u||X);$
$\sigma = SID_u + xh_3(ID_u||R_u||X||T_u)$
$(M, N, \sigma, T_u)$
$(M, N, \sigma, T_u)$
$g_x' = e(M, SID_{ms});$
$ID_u'||R_u'||X' = h_2(g_x') \oplus N;$
$W = R_u' + h_0(ID_u'||R_u')P_{pub};$
$Check\ \sigma P = W + h_3(ID_u'||R_u'||X'||T_u)X'?$
$y \leftarrow Z_q^*, Y = yP;$
$t = h_4(ID_u'||ID_{ms}||X'||Y||T_{ms});$
$K_{ms-u} = y(tX' + W);$
$SK_{ms-u} = h_5(K_{ms-u}||ID_u'||ID_{ms}||X'||Y)$
$(t, Y, T_{ms})$
$t = h_4(ID'||ID_{ms}||X||Y||T_{ms})$ ?
$K_{u-ms} = (SID_u + xt)Y;$
$SK_{u-ms} = h_5(K_{u-ms}||ID_u||ID_{ms}||X||Y)$

Fig. 5. Mutual authentication.

holds. If not, $U$ aborts the session. Otherwise, $U$ calculates $K_{u-\mathrm{ms}} = (SID_u + xt)P$, and sets the session key $\mathrm{SK}_{u-\mathrm{ms}} = h_5(K_{u-\mathrm{ms}}||\mathrm{ID}_u||\mathrm{ID}_{\mathrm{ms}}||X||Y)$.

From the aforementioned process we can see

$$\begin{aligned} K_{\mathrm{ms}-u} &= y(tX' + W) \\ &= y(h_4(\mathrm{ID}_u'||\mathrm{ID}_{\mathrm{ms}}||X||Y||T_{\mathrm{ms}})xP + r_uP \\ &\quad + h_0(\mathrm{ID}_u'||R_u')P_{\mathrm{pub}}) \\ &= h_4(\mathrm{ID}_u||\mathrm{ID}_{\mathrm{ms}}||X||Y||T_{\mathrm{ms}})xyP + r_uyP + h_uysP \\ &= (h_4(\mathrm{ID}_u||\mathrm{ID}_{\mathrm{ms}}||X||Y||T_{\mathrm{ms}})xy + r_uy + syh_u)P \end{aligned} \tag{3}$$

and

$$\begin{aligned} K_{u-\mathrm{ms}} &= (SID_u + xt)Y \\ &= (r_u + sh_u + xh_4(\mathrm{ID}_u||\mathrm{ID}_{\mathrm{ms}}||X||Y||T_{\mathrm{ms}}))yP \\ &= (r_uy + syh_u + h_4(\mathrm{ID}_u||\mathrm{ID}_{\mathrm{ms}}||X||Y||T_{\mathrm{ms}})xy)P. \end{aligned} \tag{4}$$

So, $K_{\mathrm{ms}-u} = K_{u-\mathrm{ms}}$, and thus $\mathrm{SK}_{\mathrm{ms}-u} = \mathrm{SK}_{u-\mathrm{ms}}$. The coincidence of the produced session key are guaranteed.

The authentication phase is depicted in Fig. 5.

## VI. Security Analysis

We first prove the proposed protocol is $AKA$-secure and $MA$-secure under the security model defined in Section IV-C, provided that DL, CDH, and $k$-mBIDH assumptions hold and all the hash functions are simulated as random oracles. Then, we explain how the proposed protocol satisfies the security requirements mentioned in Section IV-B.

### A. Provable Security

To illustrate that the proposed protocol provides mutual authentication, we first prove that $\mathcal{A}$ cannot forge a legal login message of some target mobile user, even if he/she can extract private keys of other mobile users and MEC servers. As the following lemma shows.

*Lemma 1:* Suppose there is an adversary $\mathcal{A}$, which can generate a valid login message in the protocol with nonnegligible probability $\epsilon$, then there is challenger $\mathcal{C}$, which can solve the DL problem with probability

$$\epsilon_1 \geq \left(1 - \frac{1}{q}\right)^{q_e} \left(1 - \frac{1}{q_e}\right) \left(1 - \frac{1}{q_s}\right) \frac{1}{q_e} \epsilon$$

where $q_e, q_s$ denote the upper bound of the Send and Extract queries made by $\mathcal{A}$, respectively.

*Proof:* Suppose there is a DL instance $P, Q = sP \in G$, where $s$ is unknown to $\mathcal{C}$. $\mathcal{C}$ runs the Setup algorithm, generates system parameters $p, q, G, G_T, e, P, \hat{s}$, and sets $P_{\mathrm{pub}} = Q, \hat{P}_{\mathrm{pub}} = \hat{s}P$. $\mathcal{C}$ maintains six hash lists $L_{h_i}$ $(i = 0, 1, 2, 3, 4, 5)$ to record the outputs of the random oracles, a user list $L_u$ and an MEC server list $L_{\mathrm{ms}}$ to record the private keys returns by the Extract oracle, respectively, and a list $L_s$ to record information exchanged in the specific instance during the simulation. All the lists are initially empty. $\mathcal{C}$ publishes two groups of identities $\mathrm{ID}_U = \{\mathrm{ID}_{u_1}, \mathrm{ID}_{u_2}, \ldots, \mathrm{ID}_{u_{q_e}}\}$ and $\mathrm{ID}_{\mathrm{MS}} = \{\mathrm{ID}_{\mathrm{ms}_1}, \mathrm{ID}_{\mathrm{ms}_2}, \ldots, \mathrm{ID}_{\mathrm{ms}_{q_e}}\}$. $\mathcal{A}$ picks an $\mathrm{ID}_{u_i^*}$ from $\mathrm{ID}_U$ as the target user.

$\mathcal{C}$ runs $\mathcal{A}$ as a subroutine and answers $\mathcal{A}$'s queries as follows. Without loss of generality, we assume that $\mathcal{A}$ always queries identities in the aforementioned group and all the needed hash oracles have been queried before other related oracles being queried.

