The Journal of Supercomputing (2020) 76:8729–8755
https://doi.org/10.1007/s11227-020-03160-x

A secure biometric-based authentication protocol for global mobility networks in smart cities

Meysam Ghahramani[1] · **Reza Javidan**[1] · **Mohammad Shojafar**[2]

Published online: 21 January 2020

Abstract

Smart city is an important concept in urban development. The use of information and communication technology to promote quality of life and the management of natural resources is one of the main goals in smart cities. On the other hand, at any time, thousands of mobile users send a variety of information on the network, and this is the main challenge in smart cities. To overcome this challenge and collect data from roaming users, the global mobility network (GLOMONET) is a good approach for information transfer. Consequently, designing a secure protocol for GLOMONET is essential. The main intention of this paper is to provide a secure protocol for GLOMONET in smart cities. To do this, we design a protocol that is based on Li et al.'s protocol, which is not safe against our proposed attacks. Our protocol inherits all the benefits of the previous one; it is entirely secure and does not impose any more communication overhead. We formally analyze the protocol using BAN logic and compare it to similar ones in terms of performance and security, which shows the efficiency of our protocol. Our proposed protocol enables mobile users and foreign agents to share a secret key in 6.1 ms with 428 bytes communication overhead, which improves the time complexity of the previous protocol to 53%.

Keywords Smart city · Secure protocol · Mobility network · Formal and informal security analysis · Impersonation attack

✉ Mohammad Shojafar
m.shojafar@surrey.ac.uk; m.shojafar@ieee.org

Meysam Ghahramani
m.ghahramani@sutech.ac.ir

Reza Javidan
javidan@sutech.ac.ir

[1] Computer Engineering and IT Department, Shiraz University of Technology, Shiraz, Iran

[2] ICS/5GIC, University of Surrey, Guildford GU2 7XH, UK

1 Introduction

The larger distribution of the Internet and mobile devices among citizens, the dimensions of cities and the need for energy consumption are reasons for studying smart cities [7]. The definitions of smart cities are various. According to [6], "A smart city is a well-defined geographical area, in which high technologies such as ICT, logistic, energy production, and so on, cooperate to create benefits for citizens in terms of well-being, inclusion and participation, environmental quality, intelligent development; it is governed by a well-defined pool of subjects, able to state the rules and policy for the city government and development".

1.1 Smart cities and authentication of mobile users

Internet of Things (IoT) is one of the newest concepts that has gained a lot of attention in recent years, despite its rapid development. IoT applications include Health Care, Home Automation, and Intelligent Transport Systems. Smart cities form based on widespread applications of IoTs [5]. Smart cities have various types with different dimensions, such as Smart Governance, Smart Economy, Smart Environment, Smart People, Smart Living, and Smart Mobility [7]. Smart cities can provide people with services like Smart Traffic Lights, Smart Parking, and Remote Health Monitors. These are examples that are now available to many people. In addition to these services, other services in smart cities are very challenging and require to design them carefully. Connected cars and smart public transit are examples of such services [21].

In smart cities, some users send up a lot of information at any time and move from one place to another. In such a situation, we should maintain the connection between mobile wireless devices. In this case, it is suggested to use the global mobility network (GLOMONET) that allows authorized users to gain roaming service at any location. To do so, a mobile user (*MU*) registers at home agent (*HA*), and then *MU* can get access to the services when he/she roams into a foreign agent (*FA*) [39]. *FA*'s establish communications with *HA*s over the Internet, but the question of how mobile users can trust *FA* is a severe challenge. Fortunately, there are solutions to this challenge, one of which is using authentication protocols. In this paper, our goal is to address this issue and present our possible solution for it. We give a summary of these topics and how this article relates to smart cities in Fig. 1. The authors in [27] delineate IoT security issues.

According to Fig. 1, we aim to examine the smart cities, which includes applications of IoT in two dimensions and services categories (see the green hexagon components presented in Fig. 1). This figure shows a secure protocol for GLOMONET that falls into "Smart Mobility" and "Security and Privacy" in terms of dimensions and services. GLOMONET can manage the interaction between the FA and HA by the help of the authentication protocol presented on it.

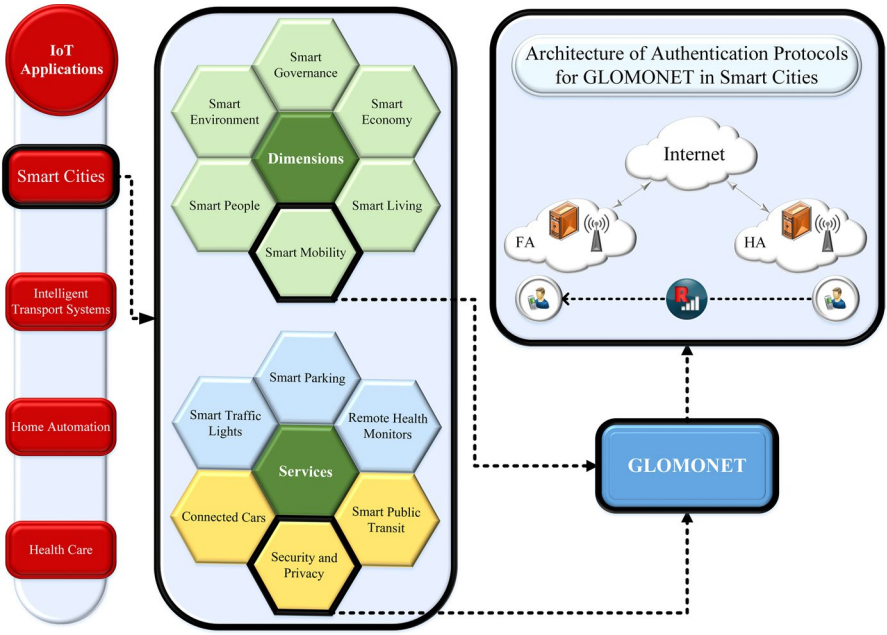

Fig. 1 Smart cities and authentication. *HA* = Home agent; *FA* = foreign agent

1.2 Motivation

It is essential to provide a secure authentication protocol for smart cities. On the one hand, we require to identify the features of a secure protocol. Motivated by these considerations, we propose a novel secure protocol that includes reliable features applied in the smart city environment. To do so, in Sect. 5, we first highlight some vulnerable features implemented in the smart city environment and describe how to attack them. Moreover, some of these features may be interdependent so that any defect in one of them may compromise the security of others. Additionally, the scholars suggest several protocols and prove their security by formal methods. However, some other researchers discover such protocol vulnerabilities and present their novel solutions against them that we describe them in the next section.

Beside, optimizing energy consumption in smart cities is necessary because most network-connected devices suffer from limited resources. So far, protocols based on a bilinear pairing have been proposed that is more than 7 times slower than an elliptic curve multiplication. As a result, using a secure elliptic curve based authentication protocol can be a good solution to the problem.

