An efficient and provably secure authenticated key agreement scheme for mobile edge computing

Mohammad Amin Rakeei and Farokhlagha Moazami

***Abstract*— Though Mobile Cloud Computing (MCC) and Mobile Edge Computing (MEC) technologies have brought more convenience to mobile services over past few years, but security concerns like mutual authentication, user anonymity, user untraceability, etc., have yet remained unresolved. In recent years, many efforts have been made to design security protocols in the context of MCC and MEC, but most of them are prone to security threats. In this paper, we analyze Jia *et al.*'s scheme, one of the latest authentication protocols for MEC environment and we show this scheme is vulnerable to user impersonation and ephemeral secret leakage attacks. Further, we demonstrate that the aforementioned attacks can be similarly applied to Li *et al.*'s scheme which recently derived from Jia *et al.*'s protocol. In this paper, we propose a provably secure authenticated key agreement protocol on the basis of Jia *et al.*'s scheme that not only withstands security weaknesses of it, but also offers low computational and communicational costs compared to the other related schemes. As a formal security proof, we simulate our scheme with widely used AVISPA tool. Moreover, we show the scalability and practicality of our scheme in a MEC environment through NS-3 simulation.**

***Index Terms*—Mobile edge computing (MEC), authentication, provable security, AVISPA, NS-3.**

I. Introduction

IN the last recent years, with the significant increase in mobile devices like smartphones and also the exponential growth of Internet of Things (IoT) technology, two new paradigms are proposed by the researchers on the baseline of cloud computing concept; Mobile Cloud Computing (MCC) and Mobile Edge Computing (MEC). Although both techniques along with Fog Computing [29] share lots of similarities in many aspects, but they differ in some key features like server hardware, deployment, service latency, mobility, system management, server location, etc. that makes them suitable in different scenarios [30].

In the literature of MEC, the cloud resources, i.e. powerful computing capacity and storages are brought to the edge of the network (base stations and access points) close to the mobile devices. This provides a resource-constrained end user to have a low latency, high bandwidth, energy efficient, location and context awareness communication [31] that are highly required in new services such as IoT, augmented reality, vehicular communication, live video streaming, etc. [32].

Although MEC technology offers many promising advantages, but it is not mature enough to deal with all challenges specifically in terms of security. Many researches have done to identify major security threats in MEC-oriented systems [32], [35], [36]. To successfully handle these threats, it necessitates to utilize a solid security mechanism that can be applied effectively in context of MEC. An authentication protocol has been considered as an appropriate solution to secure MEC resources against security threats [31]. So designing an authentication and key agreement (AKA) protocol has received much attention in recent years.

In an MEC system, a mobile user may get service from different MEC servers due to its mobility. So there must be some mechanisms to ease the communication with several MEC servers without heavy overheads. A distributed AKA protocol with a Single-Sign-On (SSO) functionality is a highly preferable option to fulfil aforementioned requirements. Due to the open nature of the wireless communication channel existing in MEC environment, it also faces most of security threats in this type of network [33], [34]. Therefore a desirable AKA protocol must also provide some key security features in wireless network area like mutual authentication, user privacy, anonymity, etc. By the way as the cloud resources are moved toward the edge of the network, the security of MEC server are becoming so crucial since the MEC resources like storages and processing units located in the edge server may be compromised and even controlled by adversaries and the security of a group of mobile users may be threatened [27]. This implies that an AKA protocol must consider this type of practical vulnerability. Another important challenge that must be taken into account is resource limitation of mobile devices. Hence, an AKA protocol must be as lightweight as possible on user side.

The present paper aims to propose an AKA protocol which addresses all above-mentioned challenges. Our protocol is equipped with the identity-based cryptography [14] to handle mutual authentication in an efficient way [13].

A. MEC Architecture

Unlike the traditional centralized cloud servers in a MCC context, the MEC ecosystem comprises geographically distributed edge servers. This topology enables an edge server to deliver a high quality service with low latency to its local customers. Fig.1 depicts the architecture of a MEC environment.

M. A. Rakeei is with the Cyberspace Research Institute, Shahid Beheshti University, Tehran, Iran (e-mail: m.rakeei@mail.sbu.ac.ir).

F. Moazami is with Cyberspace Research Institute, Shahid Beheshti University, Tehran, Iran (e-mail: f_moazemi@sbu.ac.ir).

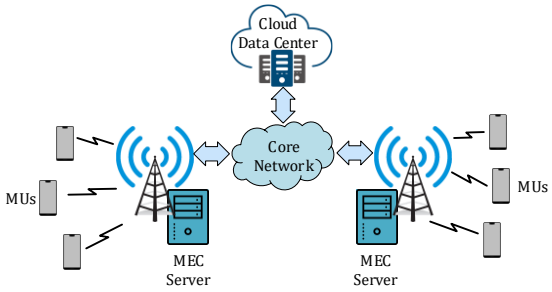


Fig. 1. MEC architecture

B. Motivation and Contributions

The main contributions of this paper are as follows:

- We analyze Jia *et al.*'s scheme [18] and show its vulnerabilities to user impersonation and ESL attacks. We also briefly mention the same security issues of Li *et al.*'s scheme [19], which designed very similar to Jia *et al.*'s protocol.
- We propose a provably secure biometric-based AKA protocol for MEC environment inspired by Jia *et al.*'s scheme. Our scheme supports distributed authentication as well as efficient revocation and password update phase.
- We present a security analysis of the proposed scheme with both formal and informal methods.
- By performing a comparison between our protocol and some recent related schemes, we show that the new scheme proposed in this paper, provides more security properties with low computational and communicational costs and so is more applicable in practice.
- We simulate the proposed protocol using AVISPA tool.
- We show the scalability and practicality of our scheme in MEC context through NS-3 simulation.

C. Organization of the paper

In this paper, first we discuss some preliminaries in section II. In section III, we review Jia *et al.*'s protocol in brief and present a detailed cryptanalysis of it. We also briefly mention the same security issues of Li *et al.*'s scheme in this section. In section IV, we propose an identity-based AKA protocol suitable for MEC environment. In section V, we prove the security of our proposed scheme in both formal and informal manners. Also, the simulation of our scheme using AVISPA is presented in this section. In section VI, we compare the performance of our scheme in terms of communicational and computational costs to several recently proposed schemes for MCC and MEC environments. In section VII, using NS-3 simulation, we show the practical perspective of our proposed scheme. Finally, we conclude this paper in section VIII.

II. Background

A. Related Work

In recent years, many attempts have been made in order to design secure authentication protocols in context of distributed MCC that enable entities to authenticate each other without presence of the trusted third party. Although MCC has different usability compared to MEC, but most of these schemes is applicable to MEC with a bit of consideration.

In 2015, Tsai and Lo [12] introduced one of the first efficient authentication schemes for distributed MCC. Their scheme takes advantage of an identity-based cryptography and supports key exchange and user privacy without an online trusted third party. It enables a mobile user to establish authentication session with multiple MCC servers using one single private key. A few years later, Jiang *et al.* [24] showed that Tsai and Lo's scheme is prone to service provider impersonation attack and does not achieve mutual authentication. Jiang *et al.* stated that their protocol fails to resist wrong password/login attack and also does not support user revocation and re-registration mechanism. A number of schemes have been proposed in the literature motivated by Tsai and Lo's scheme [4], [11], [37]. But none of these protocols yielded a better performance.

In 2018, Amin *et al*. [42] designed an authentication protocol for IoT devices in a distributed MCC environment. In their scheme, a central control server must be present in every authentication session. Later, Zhou *et al*. [28] pointed out that this scheme suffers from user traceability and offline guessing attack. Zhou *et al*. also proposed a more secure authentication scheme for distributed MCC, but similar to [42], the presence of a cloud center in the authentication phase, makes their scheme less practical in a distributed environment. Ghaffar *et al*. [38] designed an authentication protocol for cyber physical systems that was relatively secure and efficient compared to those at [28] and [42], but it missed the functionality in a multi-server environment.

He *et al*. [8] proposed a privacy-aware authentication scheme on the baseline of Tsai and Lo's protocol. Their scheme solved the security weaknesses of Tsai and Lo's protocol as well as enhancing its performance by removing heavy bilinear pairing function from the user side. However, Xiong *et al.* [10] pointed out that their scheme suffers from wrong password/login attack and does not offer key revocation facility. Xiong *et al.* designed a privacy-aware mutual authentication scheme that supports multi-factor security along with user key revocation.

In 2017, Odelu *et al.* [9] proposed a provably secure biometric-based AKA protocol using identity-based encryption technique for distributed MCC services. As they claimed, their scheme achieves session-key security (SK-security) [3] which supposed to be an important requirement in a distributed environment. They also applied the same method as Tseng *et al*. [1] proposed to provide key revocation mechanism.

Jia *et al*. [18] designed a new anonymous authentication protocol in the context of MEC. Their scheme also benefits from identity-based cryptosystem to gain efficient distributed authentication property. Also, it offers two round messages authentication phase that is suitable from communication point of view. In this paper we pointed out that this scheme is vulnerable to user impersonation and ephemeral secret leakage (ESL) attacks. The proposed scheme by Jia *et al.* does not support key revocation and re-registration mechanism as well.

In 2020, Li *et al*. [19] proposed an anonymous AKA protocol on the basis of Jia *et al*.'s scheme. They claimed that their scheme is provably secure and resists against known attacks. But we show that their scheme carries the same weaknesses as Jia *et al.*'s protocol. By the way, a registration center must be online on every authentication session that

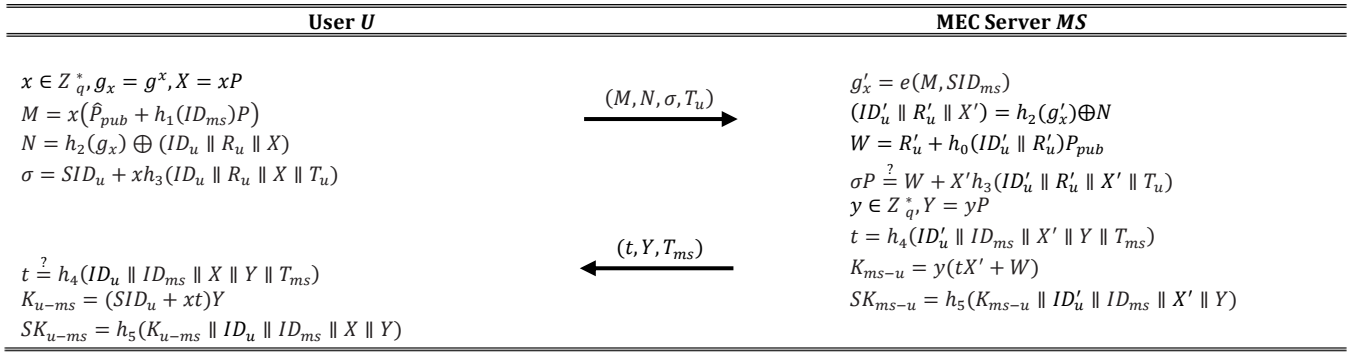


Fig. 2. Authentication phase Jia et al.'s Scheme.