1) $h_i(m)$*:* When $\mathcal{A}$ makes a hash query $h_i(m)$ on input $m$, $\mathcal{C}$ first looks up the list $L_{h_i}$ for the entry $(m, h_i(m))$ and returns $h_i(m)$ to $\mathcal{A}$ if the entry exists. Otherwise, $\mathcal{C}$ randomly chooses $r$ in the domain of $h_i()$ and sets $h_i(m) = r$. $\mathcal{C}$ returns $h_i(m)$ to $\mathcal{A}$ and adds $(m, h_i(m))$ to $L_{h_i}$.
2) $Extract\,(ID)$: There are two kinds of Extract queries $\mathcal{A}$ can make, as listed in the following.
   a) $Extract\,(ID_{u_i})$: If $u_i \neq u_i^*$, $\mathcal{C}$ randomly chooses $r_{u_i}, h_{u_i} \in Z_q^*$, and sets $R_{u_i} = r_{u_i}P - h_{u_i}P_{\text{pub}}$. $\mathcal{C}$ looks up the list $L_{h_0}$, if there exists an entry for $(\mathrm{ID}_{u_i}, R_{u_i})$ and $h_0(\mathrm{ID}_{u_i}||R_{u_i}) \neq h_{u_i}$, then $\mathcal{C}$ aborts the simulation. Otherwise, $\mathcal{C}$ sets $SID_{u_i} = r_{u_i}$. Obviously $(SID_{u_i}, R_{u_i})$ is a valid private key, since $SID_{u_i}P = R_{u_i} + h_{u_i}P_{\text{pub}}$. $\mathcal{C}$ returns $(R_{u_i}, SID_{u_i})$ to $\mathcal{A}$, and inserts $(\mathrm{ID}_{u_i}, h_{u_i}, R_{u_i}, SID_{u_i})$ and $(\mathrm{ID}_{u_i}, R_{u_i}, h_{u_i})$ into the list $L_u$ and $L_{h_0}$, respectively. If $u_i = u_i^*$, $\mathcal{C}$ simply rejects the query and aborts the game.
   b) $Extract\,(ID_{\mathrm{ms}_j})$: When $\mathcal{A}$ makes such a query on input $\mathrm{ID}_{\mathrm{ms}_j}$, $\mathcal{C}$ retrieves $L_{h_1}$ for $(\mathrm{ID}_{\mathrm{ms}_j}, h_{\mathrm{ms}_j})$, and computes $SID_{\mathrm{ms}_j} = \frac{1}{\hat{s}+h_{\mathrm{ms}_j}}P$. $\mathcal{C}$ returns $SID_{\mathrm{ms}_j}$ to $\mathcal{A}$, and inserts $(\mathrm{ID}_{\mathrm{ms}_j}, SID_{\mathrm{ms}_j})$ into $L_{\mathrm{ms}}$.
3) $Send\,(P^i, Msg)$: According to the specification of the protocol, $\mathcal{A}$ can launch three types of Send queries to simulate the ability of active attack.
   a) $Send\,(U_i^k, Start)$: If $\mathcal{A}$ launches such a query, $\mathcal{C}$ checks if $u_i = u_i^*$. If yes, $\mathcal{C}$ returns a "⊥" and aborts. Otherwise, $\mathcal{C}$ continues to check if there is an entry for $ID_{u_i}$ in $L_u$. If yes, $\mathcal{C}$ extracts $SID_{u_i}$ from $L_u$, or else $\mathcal{C}$ generates a private key $SID_{u_i}$ as it does in Extract$(\mathrm{ID}_{u_i})$ query, and adds $(\mathrm{ID}_{u_i}, h_{u_i}, R_{u_i}, SID_{u_i})$ to $L_u$. With this private key $SID_{u_i}$, $\mathcal{C}$ randomly chooses $x \in Z_q^*$, computes $X = xP$ and $M, N, \sigma, T_{u_i}$ as described in the protocol. $\mathcal{C}$ records $(\mathrm{ID}_{u_i}, k, x, X)$ in $L_s$, and returns $(M, N, \sigma, T_{u_i})$ to $\mathcal{A}$.
   b) $Send\,(MS_j^l, (M, N, \sigma, T_u))$: $\mathcal{C}$ first checks if $\mathrm{ID}_{\mathrm{ms}_j}$ in the $L_{\mathrm{ms}}$ list. If not, $\mathcal{C}$ generates a private key $SID_{\mathrm{ms}_j}$ for $\mathrm{ms}_j$ as it does in the Extract$(\mathrm{ID}_{\mathrm{ms}_j})$ query, and adds $(\mathrm{ID}_{\mathrm{ms}_j}, h_{\mathrm{ms}_j}, SID_{\mathrm{ms}_j})$ to $L_{\mathrm{ms}}$. $\mathcal{C}$ then calculates $g_x = e(M, SID_{\mathrm{ms}})$ and extracts $\mathrm{ID}_{u_i}, R_{u_i}, X$ via $\mathrm{ID}_{u_i}||R_{u_i}||X = N \oplus h_2(g_x)$. $\mathcal{C}$ computes $W = R_{u_i} + h_0(\mathrm{ID}_{u_i}||R_{u_i})P_{\text{pub}}$, and verifies if $\sigma P = W + h_3(\mathrm{ID}_{u_i}||R_{u_i}||X||T_{u_i})X$ holds. If not, $\mathcal{C}$ rejects the message. Otherwise, if $u_i \neq u_i^*$, $\mathcal{C}$ randomly chooses $y \in Z_q^*$ and produces the response $(t, Y, T_{\mathrm{ms}})$ as in the real protocol and returns it to $\mathcal{A}$. If $u_i = u_i^*$, then $\mathcal{A}$ successfully forge a legal login message and wins the game.
   c) $Send\,(U_i^k, (t, Y, T_{ms}))$: On receiving this query, $\mathcal{C}$ retrieves $(\mathrm{ID}_{u_i}, k, x, X)$ in $L_s$ and verifies if $t = h_4(\mathrm{ID}_{u_i}||\mathrm{ID}_{\mathrm{ms}_j}||X||Y||T_{\mathrm{ms}_j})$ holds. If not, $\mathcal{C}$ rejects the message. Otherwise, $\mathcal{C}$ authenticated $\mathcal{A}$.
4) $Reveal\,(\Pi_P^i)$. $\mathcal{C}$ responses with the correct session key SK if SK is accepted. Otherwise returns a "⊥."

Suppose that $\mathcal{A}$ successfully submits a valid login message, e.g., $\mathcal{A}$ issues a Send$(\mathrm{MS}_j^l, (M, N, \sigma, T_{u_i}))$ query with $u_i = u_i^*$ and passes the verification. Then, there is an equation

$$\sigma P = R_{u_i^*} + h_{u_i^*}P_{\text{pub}} + vX \tag{5}$$

where $v{=}h_3(\mathrm{ID}_{u_i}||R_{u_i}||X||T_{u_i})$. By applying forking lemma, $\mathcal{A}$ and $\mathcal{C}$ proceeds the aforementioned procession again, with the same input randomness and different hash oracle responses, $\mathcal{A}$ may submit another legal message $(M', N', \sigma', T'_{u_i})$, such that

$$\sigma' P = R_{u_i^*} + h'_{u_i^*}P_{\text{pub}} + vX. \tag{6}$$

By (5) and (6), we have the following:

$$(\sigma - \sigma')P = (h_{u_i^*} - h'_{u_i^*})P_{\text{pub}} = (h_{u_i^*} - h'_{u_i^*})sP.$$

$\mathcal{C}$ outputs $(\sigma - \sigma')(h_{u_i^*} - h'_{u_i^*})^{-1} \mod q$ as the solution to the DL problem.

In order to evaluate the advantage of $\mathcal{C}$, we define the following events.

