Computer Communications 186 (2022) 12–21

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

A seamless anonymous authentication protocol for mobile edge computing infrastructure

Khalid Mahmood [a], Muhammad Faizan Ayub [b], Syed Zohaib Hassan [b], Zahid Ghaffar [b], Zhihan Lv [c], Shehzad Ashraf Chaudhry [d,*]

[a] *Future Technology Research Center, National Yunlin University of Science and Technology, Yunlin, 64002, Taiwan, R.O.C*
[b] *Department of Computer Science, COMSATS University Islamabad, Sahiwal Campus, 57000, Pakistan*
[c] *Department of Game Design, Faculty of Arts, Uppsala University, Uppsala, Sweden*
[d] *Department of Computer Engineering, Faculty of Engineering and Architecture, Istanbul Gelisim University, Istanbul, Turkey*

ARTICLE INFO

Keywords:
Authentication
Authentication protocol
Mutual authentication
Mobile edge computing

ABSTRACT

Mobile Edge Computing (MEC) accommodates processing and data storage and manipulation capabilities across the scope of wireless network. In MEC environment, MEC servers along with the computing and storage capabilities are distributed at the edge of the network. However, due to the broad range of wireless communication, the fulfillment of security requirements still remain a challenging task in the for MEC environment. With the expeditious traffic expansion and growing end user requirements, the classic security protocols cannot encounter the innovative requirements of lightweightness and real-time communication. To meet these requirements, we have proposed an authentication protocol for the MEC environment. Our proposed protocol stipulates secure and efficient communication for all of the intended entities. Meanwhile, during its execution user anonymity remains intact. Moreover, our protocol is proven to be secure under the assumptions of formal security model. Additionally in this article, we have described the security properties of our protocol that it offers resistance against impersonation, session key computation and forward and backward secrecy attacks. The comparative analysis of time consumption and computation overheads are presented at the end of the paper, which is an evidence that our proposed protocol outperforms prior to various existing MEC protocols.

1. Introduction

In the ancient days of centralized data processing, cloud centers accomplish the task of processing and manipulating massive data produced by the devices. Some problems become non-negligible when the target data rises to an exceptional degree, for example bandwidth load and communication delay. Edge Computing has attracted a great deal of attention from both academic community and the communications engineering. Edge networks have adequate capability to process data locally in the edge computing model. Once processing is completed, the edge network sends the compiled results to the cloud computing center [1–4]. Edge computing not only minimizes the peril of sensitive data leakage but it also alleviate bandwidth demand in network communication.

In the modern era, Mobile Edge Computing (MEC) is the latest network architecture. MEC offers computing power and information storage features in the specified area of the wireless access network near to the end users. A typical MEC scenario is shown in Fig. 1, it is depicted that in the MEC paradigm, an edge server provides different services including data processing and conducts a preliminary analysis of data between edge devices and cloud computing centers. Edge server compiles all the computing tasks of data that are produced at the edge. In the MEC environment, the mobile edge devices bears insignificant computing power. MEC behaves differently as compared to Conventional Cloud Computing (CCC) in terms of end-to-end delay, network bandwidth, and data processing. MEC provides higher bandwidth and lower communication delay. These characteristics accredit MEC an adoptable technology to dynamic application requirements and consequently cut down the resource consumption at user side.

The rapid expansion of MEC technologies is compelling the demand for the latest type of security features. Authentic cryptographic approaches should be adapted according to modern functionalities of the MEC environment. In this infrastructure, mobile devices are involved in two-way exchange of data, where they act not only as data users but also as data providers. Thus, there are hurdles with identity authentication and access control when the process of data outsourcing is conducted. Moreover, in different stages, data confidentiality is a serious security threat. For example, patients wear medical smart

[*] Corresponding author.
E-mail address: sashraf@gelisim.edu.tr (S.A. Chaudhry).

https://doi.org/10.1016/j.comcom.2022.01.005
Received 30 May 2021; Received in revised form 24 July 2021; Accepted 5 January 2022
Available online 12 January 2022

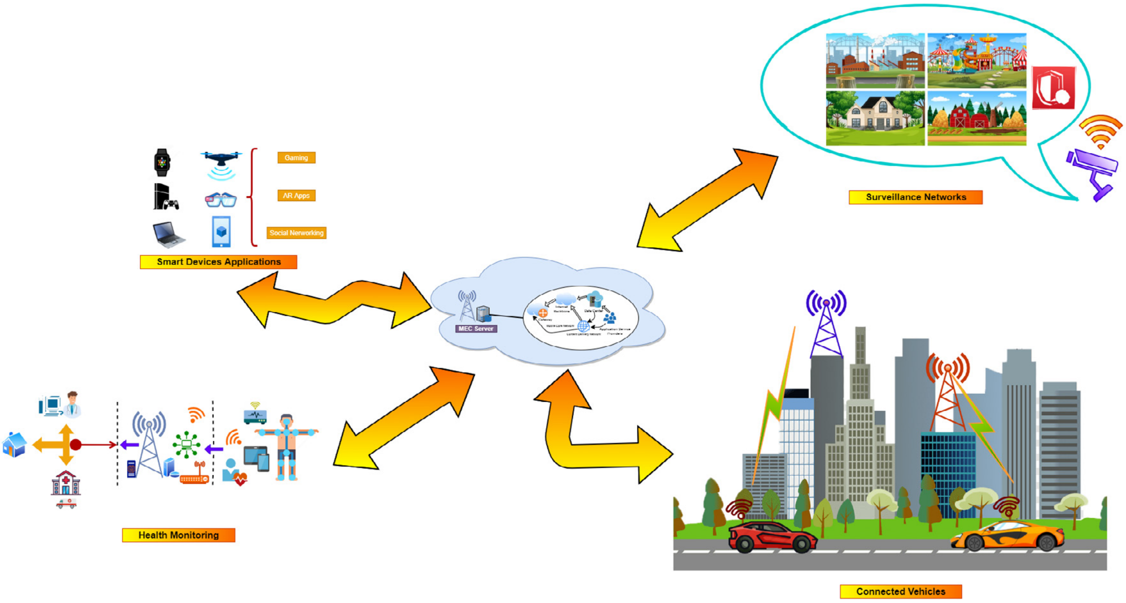

Fig. 1. MEC Generic Environment.

devices that has the capability to get private information. Whereas, the edge data center conduct the processing on the collected data through doctor and medical smart devices. However, there are some sensitive issues arises like how to protect the patient's crucial information and what is the amount of edge data centers have access to authority. Moreover, in this modern era the challenge of identity authentication also appears due to multiple communication networks. When the computing resources are shifted towards the edge, then real-time demands for data manipulation needs to be improved. For example, an Industrial control system (ICS) usually demands higher anticipation for real-time performance. These modern threats must be taken into consideration.

The existing solutions are found incapable to overcome the recent challenges. A well-known solution is Authenticated Key Exchange (AKE) protocols. The concept of AKE has acknowledged great consideration when Paterson and Al-Riyami suggested the first authentication less AKE protocol. Mostly, the AKE protocol is considered effective and suitable for solving network communication security and information security challenges. AKE is a well-known theoretical framework for establishing a network security information system [5]. AKE ensures that only registered entities obtain the session key. It also provides security for further communication, and make sure that unauthorized entities cannot be capable to intercept the shared information on the public channel. The classical model of AKE protocol has become unsuitable for the contemporary architectures due to expeditious evolution in the application development.

1.1. Motivation and contributions

In MEC environment, first issue is that the edge devices are lightweight. We generally consider symmetric ciphers for their least computing cost. However, in communication procedure, symmetric ciphers are incapable to offer user anonymity [6]. Trusted third party is another challenge which is not suitable and should have to be taken into consideration. Numerous AKE protocols for the MEC environment have been proposed by the researchers. Such as, Jia et al. proposed an advance efficient anonymous authenticated protocol for MEC [7]. We found some serious issues after studying their protocol. In this scenario, we defined a new secure authentication infrastructure and introduced a secure identity-based AKE protocol for the MEC environment with aided feature of anonymity. To ensure the anonymity, the proposed protocol does not need to share the user's identity information on the public channel. Moreover, we have conducted a comparison between our proposed protocol and the related protocols on behalf of communication cost and computational cost. Which, demonstrates that our protocol performs superior as compare to the numerous previous protocols. The essential contributions of our paper are as follows:

- We depict a latest authentication infrastructure for MEC terrain. The devised protocol offers efficient and secure communication between MEC server and the lightweight devices.
- We proposed a new AKE protocol which is utilizing identity-based cryptography. The user identification is not broadcasted in plain text in the authentication phase. User anonymity is depends on secure one-way hash function.
- Our proposed protocol's security is substantiated under the Random Oracle Model (ROM), with security characteristics analysis.