TABLE I
SYSTEM PARAMETERS NOTATIONS

Notations	Description
U	Mobile user
MS	MEC server
RC	Registration center
MD	Mobile device
ID_u	Identity of user
ID_{ms}	Identity of MEC server
p, q	Large prime numbers
$E(F_p)$	Elliptic curve over finite field F_p
G	An additive cyclic group consisting $E(F_p)$ points
G_T	An multiplicative cyclic group
P	Generator of G
$s, \hat{s}$	Private keys of registration center
$P_{pub}, \hat{P}_{pub}$	Public keys of registration center
r_u, x, y	Random numbers in Z_q^*
(SID_u, R_u)	Private keys of mobile user
SID_{ms}	Private key of MEC server
pw_u	Password of mobile user
bio_u	Biometric information of mobile user
$Gen()$	Generation function of fuzzy extractor
$Rep()$	Reproduction function of fuzzy extractor
$h_i(i = 0,1, \dots, 7)$	Hash function
T_u	Timestamp of mobile user
T_{ms}	Timestamp of MEC server
SK_{u-ms}	Session key of mobile user
SK_{ms-u}	Session key of MEC server

makes their scheme less practical in multi-server environment like MEC. In 2020, Irshad *et al* [41] suggested a new pairing-free authentication scheme for distributed MCC context and demonstrated its better performance compared to the prior schemes such as [7] and [37].

B. Mathematical Preliminaries

Based on the parameters definition in table I, it is shown that the following mathematical assumptions can be considered as hard problems [14], [15], [17]:

1) *Elliptic curve discrete logarithm* (ECDL): given an element $Q \in G$ and $Q = xP$, then it is hard to find $x \in Z_q^*$.
2) *Decisional Diffie–Hellman* (DDH): given elements $g^x, g^y \in G_T$, then it is hard to determine if another given element $h \in G_T$ is equal to g^{xy} where x, y are unknown elements in Z_q^*.
3) *Computational Diffie–Hellman* (CDH): given elements $g^x, g^y \in G_T$, then it is hard to compute g^{xy} where x, y are unknown elements in Z_q^*.
4) *k-Modified bilinear inverse Diffie–Hellman* (k-mBIDH): given $\tau P, sP \in G$, $\tau, s \in Z_q^*$, $\alpha_1, \alpha_2, \dots, \alpha_k \in Z_q^*$, $\frac{1}{s+\alpha_1}P, \frac{1}{s+\alpha_2}P, \dots, \frac{1}{s+\alpha_k}P$, then it is hard to compute $e(P,P)^{\frac{\tau}{s+\alpha}}$ for $\alpha \notin \{\alpha_1, \alpha_2, \dots, \alpha_k\}$.

III. REVIEW OF JIA ET AL.'S SCHEME

In this section, we review Jia *et al.*'s scheme. There are three participants in this scheme with following roles: (i) RC: registration center that is responsible for generating some security parameters and publishing them to another parties via a secure channel. (ii) U: mobile user who possesses limited computational resources. He registers with RC and receives his private keys via secure channel. He can also run an authentication process with MEC server through a public channel without help of RC. (iii) MS: MEC server with a high computational power that registers to RC with its identity to present services to authenticated mobile users.

Their scheme has composed of three phases: system setup, registration and authentication phases. Here, we only discuss about the authentication phase where we found that the scheme suffers from user impersonation and ESL attacks.

A. Description of Jia et al.'s scheme

After system setup phase, U and MS can register with RC and then, perform authentication phase as shown in Fig. 2.

B. Cryptanalysis of Jia et al.'s scheme

Now, we show that how a malicious MEC server can impersonate a specific user by performing a parallel session attack. Also we demonstrate that Jia *et al.*'s scheme is insecure when ephemeral secrets are leaked by a session exposure attacks. Moreover, absence of the re-registration and revocation mechanism also makes the scheme less practical.

1) User Impersonation Attack

As discussed in section I, in a MEC ecosystem, an edge server may be compromised and controlled by an adversary in practice. Also, there is no guarantee that all MEC servers behave honestly. Jia *et al.*'s scheme claimed that, no external and internal adversary can generate a legitimate login message (M, N, σ, T_u) without having U's private key SID_u, and so the protocol is secure against user impersonation attack and provides mutual authentication. To refute this claim, we show how a malicious MEC server or a MEC server that controlled by an adversary can forge a login message without

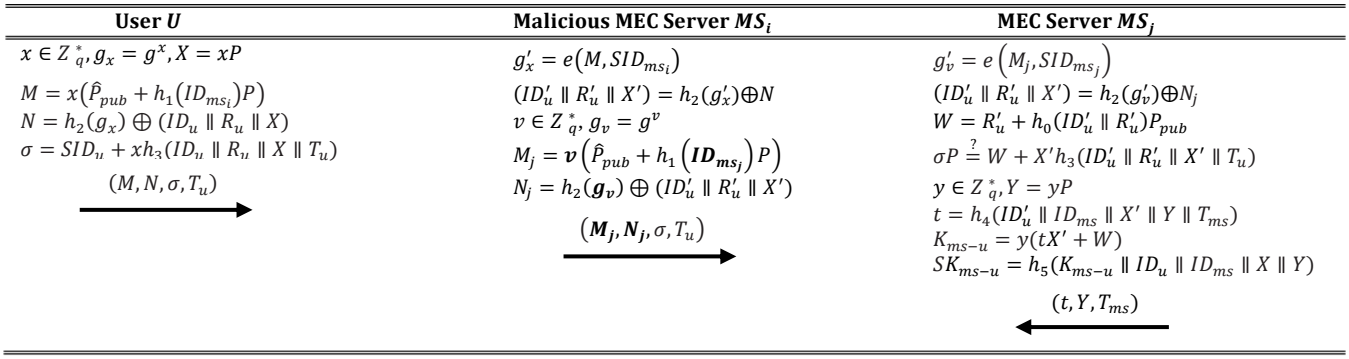


Fig. 3. User impersonation in Jia et al.'s scheme using a parallel session attack.

knowing SID_u. We show that this adversary can perform a parallel session attack and impersonate user U to establish an authentication session with any arbitrary MEC server. Without loss of generality, we only discuss about malicious MEC server scenario. Let's assume user U wants to establish a session with MEC server MS_i. Below steps show how MS_i can impersonate user U to establish a session with an arbitrary MEC server MS_j.

1) MS_i selects a random number $v \in Z_q^*$ and computes value $g_v = g^v$. Then he calculates $M_j = v\left(\hat{P}_{pub} + h_1\left(ID_{ms_j}\right)P\right)$.
2) Upon receiving login message (M, N, σ, T_u) from U, MS_i retrieves values ID'_u, R'_u, X' and computes $N_j = h_2(g_v) \oplus (ID'_u \| R'_u \| X')$.
3) MS_i sends message (M_j, N_j, σ, T_u) to MS_j.

When MS_j receives message (M_j, N_j, σ, T_u) first it computes the bilinear pairing function for M_j, SID_{ms_j} inputs as follow:

$$\begin{aligned} g'_v &= e\left(M_j, SID_{ms_j}\right) \\ &= e\left(v\left(\hat{s}P + h_1\left(ID_{ms_j}\right)P\right), \frac{1}{\hat{s} + h_1\left(ID_{ms_j}\right)}P\right) \\ &= e\left(v(\hat{s} + h_1(ID_{ms}))P, \frac{1}{\hat{s} + h_1\left(ID_{ms_j}\right)}P\right) \\ &= e(P,P)^{v\left(\hat{s}+h_1\left(ID_{ms_j}\right), \frac{1}{\hat{s}+h_1\left(ID_{ms_j}\right)}\right)} \\ &= e(P,P)^v \\ &= g_v \end{aligned} \quad (1)$$

Then it calculates $N_j \oplus h_2(g_v) = (ID'_u \| R'_u \| X')$ and retrieves $(ID'_u \| R'_u \| V')$. Now it computes $W = R'_u + h_0(ID'_u \| R'_u)P_{pub}$ and checks whether the U's signature verifies or not. Clearly in signature verifier equation $\sigma P = W + X'h_3(ID'_u \| R'_u \| X' \| T_u)$, all parameters calculated by MS_j are exactly what U is supposed to send to MS_i. So MS_j successfully authenticates U and generates a new session key for U. It means MS_i has impersonated U and on behalf of him has established a valid session with MS_j. This attack can easily be done by any MEC server like MS_i who participates in a session with a user like U. Note that, the aforementioned attack can easily be performed in practice. The only thing MS_i must consider is timestamp freshness. To ensure the user's timestamp T_u remains fresh, MS_i must send the login message to MS_j as soon as he gets the login message from U. MS_i can perform this process in real time, since the parameter M_j could be pre-calculated and stored in look-up table. Also the parameters σ and T_u will be forwarded without any modification. The only parameter MS_i must compute in real time is N_j and preparing it costs executing of a bilinear pairing function and an xor operation. So MS_i can practically impersonate U. This attack also implies that Jia *et al.*'s scheme could not provide mutual authentication.

It is worth to notice, that this impersonation attack can be similarly applied to Li et al. [18] where a malicious MEC server can extract $h_0(ID'_u \| SID'_u)$ and follows the above-mentioned procedure to impersonate a specific user. In Li *et al.*'s scheme, after performing such user impersonation attack, a malicious MEC server not only impersonates a user to any arbitrary MEC server, but also knows the agreed session key and can communicate with this server on behalf of the victim user.