1) $E_1$: The simulation process was not aborted in midway.
2) $E_2$: $\mathcal{A}$ submits a Send$(\mathrm{MS}_j^l, \mathrm{Msg})$ query, where $\mathrm{Msg} = (M, N, \sigma, T)$ is a legal login message of user $U_i$ and Extract$(\mathrm{ID}_{u_i})$ has never been queried before.
3) $E_3$: In the forged login message, $\mathrm{ID}_{u_i} = \mathrm{ID}_{u_i^*}$.

$\mathcal{C}$ will abort the simulation in three cases.

1) There is a $h_0$ hash collusion in the Extract$(\mathrm{ID}_{u_i})$ query, Since $r_{u_i}, h_{u_i}$ are all selected randomly, the probability is $\frac{1}{q}$.
2) $\mathcal{A}$ queries Extract$(\mathrm{ID}_{u_i^*})$. The probability is $\frac{1}{q_e}$.
3) $\mathcal{A}$ issues a Send$(U_i^k, \mathrm{Start})$ query with $u_i = u_i^*$. The probability is $\frac{1}{q_s}$.

Therefore

$$Pr[E_1] = \left(1 - \frac{1}{q}\right)^{q_e}\left(1 - \frac{1}{q_e}\right)\left(1 - \frac{1}{q_s}\right).$$

It is obvious that

$$Pr[E_2|E_1] \geq \epsilon$$

and

$$Pr[E_3|E_2 \wedge E_1] = \frac{1}{q_e}.$$

Suppose the probability that $\mathcal{C}$ solves DL problem is $\epsilon_1$. From the aforementioned analysis, we have

$$\begin{aligned}\epsilon_1 &= Pr[E_1 \wedge E_2 \wedge E_3]\\ &= Pr[E_1]Pr[E_2|E_1]Pr[E_3|E_2 \wedge E_1]\\ &= \left(1 - \frac{1}{q}\right)^{q_e}\left(1 - \frac{1}{q_e}\right)\left(1 - \frac{1}{q_s}\right)\frac{1}{q_e}\epsilon\end{aligned} \tag{7}$$

as desired.

Next, we show that it is infeasible for an adversary $\mathcal{A}$ to forge a valid response message of a MEC server, even if he/she can corrupt other mobile users and MEC servers. ■

*Lemma 2:* Suppose there is an adversary $\mathcal{A}$, which can impersonate a MEC server and forge a response message with nonnegligible probability $\epsilon$, then there is challenger $\mathcal{C}$, which can solve the $k$-mBIDH problem with probability

$$\epsilon_1 \geq \left(1 - \frac{1}{q}\right)^{q_e} \left(1 - \frac{1}{q_e}\right) \left(1 - \frac{1}{q_s}\right) \frac{1}{q_{h_3}} \epsilon$$

where $q_{h_3}, q_s, q_e$ denotes the upper bound of the $h_3$, Send, and Extract queries made by $\mathcal{A}$, respectively.

*Proof:* Suppose there is a $k$-mBIDH instance $P, sP, \tau P$, $\{e_1, e_2, \ldots, e_k \in Z_q^*\}$, and $\frac{1}{s+e_1}P, \frac{1}{s+e_2}P, \ldots, \frac{1}{s+e_k}P$, where $s, \tau$ is unknown to $\mathcal{C}$. $\mathcal{C}$ tries to compute $e(P,P)^{\frac{\tau}{s+e^*}}$ for some $e^* \notin \{e_1, e_2, \ldots, e_k\}$. Here, we assume that $k \geq q_e$.

$\mathcal{C}$ runs the setup algorithm, generates system parameters $p, q, G, G_T, P$, and sets $P_{\text{pub}} = sP$. $\mathcal{C}$ maintains six hash lists $L_{h_i}$ $(i = 0, 1, 2, 3, 4, 5)$, a user list $L_u$, a MEC server list $L_{\text{ms}}$. All lists are initially empty. $\mathcal{C}$ publishes two groups of identities $\text{ID}_U = \{\text{ID}_{u_1}, \text{ID}_{u_2}, \ldots, \text{ID}_{u_{qe}}\}$ and $\text{ID}_{\text{MS}} = \{\text{ID}_{\text{ms}_1}, \text{ID}_{\text{ms}_2}, \ldots, \text{ID}_{\text{ms}_{q_e}}\}$. $\mathcal{A}$ picks an $\text{ID}_{\text{ms}_j^*}$ from $\text{ID}_{\text{MS}}$ as the target server. Without loss of generality, we assume that all the needed hash oracles have been asked before other related oracles being queried.

The $h_i$ $(i = 0, 2, 3, 4, 5)$, Reveal, and Test oracles are simulated in the same way with lemma 1. $h_1$, Extract, and Send oracles are simulated as follows.

1) $h_1(\text{ID}_{\text{ms}_j})$: On receiving this query, $\mathcal{C}$ checks if $\text{ID}_{\text{ms}_j^*} = \text{ID}_{\text{ms}_j}$. If not, $\mathcal{C}$ sets $h_1(\text{ID}_{\text{ms}_j}) = e_j$. Otherwise, $h_1(\text{ID}_{\text{ms}_j}) = e^*$. $\mathcal{C}$ inserts $(\text{ID}_{\text{ms}_j}, h_1(\text{ID}_{\text{ms}_j}))$ into $L_{h_1}$.
2) $Extract\,(ID)$: There are two kinds of Extract queries $\mathcal{A}$ can make, as listed below.
    a) $Extract\,(ID_{u_i})$: $\mathcal{C}$ randomly chooses $r_{u_i}, h_{u_i} \in Z_q^*$, and sets $R_{u_i} = r_{u_i}P - h_{u_i}P_{\text{pub}}$. $\mathcal{C}$ looks up the list $L_{h_0}$, if there exists an entry for $(\text{ID}_{u_i}, R_{u_i})$ and $h_0(\text{ID}_{u_i}||R_{u_i}) \neq h_{u_i}$, then $\mathcal{C}$ aborts the simulation. Otherwise, $\mathcal{C}$ sets $SID_{u_i} = r_{u_i}$ and returns $(R_{u_i}, SID_{u_i})$ to $\mathcal{A}$, and inserts $(\text{ID}_{u_i}, h_{u_i}, R_{u_i}, SID_{u_i})$ and $(\text{ID}_{u_i}, R_{u_i}, h_{u_i})$ into the list $L_u$ and $L_{h_0}$, respectively.
    b) $Extract(ID_{ms_j})$: If $\text{ID}_{\text{ms}_j} \neq \text{ID}_{\text{ms}_j^*}$, $\mathcal{C}$ sets $h_1(\text{ID}_{\text{ms}_j}) = e_j$, $SID_{\text{ms}_j} = \frac{1}{s+e_j}P$. $\mathcal{C}$ returns $SID_{\text{ms}_j}$ to $\mathcal{A}$, and inserts $(\text{ID}_{\text{ms}_j}, e_j)$ into $L_{h_1}$ and $(\text{ID}_{\text{ms}_j}, SID_{\text{ms}_j})$ into $L_{\text{ms}}$. If $\text{ID}_{\text{ms}_j} = \text{ID}_{\text{ms}_j^*}$, $\mathcal{C}$ sets $h_1(\text{ID}_{\text{ms}_j^*}) = e^*$, adds $(\text{ID}_{\text{ms}_j^*}, h_1(\text{ID}_{\text{ms}_j^*}))$ into $L_{h_1}$, and returns a "$\perp$."
3) $Send(P^i, Msg)$: According to the specification of the protocol, $\mathcal{A}$ can launch three types of Send queries to simulate the ability of active attack.
    a) $Send(U_i^k, Start)$: If the partner $U_i$'s partner is $\text{MS}_j^*$, $\mathcal{C}$ sets $M = \tau P$, and computes $N, \sigma, T_{u_i}$ in a regular way and returns $(M, N, \sigma, T_{u_i})$. Otherwise, $\mathcal{C}$ extracts $SID_{u_i}$ from $L_u$, and generates a login message according to the rules in the protocol and records $(\text{ID}_{u_i}, k, x, X)$ in $L_s$.
    b) $Send(MS_j^l, (M, N, \sigma, T_u))$: $\mathcal{C}$ first checks if $\text{ID}_{\text{ms}_j} = \text{ID}_{\text{ms}_j^*}$. If not, $\mathcal{C}$ extracts $SID_{\text{ms}_j}$ from $L_{\text{ms}_j}$ and verifies the validity of the signature. $\mathcal{C}$ then randomly chooses $y \in Z_q^*$, computes $t, Y, T_{\text{ms}_j}$ as the protocol describes, and returns it to $\mathcal{A}$. If not, $\mathcal{C}$ rejects the message.
    c) $Send(U_i^k, (t, Y, T_{ms}))$: On receiving this query, $\mathcal{C}$ retrieves $(\text{ID}_{u_i}, k, x, X)$ in $L_s$ and verifies if $t = h_4(\text{ID}_{u_i}||\text{ID}_{\text{ms}_j}||X||Y||T_{\text{ms}_j})$ holds. If not, $\mathcal{C}$ rejects the message. Otherwise, $\mathcal{C}$ authenticated $\mathcal{A}$. And what is more, if the partner is $\Pi_{\text{ms}_j^*}^l$, then $\mathcal{A}$ successfully forge a response message.