1.3 The main goal and contribution of the paper

Techniques like deep learning are used to increase security on the IoT. This technique relates to our work in two ways. On the one hand, authentication and key agreement protocols have phases such as registration, login, and security information change phases, in which there is information such as users' biometric information that may change over time. Deep learning techniques can be used to validate this information. Once this information is verified by the system, users enter the authentication and key agreement phase and are allowed to send information to each other on the network. On the other hand, deep learning techniques can detect malicious behaviors by examining network data. This process is illustrated in Fig. 2. Our analysis shows that even if deep learning methods do their job well and the users use encrypted messages, if there is any vulnerability in the protocol, the attackers are still able to threaten users' privacy. We found that Li et al.'s protocol [22] is not secure. Therefore, the intention of this paper is to reveal the vulnerabilities of Li et al.'s protocol, which have not been revealed yet. Fortunately, this interesting protocol can be modified with the least possible changes, and this will eliminate the need for a new protocol and its re-analysis. The results of this article can be summarized as follows:

- In this paper, we analyze one of the authentication protocols for GLOMONET in smart cities.
- We discover that the work in [22] suffers from multiple vulnerabilities such as: offline guess attack, lacking of user anonymity and unlinkability, home agent *HA* and foreign agent *FA* impersonation attack, insecure key distribution, and malicious foreign agent *FA* attack.

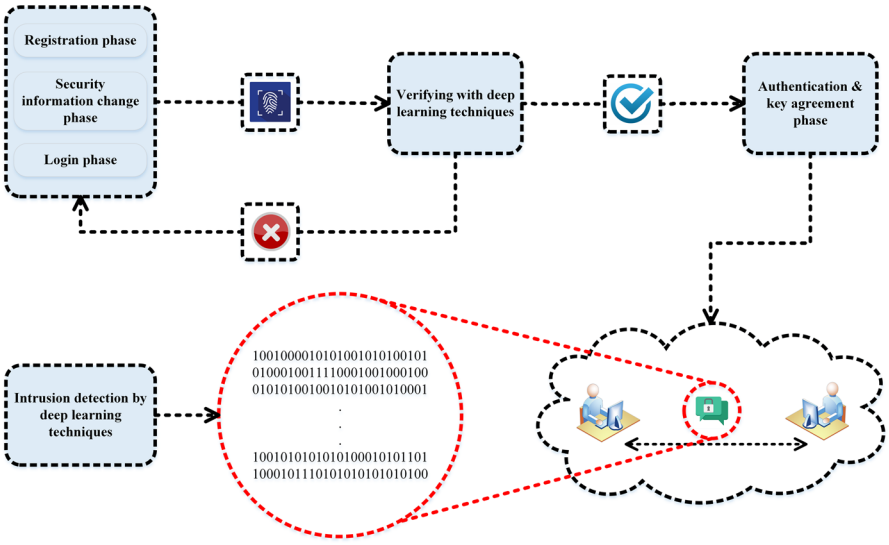

Fig. 2 Deep learning and authentication

- We propose an alternative protocol to counter the proposed attacks and compare our protocol with similar ones in terms of security and efficiency.
- We show that our method is not only able to withstand the proposed attacks, but its time complexity is approximately 53% of the previous one.

1.4 Roadmap

In Sect. 3, we introduce the problem and assumptions which are necessary to reading this paper. Section 4 reviews the Li et al.'s scheme. Section 5 analyzes the scheme and explains the discovered attacks and shows that such attacks are practical. Section 6 proposes an enhanced version. In Sect. 7, the security of our protocol is proven; we compare our proposed protocol with other similar ones in this section. Finally, the conclusion is presented in Sect. 8. The symbols used in this article are summarized in Table 1.

2 Related works

In this section, we will give a brief overview of the existing secure protocol in smart cities. Authentication protocols are not limited to smart cities and can also be used to provide security for wireless body area networks [30], cloud computing services [13], e-Health systems [1], multi-server environments [3] and wireless sensor networks [20]. Numerous authentication protocols have been proposed so far, in which

Table 1 Notations

Notation	Description
HA, MU, FA	Home agent, mobile user, and foreign agent
K_{FA}	The secret key of FA which is generated by HA
ID_{MU}, PW_{MU}	Identity and password of MU
Gen, Rep	Randomized procedures of fuzzy extractor
ID_{HA}, ID_{FA}	Identity of HA and FA
p	A large prime number
s	The secret key of HA
$e : G_1 \times G_1 \rightarrow G_2$	A bilinear pairing
P	A generator of G_1
$P_{pub_{FA}}, P_{pub_{HA}}$	Public keys of FA and HA
$\oplus$	Xor operation
$H(.), h(.)$	Hash functions
$\|\|$	Concatenation operator
T_{sym}, T_h	Time needed to compute symmetric encryption and hash function
T_p, T_{ecc}	Time needed to bilinear pairing and elliptic curve point multiplication operators
l_A	The length of A

several parameters are important. These parameters can generally be divided into *three* categories of *factors*, *performance*, and *method of analysis*. The security of a protocol can be based on secrets such as passwords that the user knows, things like the smart card that holds them, or the intrinsic information of the user such as his/her fingerprint. If a protocol has all of these factors, it is called a *three-factor protocol*.

Another parameter to note is the protocol analysis method. Protocols can be formally or informally analyzed. BAN logic is one of the most popular formal methods used in many protocols. For example, the authors in [3, 20, 29] have used this method to prove the mutual authentication feature of their proposed protocol. Other formal methods include the random oracle method used in references [15, 30]. Another way to test protocol security is to use tools such as AVISPA and ProVerif that authors of [1, 35] have used.

Unfortunately, formal proofs do not necessarily guarantee resistance to all known attacks. For example, the authors of [16], using BAN logic, demonstrated that their protocol ensures mutual authentication, while after careful analysis, the vulnerabilities of this protocol became apparent in [19]. Similarly, the authors of [12] applied BAN logic to prove the security of their protocol and indicted that based on their achievements by implementation with AVISPA method, their protocol is safe while the authors in [2] report its weaknesses. Lastly, although the security of the protocol presented in [33] was demonstrated using random oracles, the interesting analysis performed in [18] confirmed the shortcomings of the previous one.

The last parameter to address here is protocol performance. It is necessary to see what function the protocol uses in order to calculate the efficiency of a protocol. The time required to execute these functions and their output length are two important parameters in the performance of the protocol because a protocol will send messages over the network that will impose too much pressure on the network over a long period. Common functions used in various protocols include hash functions, symmetric encryption, and message authentication codes, elliptic curve cryptography, bilinear pairings, chaotic maps, and problems such as discrete logarithm problem. We show these functions with $H(\alpha)$, $E(\alpha)$, $ECC(\alpha)$, $BP(\alpha)$, $T(\alpha)$, and $DLP(\alpha)$, respectively. All of these functions are based on a hard problem that ensures protocol security. In all of these functions, it is easy to compute $f(\alpha)$ with f and α while finding α using f and $f(\alpha)$ is considered impossible, where $f \in \{H, E, ECC, BP, T, DLP\}$. Using each of these functions has its disadvantages and benefits. For example, functions E and H have a much lower time complexity than other ones. On the other hand, secure communications using these functions require members to share information in advance. Other functions can eliminate this need while suffering from more time complexity. In Table 2, we compare some recent authentication protocols with our proposed one.

In 2008, Wu et al. suggested a cryptographic protocol for GLOMONET [36]. He et al. investigated and analyzed the GLOMONET and revealed a registered user at home agent *HA* can obtain the identities of other users at the same home agent *HA*. They also showed that the protocol is vulnerable to replay and impersonation attacks [17]. Similarly, they offered an anonymous scheme for roaming in GLOMONET environments. Later on, the authors in [14] proposed a protocol for GLOMONET.