2) *Ephemeral Secret Leakage Attack*

As pointed out in [2]-[6], authentication scheme should guarantee that leakage of any temporary information of a session, does not compromise security of other secrets and also other sessions. The assumption of ephemeral secret leakage is possible in practice; since generation of random numbers takes occur using external random sources that may be controlled by an adversary. Also ephemeral secrets on the user side, usually are pre-computed and stored in insecure storages to speed up protocol execution. With this in mind, let's assume that the ephemeral secret of a session on the user side x, revealed to an adversary $\mathcal{A}$, with an ESL attack. It means that $\mathcal{A}$ can compute values $X = xP$ and $g_x = g^x$ effectively. Also we assume that, $\mathcal{A}$ has the full control of insecure channel between user U and MEC server MS. So he knows the corresponding (M, N, σ, T_u) message to this leaked x. By knowing these values, $\mathcal{A}$ can computes values ID_u, R_u, X from the equation $N = h_2(g_x) \oplus (ID_u \| R_u \| X)$. Now he can successfully retrieves SID_u, from equation $\sigma = SID_u + xh_3(ID_u \| R_u \| X \| T_u)$. It means the private keys of user U, i.e. (SID_u, R_u) and also session key SK_{u-ms} is known by $\mathcal{A}$. So $\mathcal{A}$ can establish any session with an arbitrary MEC server.

3) *Absence of User Key Revocation and Update*

One of the functionality features an authentication protocol could offer is private key update and revocation particularly on

User U		MEC Server MS
$x \in Z_q^*, g_x = g^x, X = (g_x \bmod q)P$		$g'_x = e(M, SID_{ms})$
$M = x\left(\hat{P}_{pub} + h_1(ID_{ms})P\right)$		$(ID'_u \parallel R'_u) = h_2(g'_x) \oplus N$
$N = h_2(g_x) \oplus (ID_u \parallel R_u)$	(M, N, σ, T_u) →	$W = R'_u + h_0(ID'_u \parallel R'_u)P_{pub}$
$\sigma = (g_x \bmod q) + SID_u h_3(ID_u \parallel R_u \parallel X \parallel SID_u P \parallel T_u)$		$\sigma P \stackrel{?}{=} X' + Wh_3(ID'_u \parallel R'_u \parallel X' \parallel W \parallel T_u)$
		$y \in Z_q^*, Y = g^y$
		$t = h_4(ID'_u \parallel ID_{ms} \parallel X' \parallel Y \parallel T_{ms})$
$t \stackrel{?}{=} h_4(ID_u \parallel ID_{ms} \parallel X \parallel Y \parallel T_{ms})$		$K_{ms-u} = (g'^y_x \bmod q)tW$
$K_{u-ms} = (Y^x \bmod q)tSID_u P$	(t, Y, T_{ms}) ←	$SK_{ms-u} = h_5(K_{ms-u} \parallel ID'_u \parallel ID_{ms} \parallel X' \parallel Y)$
$SK_{u-ms} = h_5(K_{u-ms} \parallel ID_u \parallel ID_{ms} \parallel X \parallel Y)$		

Fig. 4. Mutual authentication in our proposed scheme.

the user side [9], [10], [24]. Unfortunately, Jia *et al.*'s scheme does not support this facility.

IV. Our Proposed Scheme

In this section, we introduce a new authentication key agreement protocol inspired by Jia *et al.*'s scheme which addresses above-mentioned security weaknesses and design flaws. Our scheme consists of three participants same as Jia *et al.*'s scheme and the following phases:

A. System Setup

In this phase, RC selects its public and private keys and also security parameters as follows:

1) RC chooses a bilinear map $e: G \times G \to G_T$ and computes $g = e(P, P)$.
2) RC selects two random numbers $s, \hat{s} \in Z_q^*$ and computes $P_{pub} = sP, \hat{P}_{pub} = \hat{s}P$.
3) RC selects eight secure hash functions with following definitions $h_0: \{0,1\}^* \times G \to Z_q^*$, $h_1: \{0,1\}^* \to Z_q^*$, $h_2: G_T \to \{0,1\}^* \times G$, $h_3: \{0,1\}^* \times G \times G \times G \times \{0,1\}^* \to Z_q^*$, $h_4: \{0,1\}^* \times \{0,1\}^* \times G \times G_T \times \{0,1\}^* \to Z_q^*$, $h_5: G \times \{0,1\}^* \times \{0,1\}^* \times G \times G_T \to \{0,1\}^*$, $h_6: \{0,1\}^* \times \{0,1\}^* \times Z_q^* \to Z_q^*$, $h_7: Z_q^* \times \{0,1\}^* \to \{0,1\}^*$.
4) RC publishes $G, G_T, e, P, P_{pub}, \hat{P}_{pub}, g, h_0, h_1, h_2, h_3, h_4, h_5, h_6, h_7$ parameters and stores $s, \hat{s}$ as its private keys.

B. Registration

In this phase, both user and MEC server send their requests for a new private key and RC provides them secret parameters. Here we assume that RC communicates with all users and MEC servers through a secure channel. The registration phase at each sides are as follows:

- ***User Registration***: a user U imprints his biometric bio_u and computes $(\gamma_u, \theta_u) = Gen(bio_u)$ where Gen is generation function of a fuzzy extractor [16]. Then he computes $D = h_7(\gamma_u \parallel ID_u)$ and sends this value and its identity ID_u, to RC. In response, RC first checks the existence of identity ID_u in its table. If exists, then it rejects the registration request. Otherwise, RC chooses a random number $r_u \in Z_q^*$ and calculates values $R_u = r_u P$, $h_u = h_0(ID_u \parallel R_u)$ and $SID_u = (r_u + sh_u) \bmod q$ then RC stores $(ID_u, D, \text{"1"})$ pair in a table named user registration table where "1" label stands for active state of user U. Now RC sends values (SID_u, R_u) to user U. Upon receiving (SID_u, R_u), U stores values $S = SID_u \oplus C$, $R_x = R_{xu} \oplus h_7(C \parallel ID_u)$, $R_y = R_{yu} \oplus h_7(C \parallel ID_u)$, $V = h_1(C)$ and θ_u on the mobile device storage where $C = h_6(ID_u \parallel pw_u \parallel \gamma_u)$ and $R_u = (R_{xu}, R_{yu})$.
- ***MEC Server Registration***: MS sends its registration request to RC. Upon receiving request from MS, RC chooses a unique identity ID_{ms} and computes values $h_{ms} = h_1(ID_{ms})$, $SID_{ms} = \frac{1}{\hat{s} + h_{ms}} P$. Then RC sends SID_{ms} as the private key of MS.

C. Authenntication

In this phase, user U and MEC server MS can authenticate each other and agree on a secure shared key. Following steps describe this procedure:

1) U inputs his identity ID_u, his password pw_u and his scanned biometric bio_u into the mobile device MD_u. MD_u calculates $\gamma_u = Rep(bio_u, \theta_u)$ and checks whether the equation $V = h_1(C)$ holds or not, where $C = h_6(ID_u \parallel pw_u \parallel \gamma_u)$ and Rep is reproduction function of the fuzzy extractor. If holds, it retrieves $SID_u = S \oplus C$, $R_{xu} = R_x \oplus h_7(C \parallel ID_u)$, $R_{yu} = R_y \oplus h_7(C \parallel ID_u)$ and returns (SID_u, R_u) to U where $R_u = (R_{xu}, R_{yu})$. Now U chooses a random number $x \in Z_q^*$ and computes values $g_x = g^x, X = (g_x \bmod q)P$. Then he computes $M = x\left(\hat{P}_{pub} + h_1(ID_{ms})P\right)$, $N = h_2(g_x) \oplus (ID_u \parallel R_u)$, $\sigma = (g_x \bmod q) + SID_u h_3(ID_u \parallel R_u \parallel X \parallel SID_u P \parallel T_u)$ and sends the message $Msg_1 = (M, N, \sigma, T_u)$ to MS via a public channel, where T_u is the current timestamp of U.
2) Upon receiving $Msg_1 = (M, N, \sigma, T_u)$, first, MS checks the freshness of timestamp T_u. If it was not fresh, it terminates the session with U. Otherwise, it calculates g'_x using the following equation:

$$\begin{aligned} g'_x &= e(M, SID_{ms}) \\ &= e\left(x(\hat{s}P + h_1(ID_{ms})P), \frac{1}{\hat{s} + h_1(ID_{ms})} P\right) \\ &= e\left(x\left(\hat{s} + h_1(ID_{ms})\right)P, \frac{1}{\hat{s} + h_1(ID_{ms})} P\right) \\ &= e(P, P)^{x\left(\hat{s} + h_1(ID_{ms}), \frac{1}{\hat{s} + h_1(ID_{ms})}\right)} \\ &= e(P, P)^x \\ &= g_x \end{aligned} \quad (2)$$

Once MS knows g_x, it can retrieve values ID'_u, R'_u from equation $(ID'_u \parallel R'_u) = h_2(g'_x) \oplus N$. Afterward, it checks its revocation list. If ID_u exists in this table, it terminates the session. Otherwise, MS computes $W = R'_u +$

$h_0(ID'_u \parallel R'_u)P_{pub}$ and checks the validity of equation $\sigma P = X' + Wh_3(ID'_u \parallel R'_u \parallel X' \parallel W \parallel T_u)$ where $X' = (g'_x \bmod q)P$.

$$\begin{aligned}\sigma P &= \big((g_x \bmod q) + SID_u h_3(ID_u \parallel R_u \parallel X \\ &\qquad \parallel SID_u P \parallel T_u)\big)P \\ &= X + \big((r_u + sh_0(ID_u \parallel R_u)h_3(ID_u \parallel R_u \parallel X \\ &\qquad \parallel SID_u P \parallel T_u)\big)P \\ &= X + \big(R_u + h_0(ID_u \parallel R_u)P_{pub}\big)h_3(ID_u \parallel R_u \\ &\qquad \parallel X \parallel W \parallel T_u) \\ &= X + Wh_3(ID_u \parallel R_u \parallel X \parallel W \parallel T_u) \qquad (3)\end{aligned}$$

If (3) does not hold, MS terminates the session. Otherwise, it means U is authenticated. Now MS selects a random number $y \in Z_q^*$ and then computes values $Y = g^y$ and $t = h_4(ID'_u \parallel ID_{ms} \parallel X' \parallel Y \parallel T_{ms})$ where T_{ms} is the current timestamp of MS. It also calculates $K_{ms-u} = (g'^y_x \bmod q)tW$ and the session key $SK_{ms-u} = h_5(K_{ms-u} \parallel ID'_u \parallel ID_{ms} \parallel X' \parallel Y)$. Finally, MS sends the message $Msg_2 = (t, Y, T_{ms})$ to U via a public channel.

3) After receiving $Msg_2 = (t, Y, T_{ms})$, U first checks the freshness of timestamp T_{ms}, then he validates the equation $t = h_4(ID_u \parallel ID_{ms} \parallel X \parallel Y \parallel T_{ms})$. If it does not hold, he terminates the session. Otherwise MS is authenticated for U and he computes $K_{u-ms} = (Y^x \bmod q)tSID_u P$ and the session key $SK_{u-ms} = h_5(K_{u-ms} \parallel ID_u \parallel ID_{ms} \parallel X \parallel Y)$.