At the end of the simulation, suppose $\mathcal{A}$ submits a $\text{Send}(U_i^k, \text{Msg})$ with a legal response message $(t, Y, T_{\text{ms}})$, and the partner is $\Pi_{\text{ms}_j^*}^l$, then $\mathcal{C}$ can solve the $k$-mBIDH problem in the following way. $\mathcal{C}$ randomly picks a tuple $(g_x, h_2(g_x))$ in $L_{h_2}$ and outputs $g_x$ as the solution of $k$-mBIDH problem. This is because if $\mathcal{A}$ submits a valid authenticator $t = h_4(\text{ID}_{u_i}||\text{ID}_{\text{ms}_j^*}||X||Y)$, he/she must have recover correct $\text{ID}_{u_i}$ from $\text{ID}_{u_i}||R_{u_i} = h_2(g_x) \oplus N$, thus he/she must have queried $g_x$ on $h_2()$ oracle. And we have

$$g_x = e\left(M, SID_{\text{ms}_j^*}\right) = e\left(\tau P, \frac{1}{s+e^*}P\right) = e(P,P)^{\frac{\tau}{s+e^*}}.$$

In order to analyze the advantage of $\mathcal{C}$, we define the following events.

1) $E_1$: The simulation is not aborted.
2) $E_2$: $\mathcal{A}$ submits a $\text{Send}(U_i^k, \text{Msg})$ query, where $\text{Msg} = (t, Y, T_{\text{ms}})$ is a legal response message of MEC server $\text{ms}_j$ and $\text{Extract}(\text{ID}_{\text{ms}_j})$ has never been queried before.
3) $E_3$: $\text{ID}_{\text{ms}_j} = \text{ID}_{\text{ms}_j^*}$.
4) $E_4$: $\mathcal{C}$ picked a correct tuple.

From the aforementioned analysis we can see that $Pr[E_1] = (1 - \frac{1}{q})^{q_e}(1 - \frac{1}{q_e})(1 - \frac{1}{q_s})$, $Pr[E_2|E_1] \geq \epsilon$, $Pr[E_3 |E_2 \wedge E_1] = \frac{1}{q_e}$, $Pr[E_4|E_3 \wedge E_2 \wedge E_1] = \frac{1}{q_{h\,2}}$. Therefore

$$\begin{aligned} & Pr[E_1 \wedge E_2 \wedge E_3 \wedge E_4] \\ & = Pr[E_4|E_3 \wedge E_2 \wedge E_1] Pr[E_3|E_2 \wedge E_1] Pr[E_2|E_1] Pr[E_1] \\ & \geq \left(1 - \frac{1}{q}\right)^{q_e} \left(1 - \frac{1}{q_e}\right) \left(1 - \frac{1}{q_s}\right) \frac{1}{q_{h_2}} \epsilon \end{aligned}$$

which concludes the proof. ■

Lemmas 1 and 2 illustrate that it is infeasible for an efficient adversary to produce a legal login or response message. In other words, $U$ and the MS can authenticate each other by executing the protocol. Therefore, we get the following theorem.

*Theorem 1:* The proposed protocol is MA-secure suppose DL problem and $k$-mBIDH problem is hard.

The following theorem shows that the proposed protocol is AKA-secure provided that CDH problem is hard.

*Theorem 2:* Suppose there is an adversary $\mathcal{A}$, which wins the AKA game and outputs a correct $b' = b$ with nonnegligible probability, then there is a challenger $\mathcal{C}$, which can solve the CDH problem with nonnegligible probability.

*Proof:* Suppose $\mathcal{A}$ wins the AKA game with an advantage $\epsilon$. We define the following events.

1) $E_{sk}$: $\mathcal{A}$ gets a correct session key in the response of Test query.
2) $E_U$: A Test query to the instance $\Pi_{U_i}$ is successfully invoked.
3) $E_{MS}$: A Test query to the instance $\Pi_{\text{MS}_j}$ is successfully invoked.

4) $E_{u-ms}$: $\mathcal{A}$ successfully breaks the user-to-MEC authentication.

5) $E_{ms-u}$: $\mathcal{A}$ successfully breaks the MEC-top-user authentication.