1.2. Roadmap of article

In the subsequent sections, we have described the related work in Section 2 and cryptographic preliminaries are discussed in Section 3. Whereas, our proposed protocol is presented in Section 4. The security analysis and performance evaluation of proposed protocol is described in Sections 5 and 6, respectively. Lastly, we have given conclusive remarks in Section 7.

2. Related work

Security challenges of MEC environments have been extensively explored and evaluated in the literature. Recently, Roman et al. [8] evaluated these security challenges for all computing infrastructures like Fog Computing (FC), mobile cloud computing and last but not the least mobile edge computing. Their analysis revealed the fact that most of the edge paradigms exhibits the identical behavior. The potential similarity is also demonstrated in the security structure. Mollah et al. [9] revealed the security flaws in the MEC terrain and conduct a comparison with modern works as per diverse privacy prerequisites. Ahmad and Rehmani [10] presented versatile applications where MEC infrastructure can be deployed. Furthermore, they debated the possibilities that MEC technology could pull it. Almajali et al. [5] appraised the modern existing authentication protocols under the MEC infrastructure. As a result of the huge adaptability of users in edge computing terrain, authentication protocols should concentrate on optimization in terms of flexibility and efficiency. In reply to these problems, researchers have proposed their own ideas. Thereafter, we will concisely review several authentication protocols in MEC terrain. Tsai and Lo [6] introduced an authentication protocol for distributed MEC environment in 2015. Tsai and Lo [6] claimed that their protocol delivered security capabilities and mutual authentication. Anyhow, after three years, Jiang et al. [7] proved that Tsai and Lo [6] protocol does not meet these objectives and highlighted few design shortcomings. Their improved suggestions were very helpful for other researchers to evade these flaws in their future work. These improved suggestions are very useful and played a vital role for future generations in the study of MEC terrain.

Researchers have presented numerous improvements. Irshad et al. [11] proposed an authentication protocol for a multi-server system in a MEC terrain. Their proposed protocol depends on bilinear pairing functions. Anyhow, Xiong et al. [12] highlighted the Irshad et al. [11] protocol weaknesses including revocation and user registration phase. To overcome these issues, Xiong et al. [12] introduced the enhanced authentication protocol for the distributed MEC terrain. They proved, that their proposed protocol is enough capable to defend numerous categories of cyber-attacks. Moreover, Li et al. [13] introduced an authentication protocol for mobile gadgets that are based on an elliptic curve. They claimed that their protocol provide perfect forward secrecy and kept user information secure from adversaries. However, numerous researchers are also very solicitous. Xu et al. [14] tackled a challenging problems of MEC terrain. Xu et al. [14] introduced a secure handoff protocol for user authentication. They claimed that their protocol can provide universality, efficiency and robust security. Kaur et al. [15] introduced a lightweight and efficient authentication protocol for MEC terrain. Their protocol design was based on one-way hash functions, cascading operations, and elliptic curves. Their protocol also used the advantages of random numbers and difficult logistic problems to safeguard numerous well-known attacks, for example, replay, a man in the middle, and camouflage attacks, etc. Their protocol consumes low communication and computation costs, which is more appropriate for resource constrained MEC applications.

In resource constraint environment designing of solutions and their implementation is a big challenge on lightweight gadgets that should be taken into consideration. Ibrahim [16] presented an efficient authentication protocol for Fog Computing terrain. They claimed that fog users can easily authenticate with an entirely new fog server without the re-registration phase in the FC environment. Moreover, their protocol did not utilize high communication costs, so it was a perfect fit for smart cards and small devices. Furthermore, Ke et al. [17] concentrate on the energy efficiency protocol for MEC applications in the Internet of Things (IoT) infrastructure. They introduced a mobility aware hierarchical MEC protocol for low-latency devices in an IoT environment. Experimental evaluation shows that their proposed protocol brought great impact in terms of performance such as improving efficiency and reduce the latency. Afterwards, Chen et al. [18] introduced an (Authenticated and Key Exchange) AKE protocol for the FC, which based on user's password and identity. Wang et al. [19] introduced an anonymous two-factor authentication protocol. They elaborated only two problems for designing anonymous two-factor based authentication protocol in the MEC environment. The delay problem in MEC environment is a big issue. Intharawijitr et al. [20] upset about that problem and they explored the reasons which may cause the two types of latencies of communication and computing in MEC. They introduced a new protocol for MEC applications which is enough capable to manage the estimated different delays and dynamically self-selected based on actual requirements. Messous et al. [21] elected the game theory model for Unmanned Aerial Vehicles (UAVs) to manage the variation between energy and latency. Ansari and Sun [22] proposed a Mobile Edge (ME) Internet of Things (IoT) infrastructure. They have also introduced the social and semantic IoT technology to supervise the unauthorized access control for IoT environment. Their research brought revolutionary benefits and encouragement to the other researchers.

Table 1
Notations table.

Notations	Description
$\mathcal{MU}_u$	The mobile user
ID_u	Identity of mobile user
SID_u	Private key of mobile user
$\mathcal{MEC}_s$	MEC Server
ID_s	Identity of MEC server
SID_s	Private key of MEC server
$\mathcal{RC}$	Registration Center
$\hat{s}$	Private key of MEC server
G	An additive cyclic group
G_T	A multiplicative cyclic group
P	An Elliptic Point on G
p, q	Large prime numbers
x, y	Random numbers in Z_q^*
$\hat{P}_{ub}$	Public key of $\mathcal{RC}$
$Gen()$	Secure identity extractor
SK	Shared Session Key
$h(.)$	secure one-way hash function, where $h : \{0,1\}^* \rightarrow Z_n^*$
$\oplus$	bitwise XOR operation
$\|$	concatenation operation

In the MEC terrain, untraceability of user identity is also a significant part. Such as, when we make payments on the internet, we expect that our personal information is hard to be tracked rather than payment information. As we know that in the MEC devices have less computing capabilities. So, some part of the identity encryption function can be transferred on the mobile gadgets for the sake of accomplishment. Tan [23] introduced a Proxy Blind Signature (PBS) protocol which is an identity based protocol without using pairing techniques. In their article, they have demonstrated that their protocol is secure under the Random Oracle Model (ROM) which is based on Discrete Logarithm (DL). After a while, Zhu et al. [23] also introduced an identity based proxy blind signature protocol, which is based on number theorem. Their protocol provide independence public key architecture because of which it can resists adversarial quantum computer threats. After critically reviewing all the source articles, we can conclude that each of them has at least one flaw remained unresolved. So, we have presented a secure and lightweight authentication protocol for user and MEC server environment which provides resistance against numerous attacks like replay, user and server impersonation, man-in-middle and establishment of private key etc, in a quite reasonable way.

3. Preliminaries

In this section, we illustrate the fundamental concepts related to Elliptic Curve Cryptography (ECC), the conventional adversarial model and primitive notations presented in Table 1.

3.1. Elliptic curve cryptography

Suppose p, q are two long prime numbers. E/F_p is an elliptic curve determined using the equation $y^2 = x^3 + ax + b(mod\,p)$, where $y, x, a, b \in F_p$. Moreover, an additive group is formed using the point addition approach by setting up each points on the curve and a "point at infinity" denoted as O. Here, consider G is a subgroup of order q and P is used to generate G. Whereas, the scalar multiplication is specified as $nP = P + P + \cdots + P(ntimes)$, while $n \in Z_q$. [24,25]

3.2. Complexity assumptions

In this section, we have illustrate few hard mathematical problems. Considering that p, q, G, G_T, P, e are already defined above. The illustrated complexity assumptions represents the basis of security in devised protocol.