D. Password Update

Anytime U wants to update his password, he should follow below steps:

1) U inputs his identity ID_u , his old password pw_u and his biometric bio_u into the mobile device MD_u. Then MD_u computes $\gamma_u = Rep(bio_u, \theta_u)$, $C = h_6(ID_u \parallel pw_u \parallel \gamma_u)$ and checks whether equation $V = h_1(C)$ holds or not. If not, MD_u aborts the update request. Otherwise it continues.
2) MD_u asks U to enter a new password pw_u^* and a new biometric bio_u^*. Then it computes values $(\gamma_u^*, \theta_u^*) = Gen(bio_u^*)$, $C^* = h_6(ID_u \parallel pw_u^* \parallel \gamma_u^*)$, $V^* = h_1(C^*)$, $S^* = S \oplus C \oplus C^*$, $R_x^* = R_x \oplus h_7(C \parallel ID_u) \oplus h_7(C^* \parallel ID_u)$ and $R_y^* = R_y \oplus h_7(C \parallel ID_u) \oplus h_7(C^* \parallel ID_u)$ and replaces values V, S, R_x and R_y with V^*, S^*, R_x^* and R_y^* respectively.

E. User Revocation and Re-Registration

An important feature in a practical smart card-based authentication scheme is user revocation [24], [25]. If a user's mobile device is lost or stolen, or if the user's private keys are unexpectedly revealed or compromised, the user revocation enables user to revoke his identity and prevents the threat of impersonation. Beside this functionality, the user re-registration with the same identity is highly recommended [24] which allows the user to refresh his private keys and participate in authentication sessions with his old identity but renewed keys. Following steps describe the revocation and re-registration phase performed by the user U.

Step 1: U inputs his identity ID_u and imprints his biometric bio_u. Then, he generates $(\gamma_u, \theta_u) = Gen(bio_u)$ and $D = h_7(\gamma_u \parallel ID_u)$. Now he sends the revocation/re-registration request along with ID_u and D to the RC, via the secure channel.

Step 2: Upon receiving revocation/re-registration message, RC retrieves the corresponding D to the identity ID_u from its database and checks the validity of received D. If they were not the same, it aborts the request. Otherwise, if the request is revocation, RC turns the corresponding "1" label to "0" means that the user U is revoked and returns a successful message to inform U. RC also informs all MEC servers that user with identity of ID_u is revoked. If the request is re-registration, RC generates new (SID'_u, R'_u) for the user U and updates the state label to "1". Then it sends the new private keys to U through the secure channel.

V. Security Analysis

In this section we present a security analysis of our scheme in both formal and informal forms. Moreover we compare the security of our protocol with the recently proposed schemes.

A. Provable Security

We present the formal security proof of the proposed protocol. In this model, the adversary $\mathcal{A}$ is an active attacker that can listen the transmitted messages in the public channel and also can modify, replay and intercept them. Also, the adversary can communicate with the oracle $\mathcal{T}_P^i$ through the following queries. Here, $\mathcal{T}_P^i$ is the oracle of the i^{th} instance of entity P and P is the mobile user U or the MEC server MS.

- $\boldsymbol{h(m)}$: When the adversary query $h(m)$, the oracle first search if $h(m)$ has been requested before. If yes, answers the query with the previous value otherwise chooses a random number as the response of the requested query.
- $\boldsymbol{Execute(U^i, MS^j)}$: With this query, the adversary obtains all the transmitted messages between U^i and MS^j during the run of the protocol as its description.
- $\boldsymbol{Extract(ID)}$: When the adversary issues this query, the oracle returns the long term private key of the entity with the identity ID.
- $\boldsymbol{Send(P^i, Msg)}$: If the adversary issues this query and sends the message Msg to the participant P^i, then oracle $\mathcal{T}_P^i$ returns the corresponding messages according to the protocol description.
- $\boldsymbol{EKReveal(\mathcal{T}_P^i)}$: With this query the adversary obtains ephemeral secret keys of the oracle $\mathcal{T}_P^i$.
- $\boldsymbol{SKReveal(\mathcal{T}_P^i)}$: With this query the adversary obtains the session key of the oracle $\mathcal{T}_P^i$, if it has been successfully produced.

  $\boldsymbol{Test(\mathcal{T}_P^i)}$: This query returns a session key or a random value. To answer this query the oracle chooses a random bit b, if $b = 1$ answers the query with the session key and if $b = 0$ answers it with a random value. The adversary can issue only one $Test$ query.

The oracle instances U and MS are said to be partner provided that they can authenticate each other and successfully share a session key SK such that no other instance accept SK. A session key SK is called fresh if it is established between U and MS without issuing $EKReveal$ and $SKReveal$ queries to U and MS. In the semantic security, the adversary $\mathcal{A}$ aims to distinguish a fresh session key from a random number. The

TABLE II
SIMULATION OF ORACLES

$\boldsymbol{h_i(m)}$: For a hash query, $h_i(m)$, on input m challenger $\mathcal{C}$ first looks up the list L_{h_i}, if $(m, h_i(m))$ is in the list L_{h_i}, returns $h_i(m)$ to $\mathcal{A}$. Otherwise, $\mathcal{C}$ randomly chooses r and sets $h_i(m) = r$.Then $\mathcal{C}$ returns $h_i(m)$ to $\mathcal{A}$ and adds $(m, h_i(m))$ to L_{h_i} .
$\boldsymbol{Extract(ID_{u_i})}$: $\mathcal{C}$ answers this query as follows. If $u_i = u_i^*$, $\mathcal{C}$ rejects the query. If $u_i \neq u_i^*$, $\mathcal{C}$ chooses $r_{u_i}, h_{u_i} \in \mathbb{Z}_q^*$ randomly and computes $R_{u_i} = r_{u_i}P - h_{u_i}P_{pub}$ then $\mathcal{C}$ checks if there exists the entry (ID_{u_i}, R_{u_i}) in the list $L_{h_0}$and $h_0(ID_{u_i}\|R_{u_i}) \neq h_{u_i}$. If it is then $\mathcal{C}$ aborts the simulation. Otherwise, $\mathcal{C}$ sets $SID_{u_i} = r_{u_i}$, and consider (SID_{u_i}, R_{u_i}) as a valid private key. $\mathcal{C}$ returns (R_{u_i}, SID_{u_i}) to $\mathcal{A}$, and inserts $(ID_{u_i}, h_{u_i}, R_{u_i}, SID_{u_i})$ and $(ID_{u_i}, R_{u_i}, h_{u_i})$ into the list L_u and L_{h_0}, respectively.
$\boldsymbol{Extract\left(ID_{ms_j}\right)}$: To answer this query on input ID_{ms_j}, $\mathcal{C}$ retrieves L_{h_1} for (ID_{ms_j}, h_{ms_j}), and sets $SID_{ms_j} = \frac{1}{\hat{s}+h_{ms_j}}P$. $\mathcal{C}$ returns SID_{ms_j}as the answer of this query, and adds (ID_{ms_j}, SID_{ms_j}) into the list L_{ms}.
$\boldsymbol{Send(U_k, Start)}$: For this query, if $u_i = u_i^*$ then $\mathcal{C}$ aborts the game. Otherwise, $\mathcal{C}$ checks if there exists an entry for ID_{u_i} in L_u. If yes, $\mathcal{C}$ extracts SID_{u_i} from L_u. Else as description of $Extract(ID_{u_i})$ query, $\mathcal{C}$ generates a private key SID_{u_i} and adds $(ID_{u_i}, h_{u_i}, R_{u_i}, SID_{u_i})$ to L_u. Then $\mathcal{C}$ chooses $x \in \mathbb{Z}_q^*$ randomly and computes $g_x = g^x, X = (g_x \bmod q)P$, (M, N, σ, T_{u_i}) as described in the proposed protocol. Finally, $\mathcal{C}$ adds (ID_{u_i}, k, x, X) into the list L_s, and answers the query with tuple (M, N, σ, T_{u_i}).
$\boldsymbol{Send\left(MS_j^l, (M, N, \sigma, T_{u_i})\right)}$: To answer this query, $\mathcal{C}$ checks if ID_{ms_j} exists in the list L_{ms}. If not, $\mathcal{C}$ generates a private key SID_{ms_j} as description of the $Extract\left(ID_{ms_j}\right)$ query, and adds $(ID_{ms_j}, h_{ms_j}, SID_{ms_j})$ to the list L_{ms}. To answer the query, $\mathcal{C}$ follows the protocol and calculates $g^x = e(M, SID_{ms})$ then extracts $ID_{u_i}\|R_{u_i}$ via $ID_{u_i}\|R_{u_i} = h_2(g_x)\oplus N$ and computes $W = R_u + h_0(ID_u \| R_u)P_{pub}$, finally, checks the equation $\sigma P = X + W h_3(ID_u \| R_u \| X \| g_x \| T_u)$. If it does not hold, $\mathcal{C}$ rejects the message. Otherwise and also if $u_i \neq u_i^*$, as description of the protocol $\mathcal{C}$ randomly chooses $y \in \mathbb{Z}_q^*$ and computes (t, Y, T_{ms}). Then answers the query with computed tuple (t, Y, T_{ms}). If $u_i = u_i^*$, then $\mathcal{A}$ successfully forge a legal login message and wins the game.
$\boldsymbol{Send(U_k, (t, Y, T_{ms}))}$: For this query, $\mathcal{C}$ does as follows: first retrieves (ID_{u_i}, k, x, X) in L_s and checks the equality $t = h_4(ID_u \| ID_{ms} \| X \| Y \| T_{ms})$. If it holds $\mathcal{C}$ authenticated $\mathcal{A}$. Otherwise, $\mathcal{C}$ rejects the message.
$\boldsymbol{EKReveal(\mathcal{T}_P^i)}$: $\mathcal{C}$ responses this query with return the ephemeral secret of the participant P^i.
$\boldsymbol{SKReveal(\mathcal{T}_P^i)}$: $\mathcal{C}$ responses this query with the correct session key SK if SK is accepted. Otherwise returns a "⊥."

semantic security of the protocol is modeled by the game between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}$ that $\mathcal{A}$ can make many queries to $\mathcal{T}_P^i$. If the adversary issues a $Test\left(\mathcal{T}_P^i\right)$ query where the session key is fresh, then the oracle $\mathcal{T}_P^i$ randomly chooses $b \in \{0,1\}$, if $b = 1$ the oracle returns the session key and if $b = 0$ returns a random value. The adversary aims to rightly guess the bit b. Let $Pr[Succ]$ be the probability that the adversary $\mathcal{A}$ wins the game then the advantage of $\mathcal{A}$ in breaking the semantic security of the proposed protocol is defined as $Adv(\mathcal{A}) = |2\,Pr[Succ] - 1|$.