Table 2 Comparison of related works

Refs.	Factors	Informal analysis	Random oracle	BAN logic	Tools	*H*	*E*	*ECC*	*BP*	*T*	*DLP*	Suggested for
[15]	3	•	•	○	○	○	○	○	○	○	•	Mobile phones
[30]	2	•	•	○	○	•	•	•	•	○	○	Wireless body area
[24]	3	•	○	○	○	•	•	•	○	○	○	Industrial IoT
[38]	2	•	•	○	○	•	•	•	○	○	○	User privacy
[8]	3	•	•	○	○	•	○	○	○	○	○	Cloud-based industrial IoT
[13]	2	•	•	○	○	•	○	○	○	○	○	Cloud computing services
[31]	2	•	•	○	○	•	○	○	○	○	○	Social IoT
[1]	3	•	○	○	•	•	○	○	○	○	○	e-Health systems
[35]	3	•	•	○	•	•	•	○	○	○	○	Generic IoT
[32]	1	•	○	○	○	○	•	○	○	○	○	Smart homes
[3]	3	•	•	•	•	•	○	○	○	•	○	Multi-server environment
[29]	3	•	•	•	•	•	○	○	○	•	○	Crowd sourcing IoT
[20]	2	•	○	•	○	•	•	○	○	○	○	Wireless sensor networks
[22]	3	•	•	○	○	•	○	•	•	○	○	Smart cities
This paper	3	•	○	•	○	•	○	•	○	○	○	Smart cities

However, the lack of security of this protocol was proven by Li et al. in 2017. They found that the presented protocol in [14] lacks wrong password detection and session key update mechanisms and is vulnerable to a denial-of-service attack. They also reported other protocol's vulnerabilities and proposed a biometric authentication protocol for the smart city [22]. The scheme presented in [22] suffers from several serious vulnerabilities in which we present them in Sect. 5.

3 Problem definition and assumptions

The goal of our work is to present a secure authentication protocol for global mobility networks in smart city. In such networks, there is a home agent *HA* which mobile users *MU*s can communicate with it. Unfortunately, the range of home agent services is limited and it is sometimes necessary for *MU* to leave and communicate remotely. Foreign agents *FA*'s can be used in such situations. Moreover, there must be a mechanism to assure the user that *FA* is valid. For this reason, the mechanism should benefit from the registration phase. In this phase, *FA* and *MU* register and home agent *HA* provide information to them which is used for authentication and key agreement. Foreign agent *FA* in the registration phase receives K_{FA} and distributes $K_{FA} \times P$ as its public key.

Similarly, mobile user *MU* receives a smart card/device containing the information necessary to authenticate. After registering, mobile user *MU* can communicate with home agent *HA* by the help of *FA* whenever he investigates outside the coverage area of home agent *HA*. Our proposed protocol determines the

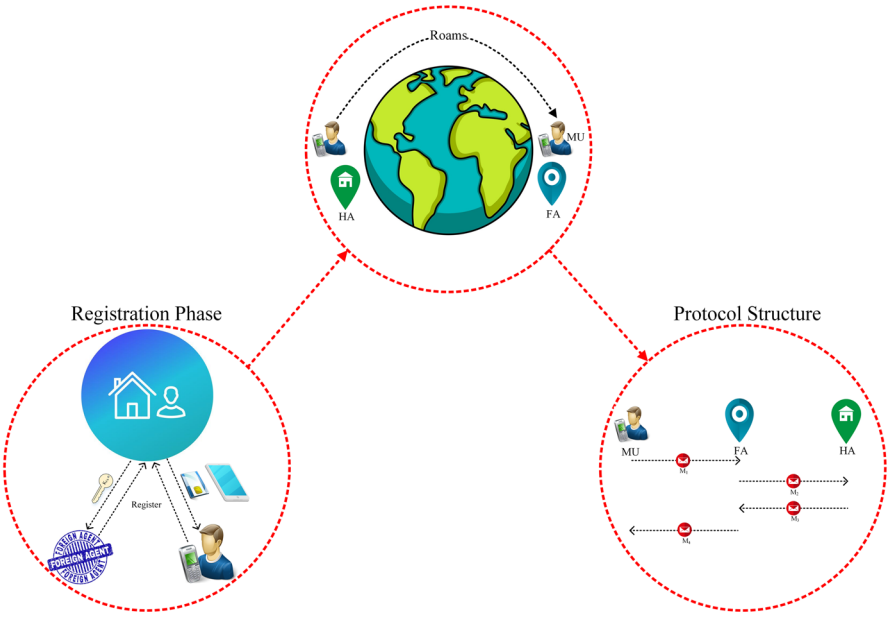

Fig. 3 Problem definition. HA = Home agent; MU = mobile user; FA = foreign agent

structure of communicated messages between *MU*, *FA*, and *HA*. In the protocol, four messages are transmitted over the network; after successful sending of these messages, *FA* and *MU* are able to initiate their own encrypted communications under a secret key. This process is illustrated in Fig. 3.

Note that the content of these four messages must be such that the protocol is highly secure and no adversary can become a threat to the network. For this reason, the protocol uses concepts such as bilinear pairing, fuzzy extractor, elliptic curves, and hash functions. In the remainder of this section, these concepts are briefly introduced and the security requirements of an authentication protocol and the capabilities of adversaries are discussed in detail in Sect. 5.

Elliptic curve: Let K be an arbitrary field; an elliptic curve E over K is defined by an equation $E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$, where $a_1, a_2, a_3, a_4, a_6 \in K$, and $\Delta \neq 0$ is the discriminant of E. In this paper, we focus on $K = GF(p)$, where p is a large prime number such that E is a secure curve. In this case, E is isomorphic to $y^2 = x^3 + ax + b$, where $a, b \in GF(p)$ and $\Delta = -16(4a^3 + 27b^2)$.

In the case of secure elliptic curves, there are three well-known problems that play an important role in establishing the security of protocols. These problems are as follows:

- Given an integer x and a point P over E, it is easy to compute point $ECC(x) = x \times P$. This problem is called the multiplication over elliptic curve.
- Consider points P and $ECC(x)$ over E, finding the integer x is intractable which is called the Elliptic Curve Discrete Logarithm Problem (ECDLP).
- Consider points P, $ECC(x)$, and $ECC(y)$ over E, it is intractable to compute $ECC(x \times y)$. This problem is called the Elliptic Curve Diffie–Hellman Problem (ECDHP).

For more details about elliptic curves and related problems, please see [34].

Another concept that we need to introduce in this article is the bilinear pairing. Li et al. used this concept to enhance the security of the Gope and Hwang's protocol.

Bilinear pairing: A bilinear pairing on (G_1, G_2) is a map $e : G_1 \times G_1 \rightarrow G_2$ that satisfies several conditions. Among these conditions, we would like to introduce one of them as $e(P, P)^{\alpha\times\beta} = e(\alpha \times P, \beta \times P) = e(\alpha \times \beta \times P, P) = e(P, \alpha \times \beta \times P)$. Similar to elliptic curves, in bilinear pairing we have three problems as follows:

- Given an integer x and $e(P, P)$, it is easy to compute $BP(x) = e(x \times P, P) = e(P, x \times P)$.
- Given $BP(x)$ and $e(P, P)$, finding the integer x is intractable.
- Given $e(P, P)$, $BP(x)$, and $BP(y)$, it is intractable to compute $BP(x \times y)$.

For more details about bilinear pairing, we refer the interested readers to [26].

In this article, we also use the hash functions. These functions are represented by h and H in this paper and the following assumptions are made for them:

- Hash functions are easily computable and publicly available to everyone.
- Given $h(x)$, find x is intractable.
- Given x, find $y \neq x$ such that $h(x) = h(y)$ is intractable.
- Finding two distinct x and y such that $h(x) = h(y)$ is very hard.