(1) *Discrete logarithm (DL) problem:* In this problem, we have given an element $Q \in G$ and we have to find x whereas $Q = xP$.
(2) *Computational Diffie–Hellman (CDH) problem:* In this problem, we have given two elements $aP, bP \in G$. While $a, b \in Z_q$ are not defined. Compute abP.

3.3. Adversarial model

In this paper, we have illustrated the conventional adversarial model. As [26–31] have discussed attacker capabilities in their research, those capabilities are described as follows:

1- $\mathcal{A}$ can have access to all messages transmitted over public communication channel. $\mathcal{A}$ also have the ability of modifying, blocking, intercepting, replaying and deleting the messages.
2- $\mathcal{A}$ can guess the identity and password of the user and drone in polynomial time via dictionary attack.
3- $\mathcal{A}$ can use malicious devices to whether intercept the password or excerpt the relevant parameters from any mobile device. However, $\mathcal{A}$ cannot execute both actions simultaneously.
4- $\mathcal{A}$ can be a malicious or legal user of the MEC system.

4. Proposed scheme

The proposed seamless authentication scheme for mobile edge computing architecture is summarized in following phases:

4.1. Setup phase

In this phase, registration center $\mathcal{RC}$ sets up all the public parameters of the system. The setup phase is illustrated below:

1. Initially, $\mathcal{RC}$ selects a multiplicative cyclic group G_T and an additive cyclic group G with the similar sequence q. Whereas, P is an elliptic point on G.
2. Then, $\mathcal{RC}$ computes a public parameter as: $\hat{P}_{ub} = \hat{s}P$ respectively. Next, $\mathcal{RC}$ selects a one-way hash functions: $h_0 : \{0,1\}^* \times G \rightarrow Z_q^*$.
3. In the end, $\mathcal{RC}$ publishes all the computed public parameters $(G, G_T, e, P, P_{pub}, \hat{P}_{pub}, g, h_0)$.

4.2. Server registration phase

In this phase, Mobile Edge Computing (MEC) server $\mathcal{MEC}_s$ registers himself at RC and gets his private key SID_s. During this phase, all the messages are communicated over a private channel. The server registration phase is described as below:

1. Firstly, it is to be noted that the identity ID_{mec} of MEC_s is known publicly. Whenever $\mathcal{MEC}_s$ forwards a ‘ registration request message to $\mathcal{RC}$, $\mathcal{RC}$ will compute

$$h_{mec} = h_1(ID_{mec}) \tag{1}$$

$$SID_{mec} = \frac{1}{\hat{s} + h_{mec}} P \tag{2}$$

2. At last, $\mathcal{RC}$ transmits SID_{mec} to $\mathcal{MEC}_s$ as its private key.

4.3. Mobile user registration phase

In this phase, mobile user $\mathcal{MU}_{mec}$ registers himself at the RC and receives his private key SID_u. During this phase, all the messages are communicated over a private channel. The mobile user registration phase is described as below:

1. First, mobile user $\mathcal{MU}_{mec}$ sends a registration request with its identity ID_u to $\mathcal{RC}$. Next, $\mathcal{RC}$ chooses a random number $r_u \in Z_q^*$. After that, $\mathcal{RC}$ computes

$$R_u = r_u P \tag{3}$$

$$SID_u = (r_u + sh_u) mod q \tag{4}$$

2. At last, $\mathcal{RC}$ forwards SID_{mec} to $\mathcal{MU}_{mec}$ as its private key.

After the registration of all users, $\mathcal{RC}$ will keep the binaries of legitimate users (ID_u, v), where the value of v is calculated as: $v = h_0(ID_u \parallel SID_u)$.

4.4. Authentication and key agreement phase

Whenever, $\mathcal{U}_m$ needs to communicate with $\mathcal{D}_n$, he have to perform authentication and key agreement phase. The detail description of authentication and key agreement phase is described as below and presented in Fig. 2:

1. On the mobile user $\mathcal{MU}_u$ side, first $\mathcal{MU}_u$ selects a random number $x \in Z_q^*$ and computes

$$\alpha = xP \tag{5}$$

$$\lambda = x\hat{P}_{ub} \tag{6}$$

$$\beta = h(ID_u \parallel SID_u) \tag{7}$$

After that, $\mathcal{MU}_u$ encrypts the value of β using α and assign it to γ as: $\gamma = Enc_\alpha(\beta)$. Furthermore, $\mathcal{MU}_u$ computes $w = h(ID_{mec} \| \alpha \| \beta)$ and sends $\{\lambda, \gamma, w\}$ to $\mathcal{MEC}_s$ over a public channel.
2. After getting message $\{\lambda, \gamma, w\}$ from $\mathcal{MU}_u$, $\mathcal{MEC}_s$ computes $\alpha = \lambda SID_{mec}^{-1}$. Afterwards, $\mathcal{MEC}_s$ decrypts γ using α and assign it to β. Then, $\mathcal{MEC}_s$ authenticates $\mathcal{MU}_u$ using $w \overset{?}{=} h(ID_{mec} \| \alpha \| \beta)$. If it does not hold true then session will terminate. Otherwise, $\mathcal{MEC}_s$ further selects a random number $y \epsilon Z_q^*$ and computes $M = yP$. In the end, $\mathcal{MEC}_s$ forwards $\{Enc_{SID_{mec}}(M \parallel \beta)\}$ to $\mathcal{RC}$ through a public channel.
3. Whenever $\mathcal{RC}$ gets $\{Enc_{SID_{mec}}(M \parallel \beta)\}$ from $\mathcal{MEC}_s$, first $\mathcal{RC}$ decrypts the values of M and β using the private key SID_{mec} of $\mathcal{MEC}_s$. Then, $\mathcal{RC}$ verifies $Verifies\ \beta?$. If β does not verified then session will terminate. Otherwise, $\mathcal{RC}$ forwards $\{Enc_{SID_{mec}}(M \parallel RP_u)\}$ to $\mathcal{MEC}_s$ on a public channel.
4. On receiving $\{Enc_{SID_{mec}}(M \parallel RP_u)\}$ from $\mathcal{RC}$, $\mathcal{MEC}_s$ first decrypts the values of M and β using its own private key SID_{mec}. Then, $\mathcal{MEC}_s$ computes

$$N = M \oplus \alpha \tag{8}$$

$$Q = h(SID_{mec} \parallel M) \oplus RP_u \tag{9}$$

$$SK = h(h(SID_{mec} \parallel M) \| RP_u \| \alpha y \| h(SID_{mec} \| M)) \tag{10}$$

After that, $\mathcal{MEC}_s$ sends $\{N, Q\}$ to $\mathcal{MU}_u$ over the public channel.
5. When $\mathcal{MU}_u$ gets $\{N, Q\}$ from $\mathcal{MEC}_s$, $\mathcal{MU}_u$ computes

$$M = N \oplus \alpha \tag{11}$$

$$R = Q \oplus Gen(ID_u \parallel SID_u) \tag{12}$$

Then, $\mathcal{MU}_u$ verifies $\mathcal{MEC}_s$ using $SK \overset{?}{=} h(R \| Gen(ID_u \| SID_u) \| Mx \| R)$. If $\mathcal{MEC}_s$ does not verified the session will terminate instantly. Otherwise, $\mathcal{MU}_u$ will verify $\mathcal{MEC}_s$ successfully and share the common session key with $\mathcal{MEC}_s$.

5. Security analysis

In this section, we have presented the formal and informal security analysis of the devised protocol. In order to prove the security of devised protocol, we have utilized the assumptions of widely used Random Oracle Model (ROM). Moreover, informal security analysis demonstrates that the devised protocol provides resistance against numerous security attacks including user impersonation and server impersonation attack.