Let E_{u-ms} and E_{ms-u} be the event that $\mathcal{A}$ breaks the user-to-server and the server-to-user authentication, respectively. The proposed protocol achieves mutual authentication if $Pr[E_{u-ms}]$ and $Pr[E_{ms-u}]$ are negligible. In the sequel, we show that the proposed protocol in the random oracle model achieves mutual authentication and is semantically secure when the probability that $\mathcal{A}$ can break ECDL, k-mBIDH and DDH problems are negligible.

Theorem1: Let the adversary $\mathcal{A}$ break user-to-server authentication with probability ε, assume the adversary can query at most q_e and q_s, $Extract$ and $Send$ queries, respectively. Then the challenger $\mathcal{C}$ can solve the ECDL problem with probability

$$\varepsilon_1 \geq \left(1-\frac{1}{q}\right)^{q_e}\left(1-\frac{1}{q_e}\right)\left(1-\frac{1}{q_s}\right)\frac{1}{q_e}\varepsilon \qquad (4)$$

Proof: Let $P, Q = sP \in G$ be an ECDL instance that s is unknown to $\mathcal{C}$ and $\mathcal{C}$ generates $p, q, G, G_T, e, P, \hat{s}$ as system parameters and set $P_{pub} = Q, \hat{P}_{pub} = \hat{s}P$. The challenger consider two sets of identities as $ID_U = \left\{(BIO_{u_1}, pw_{u_1}, ID_{u_1}), (BIO_{u_2}, pw_{u_2}, ID_{u_2}), \cdots, (BIO_{u_{q_e}}, pw_{u_{q_e}}, ID_{u_{q_e}})\right\}$ and $ID_{MS} = \left\{ID_{ms_1}, ID_{ms_2}, \cdots, ID_{ms_{q_e}}\right\}$. The adversary chooses $ID_{u_i}^*$ from the set ID_U as the target user. The simulation of all queries is as Table II. Also, let L_{h_i} save the answers of the random oracle h_i and L_u, L_{ms} maintain answers of $Extract$ query on user U and MS, respectively. Finally let L_s record the transcript in the channel. Suppose that $\mathcal{A}$ successfully generates a valid login message and issues a $Send\left(MS_j^l, (M, N, \sigma, T_{u_i})\right)$ query with $u_i = u_i^*$, so

$$\sigma P = X + Wv, \qquad (5)$$

where $v = h_3\left(ID_{u_i} \| R_{u_i} \| X \| SID_{u_i}P \| T_{u_i}\right)$ and $W = R_u + h_0(ID_u||R_u)P_{pub}$. Let $(M, N, \sigma', T_{u_i}')$ be another valid login message produced by $\mathcal{A}$ then

$$\sigma' P = X + Wv'. \qquad (6)$$

By (5) and (6), the following equation holds:

$$(\sigma - \sigma')P = W(v - v'). \qquad (7)$$

Hence

$$(\sigma - \sigma')P = ((r_u + h_0(ID_u \| R_u)s)(v - v'))P \qquad (8)$$

Therefore

$$(\sigma - \sigma')(v - v')^{-1} = r_u + h_0(ID_u \| R_u)s \qquad (9)$$

So, $\left((\sigma - \sigma')(v - v')^{-1} - r_u\right)(h_0(ID_u \| R_u)^{-1}$can be considered as the solution of the ECDL problem.

We can evaluate the advantage of $\mathcal{C}$ as follows. The simulation can be aborted if:

1) There exists a hash collision for h_0 when the adversary query $Extract(ID_{u_i})$, that its probability is $\frac{1}{q}$.
2) The adversary queries $Extract(ID_{u_i}^*)$, this occurs with probability $\frac{1}{q_e}$.
3) The adversary issues a $Send(U_k, Start)$ query in which $u_i = u_i^*$ that its probability is $\frac{1}{q_s}$.

Therefore, if we assume that A_1 is the event of aborting the game then

$$Pr[A_1] = \left(1-\frac{1}{q}\right)^{q_e}\left(1-\frac{1}{q_e}\right)\left(1-\frac{1}{q_s}\right) \qquad (10)$$

Let A_2 be the event of forging the login message (M, N, σ, T_{u_i}) such that $Extract(ID_{u_i})$ has never been issued before and A_3 be the event that in the forged login message we have $ID_{u_i} = ID_{u_i}^*$. Easily one can see

$$Pr[A_2|A_1] \geq \varepsilon \qquad (11)$$

and

$$Pr[A_3|A_2 \wedge A_1] = \frac{1}{q_e} \qquad (12)$$

Let $\varepsilon_1$be the probability of solving ECDL problem, then

$$\varepsilon_1 \geq \left(1-\tfrac{1}{q}\right)^{q_e}\left(1-\tfrac{1}{q_e}\right)\left(1-\tfrac{1}{q_s}\right)\tfrac{1}{q_e}\varepsilon. \blacksquare$$

Theorem 2: Let the adversary $\mathcal{A}$ break server-to-user authentication of the proposed protocol with probability ε and the adversary query at most q_s, q_e, q_{h_2}, $Send$, $Extract$ and h_2 queries, respectively, then $\mathcal{C}$ can solve the k-mBIDH problem with probability

$$\varepsilon_1 \geq \left(1-\frac{1}{q}\right)^{q_e}\left(1-\frac{1}{q_e}\right)\left(1-\frac{1}{q_s}\right)\frac{1}{q_{h_2}}\varepsilon.$$

TABLE III
SIMULATION OF ORACLES

$\boldsymbol{h_1}\left(\boldsymbol{ID_{ms_j}}\right)$: When the adversary issues this query, $\mathcal{C}$ checks $ID_{ms_j} = ID_{ms_j^*}$. If the equality not hold, $\mathcal{C}$ sets $h_1\left(ID_{ms_j}\right) = e_j$. Otherwise, sets $h_1\left(ID_{ms_j}\right) = e^*$ and inserts $\left(ID_{ms_j}, h_1\left(ID_{ms_j}\right)\right)$ into L_{h_1}.
$\boldsymbol{Extract}(\boldsymbol{ID_{u_i}})$: The challenger to answer this query chooses random numbers $r_{u_i}, h_{u_i} \in \mathbb{Z}_q^*$, and sets $R_{u_i} = r_{u_i}P - h_{u_i}P_{pub}$. For this (ID_{u_i}, R_{u_i}) challenger checks the list L_{h_0}, if finds an entry such that $h_0(ID_{u_i} \| R_{u_i}) \neq h_{u_i}$ then aborts the simulation. Otherwise, $\mathcal{C}$ sets $SID_{u_i} = r_{u_i} + sh_{u_i}$ and answers the query with (R_{u_i}, SID_{u_i}). Also add $(ID_{u_i}, h_{u_i}, R_{u_i}, SID_{u_i})$ and $(ID_{u_i}, R_{u_i}, h_{u_i})$ into the list L_u and L_{h_0}, respectively.
$\boldsymbol{Extract}\left(\boldsymbol{ID_{ms_j}}\right)$: In this query, if $ID_{ms_j} \neq ID_{ms_j^*}$, the challenger sets $h_1\left(ID_{ms_j}\right) = e_j$ and $SID_{ms_j} = \frac{1}{s+e_j}P$ and answers the query with SID_{ms_j} and add (ID_{ms_j}, e_j) and (ID_{ms_j}, SID_{ms_j}) into L_{h_1} and L_{ms}, respectively. If $ID_{ms_j} = ID_{ms_j^*}$, $\mathcal{C}$ returns a "$\perp$" and sets $h_1\left(ID_{ms_j^*}\right) = e^*$ and inserts $(ID_{ms_j^*}, h_1\left(ID_{ms_j^*}\right))$ into L_{h_1}.
$\boldsymbol{Send}(\boldsymbol{U_i^k}, \boldsymbol{Start})$: If the partner of U_i is ms_j^* then $\mathcal{C}$ sets $M = \tau P$, and as description of the protocol computes (M, N, σ, T_{u_i}). Otherwise, $\mathcal{C}$ lookup L_u for SID_{u_i} and obtain the private key of the user then generates a login message as description of the protocol and saves transcript messages in L_s.
$\boldsymbol{Send}\left(\boldsymbol{MS_j^l}, (\boldsymbol{M}, \boldsymbol{N}, \boldsymbol{\sigma}, \boldsymbol{T_{u_i}})\right)$: In this query if $ID_{ms_j} \neq ID_{ms_j^*}$, $\mathcal{C}$ obtains SID_{ms_j}form L_{ms_j} and verifies the signature. Then chooses a random number $y \in \mathbb{Z}_q^*$ and as description of the protocol computes (t, Y, T_{ms}), and answers the query with this (t, Y, T_{ms}). Otherwise the challenger rejects the message.
$\boldsymbol{Send}(\boldsymbol{U_i^k}, (\boldsymbol{t}, \boldsymbol{Y}, \boldsymbol{T_{ms}}))$: To answer this query challenger find the corresponding tuple in L_s and checks equality $t = h_4(ID_u \| ID_{ms} \| X \| Y \| T_{ms})$, if it is hold then the adversary authenticated as a legal partner and if the partner is the partner $\mathcal{T}_{ms_j^*}^l$, then $\mathcal{A}$ wins the game. Otherwise $\mathcal{C}$ rejects the message.