The last concept introduced in this section is a fuzzy extractor that is used to store and retrieve biometrics information safely because these inputs are noisy and we need to convert noisy inputs into reliably reproducible random strings.

Fuzzy extractor: A fuzzy extractor is a pair of randomized procedures, *Gen* and *Rep* such that $(R, P) \leftarrow Gen(w)$ and $Rep(w, P) = R$. If $distance(w, w')$ is less than a threshold t, then $Rep(w', P) = R$. This feature is useful for biometric information because such information may change over time. For more information about fuzzy extractor, please see [9].

Now, we are ready to review Li et al.'s protocol.

4 Review of Li et al.'s protocol

In 2017, Li et al. presented a biometric-based three-factor authentication scheme for global mobility networks in the smart city [22]. This protocol has several phases, which are based on elliptic curves and related problems, bilinear pairing, and fuzzy extractor. These concepts are introduced in the previous section. The protocol consists of three phases: registration phase, authentication and key agreement phase, and password change phase; moreover, this protocol has a mechanism for updating the session key. In this section, we review the first two phases. For more details about this protocol, please see [22].

4.1 Registration phase

In this phase, *FA* chooses an identity ID_{FA} and sends it to *HA*, and *HA* computes $K_{FA} = \frac{P}{H(ID_{FA})+s}$ and sends it to *FA* secretly. In this case, $H : \{0,1\}^* \rightarrow Z_p^*$ is a secure hash function which maps arbitrary length inputs to an element of the group Z_p^*, where p is a large prime number. Similarly, *MU* chooses an identity ID_{MU}, password PW_{MU}, the random number r, computes $HPW_{MU} = h(PW_{MU} \| r)$, and sends $\{ID_{MU}, HPW_{MU}, R_{MU}\}$ to *HA* via a secure channel, where R_{MU} is fingerprint information that is extracted by the fuzzy extractor. Then, *HA* computes $B_1 = h(ID_{MU} \| HPW_{MU} \| R_{MU})$, $B_2 = h(ID_{MU} \| s)$, $B_3 = h(ID_{MU} \| R_{MU}) \oplus B_2$, and sends $\{B_1, B_3\}$ to *MU* secretly. Finally, *MU* stores $\{G_1, G_2, P, P_{pub_{HA}}, H(.), h(.), e(P, P), Rep, Gen, P_{MU}, B_1, B_3, r\}$ to the mobile device.

4.2 Authentication and key agreement phase

At the beginning of this phase, *MU* inputs ID_{MU}, PW_{MU}, imprints the fingerprint information B'_{MU}, and the mobile device calculates $R'_{MU} = Rep(B'_{MU}, P_{MU})$ and

$B'_1 = h(ID_{MU}\|h(PW_{MU}\|r)\| R'_{MU})$. If $B'_1 = B_1$, the mobile device generates two random numbers $a_i, b_i \in Z_p^*$, computes $B_2 = B_3 \oplus h(ID_{MU}\|R_{MU})$, $D_1 = e(P,P)^{a_i}$, $D_2 = a_i \times (H(ID_{FA}) \times P + P_{pub_{HA}})$, $D_3 = b_i \times P$, $D_4 = h(ID_{FA}\|ID_{HA}\|D_1\|D_3)$, $D_5 = ID_{MU} \oplus h(b_i \times P_{pub_{HA}})$, $D_6 = h(B_2 \|ID_{FA} \|ID_{HA}\|D_3)$, and sends $M_1 = \{ID_{HA}, D_2, D_3, D_4, D_5, D_6\}$ to *FA* via a public channel. Otherwise, *MU* is not valid.

After getting message M_1, *FA* computes $D'_1 = e(D_2, K_{FA})$ and $D'_4 = h(ID_{FA}\|ID_{HA}\|D'_1\|D_3)$. If D'_4 is equal to D_4, *FA* generates a random number $c_i \in Z_p^*$, calculates $D_7 = c_i \times P$, $D_8 = c_i \times P_{pub_{HA}}$, $D_9 = h(ID_{FA}\|ID_{HA}\|D_3\|D_5\|D_6\|D_8)$, and sends $M_2 = \{D_3, D_5, D_6, D_7, D_9, ID_{FA}\}$ to *HA* via a public channel.

When receiving message M_2, *HA* computes $ID'_{MU} = D_5 \oplus h(s \times D_3)$, $B'_2 = h(ID'_{MU}\|s)$, and $D'_6 = h(B'_2\|ID_{FA}\|ID_{HA}\|D_3)$. If *MU* is valid, then D'_6 must be equal to D_6. *HA* verifies this and calculates $D'_8 = s \times D_7$ and $D'_9 = h(ID_{FA}\|ID_{HA}\|D_3\|D_5\|D_6\|D'_8)$. Similarly, if *FA* is valid, then D'_9 must be equal to D_9. *HA* checks this and computes $D_{10} = h(ID'_{MU}\|ID_{FA}\|D_3\|D_7)$ and $D_{11} = h(ID_{HA}\|ID_{FA}\|D_3\|D_5\|D'_8\|D_{10})$. At last, *HA* sends $M_3 = \{D_{10}, D_{11}\}$ to *FA*.

After receiving the message M_3, *FA* calculates $D'_{11} = h(ID_{HA}\|ID_{FA}\|D_3\|D_5\|D_8\|D_{10})$ and verifies $D'_{11}? = D_{11}$. If so, *HA* is valid and *FA* computes $SK_{FM} = h(c_i \times D_3)$, $D_{12} = h(SK_{FM} \|D'_1 \|D_3 \|D_7\|D_{10})$, and sends $M_4 = \{D_7, D_{10}, D_{12}\}$ to *MU*. Finally, *MU* calculates $D'_{10} = h(ID_{MU}\|ID_{FA}\|D_3\|D_7)$ and checks $D'_{10}? = D_{10}$. This equality means that *FA* is valid, and *MU* computes $SK_{MF} = h(b_i \times D_7)$ as a secret key for future session with *HA*. If the secret key is valid, $h(SK_{MF}\|D_1\|D_3\| D_7\| D'_{10})$ must be equal to D_{12}. *MU* checks this in the final step of the protocol.

In the next section, we analyze this protocol and propose several attacks.

5 Proposed attacks

As discussed in Sect. 3, the purpose of this article is to present a secure protocol. For this reason, we must know the capabilities of an adversary and be aware of the features that a secure protocol should have. As a result, this section will discuss these cases and then presents the vulnerabilities of Li et al.'s protocol.

We need a threat model to evaluate the capabilities of an adversary. This paper uses the Dolev–Yao threat model [10] which assumes:

- All network communications are transmitted through unsafe channels that are always subject to eavesdropping.
- There are always adversaries who eavesdrop on all conversations, store them in their database, and this information is always available to them.
- Adversaries can prevent the messages of legitimate users from reaching the destination for a limited time.
- Adversaries are also able to make changes to the messages of authorized users and send messages to behalf users.

To resist against the adversary, the protocol structure must be such that none of the above is a threat to the protocol. Moreover, a secure protocol should have the following features:

Preserve Anonymity and Unlinkability: Many users prefer to do their work anonymously, so this is one of the important features that should be carefully considered. In the case of anonymity, the protocol structure must be such that the adversary $\mathscr{A}$ cannot find ID_{MU} using messages sent over public networks. More rigorously, $\mathscr{A}$ should cannot detect a relation between two different messages sent by a user. In other words, it should not be clear how many messages belong to a specific user even if the identity of the user is not disclosed. This feature is called Unlinkability.