5.1. Formal security analysis

This subsection illustrates a theorem which proves that the devised protocol provides session key agreement and mutual authentication

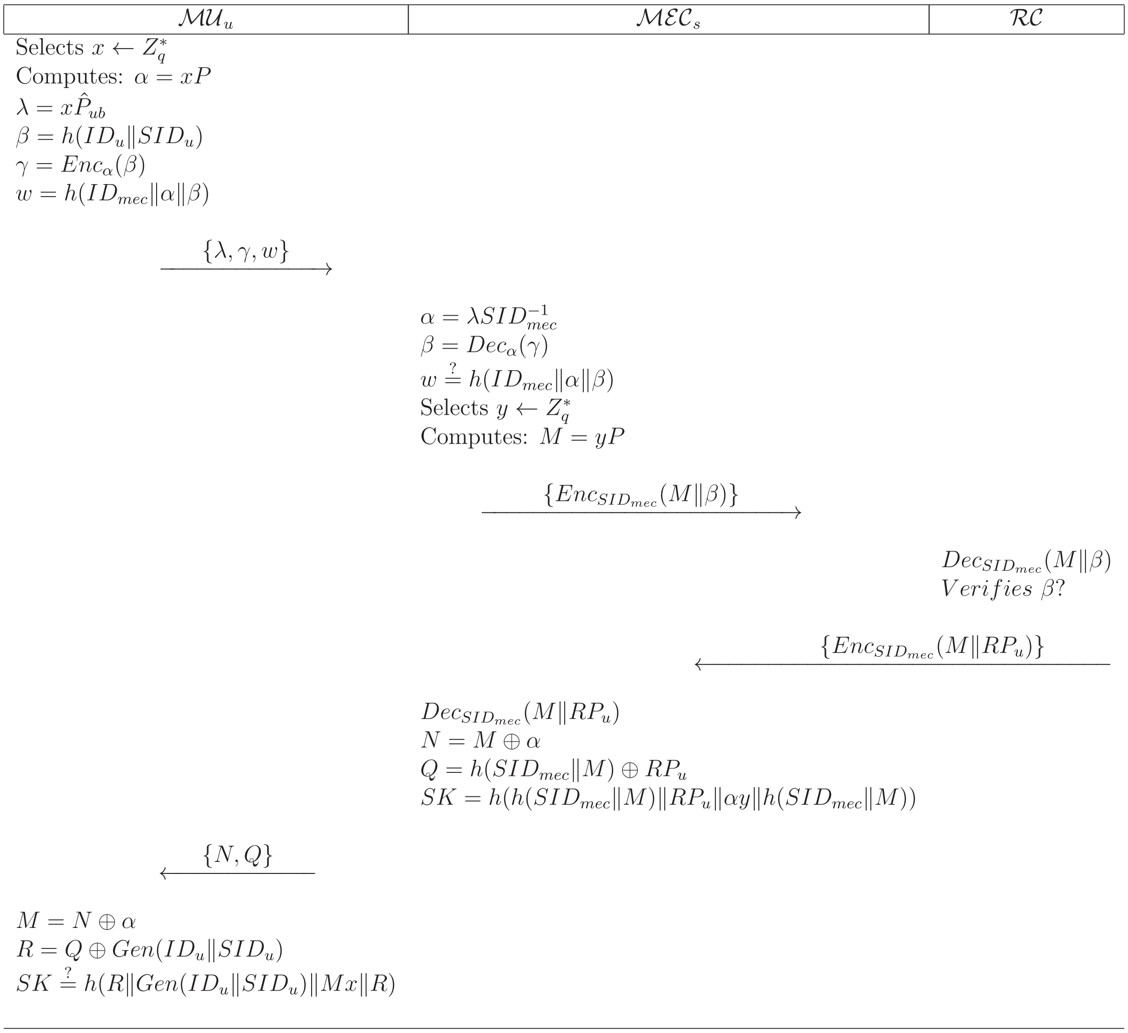
$\mathcal{MU}_u$
$\mathcal{MEC}_s$
$\mathcal{RC}$
Selects $x \leftarrow Z_q^*$
Computes: $\alpha = xP$
$\lambda = x\hat{P}_{ub}$
$\beta = h(ID_u \| SID_u)$
$\gamma = Enc_\alpha(\beta)$
$w = h(ID_{mec} \| \alpha \| \beta)$
$\{\lambda, \gamma, w\}$
$\alpha = \lambda SID_{mec}^{-1}$
$\beta = Dec_\alpha(\gamma)$
$w \overset{?}{=} h(ID_{mec} \| \alpha \| \beta)$
Selects $y \leftarrow Z_q^*$
Computes: $M = yP$
$\{Enc_{SID_{mec}}(M \| \beta)\}$
$Dec_{SID_{mec}}(M \| \beta)$
$Verifies\ \beta?$
$\{Enc_{SID_{mec}}(M \| RP_u)\}$
$Dec_{SID_{mec}}(M \| RP_u)$
$N = M \oplus \alpha$
$Q = h(SID_{mec} \| M) \oplus RP_u$
$SK = h(h(SID_{mec} \| M) \| RP_u \| \alpha y \| h(SID_{mec} \| M))$
$\{N, Q\}$
$M = N \oplus \alpha$
$R = Q \oplus Gen(ID_u \| SID_u)$
$SK \overset{?}{=} h(R \| Gen(ID_u \| SID_u) \| Mx \| R)$

Fig. 2. Authentication and Key Agreement Phase.

under the assumptions of well-known Random Oracle Model (ROM). The theorem proves that if $\mathcal{A}$ being successful to impersonate user under the consideration of Elliptic Curve Discrete Logarithm Problem (ECDLP) then $\mathcal{A}$ still can never to be able to interchange messages with $\mathcal{MU}_u$. In addition, we represent Π_U^i as the *ith* entity U, where $U \in (\mathcal{MU}_u, \mathcal{MEC}_s)$. Moreover, ROM has various queries for simulating an active attack, such as *hash*, *Extract*, *Send*, *Reveal*, *Corrupt* and *Test*. All of these queries are illustrated below:

- $H(msg_i)$: When $\mathcal{A}$ executes a hash query against an oracle Π_U^i, then in response entity will return a random number a_i as (msg_i, a_i). After that, $\mathcal{A}$ will keep (msg_i, a_i) in empty hash list h_{lt}.
- $Extract(ID_u)$: $\mathcal{A}$ can utilize this query to obtain the private key of $\mathcal{MU}_u$ using his identity ID_u.
- $Send(\Pi_U^i, msg)$: If $\mathcal{A}$ wants to performs a *Send* query, then he has to render a message *msg* as a query to an oracle Π_U^i and in respond $\mathcal{A}$ will get a *msg*.
- $Reveal(\Pi_U^i)$: $\mathcal{A}$ can use *Reveal* query to set up a session key with any specific oracle. Additionally, $\mathcal{A}$ can obtain the session key during the simulation process. Otherwise, the oracle will return null to $\mathcal{A}$ in response.
- $Corrupt(\Pi_U^i)$: In Authentication and Key Agreement (AKA) phase of devised protocol, $\mathcal{A}$ can execute *Corrupt* query, when $\mathcal{MU}_u$ tries to set up a session key with $\mathcal{MEC}_s$. Other than that, $\mathcal{A}$ can get the session key in response of *Corrupt* query.
- $Test(\Pi_U^i)$: *Test* query is used to measure the semantic security of the session key. When $\mathcal{A}$ renders *Test* query against an oracle Π_U^i, the oracle will respond with an arbitrary number b. If $b = 1$, the session key will also be send to $\mathcal{A}$ with response. Otherwise, a random string will be generated.

Theorem 1. *D represents the Uniformed dictionary of passwords with a size –D–. Whereas, Π presents the enhanced edition of protocol. If we suppose that bit-wise one-way hash function is specified as Random Oracle Model (ROM), then we have:*

$$Adt_{\Pi,D}(Adt) \leq \frac{q_{hq}^3 + (q_f + q_r)^2}{2^l} + \frac{q_{qh}}{2_l} + \frac{q_f}{|D|} \tag{13}$$

In above equation, q_r denotes Execute queries, q_{hq} denotes hashed queries and q_f denotes the Send queries.

Proof. The illustrated proof consists of four games. These games are known as game fusion. Whereas, all four games begins from $\mathcal{GA}_0$ and finishes at $\mathcal{GA}_3$ and $\mathcal{A}$ has no benefit of any sequence of games. Since, each $\mathcal{GA}_v (0 \leq v \leq 3)$, $Succ_v$ denoted as an identical event. Moreover, in each session $\mathcal{A}$ will do several attempts to identify the game.