Proof: Let $P, sP, \tau P, \{e_1, e_2, \cdots, e_k \in \mathbb{Z}_q^*\}$, and $\frac{1}{s+e_1}P, \frac{1}{s+e_2}P, \cdots, \frac{1}{s+e_k}P$, be a k-mBIDH instance such that s, τ is unknown to $\mathcal{C}$. The goal of $\mathcal{C}$ is computing $e(p,p)^{\frac{\tau}{s+e^*}}$ for some $e^* \in \{e_1, e_2, \cdots, e_k\}$. First, $\mathcal{C}$ generates $p, q, G, G_T, e, P, \hat{s}$ as system parameters and sets $P_{pub} = sP$ and let L_{h_i}, L_u, L_{ms} and L_s be as the proof of Theorem 1 and $\mathcal{A}$ chooses $ID_{ms_j^*}$ from the set ID_{MS} as the target server. The $h_i, (i = 0, 2, 3, \dots, 7)$, $EKReveal$, $SKReveal$ and $Test$ oracles are simulated in the same way with Table II. h_1, $Extract$, and $Send$ oracles are simulated as Table III. At the end of the simulation, suppose $\mathcal{A}$ submits a $Send(U_i^k, Msg)$ with a legal response message (t, Y, T_{ms}) that the partner is $\mathcal{T}_{ms_j^*}^l$, in the sequel we show that $\mathcal{C}$ can solve the k-mBIDH problem with using the adversary as a subprogram. Let $\mathcal{A}$ submit a valid login message that lead to the equality $t = h_4\left(ID_{u_i} \| ID_{ms_j^*} \| X \| Y \| T_{ms}\right)$. This means that the adversary must have recover correct ID_{u_i} from $\left(ID_{u_i} \| R_{u_i}\right) = h_2(g_x) \oplus N$, thus he must have queried g_x on h_2 oracle. On the other hand

$$g_x = e\left(M, SID_{ms_j^*}\right) = e\left(\tau P, \frac{1}{s+e^*}P\right) = e(P,P)^{\frac{\tau}{s+e^*}P} \quad (13)$$

The challenger randomly chooses $(g_x, h_2(g_x))$ in the list L_{h_2} and considers g_x as the solution of k-mBIDH problem.

The advantage of $\mathcal{C}$ can be computed as follows. Let A_1be the event that the simulation is not aborted. Similar to the proof of Theorem 1, one can see that

$$Pr[A_1] = \left(1 - \frac{1}{q}\right)^{q_e}\left(1 - \frac{1}{q_e}\right)\left(1 - \frac{1}{q_s}\right) \quad (14)$$

Let A_2 be the event that the adversary successfully issues a $Send(U_i^k, (t, Y, T_{ms}))$ query that (t, Y, T_{ms}) can be accepted as a legal login message such that $Extract\left(ID_{ms_j}\right)$ has never been queried before. Finally, let A_3 and A_4 be the event that in the forged login message, $ID_{ms_j} = ID_{ms_j^*}$ and the challenger $\mathcal{C}$ chooses a correct g_x, respectively. It is not difficult to see that

$$Pr[A_2|A_1] \geq \varepsilon,$$

and

$$Pr[A_3|A_2 \wedge A_1] = \frac{1}{q_e},$$
$$Pr[A_4|A_3 \wedge A_2 \wedge A_1] = \frac{1}{q_{h_2}} \quad (15)$$

Therefore

$$Pr[A_1 \wedge A_2 \wedge A_3 \wedge A_4] = Pr[A_4|A_3 \wedge A_2 \wedge A_1]\Pr[A_3|A_2 \wedge A_1]\Pr[A_2|A_1]Pr[A_1] \geq \left(1 - \frac{1}{q}\right)^{q_e}\left(1 - \frac{1}{q_e}\right)\left(1 - \frac{1}{q_s}\right)\frac{1}{q_e}\frac{1}{q_{h_2}}\varepsilon.$$

Hence the desired can be concluded. ■

Theorem 3: The proposed scheme is semantically secure if the probability of solving DDH problem is non negligible.

Proof: Suppose the adversary wins the game with non negligible advantage ε. Let E_{sk} denote the event that the adversary obtains a correct session key in the $Test$ query. Since $\mathcal{A}$ outputs a correct b with the probability at least $\frac{1}{2}$, we have $Pr[E_{sk}] \geq \frac{\varepsilon}{2}$. Let E_U and E_{MS} be the events that $Test(\mathcal{T}_U^i)$ and $Test(\mathcal{T}_{MS}^j)$ are queried successfully, respectively. Then, the following relations hold:

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

Hence

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

Also, the event $E_{MS} \wedge \neg E_{u-ms}$ is equal to E_U, hence

$$Pr[E_{sk} \wedge E_U] \geq \frac{1}{2}\left(\frac{\varepsilon}{2} - Pr[E_{u-ms}]\right) \quad (16)$$

By Theorem 1, $Pr[E_{u-ms}]$ is negligible and can be ignored. $E_{sk} \wedge E_U$ is the event that the adversary obtains the session key and impersonates U. So the adversary obtains $K_{u-ms} = (Y^x \bmod q)tSID_uP$ and can use it to solve DDH problem.

B. Informal Security Analysis

As an informal security analysis, we show how our proposed scheme meets security requirements for an authenticated key agreement protocol in MEC environment [18], [23], [26].

1) *Mutual Authentication*: In section V-A, we have formally proved that the proposed authentication protocol provides mutual authentication. As an informal way of proof, it is clear that only MS can extract the parameter g_x from M and generates valid response message to U. So U can authenticate MS when he checks the correctness of $t = h_4(ID_u \| ID_{ms} \| X \| Y \| T_{ms})$. Also, the valid signature σ only can be generated by U using his private key. So MS can authenticate U as well.
2) *User anonymity*: The user's identity appears in three parameters passed through the channel N, σ and t. As long as the hash functions remain safe, no adversary like $\mathcal{A}$

can extract ID_u from σ and t. Moreover, the hardness of k-mBIDH problem assures that $\mathcal{A}$ cannot compute g_x from M, and consequently is unable to find N. It means that the user's identity is anonymous to adversary.

3) *User untraceability*: In every authentication session of the proposed protocol, U generates a new nonce x which makes parameters N, σ and t fresh. So the user's identity could not be traced across various sessions.
4) *Session key agreement*: As shown in section IV, after a successful authentication session, both parties agree on a same session key. We proved in section V-A that no adversary can know the agreed session key and so the scheme provides session key agreement.
5) *Perfect forward secrecy*: perfect forward secrecy ensures that previous session keys remain secure even if both private keys of user and service provider are leaked to an adversary. If an adversary $\mathcal{A}$ knows SID_u and SID_{ms}, he can obtain $g_x = g^x$ from $g_x = e(M, SID_{ms})$. If he want to compute $K_{u-ms} = (Y^x \bmod q)tSID_uP$ or $K_{ms-u} = (g_x'^y \bmod q)tW$, he must calculate $Y^x = g_x'^y = g^{xy}$ or $g_x'^y = g^{xy}$ from g^x and g^y that is the solution of CDH problem and assumed to be a hard problem. So our scheme provides perfect forward secrecy.
6) *User impersonation attack*: As we proved in section V-A, no adversary can generate a valid signature σ without knowing user's private key SID_u and his identity ID_u. So the scheme withstand user impersonation attack.
7) *Server impersonation attack*: If an adversary $\mathcal{A}$ wants to impersonate a MEC server MS, he must send a valid response message to the user U. In order to do that, $\mathcal{A}$ needs to know ID_u and X to generate a legitimate $t = h_4(ID_u \parallel ID_{ms} \parallel X \parallel Y \parallel T_{ms})$. It is infeasible, because no one except MS can extract ID_u and X from login message. So our scheme resists server impersonation attack. By the way, in section V-A, we proved the resistance of our scheme to server impersonation attack.
8) *User impersonation attack by malicious MEC server*: The main reason Jia *et al.*'s scheme is subject to user impersonation attack by malicious MEC server is that there is no direct dependency between user's signature σ and MEC server's identity that enables adversary $\mathcal{A}$ to replay σ to another MEC server. In our proposed protocol, g_x depends on S_{ms}, X depends on g_x and σP depends on X. So in a chaining relation, σP depends on S_{ms} and it means only MEC server MS who knows S_{ms} can validate signature σ. So, $\mathcal{A}$ can no longer pass σ to any other MEC server.
9) *Ephemeral secret leakage attack*: Let's assume that the ephemeral secret of user on an authentication session, means x is revealed to an adversary $\mathcal{A}$. We claim that, the session key and both private keys remain secure against $\mathcal{A}$. If $\mathcal{A}$ wants to obtain the session key corresponds to this revealed secret, he needs to know SID_u or SID_{ms}. It is impossible in practice while ECDL problem considered as a hard one to solve. By the way, computing SID_u from signature σ is also infeasible, because $SID_u h_3(ID_u \parallel R_u \parallel X \parallel SID_uP \parallel T_u)$ is a random number if we assume h_3 is a secure hash function. It implies that despite Jia *et al.*'s scheme, our proposed protocol is secure against ESL attack.
10) *Privileged insider attack*: In the user registration phase, user U sends $D = h_7(\gamma_u \parallel ID_u)$ and his identity to RC, and RC saves these values in its table. If any insider adversary accesses this table, he cannot extract the biometric value due to one-way property of hash function.
11) *Password guessing attack*: Let's assume that an adversary $\mathcal{A}$ has the lost/stolen mobile device. Furthermore, we assume that he has the full access of all stored data in the mobile storage specially $V = h_1(C) = h_1\big(h_6(ID_u \parallel pw_u \parallel \gamma_u)\big)$. However, it is impossible to drive password, because he does not know the user's identity ID_u and also the biometric information of the user γ_u. Note that, due to the property of the fuzzy extractor, $\mathcal{A}$ also is unable to reproduce γ_u from θ_u.
12) *Stolen verifier attack*: All parameters S, V, R_x, R_y stored in the mobile device storage are masked with $C = h_6(ID_u \parallel pw_u \parallel \gamma_u)$ and as long as hash function remains secure, an adversary can extract no data even if he has the full control of mobile device storage.
13) *Replay attack*: According to the description of the protocol, user and MEC server use new random nonces and fresh timestamps in every authentication session. So checking the freshness of received messages, prevents replay attack.
14) *Man-in-the-middle (MITM) attack*: In the authentication phase, first, the MEC server MS extracts the user's identity ID_u using its private key and it is clear that no adversary without knowing this key can found ID_u. Then, MS verifies the user's signature σ generated using user's private key SID_u. Since σ contains ID_u and this parameter can only be extracted by MS, signature validation makes it sure that anyone who sends the message consisting identity ID_u, owns the legitimate private key SID_u. Hence, our protocol resists man-in-the-middle attack. By the way, in section V-D, we formally prove the resistance of our protocol to MITM attack.