Resistance to privileged insider attack: In most authentication protocols, there is a registration phase. This phase assumes that communications are transmitted by a secure channel. This security can be ensured by visiting users in person. Now suppose there is an adversary to see the user requesting a home agent to get a smart card/device. The request must be structured in such a way that it does not pose any threat to the protocol if it is possessed by the adversary. This states that the protocol is secure against the privileged insider attack.

Forward and Backward security: During the execution of the protocol, several session keys are generated by the members involved in the protocol. The structure of these keys should be such that the disclosure of a key does not lead to the threat of the previous keys, and does not compromise the security of the keys that will be generated in the future. If the previous keys are secure, having the current key, we say that the forward security is guaranteed, and if the future keys are secure, the backward security is guaranteed.

Resistance to offline guess attack: Another feature of a secure protocol is resistance to offline guess attack. Suppose someone has a message claiming that a specific user with ID_{MU} sent it. The authentication protocol structure must be such that it is impossible to verify this claim. Note that in this feature we don't care how ID_{MU} is found.

Resistance to impersonation attack: As discussed earlier, $\mathscr{A}$ can send messages behalf authentic network members. In this case, a secure protocol must be designed so that authentic members can detect the fake messages of $\mathscr{A}$. If $\mathscr{A}$ sends a message to the receiver R, while R believes that this message can only be generated by the sender S, we say $\mathscr{A}$ impersonated S. A robust protocol must be designed so that $\mathscr{A}$ cannot impersonate members involved in the protocol.

Secure key distribution: During the registration phase, HA generates secret keys and delivers to the foreign agents. The process of generating these keys must be such that they are not only unique but also valid agents cannot calculate each other's secret keys.

Resistance to malicious agents: The last feature discussed in this section is the behavior of the parties involved in the protocol. The structure of the protocol should be such that malicious members cannot exploit the system. For example, in our proposed protocol, the foreign agent for communicating with the mobile user must also send a message to the home agent. If *FA* behaves maliciously, it should not be able to share a secret key with *MU* without contacting *HA*.

We now present attacks that prove the weaknesses of Li et al.'s scheme.

5.1 Offline guess attack

In this attack, if an adversary guesses the user's password or identity, there is a way to verify it. In authentication and key agreement phase, *MU* sends M_1 to *FA*, *FA* submits M_2 to *HA*, *HA* sends M_3 to *FA* and *FA* submits M_4 to *MU*. Note that all of these messages are sent via a public channel. Therefore, ID_{FA}, D_3, D_7, and D_{10} are not secret.

Suppose the adversary wants to know whether ID'_{MU} belongs to *MU*. To do so, he/she verifies $h(ID'_{MU}\|ID_{FA}\|D_3\|D_7)? = D_{10}$, and this means that the protocol is vulnerable to this attack. But what are the risks of the success of this attack? Password, *ID* and biometric information such as fingerprint have been used to increase security in this protocol. It is shown below that the only important security parameter in this protocol is ID_{MU}. Let's see what can be done by knowing ID_{MU}.

5.2 Lacking user anonymity and unlinkability

The authors of [22] claimed that to calculate ID_{MU}, b_i or s is required which is means that even if ID_{MU} is correctly guessed, it is necessary to solve ECDLP to check it. As mentioned above, it is not the case.

Now suppose that $M_1^j = \{ID_{HA}, D_2^j, D_3^j, D_4^j, D_5^j, D_6^j\}$, $M_2^j = \{D_3^j, D_5^j, D_6^j, D_7^j, D_9^j\}$, $M_3^j = \{D_{10}^j, D_{11}^j\}$ and $M_4^j = \{D_7^j, D_{10}^j, D_{12}^j\}$ have been sent through the public channel, where $1 \le j \le n$ and n is the number of sessions. Let ID_{MU_i} be the identity of MU_i, obviously MU_i has sent $\{M_1^j : 1 \le j \le n | h(ID_{MU_i}\|ID_{FA}\|D_3^j\|D_7^j) = D_{10}^j\}$. So, the protocol has not these two security properties.

5.3 Home agent impersonation attack

In offline guess attack, we showed that this attack is practical and adversary can find ID_{MU}. Let us see that what happens if *FA* plays the role of adversary and computes ID_{MU}. If *FA* knows ID_{MU}, it can easily impersonate *HA*. Let *MU* sends M_1 to *FA*. *FA* computes $D'_1 = e(D_2, K_{FA})$, generates a random number $c_i \in Z_p^*$, calculates $D_7 = c_i \times P$, $D_{10} = h(ID_{MU}\|ID_{FA}\|D_3\|D_7)$, $SK_{FM} = h(c_i \times D_3)$, $D_{12} = h(SK_{FM}\|D'_1\|D_3\|D_7\|D_{10})$, and submits $M_4 = \{D_7, D_{10}, D_{12}\}$ to *MU*. After receiving M_4 from *FA*, *MU* computes $D'_{10} = h(ID_{MU}\|ID_{FA}\|D_3\|D_7)$, $SK_{MF} = h(b_i \times D_7)$, $D'_{12} = h(SK_{MF}\|D_1\|D_3\|D_7\|D'_{10})$, and verifies $D'_{10}? = D_{10}$ and

$D'_{12}? = D_{12}$. In this state, $D'_{10} = D_{10}$ and $D'_{12} = D_{12}$. Therefore, *FA* can deceive *MU* and communicate with it without communicating with *HA*. Therefore, the attack is quite practical.

5.4 Foreign agent impersonation attack

In authentication phase, *FA* sends D_3, D_5, D_6, D_7, D_9, and ID_{FA} to the home agent. Note that D_3, D_5, D_6, ID_{HA}, and ID_{FA} are values the adversary possesses because D_3, D_5, ID_{HA}, and D_6 are sent by the user through an unsecured channel to *FA* and ID_{FA} is a public value that is available to everyone. For impersonating *HA*, it is enough for the adversary to calculate valid values D_7, D_8, and D_9. On the other hand, we have: $D_7 = c_i \times P, D_8 = c_i \times P_{pub_{HA}}$ and $D_9 = h(ID_{FA} \| ID_{HA} \| D_3 \| D_5 \| D_6 \| D_8)$.

In addition, P and $P_{pub_{HA}}$ are also public values, so the adversary can easily generate an arbitrary random number c_i and after calculating the valid message $M_2 = \{D_3, D_5, D_6, D_7, D_9, ID_{FA}\}$, sends this message to the home agent. In this case, everything looks right from the point of view of the home agent, and *HA* believes it is dealing with *FA*. As a result, the adversary can easily impersonate *FA*.

The reason for this attack is that there is no secret parameter shared only between the home and foreign agents in D_9. To resist this attack, we must include a secret value in D_9 so that calculating a valid D_9 is only possible for *FA* and *HA*. This will be discussed in detail in Sect. 7.

5.5 Insecure key distribution

Li et al. introduced two cases in the registration phase: one for foreign agent registration and the other for user registration. In the first case, the foreign agent selects an ID_{FA} and sends it to *HA*. In this case, *HA* sends $K_{FA} = \frac{P}{H(ID_{FA})+s}$ to the *FA* via a secure channel. This key is later used as a secret parameter. Using this parameter, the user realizes that the *FA* is valid.