- $\mathcal{GA}_0$: In $\mathcal{GA}_0$, each Π_U^k is performed in ROM. According to the concept of $Succ_v$, $\mathcal{A}$ seeks to pick the value of b through executing the *Test* query. In the end, we will get the following:

$$Adt_{\Pi,D}(A) = 3|Pr[Succ0] - 1| \tag{14}$$

- $\mathcal{GA}_1$: $\mathcal{GA}_1$ is almost similar to the $\mathcal{GA}_0$. However, the minor difference between them is that ROM produced a list of hash h_{lt}.

All of the involved entities in h_{lt} are also resides in the form of (OP,AP). Moreover, $\mathcal{GA}_1$ can identify OP in the case, if an involved entity resides in (OP,AP) also viewed in h_{lt}. Otherwise, an arbitrary OP $\in 0, 1$ will be send to $\mathcal{A}$. Moreover, the new entry (OP,AP) will also be added in h_{lt}. Whereas, the identities of server $\mathcal{A}$ and client L are utilized for various queries like $Test$, $Reveal$, $Corrupt$, $Execute$ and $Send$. Therefore, it can be stated that the game provides resistance against several attacks. So, we got:

$$Pr[Succ_0] = Pr[Succ_1] \tag{15}$$

- $\boldsymbol{\mathcal{GA}_2}$ $\mathcal{GA}_2$ has all of the computations that are performed in $\mathcal{GA}_1$. $\mathcal{GA}_2$ could be refused if any interception appears between communicant S and hash h. Mostly, the chances of conflict can be occurred due to the presence of entities in $(q_f + q_r)^4/4^{l+1}$. While q_{hq} sets the possibility of all hashed queries. In the same way, the chances of conflict in the result of all hashed oracles will be $q_{hq}^4/4^{l+1}$, whereas q_f sets as the peak number of queries can be $Send$ to the oracle and q_r sets as the peak number of $Send$ queries against oracle. l indicates the length of arbitrary bits. According to this scenario, we will get the following output:

$$|Pr[Succ_2] - Pr[Succ_1]| \leq \frac{q_{qh}^4 + (q_f + q_r)^4}{4^{l+1}} \tag{16}$$

- $\boldsymbol{\mathcal{GA}_3}$ In $\mathcal{GA}_3$, the execution of all queries are altered according to the chosen sessions in the $\mathcal{GA}_3$. Whereas, the evaluation of SK is updated to enabled. Due to which, SK will be totally anonymous from the private keys used in the executions. When Π_C^i, N, Q are executed using the $Send$ query, then the private keys SID_u, SID_{mec} are needed. Afterwards, we can compute $SK = h(R\|Gen(ID_u\|SID_u)\|Mx\|R)$. There are two different cases where both $\mathcal{GA}_2$ and $\mathcal{GA}_3$ are different to each other. These cases are described below:
  - **Case LA 1:** $\mathcal{A}$ asked for (N, Q) from $hq4$ and this event can be executed if $q_{hq}/2^l$.
  - **Case LA 2:** If $\mathcal{A}$ forwards query without $Send(\Pi_C^i, N, Q)$ and try to deceive the client, in this way, $\mathcal{A}$ should not be permitted to disclose the private parameter ID_u of the mobile user. The primary difference between both games $\mathcal{GA}_2$ and $\mathcal{GA}_3$ is:

  $$|Pr[Succ_3] - Pr[Succ_2]| \leq \frac{q_{hq}}{2^l} + \frac{q_f}{|D|} \tag{17}$$

  On the other hand,

  $$Pr[Succ_3] = 0.5 \tag{18}$$

The overall output of all equations will be:

$$\begin{aligned} Adt_{\Pi,D}(A) &= 3|Pr[Succ0] - 1| \\ &= 4|Pr[Succ_0] - Pr[Succ_3]| \\ \leq 2(|Pr[Succ_1] - Pr[Succ_4]| &+ Pr[Succ_4] - Pr[Succ_3]) \\ &\leq \frac{q_{hq}^2 + (q_f + q_r)^4}{4^l} + \frac{q_{hq}}{2^l} + \frac{q_f}{|D|} \end{aligned} \tag{19}$$

Therefore, we have noticed that R can precisely simulate the devised protocol using the consideration of Random Oracle Model (ROM). According to the illustration of games described above, we can conclude that R successfully accomplish if and only if $\mathcal{A}$ succeed to simulate and R failed. Moreover, if $\mathcal{A}$ has not commenced one of the queries $Reveal(\Pi_C^i)$ or $Corrupt(ID_c)$. Then R will never be able to stop the simulation. This happens because $N \ni \{1, 2, 3, \dots\dots, qj\}$ and $M \ni \{1, 2, 3, \dots\dots, qi\}$; the event $N = n$ true with the possibility of $1/qj$. While the event $M = m$ true with the possibility of $1/qi$. Thereby, $\mathcal{A}$ chooses Π_s^i and it is supposed as the challenge oracle with the possibility of $1/qiqs$. While R failed to stop during simulation. In the same way, R successfully accomplish the game with the possibility of $\in /qiqs$; where $\in$ is assumed as non-negligible. Moreover, $\in$ is also suitable for $\in /qiqs$ which means that R is capable to sort out the ECDLP problem with non-insignificant possibility through negating the hardness attribute of ECDLP. Hence, it can be said that the devised protocol is capable to provide mutual authentication between server and client.

5.2. Informal security analysis

This subsection presents the informal security analysis of devised authentication mechanism under the threat model illustrated in Section 3.3. Moreover, the consequential subsections show how the devised protocol provide robustness against various security attacks.

5.2.1. Mutual authentication

In the devised protocol, $\mathcal{MU}_u$ initiates login request $\{\lambda, \gamma, w\}$ to $\mathcal{MEC}_s$. Whenever, $\mathcal{MEC}_s$ receives the request from $\mathcal{MU}_u$, $\mathcal{MEC}_s$ authenticates $\mathcal{MU}_u$ by verifying $w \overset{?}{=} h(ID_{mec}\|\alpha\|\beta)$. The calculations of γ and w needs the identity ID_u and private key SID_u of $\mathcal{MU}_u$. As ID_u, SID_u are only known to $\mathcal{MU}_u$ and ID_s is only in the access of $\mathcal{MEC}_s$; therefore, $\mathcal{A}$ can never be able to access the ID_u, ID_s and SID_u. Thus, only legal $\mathcal{MU}_u$ can be authenticated by $\mathcal{MEC}_s$. Moreover, when $\mathcal{MEC}_s$ sends a request message $\{N, Q\}$ to $\mathcal{MU}_u$, then $\mathcal{MU}_u$ authenticates $\mathcal{MEC}_s$ using $SK \overset{?}{=} h(R\|Gen(ID_u\|SID_u)\|Mx\|R)$. Here, the calculations of SK contain M and RP_u. Both M and RP_u are encrypted with the private key SID_u of $\mathcal{MEC}_s$ which is only known to $\mathcal{MEC}_s$. As a result, $\mathcal{A}$ cannot be able to compute all these values. So, only legal $\mathcal{MEC}_s$ can be authenticated by $\mathcal{MU}_u$. Thereby, it is concluded that the devised protocol provides mutual authentication among participating entities.

5.2.2. Untraceability

$\mathcal{A}$ can trace the legitimate $\mathcal{MU}_u$ through the communicated messages transmitted over the public channel. In the devised protocol, the login request $\{\lambda, \gamma, w\}$ initiated from $\mathcal{MU}_u$ contains the identity ID_u and private key SID_u of $\mathcal{MU}_u$. Whereas ID_u, SID_u are only known to $\mathcal{MU}_u$. Therefore, if $\mathcal{A}$ tries to intercept the login request $\{\lambda, \gamma, w\}$, he/she cannot discover these $\{ID_u, SID_u\}$ values correctly. Hence, in the devised protocol $\mathcal{A}$ cannot be able to trace the identity ID_u of $\mathcal{MU}_u$.

5.2.3. User anonymity

In the devised protocol, the identity ID_u and private key SID_u of $\mathcal{MU}_u$ are not used to communicate over the public channel in plaintext. During the Authentication and Key Agreement (AKA) phase of devised protocol, both γ and w computed in the login request $\{\lambda, \gamma, w\}$ needs the identity ID_u and private key SID_u of $\mathcal{MU}_u$. As ID_u and SID_u are only known to $\mathcal{MU}_u$, so $\mathcal{A}$ does not have any knowledge of them. Therefore, it is truely stated that the devised protocol ensures user anonymity.