C. Security Comparison

Security properties of our protocol and four related schemes [8], [9], [18], [19], [41] and [42] have been compared in Table IV. As shown in Table IV, He *et al.*'s [8] scheme is prone to wrong password login attack and also lacks revocation and re-registration mechanism [10]. Odelu *et al.*'s [9] scheme cannot provide multi factor security [10]. We demonstrate that Jia *et al.*'s [18] scheme is vulnerable to user impersonation attack and so fails to achieve mutual authentication. Their scheme also is insecure against ESL attack and cannot provide SK-security. Besides, it lacks revocation mechanism and also perfect forward secrecy [19]. Similarly, Li *et al.*'s [19] scheme suffers from those attack found in Jia *et al.*'s scheme. Moreover, their protocol has not the distributed authentication property as registration center must be online and take part in every authentication session. Another security weakness we found in their scheme is user traceability because the parameter $w = h_0(ID_u \parallel SID_u)$ sent in every authentication session by user, is

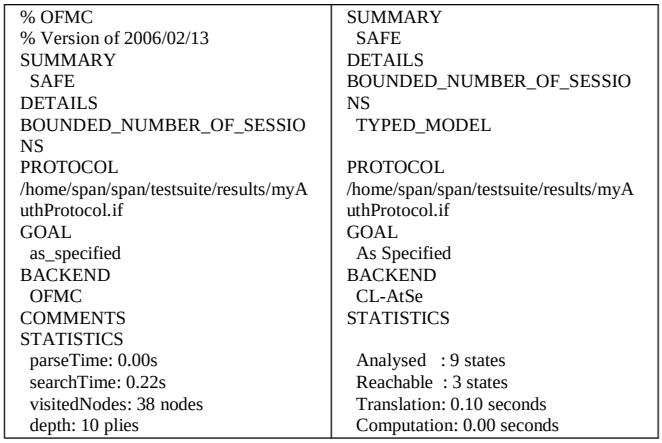


Fig. 5. AVISPA simulation results with OFMC and CL-AtSe back-ends.

only dependent on the user's identity and its private key and so could be easily traced. We found that Irshad et al.'s [41] scheme lacks untraceability, perfect forward secrecy, SK-security and user revocation. As pointed out in [38] and [41], Amin et al.'s [42] scheme is prone to ESL and password guessing attacks and also fails to achieve user anonymity. In addition, we found that in their scheme, the user untraceability is also violated. Moreover, their scheme cannot be considered as truly distributed protocol, since control server must be present in every authentication session. It is clear that our scheme, can satisfy all mentioned security requirements.

D. Simulation Using AVISPA

In this section, a formal security verification has been done by widely used AVISPA simulation tool [22]. AVISPA analyzes the protocol under given model, here Dolev-Yao threat model [23], and checks whether it is safe against replay and man-in-the-middle attacks. Our proposed scheme has been simulated with AVISPA offline tool [21] with two OFMC and CL-AtSe back-ends. The results of simulations in Fig. 5, imply that our proposed scheme is safe against replay and man-in-the-middle attacks.

VI. PERFORMANCE ANALYSIS

In this section, we compare performance of our scheme proposed in section IV, in terms of computational and communicational costs with several authentication schemes proposed for MCC and MEC environments: He *et al*. [8], Odelu *et al*. [9], Jia *et al*. [18], Li *et al*. [19], Irshad *et al*. [41] and Amin *et al*. [42].

A. Computation Cost Comparison

In order to perform a computational cost comparison, we calculate the running time of operations on both user and server sides. The execution time of various operations are exactley taken from [8] and listed in Table V. We assusme all operations not listed in Table V have negligible running time.

It is obvious from Table VI that the schemes [41] and [42] have lower computatioal costs compared to ours, but as discussed in section V-C, these schemes fail to provide some major security requirements. In schemes [8] and [9], a heavy map-to-point hash function has been used. This function can be efficiently replaced by a secure hash function while providing the desirable security. As evident from Table IV and Table VI, it is clear that our scheme has relatively low computational cost while offers more security properties compared to the others.

TABLE IV
SECURITY COMPARISON

Security Properties	[8]	[9]	[18]	[19]	[41]	[42]	Ours
Mutual Authentication	✓	✓	×	×	✓	✓	✓
User Anonymity	✓	✓	✓	✓	✓	×	✓
Untraceability	✓	✓	✓	×	×	×	✓
Perfect Forward Secrecy	✓	✓	×	✓	×	✓	✓
SSO	✓	✓	✓	✓	✓	✓	✓
Distributed Authentication	✓	✓	✓	×	✓	×	✓
Prevents User Impersonation Attack	✓	✓	×	×	✓	✓	✓
Prevents Server Impersonation Attack	✓	✓	✓	✓	✓	✓	✓
Prevents Man-In-The-Middle Attack	✓	✓	✓	✓	✓	✓	✓
Prevents Replay Attack	✓	✓	✓	✓	✓	✓	✓
SK-Security	✓	✓	×	×	×	×	✓
User Revocation and Re-Registration	×	✓	×	×	×	×	✓
Wrong Password Login/update Attack	×	✓	✓	✓	✓	✓	✓
Multi Factor Security	✓	×	✓	✓	✓	×	✓
Provable Security	✓	✓	✓	✓	✓	✓	✓

TABLE V
RUNNING TIME OF OPERATIONS (MS)

Symbol	Description	User	Server
T_{mtp}	Map to Point Hash Function	33.582	5.493
TG_b	Bilinear Pairing	32.713	5.427
TG_m	Scalar Multiplication	13.405	2.165
T_{exp}	Modular Exponentiation	2.249	0.339
TG_a	Point Addition	0.081	0.013
T_h	Hash Function	0.056	0.007

TABLE VI
COMPARISON OF COMPUTATION COSTS (MS)

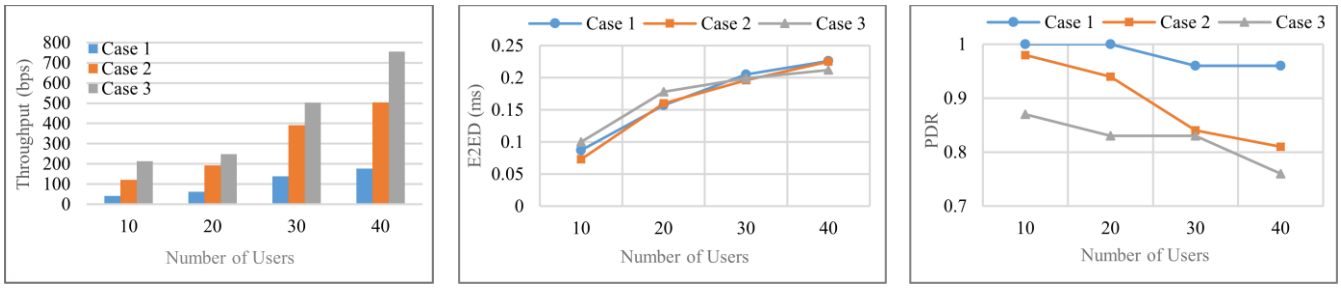
Scheme	User	Server	Total
[8]	$T_{mtp} + 3TG_m + TG_a + 3T_{exp} + 5T_h$ (80.90)	$2TG_b + TG_m + TG_a + 3T_{exp} + 5T_h$ (14.08)	94.98
[9]	$2T_{mtp} + 3TG_m + 2T_{exp} + TG_a + 5T_h$ (112.23)	$2TG_b + TG_m + TG_a + 3T_{exp} + 5T_h$ (14.08)	126.31
[18]	$4TG_m + T_{exp} + TG_a + 5T_h$ (56.23)	$TG_b + 5TG_m + 3TG_a + 5T_h$ (16.32)	72.55
[19]	$6TG_m + T_{exp} + TG_a + 5T_h$ (83.04)	$TG_b + 4TG_m + 2T_h$ (14.10)	97.14
[41]	$3TG_m + 10T_h$ (40.90)	$3TG_m + 10T_h$ (6.56)	47.46
[42]	$4TG_m + 5T_h$ (53.90)	$4TG_m + 4T_h$ (8.68)	62.58
Ours	$4TG_m + 2T_{exp} + TG_a + 8T_h$ (58.64)	$TG_b + 3TG_m + 2TG_a + 2T_{exp} + 5T_h$ (10.49)	69.13

TABLE VII
COMPARISON OF COMMUNICATION COSTS

Scheme	Costs	Length (Bits)	Number of Messages
[8]	$3\|G\| + 2\|Z_q^*\| + 2\|ID\|$	3904	4
[9]	$3\|G\| + 2\|Z_q^*\| + \|ID\|$	3648	3
[18]	$4\|G\| + 2\|Z_q^*\| + \|ID\| + 2\|T\|$	4736	2
[19]	$4\|G\| + 4\|Z_q^*\| + 2\|T\|$	4800	5
[41]	$2\|G\| + 3\|Z_q^*\|$	2528	3
[42]	$17\|Z_q^*\| + 3\|T\|$	2816	3
Ours	$3\|G\| + 2\|Z_q^*\| + \|ID\| + 2\|T\|$	3712	2

B. Communication Cost Comparison

In this section, we compare the communication cost of authentication protocols presented in [8], [9], [18] ,[19], [41] and [42] with our proposed scheme. We choose p as a 512-bits prime number so, the size of elements in the groups G and G_T denoted with $|G|$ and $|G_T|$, is 1024 bits. We also set q, as a 160-bits prime number so $|Z_q^*|$, the size of the element in Z_q^*, is 160 bits. Let the length of the identity and the timestamp used in the protocol denoted by $|ID|$ and $|T|$ be 256 and 32 bits, respectively. Table VII summarizes the communicational costs of authentication phase in different schemes. As shown in Table



(a) (b) (c)

Fig. 6. NS-3 simulation results. (a) Throughput. (b) End to end delay. (c) Packet delivery ratio

VII, schemes [41] and [42] have the lowest communication costs, but both schemes lack important security features. It is clear that our scheme significantly has lower communication cost compared with schemes [8], [18] and [19]. Moreover, the proposed scheme has almost the same communication cost as Odelu *et al.*'s scheme [9] while requiring less number of messages. So, our scheme has a desirable communication cost while achieves more security properties compared to the others.

VII. Practical Demonstration: NS-3 Simulation

In this section, we discuss the practical perspective of our proposed scheme through NS-3 [40]. NS-3 is a well-known discrete-event network simulator for internet systems. Through NS-3 simulation tool, we measure some important network parameters such as Packet Delivery Ratio (PDR), Throughput and End-to-End-Delay (E2ED).

A. Simulation Parameters

The simulation is done by NS-3 (3.30) simulator in Ubuntu 18.04 operating system. There are five MEC servers in an area of $300 \times 300\ m^2$. The initial number of mobile users is 10 and it increases with the step of 10, until reaches 40. The users are randomly allocated in the rectangle zone and can freely move with the random direction model. The simulation time is 1800 seconds and each user sends packets with an interval of 5 seconds. All entities communicate over 2.4 GHz IEEE 802.11 standard. The routing protocol is set to OLSR mode. According to the section VI-B, the length of messages Msg_1 and Msg_2 are 312 and 152 bytes, respectively. We also consider three different cases for user mobility: Case1) constant speed of 3 m/s Case2) constant speed of 10 m/s Case3) constant speed of 15 m/s.