Since the probability that $\mathcal{A}$ outputs a correct $b$ is at least 1/2, so we have $Pr[E_{\text{sk}}] \geq \frac{\epsilon}{2}$. From the constraint on the Test query, the following inequality holds:

$$\begin{aligned} Pr[E_{\text{sk}}] &= Pr[E_{\text{sk}} \wedge E_U] + Pr[E_{\text{sk}} \wedge E_{\text{MS}} \neg E_{u-\text{ms}}] \\ &\quad + Pr[E_{\text{sk}} \wedge E_{\text{MS}} \wedge \neg E_{u-\text{ms}}] \\ &\leq Pr[E_{\text{sk}} \wedge E_U] + Pr[E_{u-\text{ms}}] \\ &\quad + Pr[E_{\text{sk}} \wedge E_{\text{MS}} \wedge \neg E_{u-\text{ms}}]. \end{aligned}$$

Then, we have

$$\begin{aligned} &Pr[E_{\text{sk}} \wedge E_U] + Pr[E_{\text{sk}} \wedge E_{\text{MS}} \wedge \neg E_{u-\text{ms}}] \\ &\geq Pr[E_{\text{sk}}] - Pr[E_{u-\text{ms}}] \\ &\geq \frac{\epsilon}{2} - Pr[E_{u-\text{ms}}]. \end{aligned}$$

Since $Pr[E_{\text{MS}} \wedge \neg E_{u-\text{ms}}] = Pr[E_U]$, then

$$Pr[E_{\text{sk}} \wedge E_U] \geq \frac{\epsilon}{4} - \frac{Pr[E_{u-\text{ms}}]}{2}.$$

Note that the event $E_{\text{sk}} \wedge E_U$ means $\mathcal{A}$ successfully impersonate $U$ and gets $\text{SK}_{u-\text{ms}} = (SID_u + xt)Y$ and $\mathcal{A}$ can compute $xY = t^{-1}(K_{u-\text{ms}} - SID_u Y)$, which is the solution of CDH problem.

Suppose $\epsilon$ is nonnegligible. From Lemma 1, $Pr[E_{u-\text{ms}}]$ is negligible, so $Pr[E_{\text{sk}} \wedge E_U]$ is also nonnegligible.

Therefore, if $\mathcal{A}$ outputs the correct $b$ with an nonnegligible advantage, then $\mathcal{C}$ can solve the CDH with an nonnegligible probability either. This concludes the proof. ■

### B. Further Security Analysis

In addition to the aforementioned formal security proof, we also provide evidences to explain that the above protocol satisfies the secure requirements presented in Section IV-B, and withstands various attacks [32]–[34].

The proposed protocol provides the following security properties.

1) *No online RC:* In the proposed protocol, RC has not been involved in the mutual authentication phase at all. It only needs to be online at the registration phase.

2) *SSO:* After receiving a login request, the MEC server decrypts the login message with its own private key, then extracts the user identity and verifies the user's legality with his/her identity and the system public key. In the whole process, the MEC server need not save any personal information related to the mobile user. Therefore, the mobile user needs to register with the RC for only once, and then he/she can login to any MEC server as he/she desired.

3) *Mutual authentication:* Although MA-security has been formally proved in Section VI-A, we still provide instinct analysis for this issue. The mobile user authenticates the MEC server by encrypting his/her identity with the public key of the MEC server, so that it can only be decrypt by the target server, while the MEC server authenticates a mobile user by verifying the user's signature $\sigma$ without knowing additional personal information about the user.

4) *Session key agreement:* Equations (3) and (4) show that the mobile user and the MEC server would agree on a common session key $\text{SK} = \text{SK}_{\text{ms}-u} = \text{SK}_{u-\text{ms}}$. The hardness of the CDH problem assures that this session key will not be obtained by any other participants or adversaries.

5) *User anonymity:* In the proposed protocol, the mobile user's identity $\text{ID}_u$ is sent to MEC server in a masked form, namely, $N = h_2(g^x) \oplus (\text{ID}_u||R_u||X)$. It is infeasible for an adversary to extract $\text{ID}_u$ from $N$ without knowing about $g^x$. Moreover, if $\mathcal{A}$ wants to compute $g^x$ from $M = x(P_{\text{pub}} + h_1(\text{ID}_{\text{ms}})P)$, he/she has to solve the $k-$ mBIDH problem. $\mathcal{A}$ cannot extract $x$ from $\sigma$ either, since, $SID_u, R_u, X$ are all unknown to $\mathcal{A}$. Therefore, the proposed protocol can guarantee user's anonymity.

6) *Un-traceability:* In the proposed protocol, random numbers $x$ and $y$ are chosen in every new session, so that the exchanged messages $(M, N, \sigma)$ and $(t, Y)$ are different in each session. The adversary cannot find any relationship among these messages in two different sessions and cannot trace the mobile user's behavior. Therefore, the proposed protocol can guarantee un-traceability.

7) *Perfect forward secrecy:* Suppose the long-term private keys of both the mobile user and the MEC server are leaked and the adversary $\mathcal{A}$ has intercepted all the exchanged messages $(M, N, \sigma, T_u)$ and $(t, Y, T_{\text{ms}})$ on the channel. We might as well assume that $\text{ID}_u, X$ are also obtained by the adversary. We claim that as long as the random value $x$ and $y$ are kept secret to the adversary, the session key is safe. To obtain the session key $\text{SK} = \text{SK}_{\text{ms}-u} = h_5(K_{\text{ms}-u}||\text{ID}_u||\text{ID}_{\text{ms}},||X||Y) = \text{SK}_{u-\text{ms}} = h_5(K_{u-\text{ms}}||\text{ID}_u||\text{ID}_{\text{ms}}||X||Y)$, the adversary needs to know $K_{\text{ms}-u} = y(tX + W)$ or $K_{u-\text{ms}} = (SID_u + xt)Y$. He/she has to compute $yX$ or $xY$ even if he/she has already known $t, X, Y, W$ and $SID_u$. It is infeasible for the adversary to do so if he/she does not know the random value $x$ or $y$. The hardness comes from the CDH assumption. So the protocol provides perfect forward secrecy.

Furthermore, the proposed protocol can resist against the following attacks.

1) *User impersonation attack:* If an external or an internal adversary wants to impersonate a mobile user $U$, he/she must produce a legitimate login message $(M, N, \sigma)$, where $\sigma$ is actually a signature of $U$ on $(\text{ID}_u, R_u, X)$. The adversary cannot produce such a signature without $U$'s private key $SID_u$. So he/she cannot impersonate the mobile user $U$.

2) *MEC server impersonation attack:* In the login message $(M, N, \sigma)$, the mobile user $U$ encrypts $(\text{ID}_u, R_u, X)$ using the identity of the MEC server $\text{ID}_{\text{ms}_j}$. Without the corresponding private key $SID_{\text{ms}}$, the adversary cannot extract $(\text{ID}_u, X)$ from the login message, thus he cannot produce a correct authenticator $t = h_4(\text{ID}_u||\text{ID}_{\text{ms}}||X||Y)$. Therefore, the adversary cannot impersonate the MEC server.

3) *Stolen verifier attack:* The MEC server need not save any information related to any mobile user for mutual authentication. In other words, there is no verifier table to be

TABLE II
SECURITY COMPARISON

| Security Properties | Ref.[4] | Ref.[5] | Ours |
|---|---|---|---|
| Mutual Authentication | × | √ | √ |
| Session Key Security | × | √ | √ |
| Anonymity | × | √ | √ |
| Untraceability | × | √ | √ |
| Single-Sign-On | √ | √ | √ |
| Perfect Forward Privacy | √ | × | √ |
| Impersonation Attack | × | √ | √ |
| Man-in-the-middle | × | √ | √ |
| Replay Attack | √ | √ | √ |
| Provable Security | √ | × | √ |

stolen. Therefore, the protocol is invulnerable to stolen verifier attack.