5.2.4. User impersonation attack

If $\mathcal{A}$ tries to manipulate $\mathcal{MEC}_s$ on behalf of legal $\mathcal{MU}_u$, then this will be called as user impersonation attack. In the devised protocol, if $\mathcal{A}$ tries to forward a request message $\{\lambda, \gamma, w\}$ as a legal user, then $\mathcal{A}$ needs to compute the correct values for γ and w. Whereas, both computations $\gamma = Enc_\alpha(\beta)$ and $w = h(ID_{mec}\|\alpha\|\beta)$ contains the identity ID_u and private key SID_u of legal $\mathcal{MU}_u$ via $\beta = h(ID_u \| SID_u)$. Whereas, ID_u and SID_u are only known to $\mathcal{MU}_u$. Thereby, the devised protocol can successfully prevent the user impersonation attack.

5.2.5. Server impersonation attack

Whenever $\mathcal{A}$ tries to exploit the $\mathcal{MEC}_s$ and start to accommodates all login requests of legal users on the account of legitimate $\mathcal{MEC}_s$, then this situation is known as server impersonation attack. In the devised protocol, when $\mathcal{A}$ tries to impersonate legal $\mathcal{MEC}_s$, then $\mathcal{A}$ has to relay the request message $\{N, Q\}$ to legal $\mathcal{MU}_u$. Whereas, the computations $N = M \oplus \alpha, Q = h(SID_{mec} \parallel M) \oplus RP_u$ of N and Q needs the private key SID_{mec} of legal $\mathcal{MEC}_s$. Moreover, M is also encrypted with the private key SID_{mec} of $\mathcal{MEC}_s$ and M can only be decrypted using the $\mathcal{MEC}_s$'s private key, which is unknown to $\mathcal{A}$. Therefore, $\mathcal{A}$ cannot be able to impersonate the legal $\mathcal{MEC}_s$. Hence, the devised protocol also resists the server impersonation attack.

5.2.6. Forward and backward secrecy

If $\mathcal{A}$ becomes able to figure out the private parameters like the private keys of $\mathcal{MU}_u$ and $\mathcal{MEC}_s$, then still $\mathcal{A}$ is not capable to find out the former and future session keys. In the devised protocol, the forward and backward secrecy is gained with the assistance of x, y, P. So, to calculate the session keys of devised protocol $\mathcal{A}$ should have the knowledge of correct values of random numbers x and y. For this, $\mathcal{A}$ needs to expand these $\alpha = xP$ and $M = yP$ computations which is similar to the hard problem solving of ECDLP. While the session keys relies on the random numbers x and y. In our protocol, the random numbers are generated freshly in each session irrespectively and also their values varied for each AKA phase. Therefore, there is no possibility for $\mathcal{A}$ to breach the forward and backward secrecy of the devised protocol. Hence, the session keys in the devised protocol attains the perfect forward and backward secrecy.

5.2.7. Prevents the session key computation attack

It is quite possible that $\mathcal{A}$ intercepts all the communicated messages that are being transmitted over the insecure channel between $\mathcal{MU}_u$ and $\mathcal{MEC}_s$. After that, if $\mathcal{A}$ succeeds to compute session key SK, then this will be called as session key computation attack. However, in the devised protocol, if $\mathcal{A}$ attempts to calculate SK of any specific session, then $\mathcal{A}$ needs to compute true value of $SK = h(R \| Gen(ID_u \| SID_u) \| Mx \| R)$. Since, the computation of SK requires the identity ID_u and private key SID_u of $\mathcal{MU}_u$, which is only known to $\mathcal{MU}_u$. Hence, the devised protocol provides resistance against session key computation attack.

5.2.8. Offers no clock synchronization

In various authentication protocols, timestamp is used to validate the novelty of transmitted messages. To achieve this, all the participated entities should have to be synchronized for communication which leads towards storage and computation cost. In the devised protocol, regardless of using timestamps, we have only used random numbers which are session specific and newly generated in each session. Therefore, it is stated that the devised protocol holds no clock synchronization property.

6. Performance analysis

In this section, we have presented a performance comparison between proposed and related protocols [11,32,33]. The comparison is presented in the context of communication cost, computation cost and security features.

6.1. Implementation scenario

The proposed and related protocols consist of three entities including (1) mobile user $\mathcal{MU}_u$ (2) MEC server $\mathcal{MEC}_s$ and (3) registration center $\mathcal{RC}$. In the proposed and related protocols, the registration phase is conducted only once. Therefore, in the performance evaluation, we have excluded both mobile user and MEC server registration phases. Moreover, we have neglected some cryptographic operations including XOR and string concatenation, which have insignificant computational costs. However, to find out the experimental results, the cryptographic operations used at $\mathcal{MU}_u$'s side are implemented on a mobile device. In the same way, desktop system (DS) is used to implement the cryptographic operations of $\mathcal{MEC}_s$ end. Furthermore, the cryptographic operations, their symbols, notations, description and running time is presented in Table 2. Table 3 shows the overview of system specifications for devices on which cryptographic operations are executed.

Table 2
Cryptographic operations and their running time.

Operation	Execution time	
	Mobile device	Desktop system
T_{hf}	1.003 ms	0.0022 ms
T_{pmt}	0.234 ms	0.0026 ms
T_{sed}	0.430 ms	0.0032 ms

T_{hf}: Time requires for hash function
T_{pmt}: Time requires for point multiplication
T_{sed}: Time requires for symmetric encryption/decryption

6.2. Computation cost comparison

In this subsection, we evaluate the computation overheads of the proposed and relevant protocols of specific terrain by using the computation time of cryptographic operations explained in Table 2.

Each protocol has a registration phase which is a one time process. We have only noted the time complexity of symmetric-encryption decryption, hash, point multiplication, and cryptographic operations to compute the computation overheads of the related and proposed protocol in the Authentication and Key Agreement (AKA) phase.

In AKA phase, $\mathcal{MU}_u$ log into the devices by using his ID_u in proposed protocol. Thereafter, $\mathcal{MU}_u$ executes three hash functions, one encryption/decryption and one point multiplication operations. As a result, the accumulative computation overhead on the $\mathcal{MU}_{u's}$ side is $3T_{hf} \times 1.006$ ms $+ T_{sed} \times 0.428$ ms $+ T_{pmt} \times 0.230$ ms $= 3.676$ ms. On the other hand, when $\mathcal{MEC}_s$ receives the login request message it utilizes three hash functions, two encryption/decryption and one point multiplication operations. Therefore, the accumulative computation overheads at $\mathcal{MEC}_s$ end is $3T_{hf} \times 0.0021$ ms $+ 2T_{sed} \times 0.0034$ ms $+ T_{pmt} \times 0.0027$ ms $= 0.0158$ ms. Consequently, in our proposed protocol the total computation overhead is 3.676 ms $+ 0.0158$ ms $= 3.6918$ ms. The computation overhead of related protocols [11,32,33] is calculated in the same way which is illustrated in Table 4. Moreover, we demonstrate the computation cost for three entities of the proposed and related protocol in Fig. 3. Where, several verifiers are labeled on x-axis and computation time is displayed on y-axis. Fig. 3 clearly demonstrates that the computation overhead of proposed protocol at the side of $\mathcal{MU}_u$ and AC_pO is less than the other related protocols. While, the computation overhead of the proposed protocol at the side of $\mathcal{MEC}_s$ is slightly higher than [11] but far less than the other related protocols.

6.3. Communication overhead comparison

The communication overhead of a protocol associates with the total number of bits required to transmit the messages between all involved entities. We have provided an explicit comparison of the communication overhead of proposed and related protocols [11,32,33]. However, it is important to mention that, during the calculation of communication overhead of proposed and related protocols, we have just investigated the messages that are transmitted during the AKA phase between $\mathcal{MU}_u$ and $\mathcal{MEC}_s$.