Now suppose that n is the number of registered foreign agents in *HA*, where the i-th foreign agent FA^i has selected ID_{FA^i}. Note that ID_{FA^i} is a clear value and is available to everyone, where $i \leq n$. On the other hand, *HA* generates the secret key of FA^i as $K_{FA^i} = \frac{P}{H(ID_{FA^i})+s}$. In this case, any malicious *FA* can easily calculate all the secret key of other *FA*'s. To do this, suppose that FA^i wants to find the secret key of FA^j, where $i \neq j$.

We know that: $K_{FA^j} = \frac{P}{H(ID_{FA^j})+s} = H(ID_{FA^j})^{-1} \times P + s^{-1} \times P$. Since ID_{FA^j} is not a secret value, so to calculate K_{FA^j} one has to find $s^{-1} \times P$. On the other hand, since FA^i is already registered and has K_{FA^i}, it can easily calculates this value as follows:

$K_{FA^i} = \frac{P}{H(ID_{FA^i})+s} = H(ID_{FA^i})^{-1} \times P + s^{-1} \times P \Longrightarrow s^{-1} \times P = K_{FA^i} - H(ID_{FA^i})^{-1} \times P$.

Therefore, $K_{FA^j} = (H(ID_{FA^j})^{-1} - H(ID_{FA^i})^{-1}) \times P + K_{FA^i}$ and this means that all foreign agents can find each other's secret keys.

But this is not the end of the matter, and these attacks will also undermine the mutual authentication feature, which we will discuss in more detail below.

5.6 Malicious foreign agents

Suppose the mobile user *MU* wants to communicate with *FA* with the help of *HA*. To do this, *MU* sends the message $M_1 = \{ID_{HA}, D_1, D_2, D_3, D_4, D_5, D_6\}$ to *FA* and waits for receive $M_4 = \{D_7, D_{10}, D_{12}\}$. In this case, after receiving M_3 from *HA*, *FA* discovers *MU* is a valid user and similarly, *MU* discovers the validity of *FA* by receiving M_4 and checking D_{10} and D_{12}. In addition, the home agent can also analyze user behavior. Now we show that not only *FA*'s can communicate with users without interacting with the home agent, but they can also impersonate each other.

As stated, at the beginning of the authentication phase, *MU* sends the message $M_1 = \{ID_{HA}, D_1, D_2, D_3, D_4, D_5, D_6\}$ to the *FA* and waits for $M_4 = \{D_7, D_{10}, D_{12}\}$. After receiving M_4, *MU* first checks $D'_{10} = h(ID_{MU} \| ID_{FA} \| D_3 \| D_7)$. In this case, by checking D_{10}, *MU* finds out that *FA* has communicated with *HA* because there is ID_{MU} in D_{10} and *HA* is the only one who knows ID_{MU}.

We have shown in previous attacks that the offline guessing attack against this protocol is quite practical and everyone can calculate this value. Since $D'_{10} = h(ID_{MU} \| ID_{FA} \| D_3 \| D_7)$ and ID_{FA}, D_3, and D_7 are public values, anyone who executes the offline guess attack can generate a valid D_{10}. Consequently, it is sufficient to calculate a valid D_{12} to communicate with *MU*.

On the other hand, $D_{12} = h(h(c_i \times D_3) \| D_1 \| D_3 \| D_7 \| D_{10})$ and the only secret parameter for generating D_{12} is D_1. To find D_1, K_{FA} is needed, which is why *MU* believes that only *FA* can do this. As a result, *FA* can generate a valid D_{10} without communication with *HA* and communicate with the user. Note that K_{FA} is calculated by *HA* and delivered to *FA*. In the previous attack, we showed that all foreign agents were able to calculate each other's secret keys. As a result, a malicious *FA* can easily impersonates other foreign agents and communicate with authentic users without sending a message to the home agent.

5.7 Privileged insider attack

In the Li et al.'s Protocol, during the user registration phase, the mobile user sends a request $\{ID_{MU}, HPW_{MU}, R_{MU}\}$ to *HA*. Suppose the adversary has this message. ID_{MU} is clearly visible in this message. As stated in previous attacks, the possession of ID_{MU} can pose a number of risks, such as user anonymity violations and the success of impersonation attacks. Therefore, the protocol is vulnerable to the privileged insider attack.

6 Our proposed protocol

It was shown in the previous section that Li et al.'s protocol is vulnerable to several attacks, and the results were investigated. This section shows how to build a secure protocol with the least possible changes and without increasing the time

complexity. The password change phase and session key update of our protocol are the same as in [22]. We introduce registration phase and authentication and key agreement phase of our protocol bellow.

6.1 Registration phase of our protocol

This phase includes two cases: foreign agent registration and mobile user registration. In the first case, FA picks an identity ID_{FA} and sends it to HA, and HA computes $K_{FA} = H(ID_{FA}\|s)$ and sends it to FA secretly. Later, FA sets $P_{pub_{FA}} = H(ID_{FA}\|s) \times P$ as its public key. Similar to the previous protocol, we assumed that H is a secure hash function which maps the inputs to an element of the group Z_p^* such that recovering $H(ID_{FA}\|s)$ from $H(ID_{FA}\|s) \times P$ is intractable. Similar to the previous protocol, in the proposed protocol we used two hash functions H and h. The difference between these functions is that the output of the first one is a string where its length is less than p, while the second function length can be any arbitrary number.

In the second case, MU chooses an identity ID_{MU}, password PW_{MU}, random numbers x, y, z, and r, computes $HPW_{MU} = h(PW_{MU}\|r)$, and sends $\{X = x \times P, Y = y \times P, Z = z \times P, ID_{MU} \oplus x \times P_{pub_{HA}}, HPW_{MU} \oplus y \times P_{pub_{HA}}, R_{MU} \oplus z \times P_{pub_{HA}}\}$ to HA via a secure channel, where R_{MU} is fingerprint information that is extracted by the fuzzy extractor. Then, HA recovers $\{ID_{MU}, HPW_{MU}, R_{MU}\}$ using $\{X = x \times P, Y = y \times P, Z = z \times P, s\}$ and computes $B_1 = h(ID_{MU}\|HPW_{MU}\|R_{MU})$, $B_2 = h(ID_{MU}\|s)$, $B_3 = h(ID_{MU}\|R_{MU}) \oplus B_2$, and sends $\{B_1, B_3\}$ to MU secretly, where s is HA's secret key.

At last, MU stores $\{P, h(.), H(.), Gen, Rep, P_{MU}, B_1, B_3, r\}$ to the mobile device. This phase is depicted in Fig. 4.