4) *Man-in-the-middle attack:* From the authentication process, we can see that the mobile user's identity can be verified by the MEC server through his/her signature $\sigma$, and the MEC server has to extract the user's identity information $\mathrm{ID}_u$ and $R_u$ with its private key. An adversary cannot produce a valid message by itself without knowing corresponding secret key. Therefore, the proposed protocol withstands man-in-the-middle attack.
5) *Replay attack:* We use the timestamp in the protocol; the replayed message can be detected by checking the freshness of the corresponding timestamp.

### C. Security Comparisons

A comparison of the security properties between the proposed scheme and two privacy-preserving authentication schemes designed for MCC environment [7], [8] is provided in Table II. We use "√" and "×" to represent whether the scheme is satisfies respective security property or not. Table II mentions that Tsai *et al.*'s scheme [7] is vulnerable to impersonation attack, and thus fails to achieve mutual authentication, session key security, anonymity, un-traceability, and cannot resist man-in-the-middle attack [23]; Irshad *et al.*'s [8], [25] cannot guarantee perfect forward privacy and does not provide a formal security reduction [35]. Our scheme, however, can meet all the listed security requirements.

## VII. PERFORMANCE EVALUATION

We evaluate the performance of the AAKA protocol in terms of computational costs and communicational costs, and compare it with that of Tsai *et al.*'s [7] protocol and Irshad *et al.*'s [8] protocol. Both are identity-based privacy-preserving protocols.

In order to reflect the different computing power of the MEC servers and the mobile devices, we implement the basic operations on two different platforms. The MEC server is simulated on a cloud platform provided by Alibaba, with an Intel(R) Xeon(R) CPU E5-2630 0 @ 2.30 GHz, 1 GB RAM and Ubuntu 14.04 for 64 b operation system. The mobile device is simulated on a Google Nexus One smart phone with 2 GHz ARM CPU armeabi-v7a, 300 MiB RAM, and Android 4.4.2 operation system.

We chooses a 512-b prime number $p$ and an additive elliptic curve group $G$ over $F_p$, as well as the type-1 Ate pairing $e : G \times G \rightarrow G_T$, where $G$ and $G_T$ are groups with a 160-b prime order $q$. The execution time of the basic operations used in the schemes are listed in Table III.

TABLE III
RUNNING TIME OF BASIC OPERATIONS (MS)

| | description | Alibaba Cloud | Google Nexus |
|---|---|---|---|
| $TG_b$ | Bilinear pairing | 5.275 | 48.66 |
| $TG_m$ | Scalar multiplication | 1.97 | 19.919 |
| $TG_a$ | Point addition | 0.012 | 0.118 |
| $T_h$ | Hash function | 0.009 | 0.089 |
| $T_e$ | Modular exponentiation | 0.339 | 3.328 |

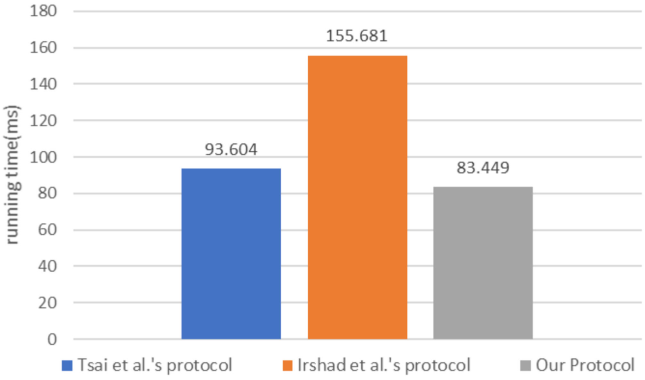

Fig. 6. Computational costs comparison: User side.

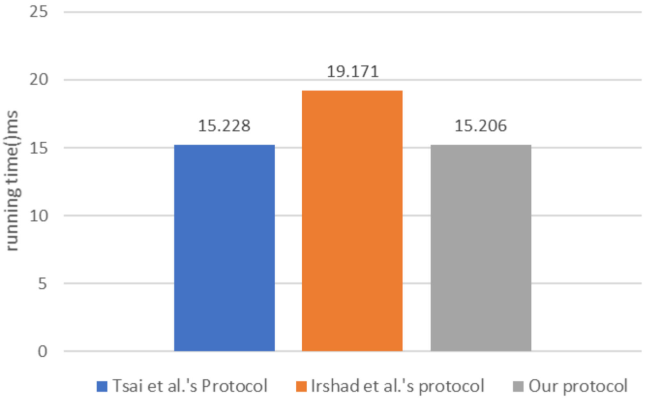

Fig. 7. Computational costs comparison: MES server side.

We evaluate the computational costs of the authentication phase, omitting the costs in the registration phase, since the registration is executed for only once and has little influence on the whole system performance. The computation costs comparison of our scheme with Tsai *et al.*'s scheme [7] and Irshad *et al.*'s scheme [8] is in the user side and the MEC server side is depicted in Figs. 6 and 7, respectively. The specific data are listed in Table IV. The table shows that on the mobile user side, Isai *et al.*'s scheme requires $5TG_m + 2TG_a + T_e + T_{inv} + 5T_h$ operations (93.604 ms), Irshad *et al.*'s scheme requires $TG_b + 5TG_m + 2TG_a + 2T_e + T_{inv} + 6T_h$ operations (155.681 ms), while our protocol only needs $4TG_m + T_e + 5T_h$ operations (83.449 ms). On the MEC server side, Tsai *et al.*'s scheme needs $2TG_b + 2TG_m + 2TG_a + 2T_e + 5T_h$ operations (15.228 ms), Irshad *et al.*'s scheme needs $2TG_b + 4TG_m + 3TG_a + 2T_e + 3T_h$ operations (19.171 ms), our scheme needs $TG_b + 5TG_m + 3TG_a + 5T_h$ operations (15.206 ms). The comparison shows

TABLE IV
COMPARISON OF COMPUTATIONAL COSTS (MS)

| Schemes | Mobile User | MEC (Mobile Cloud) Server |
|---|---|---|
| Ref.[4] | $5TG_m + 2TG_a + T_e + T_{inv} + 5T_h$ (93.604) | $2TG_b + 2TG_m + 2TG_a + 2T_e + 5T_h$ (15.228) |
| Ref. [5] | $TG_b + 5TG_m + 2TG_a + 2T_e + T_{inv} + 6T_h$ (155.681) | $2TG_b + 4TG_m + 3TG_a + 2T_e + 3T_h$ (19.171) |
| Ours | $4TG_m + T_e + 5T_h$ (83.449) | $TG_b + 5TG_m + 3TG_a + 5T_h$ (15.206) |