Table 5 shows the required bits of various communication operations which are explained in T. Limbasiya et al. [34]

In our proposed protocol, during the AKA phase, the entities $\mathcal{MU}_u$ and $\mathcal{MEC}_s$ transfer three messages $\{\lambda, \gamma, \omega\}, \{Enc_{SID_{mec}}(M \parallel \beta)\}$ and $\{N, Q\}$ mutually. The communication overhead of these messages is as follows: 160+128+256, 128 and 160+256. Therefore, in our proposed

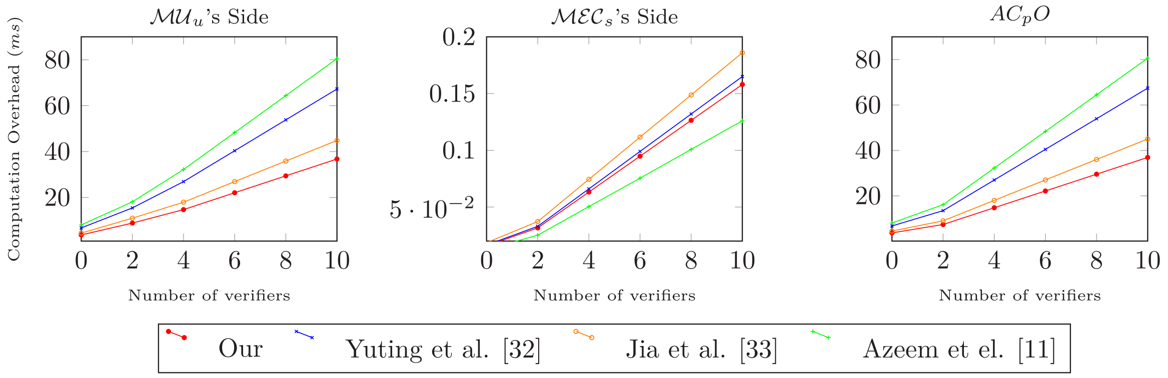

Fig. 3. Representation of Aggregated Computation Overhead.

Table 3
System specifications of devices.

Items	Specifications					
	Model	RAM	Generation	OS	Processor	Library/Language
Mobile	Samsung Galaxy S9	8 GB	–	Android	2.4 GHz	PyCrypto
System	Intel Corei5	12 GB	5th	Windows	2.5 GHz	PyCrypto

Table 4
Analysis of computation and communication overheads.

Protocols	$\mathcal{MU}_u$'s side	$\mathcal{MEC}_s$'s side	AC_pO	AC_mO
Our	$3T_{hf} + T_{sed} + T_{pmt} \approx 3.676$ ms	$3T_{hf} + 2T_{sed} + T_{pmt} \approx 0.0158$ ms	3.6918 ms	1088 bits
[32]	$6T_{hf} + 3T_{pmt} \approx 6.726$ ms	$4T_{hf} + 3T_{pmt} \approx 0.0165$ ms	6.7425 ms	2176 bits
[33]	$4T_{hf} + 2T_{pmt} \approx 4.484$ ms	$5T_{hf} + 3T_{pmt} \approx 0.0186$ ms	4.5026 ms	1504 bits
[11]	$8T_{hf} \approx 8.048$ ms	$6T_h \approx 0.0126$ ms	8.0606 ms	1184 bits

AC_pO: Aggregated Computation Overhead; AC_mO:Aggregated Communication Overhead.

Table 5
Assumptions for communication overhead.

Attribute	Symbol	Required bits
Symmetric encryption/decryption	T_{sed}	128
Identity	ID	160
XOR	$\oplus$	160
Concatenation	$\|$	160
Bi-linear pairing	b	160
Point multiplication	T_{pmt}	160
Hash function	T_{hf}	256

Table 6
Comparison of security features.

Security features	Protocols			
	Ours	[32]	[33]	[11]
User impersonation attack	Yes	No	Yes	Yes
Server impersonation attack	Yes	No	Yes	Yes
Mutual authentication	Yes	Yes	Yes	No
User anonymity	Yes	No	Yes	Yes
Untraceability	Yes	Yes	No	Yes
Forward and backward secrecy	Yes	Yes	Yes	No

Yes: Provides Resilience;
No: Does Not Provides Resilience

protocol the accumulative communication overhead is 544 + 128 + 416 = 1088 bits. The communication overhead of related protocols in the same way. The communication overhead of Azeem et al. [11], Jia et al. [33] and Yuting et al.'s [32] protocol is 1184, 1504 and 2176 bits, respectively. Moreover, the comprehensive communication overhead comparison is illustrated in Table 4.

In Fig. 4, we demonstrate the detailed communication overhead comparison for three communicating entities ($\mathcal{MU}_u$, $\mathcal{MEC}_s$ and AC_mO) of the proposed and related protocols. In Fig. 4, several related protocols are marked on the x-axis and the number of transmission bits for corresponding communicants are labeled on the y-axis for each entity. This comparison provides a brief view of the communication latency for three communicating entities of the proposed and related protocols. When the proposed and related protocols are executed numerous times for each entity, the comparison shows that the proposed protocol utilized far less communication overhead at the side of $\mathcal{MEC}_s$ and AC_mO as compared to the various related protocols. The communication cost at the side of $\mathcal{MU}_u$ of the proposed protocol is slightly higher than [11] and less than other related protocols.

6.4. Security feature comparison

In Table 6, the comparative analysis of security features among proposed and related protocols is presented. Table 6 clearly shows that the proposed protocol restricts the major security attacks whereas, related protocols [11,32,33] fails to resists various security threats including user impersonation attack, server impersonation attack. Hence, it is concluded that the proposed protocol provides more aided security features with respect to contemporary related protocols.

In the end, Table 4 shows a very clear picture of the proposed protocol performance which is more consistent and better than the related protocols. It is fact that the computation and communication overhead of the proposed protocol is less but on the other hand, related protocols have high communication and computation overheads. Furthermore, Table 6 shows that our proposed protocol provides additional security features, for example, it can easily resist impersonation and user-anonymity threats.

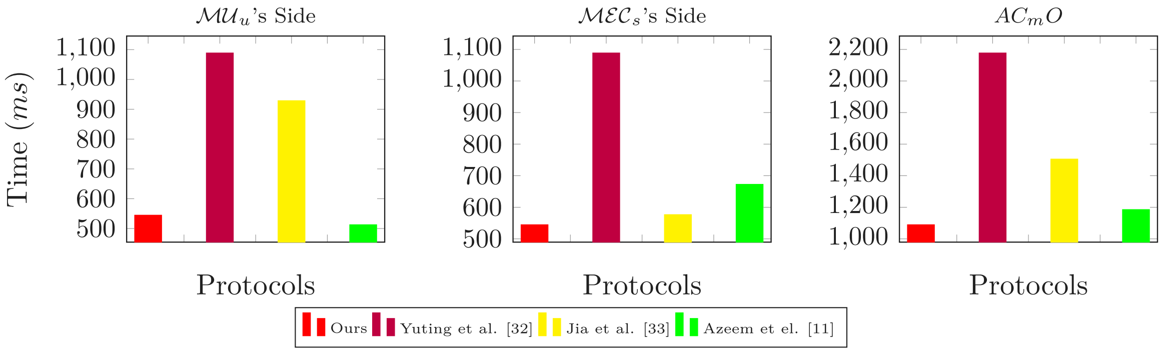

Fig. 4. Communication overhead comparison of different protocols.

7. Conclusion

In the current era, various emerging technologies are utilized to connect individuals across the worldwide. However, it is also essential to ensure secure communication among these individuals. There is an indispensable need to design authentication protocols in order to offer robust security and privacy. In this paper, we have designed a lightweight identity-based protocol for the infrastructure of mobile edge computing (MEC). The devised protocol offers the features of un-traceability and anonymity of mobile users. We validate the security requirements of the devised protocol through the well-known Random Oracle Model (ROM). Furthermore, the devised protocol is also informally examined through the provided threat model to verify its security against numerous security attacks. Additionally, the performance analysis of devised protocol presents a detailed view which shows that our protocol offers better efficiency in terms of computation and communication costs.