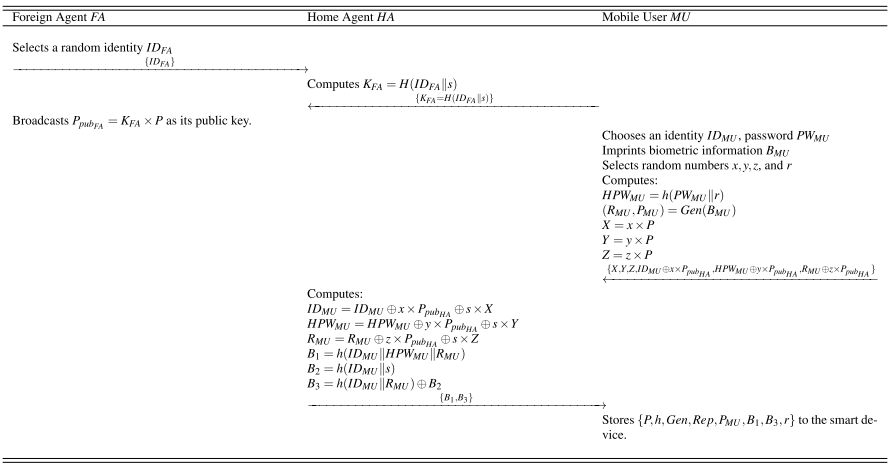

Fig. 4 Registration phase of the proposed protocol

6.2 Authentication and key agreement phase of our protocol

At the first of this phase, MU inputs ID_{MU} and PW_{MU}, imprints the fingerprint information B'_{MU}, and the smart device computes $R'_{MU} = Rep(B'_{MU}, P_{MU})$ and $B'_1 = h(ID_{MU}\|h(PW_{MU}\|r)\|R'_{MU})$. Later, the device verifies B'_1; if $B'_1 = B_1$, the mobile device generates two random numbers $a_i, b_i \in Z_p^*$, computes $B_2 = B_3 \oplus h(ID_{MU}\|R_{MU})$, $D_1 = a_i \times P_{pub_{FA}}$, $D_2 = a_i \times P$, $D_3 = b_i \times P$, $D_4 = h(ID_{FA}\|ID_{HA}\|D_1\|D_3)$, $D_5 = ID_{MU} \oplus h(b_i \times P_{pub_{HA}})$, $D_6 = h(B_2\|ID_{FA}\|ID_{HA}\|D_3)$, and submits $M_1 = \{ID_{HA}, D_2, D_3, D_4, D_5, D_6\}$ to FA by a public channel. Otherwise, MU is not valid.

When receiving this message, FA computes $D'_1 = K_{FA} \times D_2$ and $D'_4 = h(ID_{FA}\|ID_{HA}\|D'_1\|D_3)$. If D'_4 is equal to D_4, FA generates a random number $c_i \in Z_p^*$, computes $D_7 = c_i \times P$, $D_8 = c_i \times P_{pub_{HA}}$, $D_9 = h(ID_{FA}\|ID_{HA}\|D_3\|D_5\|D_6\|D_8\|K_{FA} \times D_8)$, and forwards $M_2 = \{D_3, D_5, D_6, D_7, D_9, ID_{FA}\}$ to HA by a public channel.

Upon getting message M_2, HA computes $ID'_{MU} = D_5 \oplus h(s \times D_3)$, $B'_2 = h(ID'_{MU}\|s)$, $D'_6 = h(B'_2\|ID_{FA}\|ID_{HA}\|D_3)$, and checks D'_6; if MU is valid, then D'_6 must be equal to D_6. After this, HA computes $D'_8 = s \times D_7$ and $D'_9 = h(ID_{FA}\|ID_{HA}\|D_3\|D_5\|D_6\|D'_8\|H(ID_{FA}\|s) \times D'_8)$. Similarly, If FA is valid, then D'_9 must be equal to D_9. HA checks this and computes $D_{10} = h(ID'_{MU}\|ID_{FA}\|D_3\|D_7\|s \times D_3)$ and $D_{11} = h(ID_{HA}\|ID_{FA}\|D_3\|D_5\|D'_8\|D_{10})$. Finally, HA submits $M_3 = \{D_{10}, D_{11}\}$ to FA.

After getting the message M_3 from HA, FA calculates $D'_{11} = h(ID_{HA}\|ID_{FA}\|D_3\|D_5\|D_8\|D_{10})$ and checks $D'_{11}? = D_{11}$. If so, HA is valid and FA computes $SK_{FM} = h(c_i \times D_3)$, $D_{12} = h(SK_{FM}\|D'_1\|D_3\|D_7\|D_{10})$, and forwards $M_4 = \{D_7, D_{10}, D_{12}\}$ to MU.

At last, MU computes $D'_{10} = h(ID_{MU}\|ID_{FA}\|D_3\|D_7\|b_i \times P_{pub_{HA}})$ and checks $D'_{10}? = D_{10}$. This equality means that FA is valid, and MU computes $SK_{MF} = h(b_i \times D_7)$ as a secret key for future session with HA. If the secret key is valid, $D'_{12} = h(SK_{MF}\|D_1\|D_3\|D_7\|D'_{10})$ must be equal to D_{12}; and MU checks this in the final line of the protocol.

In Fig. 5, we present the differences of our method with the previous protocol.

7 Analysis of proposed protocol

In this section, we first use the BAN logic. This logic is one of the most popular and well-known logic that has been used in many articles, as an example see [4, 11, 20, 28].

7.1 Formal analysis using BAN logic

Mutual authentication is one of the most critical features that authentication protocols should have. In this feature, the sender and receiver must be able to prove that they are what they claim. In order to formally prove the sender and receiver

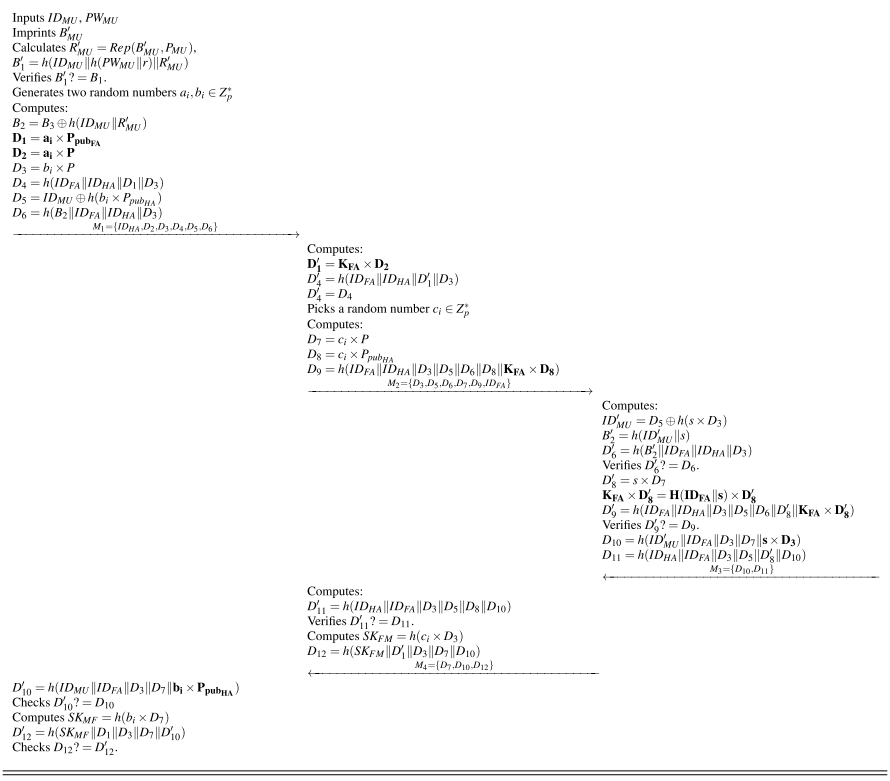

Fig. 5 Authentication and key agreement phase of the proposed protocol

Table 3 Notations of BAN logic

Notation	Description
$N_1 : \alpha \mid\equiv \beta$	α believes β
$N_2 : \#(\alpha)$	α is fresh
$N_3 : \alpha \mid\sim \beta$	α once said the statement β
$N_4 : \alpha \lhd \beta$	α sees the statement β
$N_5 : \alpha \mid\Rightarrow \beta$	α has jurisdiction over the statement β
$N_6 : (\alpha, \beta)$	α or β is one part of the (α, β)
$N_7 : \alpha \overset{SK}{\longleftrightarrow} \beta$	α and β may use the shared key SK to communicate among each other
$N_8 : \langle\alpha\rangle_\beta$	α is combined with the β

claims, we utilize BAN logic, which consists of several notations and rules. We present the results of the BAN logic of our protocol in Tables 3 and 4. It is sufficient to map the desired protocol in combination with the symbols presented in