TABLE V
COMMUNICATIONAL COSTS (BITS): A COMPARATIVE SUMMARY

| Schemes | Communication costs | Length(bit) |
|---|---|---|
| Tsai et al.'s protocol[4] | $3\|G\| + \|G_T\| + \|H\| + \|ID\|$ | 4608 |
| Irshad et al.'s protocol [5] | $4\|G\| + \|G_T\| + \|H\| + \|ID\|$ | 5632 |
| Our Scheme | $4\|G\| + 2\|T\| + 2\|Z_q\| + \|ID\|$ | 4736 |

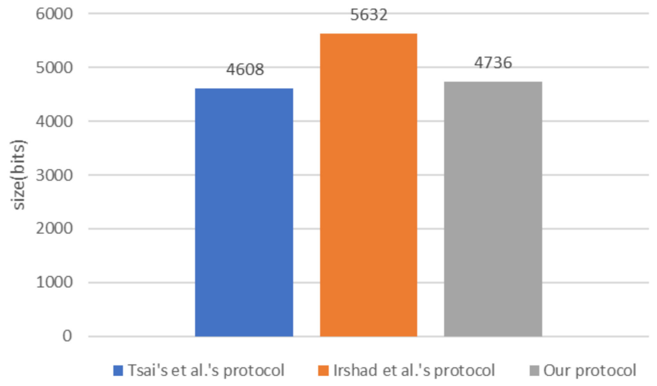

Fig. 8. Communicational costs comparison.

that our protocol has lower computational costs than the other two protocols. Moreover, we note that the time-consuming bilinear pairing operations is only executed by the MEC server, which is more suitable for mobile environment.

Table V lists the communicational costs of these three protocols. $|G|$, $|G_T|$, and $|Z_q|$ denote the size of the element in group $G$, $G_T$, and $Z_q$, respectively, which is 1024, 1024,160 b in our settings. |ID| is the length of an user's identity, and we set it as 32 B, e.g., 256 b. $|H|$ is the length of the output of a hash function, which is dependent on its domain size. For a regular hash, we assume it to be 256 b. $|T|$ denotes the length of the timestamp, which we set as 32 b. Table V and Fig. 8 shows that Tsai *et al.*'s protocol needs to transmit 4608 b during authentication, Irshad *et al.*'s protocol needs to transmit 5632 b, while our scheme needs to transmit 4736 b. Our scheme has lower communicational costs than that of Irshad *et al.*'s scheme, but a little higher than Tsai *et al.*'s scheme. But in our scheme, it needs only one round message exchange, while Tsai *et al.*'s scheme needs two rounds message exchange. Besides, in Tsai *et al.*'s and Irshad *et al.*'s scheme, it is necessary to send a login request first and we omit the communication costs of that part. Moreover, Tsai *et al.*'s protocol has been proved to be invulnerable to impersonate attacks, while our protocol can withstand it.

## VIII. CONCLUSION

The role of cryptographic protocols in secure communication remains crucial, and will be increasingly so in our interconnected society. In this paper, we presented an identity-based AAKA protocol designed to be deployed in a MEC environment. The protocol is also designed to achieve both user anonymity and nontraceability, and allows a registered mobile user to access multiple MEC servers with only a single registration. In addition, the mutual authentication between the user and the MEC server requires only a single round of message exchange. We proved the security of the proposed protocol (i.e., MA-security and AKA security) and explained how it also meets other desirable security requirements. Findings from our performance evaluation also demonstrated that the protocol does not incur significant computational and communication costs, while achieving the discussed security properties.

Future research includes extending the protocol to achieving other security properties, for example due to changes in regulation or other environmental factors.

## REFERENCES

[1] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 54–61, Apr. 2017.

[2] C.-C. Lee, M.-S. Hwang, and I.-E. Liao, "Security enhancement on a new authentication scheme with anonymity for wireless environments," *IEEE Trans. Ind. Electron.*, vol. 53, no. 5, pp. 1683–1687, Oct. 2006.

[3] C.-T. Li, C.-C. Lee, and C.-Y. Weng, "A secure chaotic maps and smart cards based password authentication and key agreement scheme with user anonymity for telecare medicine information systems," *J. Med. Syst.*, vol. 38, no. 9, pp. 1–11, 2014.

[4] C.-T. Li, C.-C. Lee, C.-Y. Weng, and C.-I. Fan, "An extended multi-server-based user authentication and key agreement scheme with user anonymity," *KSII Trans. Internet Inf. Syst.*, vol. 7, no. 1, pp. 119–131, 2013.

[5] D. Wang and P. Wang, "On the anonymity of two-factor authentication schemes for wireless sensor networks: Attacks, principle and solutions," *Comput. Netw.*, vol. 73, pp. 41–57, 2014.

[6] W. Ding, W. Li, and W. Ping, "Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks," *IEEE Trans. Ind. Inform.*, vol. 14, no. 9, pp. 4081–4092, 2018.

[7] J.-L. Tsai and N.-W. Lo, "A privacy-aware authentication scheme for distributed mobile cloud computing services," *IEEE Syst. J.*, vol. 9, no. 3, pp. 805–815, Sep. 2015.

[8] A. Irshad, M. Sher, H. F. Ahmad, B. A. Alzahrani, S. A. Chaudhry, and R. Kumar, "An improved multi-server authentication scheme for distributed mobile cloud computing services," *KSII Trans. Internet Inf. Syst.*, vol. 10, no. 12, pp. 5529–5552, 2016.

[9] M. Ali, "Green cloud on the horizon," in *Proc. Int. Conf. Cloud Comput.*, 2009, pp. 451–459.

[10] M. R. Rahimi, J. Ren, C. H. Liu, A. V. Vasilakos, and N. Venkatasubramanian, "Mobile cloud computing: A survey, state of art and future directions," *Mobile Netw. Appl.*, vol. 19, no. 2, pp. 133–143, 2014.

[11] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for Internet of things and analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments*. New York, NY, USA: Springer, 2014, pp. 169–186.

[12] P. Hu, S. Dhelim, H. Ning, and T. Qiu, "Survey on fog computing: Architecture, key technologies, applications and open issues," *J. Netw. Comput. Appl.*, vol. 98, pp. 27–42, 2017.

[13] O. A. Osanaiye, S. Chen, R. L. Zheng Yan, K.-K. R. Choo, and M. E. Dlodlo, "From cloud to fog computing: A review and a conceptual live VM migration framework," *IEEE Access*, vol. 5, pp. 8284–8300, 2017.

[14] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, Fog *et al.*: A survey and analysis of security threats and challenges," *Future Gener. Comput. Syst.*, vol. 78, pp. 680–698, 2018.

[15] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog computing for the Internet of things: Security and privacy issues," *IEEE Internet Comput.*, vol. 21, no. 2, pp. 34–42, Mar./Apr. 2017.

[16] S. Khan, S. Parkinson, and Y. Qin, "Fog computing security: A review of current applications and security solutions," *J. Cloud Comput.*, vol. 6, no. 1, 2017, Art. no. 90.