CRediT authorship contribution statement

Khalid Mahmood: Writing – original draft, Writing – revised draft. **Muhammad Faizan Ayub:** Validation, Formal analysis. **Syed Zohaib Hassan:** Conceptualization, Reviewing original and revised draft. **Zahid Ghaffar:** Conducted performance analysis and comparisons. **Zhihan Lv:** Investigation, Cryptanalysis, Validation, Informal analysis. **Shehzad Ashraf Chaudhry:** Supervision, Visualization, Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] S.A. Chaudhry, I.L. Kim, S. Rho, M.S. Farash, T. Shon, An improved anonymous authentication scheme for distributed mobile cloud computing services, Cluster Comput. 22 (1) (2019) 1595–1609.

[2] A. Irshad, S.A. Chaudhry, O.A. Alomari, K. Yahya, N. Kumar, A novel pairing-free lightweight authentication protocol for mobile cloud computing framework, IEEE Syst. J. (2020) 1–9, http://dx.doi.org/10.1109/JSYST.2020.2998721.

[3] B.D. Deebak, F. Al-Turjman, L. Mostarda, Seamless secure anonymous authentication for cloud-based mobile edge computing, Comput. Electr. Eng. 87 (2020) 106782.

[4] R.-F. Liao, H. Wen, J. Wu, F. Pan, A. Xu, H. Song, F. Xie, Y. Jiang, M. Cao, Security enhancement for mobile edge computing through physical layer authentication, IEEE Access 7 (2019) 116390–116401.

[5] S. Almajali, H.B. Salameh, M. Ayyash, H. Elgala, A framework for efficient and secured mobility of IoT devices in mobile edge computing, in: 2018 Third International Conference On Fog And Mobile Edge Computing (FMEC), IEEE, 2018, pp. 58–62.

[6] J.-L. Tsai, N.-W. Lo, A privacy-aware authentication scheme for distributed mobile cloud computing services, IEEE Syst. J. 9 (3) (2015) 805–815.

[7] Q. Jiang, J. Ma, F. Wei, On the security of a privacy-aware authentication scheme for distributed mobile cloud computing services, IEEE Syst. J. 12 (2) (2016) 2039–2042.

[8] R. Roman, J. Lopez, M. Mambo, Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges, Future Gener. Comput. Syst. 78 (2018) 680–698.

[9] M.B. Mollah, M.A.K. Azad, A. Vasilakos, Security and privacy challenges in mobile cloud computing: Survey and way ahead, J. Netw. Comput. Appl. 84 (2017) 38–54.

[10] E. Ahmed, M.H. Rehmani, Mobile edge computing: opportunities, solutions, and challenges, Elsevier, 2017.

[11] A. Irshad, M. Sher, H.F. Ahmad, B.A. Alzahrani, S.A. Chaudhry, R. Kumar, An improved multi-server authentication scheme for distributed mobile cloud computing services., TIIS 10 (12) (2016) 5529–5552.

[12] J. Li, W. Zhang, V. Dabra, K.-K.R. Choo, S. Kumari, D. Hogrefe, Aep-ppa: An anonymous, efficient and provably-secure privacy-preserving authentication protocol for mobile services in smart cities, J. Netw. Comput. Appl. 134 (2019) 52–61.

[13] J. Li, W. Zhang, V. Dabra, K.-K.R. Choo, S. Kumari, D. Hogrefe, Aep-ppa: An anonymous, efficient and provably-secure privacy-preserving authentication protocol for mobile services in smart cities, J. Netw. Comput. Appl. 134 (2019) 52–61.

[14] X. Yang, X. Huang, J.K. Liu, Efficient handover authentication with user anonymity and untraceability for mobile cloud computing, Future Gener. Comput. Syst. 62 (2016) 190–195.

[15] K. Kaur, S. Garg, G. Kaddoum, M. Guizani, D.N.K. Jayakody, A lightweight and privacy-preserving authentication protocol for mobile edge computing, in: 2019 IEEE Global Communications Conference (GLOBECOM), IEEE, 2019, pp. 1–6.

[16] M.H. Ibrahim, Octopus: an edge-fog mutual authentication scheme., IJ Netw. Secur. 18 (6) (2016) 1089–1101.

[17] K. Zhang, S. Leng, Y. He, S. Maharjan, Y. Zhang, Mobile edge computing and networking for green and low-latency internet of things, IEEE Commun. Mag. 56 (5) (2018) 39–45.

[18] C.-M. Chen, Y. Huang, K.-H. Wang, S. Kumari, M.-E. Wu, A secure authenticated and key exchange scheme for fog computing, Enterp. Inf. Syst. (2020) 1–16.

[19] D. Wang, D. He, P. Wang, C.-H. Chu, Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment, IEEE Trans. Dependable Secur. Comput. 12 (4) (2014) 428–442.

[20] K. Intharawijitr, K. Iida, H. Koga, Simulation study of low latency network architecture using mobile edge computing, IEICE Trans. Inf. Syst. 100 (5) (2017) 963–972.

[21] M.-A. Messous, H. Sedjelmaci, N. Houari, S.-M. Senouci, Computation offloading game for an UAV network in mobile edge computing, in: 2017 IEEE International Conference On Communications (ICC), IEEE, 2017, pp. 1–6.

[22] N. Ansari, X. Sun, Mobile edge computing empowers internet of things, IEICE Trans. Commun. 101 (3) (2018) 604–619.

[23] Z. Tan, Efficient pairing-free provably secure identity-based proxy blind signature scheme, Secur. Commun. Netw. 6 (5) (2013) 593–601.

[24] S.A. Chaudhry, Correcting PALK: Password-based anonymous lightweight key agreement framework for smart grid, Int. J. Electr. Power Energy Syst. 125 (2021) 106529, http://dx.doi.org/10.1016/j.ijepes.2020.106529.

[25] S.A. Chaudhry, K. Yahya, M. Karuppiah, R. Kharel, A.K. Bashir, Y.B. Zikria, Gcacs-iod: A certificate based generic access control scheme for internet of drones, Comput. Netw. 191 (2021) 107999, http://dx.doi.org/10.1016/j.comnet.2021.107999.

[26] M. Wazid, A.K. Das, N. Kumar, A.V. Vasilakos, J.J. Rodrigues, Design and analysis of secure lightweight remote user authentication and key agreement scheme in internet of drones deployment, IEEE Internet Things J. 6 (2) (2018) 3572–3584.

[27] A. Irshad, M. Usman, S.A. Chaudhry, H. Naqvi, M. Shafiq, A provably secure and efficient authenticated key agreement scheme for energy internet-based vehicle-to-grid technology framework, IEEE Trans. Ind. Appl. 56 (4) (2020) 4425–4435, http://dx.doi.org/10.1109/TIA.2020.2966160.

[28] S. Hussain, S.A. Chaudhry, O.A. Alomari, M.H. Alsharif, M.K. Khan, N. Kumar, Amassing the security: An ECC-based authentication scheme for internet of drones, IEEE Syst. J. (2021) 1–8, http://dx.doi.org/10.1109/JSYST.2021.3057047.

[29] S.A. Chaudhry, M.S. Farash, N. Kumar, M.H. Alsharif, Pflua-dIoT: A pairing free lightweight and unlinkable user access control scheme for distributed IoT environments, IEEE Syst. J. (2020) 1–8, http://dx.doi.org/10.1109/JSYST.2020.3036425.

[30] Z. Ali, S.A. Chaudhry, K. Mahmood, S. Garg, Z. Lv, Y.B. Zikria, A clogging resistant secure authentication scheme for fog computing services, Comput. Netw. 185 (2021) 107731, http://dx.doi.org/10.1016/j.comnet.2020.107731.

[31] S.A. Chaudhry, K. Yahya, F. Al-Turjman, M.H. Yang, A secure and reliable device access control scheme for IoT based sensor cloud systems, IEEE Access 8 (2020) 139244–139254, http://dx.doi.org/10.1109/ACCESS.2020.3012121.

[32] Y. Li, Q. Cheng, X. Liu, X. Li, A secure anonymous identity-based scheme in new authentication architecture for mobile edge computing, IEEE Syst. J. (2020).

[33] X. Jia, D. He, N. Kumar, K.-K.R. Choo, A provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing, IEEE Syst. J. 14 (1) (2019) 560–571.

[34] T. Limbasiya, M. Soni, S.K. Mishra, Advanced formal authentication protocol using smart cards for network applicants, Comput. Electr. Eng. 66 (2018) 50–63.