B. Throughput

Throughput means the rate of transmitted bits in the network. As shown in Fig. 6(a), in a constant user movement speed, when the number of users increases, more data are exchanged and as a result, the throughput becomes more. Also, we see that in a constant quantity of users, the throughput increases when mobility goes high.

C. End-To-End Delay

E2ED is the average time of transmitting packets. From Fig. 6(b), it is clear that the E2ED is almost constant when mobility of users changes. Also, when we add more users to the area, more congestion happens and so the E2ED becomes more.

D. Packet Delivery Ratio

PDR is the ratio of total transmitted packets to total received packets. As shown in Fig. 6(c), when mobility is fixed, if the number of users becomes more, the congestion in network consequently is increasing and so the PDR becomes less. Moreover, when the number of users is fixed, the PDR becomes less if the mobility becomes more and this is a natural relation between mobility and PDR for the chosen routing protocol.

VIII. Conclusion

In this paper, we have cryptanalyzed Jia *et al.*'s anonymous AKA scheme that has been recently proposed for MEC environment and demonstrated how this scheme is vulnerable to user impersonation and ESL attack. Inspired by Jia *et al.*'s scheme, we have presented a new AKA protocol designed in MEC context and we have shown it is secure through formal and informal security proves. The simulation of our proposed protocol using AVISPA describes that it also withstand replay and man-in-the-middle attack. The performance evaluation done in this paper shows that our scheme has a low computational and communicational costs compared to the several related schemes while provides more security properties, so is a desirable choice for implementing in MEC environment. Moreover, through NS-3 simulation, we have shown that our scheme is scalable and practical in a real MEC environment. Further work will look into utilizing blockchain capabilities to achieve more efficient and robust AKA protocol.

References

[1] Y. Tseng, S. Huang, T. Tsai and J. Ke, "List-Free ID-Based Mutual Authentication and Key Agreement Protocol for Multiserver Architectures," *IEEE Transactions on Emerging Topics in Computing*, vol. 4, no. 1, pp. 102-112, Jan.-Mar. 2016.

[2] R. Canetti and H. Krawczyk, "Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels," *Lecture Notes in Computer Science Advances in Cryptology — EUROCRYPT 2001*, pp. 453–474, 2001.

[3] A. LaMacchia Brian, Lauter Kristin and Anton Mityagin, "Stronger Security of Authenticated Key Exchange," *ProvSec 2007, LNCS. Springer, Heidelberg*, vol. 4784, pp. 1-16, Nov. 2007.

[4] 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.

[5] C.-M. Chen, Y. Huang, K.-H. Wang, S. Kumari, and M.-E. Wu, "A secure authenticated and key exchange scheme for fog computing," Enterprise Information Systems, pp. 1–16, Oct. 2020.

[6] M. Luo, Y. Zhang, M. K. Khan, and D. He, "A secure and efficient identity-based mutual authentication scheme with smart card using elliptic curve cryptography," *International Journal of Communication Systems*, vol. 30, no. 16, 2017.

[7] A. Karati, R. Amin, S. H. Islam and K.-K. R. Choo, “Provably secure and lightweight identity-based authenticated data sharing protocol for cyber-physical cloud environment”, *IEEE Trans. Cloud Comput.*

[8] D. He, N. Kumar, M. K. Khan, L. Wang and J. Shen, "Efficient Privacy-Aware Authentication Scheme for Mobile Cloud Computing Services," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1621-1631, Jun. 2018.

[9] V. Odelu, A. K. Das, S. Kumari, X. Huang, and M. Wazid, “Provably secure authenticated key agreement scheme for distributed mobile cloud computing services,” *Future Generation Computer Systems*, vol. 68, pp. 74–88, 2017.

[10] 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.

[11] H. Jannati and B. Bahrak, “An improved authentication protocol for distributed mobile cloud computing services,” *International Journal of Critical Infrastructure Protection*, vol. 19, pp. 59–67, Dec. 2017.

[12] J. Tsai and N. Lo, “A Privacy-Aware Authentication Scheme for Distributed Mobile Cloud Computing Services,” in *IEEE Systems Journal*, vol. 9, no. 3, pp. 805-815, Sep. 2015.

[13] M. Jakobsson and D. Pointcheval, “Mutual Authentication for Low-Power Mobile Devices,” *Proc. Financial Cryptography*, pp. 178-195, Feb. 2001.

[14] D. Boneh and M. Franklin. “Identity Based Encryption from the Weil Pairing,” in *Advances in Cryptology - CRYPTO*, volume 2139 of LNCS, pages 213-229. Springer, 2001.

[15] L. Chen, Z. Cheng and N. Smart, “Identity-Based Key Agreement Protocols from Pairings,” *Int'l J. Information Security*, vol. 6, pp. 213-41, 2007.

[16] Y. Dodis, L. Reyzin, and A. Smith, “Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data,” Advances in Cryptology - *EUROCRYPT 2004 Lecture Notes in Computer Science*, pp. 523–540, 2004.

[17] R. Sakai and M. Kasahara, “ID based cryptosystems with pairing on elliptic curve,” *Cryptol. ePrint Archive Report 2003/054*, 2003.

[18] X. Jia, D. He, N. Kumar, and K. K. R. Choo, “A provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing,” *IEEE Syst. J.*, vol. 14, no. 1, pp. 560–571, Mar. 2020.

[19] Y. Li, Q. Cheng, X. Liu and X. Li, "A Secure Anonymous Identity-Based Scheme in New Authentication Architecture for Mobile Edge Computing," in *IEEE Systems Journal*, Mar. 2020.

[20] AVISPA. SPAN, the Security Protocol ANimator for AVISPA. version 1.6 - Sep. 2017, [online] Available: http://people.irisa.fr/Thomas.Genet/span/. “Automated validation of internet security protocols and applications,” Mar. 2016, [online].

[21] “Automated validation of internet security protocols and applications,” Dec. 2019, [online] Available: http://www.avispa-project.org/.

[22] D. Dolev and A. Yao, “On the security of public key protocols,” in *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198-208, Mar. 1983.

[23] V. Odelu, A. K. Das, and A. Goswami, “A secure biometrics-based multiserver authentication protocol using smart cards,” *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 9, pp. 1953–1966, Sep. 2015.

[24] Q. Jiang, J. Ma and F. Wei, “On the Security of a Privacy-Aware Authentication Scheme for Distributed Mobile Cloud Computing Services,” in *IEEE Systems Journal*, vol. 12, no. 2, pp. 2039-2042, Jun. 2018.

[25] D. Xu, J. Chen and Q. Liu, “Provably secure anonymous three-factor authentication scheme for multi-server environments,” *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 2, pp. 611-627, 2019.

[26] E. Yoon and K. Yoo, “Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem,” *J. Supercomput.*, vol. 63, no. 1, pp. 235-255, Jan. 2013.

[27] 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.*, Nov. 2016.

[28] L. Zhou, X. Li, K.-H. Yeh, C. Su, and W. Chiu, ‘‘Lightweight IoT-based authentication scheme in cloud computing circumstance,’’ Future Gener. Comput. Syst., vol. 91, pp. 244–251, Feb. 2019.

[29] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog Computing and Its Role in the Internet of Things,” in *Proceedings of the ACM SIGCOMM Workshop on Mobile Cloud Computing*, Helsinki, Finland, Aug. 2012.

[30] Y. Mao, C. You, J. Zhang, K. Huang and K. B. Letaief, “A Survey on Mobile Edge Computing: The Communication Perspective,” in *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322-2358, Fourthquarter 2017.

[31] N. Abbas, Y. Zhang, A. Taherkordi and T. Skeie, “Mobile Edge Computing: A Survey,” in *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450-465, Feb. 2018.

[32] S. N. Shirazi, A. Gouglidis, A. Farshad and D. Hutchison, "The Extended Cloud: Review and Analysis of Mobile Edge Computing and Fog From a Security and Resilience Perspective," in *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2586-2595, Nov. 2017.

[33] D. Welch and S. Lathrop, “Wireless security threat taxonomy,” *IEEE Systems, Man and Cybernetics SocietyInformation Assurance Workshop*, 2003., West Point, NY, USA, 2003.

[34] I. Ahmad, S. Shahabuddin, T. Kumar, J. Okwuibe, A. Gurtov and M. Ylianttila, “Security for 5G and Beyond,” in *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3682-3722, Fourthquarter 2019.

[35] D. He, S. Chan and M. Guizani, “Security in the Internet of Things Supported by Mobile Edge Computing,” in *IEEE Communications Magazine*, vol. 56, no. 8, pp. 56-61, Aug. 2018.

[36] Lin, W. Yu, N. Zhang, X. Yang, H. Zhang and W. Zhao, “A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications,” in *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125-1142, Oct. 2017.

[37] S. Chaudhry, I. L. Kim, S. Rho, M. S. Farash and T. Shon, "An improved anonymous authentication scheme for distributed mobile cloud computing services", *Cluster Comput*., pp. 1-15, Aug. 2017.

[38] Z. Ghaffar, S. Ahmed, K. Mahmood, S. H. Islam, M. M. Hassan and G. Fortino, “An Improved Authentication Scheme for Remote Data Access and Sharing Over Cloud Storage in Cyber-Physical-Social-Systems,” in *IEEE Access*, vol. 8, pp. 47144-47160, 2020.

[39] A. Irshad, M. Usman, S. A. Chaudhry, H. Naqvi and M. Shafiq, “A Provably Secure and Efficient Authenticated Key Agreement Scheme for Energy Internet-Based Vehicle-to-Grid Technology Framework,” in *IEEE Transactions on Industry Applications*, vol. 56, no. 4, pp. 4425-4435, July-Aug. 2020.

[40] nsnam.org, “Ns-3.30,” https://www.nsnam.org/releases/ns-3-30/, 2019.

[41] A. Irshad, S. A. Chaudhry, O. A. Alomari, K. Yahya and N. Kumar, "A novel pairing-free lightweight authentication protocol for mobile cloud computing framework", *IEEE Syst. J.*, Jun. 2020.

[42] R. Amin, N. Kumar, G. P. Biswas, R. Iqbal, and V. Chang, “A light weight authentication protocol for IoT-enabled devices in distributed cloud computing environment,” *Future Gener. Comput. Syst.*, vol. 78, pp. 1005–1019, Jan. 2018.

[43] 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.