[17] M. B. Mollah, M. A. K. Azad, and A. Vasilakos, "Security and privacy challenges in mobile cloud computing: Survey and way ahead," *J. Netw. Comput. Appl.*, vol. 84, pp. 38–54, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804517300632

[18] E. Ahmed and M. H. Rehmani, "Mobile edge computing: Opportunities, solutions, and challenges," *Future Gener. Comput. Syst.*, vol. 70, pp. 59–63, 2017.

[19] T. X. Tran, M.-P. Hosseini, and D. Pompili, "Mobile edge computing: Recent efforts and five key research directions," *IEEE COMSOC MMTC Commun.-Frontiers*, vol. 12, no. 4, pp. 29–33, Jul. 2017.

[20] J. H. Yang and C. C. Chang, "An ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem," *Comput. Secur.*, vol. 28, no. 3, pp. 138–143, 2009.

[21] E.-J. Yoon and K.-Y. Yoo, "Robust ID-based remote mutual authentication with key agreement scheme for mobile devices on ECC," in *Proc. Int. Conf. Comput. Sci. Eng.*, vol. 2, 2009, pp. 633–640.

[22] X. Cao, W. Kou, and X. Du, "A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges," *Inf. Sci.*, vol. 180, no. 15, pp. 2895–2903, 2010.

[23] Q. Jiang, J. Ma, and F. Wei, "On the security of a privacy-aware authentication scheme for distributed mobile cloud computing services," *IEEE Syst. J.*, vol. 12, no. 2, pp. 2039–2042, Jun. 2018.

[24] R. Amin, S. H. Islam, G. P. Biswas, D. Giri, M. K. Khan, and N. Kumar, "A more secure and privacy-aware anonymous user authentication scheme for distributed mobile cloud computing environments," *Secur. Commun. Netw.*, vol. 9, no. 17, pp. 4650–4666, 2016.

[25] D. He, N. Kumar, M. K. Khan, L. Wang, and J. Shen, "Efficient privacy-aware authentication scheme for mobile cloud computing services," *IEEE Syst. J.*, vol. 12, no. 2, pp. 1621–1631, Jun. 2018.

[26] X. Yang, X. Huang, and J. K. Liu, "Efficient handover authentication with user anonymity and untraceability for mobile cloud computing," *Future Gener. Comput. Syst.*, vol. 62, pp. 190–195, 2016.

[27] M. H. Ibrahim, "Octopus: An edge-fog mutual authentication scheme," *IJ Netw. Secur.*, vol. 18, no. 6, pp. 1089–1101, 2016.

[28] D. He, S. Zeadally, N. Kumar, and W. Wu, "Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server architectures," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 9, pp. 2052–2064, Sep. 2016.

[29] L. Xiong, D. Peng, T. Peng, and H. Liang, "An enhanced privacy-aware authentication scheme for distributed mobile cloud computing services," *KSII Trans. Internet Inf. Syst.*, vol. 11, no. 12, pp. 6169–6187, 2017.

[30] D. Wang, H. Cheng, D. He, and P. Wang, "On the challenges in designing identity-based privacy-preserving authentication schemes for mobile devices," *IEEE Syst. J.*, vol. 12, no. 1, pp. 916–925, Mar. 2018.

[31] D. Wang, D. He, P. Wang, and C. H. Chu, "Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment," *IEEE Trans. Dependable Secure Comput.*, vol. 12, no. 4, pp. 428–442, Aug. 2015.

[32] Q. Feng, D. He, S. Zeadally, N. Kumar, and K. Liang, "Ideal lattice-based anonymous authentication protocol for mobile devices," *IEEE Syst. J.*, to be published. doi: 10.1109/JSYST.2018.2851295.

[33] D. He, Y. Zhang, D. Wang, and K. K. R. Choo, "Secure and efficient two-party signing protocol for the identity-based signature scheme in the IEEE p1363 standard for public key cryptography," *IEEE Trans. Dependable Secure Comput.*, to be published. doi: 10.1109/TDSC.2018.2857775.

[34] D. Wang, W. Li, and P. Wang, "Two birds with one stone: Two-factor authentication with security beyond conventional bound," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 4, pp. 708–722, Aug. 2018.

[35] S. Roy, S. Chatterjee, A. K. Das, S. Chattopadhyay, N. Kumar, and A. V. Vasilakos, "On the design of provably secure lightweight remote user authentication scheme for mobile cloud computing services," *IEEE Access*, vol. 5, pp. 25808–25825, 2017.

**Xiaoying Jia** received the Ph.D. degree from the Stake Key Laboratory of Information Security, graduate degree from the University of Chinese Academy of Sciences, Beijing, China, in 2012.

She is currently a Lecturer with South-Central University for Nationalities, Wuhan, China. Her research interests include applied cryptography, cloud computing, and network security.

**Debiao He** received the Ph.D. degree in applied mathematics from the School of Mathematics and Statistics, Wuhan University, Wuhan, China, in 2009.

He is currently a Professor with the School of Cyber Science and Engineering, Wuhan University. His research interests include cryptography and information security, in particular, cryptographic protocols.

**Neeraj Kumar** (M'17–SM'18) received the Ph.D. degree in computer science and engineering from Shri Mata Vaishno Devi University, Katra, India.

He is currently an Associate Professor with the Department of Computer Science and Engineering, Thapar University, Patiala, India. He has more than 300 technical research papers in leading journals such as the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, the IEEE TWPS, the IEEE SYSTEMS JOURNAL, the IEEE COMMUNICATIONS MAGAZINE, the IEEE WCMAG, the IEEE NETMAG, and conferences. His research is supported from DST, TCS, and UGC. He has guided many students leading to M.E. and Ph.D. His research interests include mobile computing, parallel/distributed computing, multiagent systems, service oriented computing, routing and security issues in mobile *ad hoc*, and sensor and mesh networks. He is recipient of best papers award from IEEE Systems Journal (2018) and IEEE ICC (2018). He is TPC member/Technical committee members of various conferences and organized various workshops in ICC, and Globecom conferences.

**Kim-Kwang Raymond Choo** (SM'15) received the Ph.D. degree in information security from the Queensland University of Technology, Brisbane, QLD, Australia, in 2006.

He currently holds the Cloud Technology Endowed Professorship at The University of Texas at San Antonio, San Antonio, TX, USA.

Dr. Choo is the recipient of the IEEE TrustCom 2018 Best Paper Award, ESORICS 2015 Best Research Paper Award, Winning Team of the Germany's University of Erlangen-Nuremberg (FAU) Digital Forensics Research Challenge 2015, 2014 Highly Commended Award by the Australia New Zealand Policing Advisory Agency, Fulbright Scholarship in 2009, 2008 Australia Day Achievement Medallion, and British Computer Society's Wilkes Award in 2008. He is a fellow of the Australian Computer Society, and the Co-Chair of IEEE Multimedia Communications Technical Committee (MMTC)'s Digital Rights Management for Multimedia Interest Group.