Table 4 Rules of BAN logic

Rule	Name
$R_1 : \frac{\alpha \mid\equiv \alpha \overset{SK}{\longleftrightarrow} \beta, \alpha \lhd \langle \gamma \rangle_{SK}}{\alpha \mid\equiv \beta \mid\sim \gamma}$	Message-meaning rule
$R_2 : \frac{\alpha \mid\equiv \#(\beta)}{\alpha \mid\equiv \#(\beta, \gamma)}$	Freshness-conjuncatenation rule
$R_3 : \frac{\alpha \mid\equiv \#(\beta), \alpha \mid\equiv \gamma \mid\sim \beta}{\alpha \mid\equiv \gamma \mid\equiv \beta}$	Nonce-verification rule
$R_4 : \frac{\alpha \mid\equiv \beta \mid\Rightarrow \gamma, \alpha \mid\equiv \beta \mid\equiv \gamma}{\alpha \mid\equiv \gamma}$	Jurisdiction rule

Table 3 and then derive the desired result using the rules presented in Table 4. In Sect. 2, we summarize other formal methods to analyze authentication protocols.

Theorem 1 *In our proposed protocol,* $MU \mid\equiv FA \overset{SK_{MF}}{\longleftrightarrow} MU$, *and* $FA \mid\equiv FA \overset{SK_{FM}}{\longleftrightarrow} MU$.

Proof According to Tables 3 and 4, and the authentication and key agreement phase (Fig. 5), the following statements can be obtained:

S_1: MU generates a_i, b_i and computes D_1, D_2, D_3, D_4, D_5 , and D_6, from R_2 we have $MU \mid\equiv \#(a_i, b_i, D_1, D_2, D_3, D_4, D_5, D_6)$.
S_2: MU sends M_1 to FA, so $FA \lhd M_1$.
S_3: FA generates c_i and computes D_7, D_8, and D_9, from R_2 we have $FA \mid\equiv \#(c_i, D_7, D_8, D_9)$.
S_4: FA submits M_2 to HA, so $HA \lhd M_2$.
S_5: HA verifies $D_6' ? = D_6$, from S_4 and R_1, we get $HA \mid\equiv MU \mid\sim D_6$.
S_6: HA sends M_3 to FA, so $FA \lhd M_3$.
S_7: FA checks $D_{11}' ? = D_{11}$, from S_6 and R_1, we have $FA \mid\equiv HA \mid\sim M_3$.
S_8: From S_5, S_7, and R_4, we have $FA \mid\equiv MU \mid\sim M_1$. Therefore, b_i is generated by MU and $FA \mid\equiv FA \overset{SK_{FM}}{\longleftrightarrow} MU$.

Now, we show $MU \mid\equiv FA \overset{SK_{MF}}{\longleftrightarrow} MU$.

S_9: FA sends M_4 to MU, so $MU \lhd M_4$.
S_{10}: MU verifies $D_{10}' ? = D_{10}$, from S_9 and R_1, we get $MU \mid\equiv HA \mid\sim M_3$.
S_{11}: From S_{10}, we get $MU \mid\equiv FA \mid\sim M_2$ and c_i is generated by FA.
S_{12}: MU verifies $D_{12}' ? = D_{12}$, from S_9, R_1, and S_{11}, we have $MU \mid\equiv FA \mid\sim M_4$. Consequently, $MU \mid\equiv FA \overset{SK_{MF}}{\longleftrightarrow} MU$, and this completes the proof.

□

7.2 Resistance to offline guess attack

In the previous section, it was shown that if an adversary guesses the ID_{MU}, not only the protocol is vulnerable to offline guess attack but also is vulnerable to anonymity,

unlinkability, and forgery attacks. In the following, resistance to the offline guess attack is proven.

Suppose an adversary guesses ID_{MU} and wants to check it out. ID_{MU} is only included in $D_5 = ID_{MU} \oplus h(b_i \times P_{pub_{HA}})$ and $D_{10} = h(ID_{MU} \| ID_{FA} \| D_3 \| D_7 \| b_i \times P_{pub_{HA}})$ $= h(ID_{MU} \| ID_{FA} \| D_3 \| D_7 \| s \times D_3)$. To be successful, the adversary must calculate $b_i \times P_{pub_{HA}}$ or $s \times D_3$ because ID_{FA}, D_3, and D_7 are public. To do this, the adversary has three options:

- Find b_i using $D_3 = b_i \times P$ and then calculate $b_i \times P_{pub_{HA}}$ which is impossible because finding b_i using $b_i \times P$ requires solving ECDLP and we assumed this would be impossible to solve.
- Find s and calculate $s \times D_3$. This is also impossible because s is the private key of HA and it is assumed that no one can calculate it except HA.
- Compute $b_i \times P_{pub_{HA}} = s \times D_3 = b_i \times s \times P$ using $D_3 = b_i \times P$ and $P_{pub_{HA}} = s \times P$ which is impossible because this requires solving ECDHP and we assumed this would be impossible to solve.

As a result, the calculation of ID_{MU} is only possible for HA. Note that knowing ID_{MU} by HA will not cause any problems and this is one of the goals of the protocol of Li et al., since home agents sometimes need to monitor user behavior [22].

7.3 Preserving anonymity and unlinkability

Note that Li et al. showed that their protocol has properties anonymity and unlinkability. In Sect. 5, it was shown that this claim is correct in the state that the protocol is safe against offline guess attack which is not analyzed in [22]. As a result, our proposed protocol preserves anonymity and unlinkability because the security of proposed protocol against offline guess attack is proved earlier.

7.4 Resistance to *HA* impersonation attack

Suppose the intention of an adversary is to impersonate HA. To do this, the adversary must send a valid D_{11} to FA, where $D_{11} = h(ID_{HA} \| ID_{FA} \| D_3 \| D_5 \| D_8 \| D_{10})$. All parameters required to compute a valid D_{11} are sent through a public channel except D_8. Therefore, calculating D_8 is required for impersonating HA. On the other hand, since $D_8 = c_i \times P_{pub_{HA}}$ and $P_{pub_{HA}}$ is a public value, the adversary can calculate D_8 by having c_i. The only possible way to calculate c_i is to use $D_7 = c_i \times P$, and this is ECDLP, which is known as a hard problem. Another way for calculate $D_8 = c_i \times s \times P$ is solving ECDHP using $D_7 = c_i \times P$ and $P_{pub_{HA}} = s \times P$ which is impossible too. As a result, the adversary cannot impersonate HA and our protocol is secure against this attack.

In addition, if a malicious FA wants to forge HA and communicate with MU without its permission, it must find a valid D_{10}. This requires ID_{MU} and $b_i \times P_{pub_{HA}}$. As we have shown, our protocol is safe against offline guess attack and it is impossible

to find ID_{MU}. Similar to the previous case, to calculate $b_i \times P_{pub_{HA}}$ we have to solve ECDLP or ECDHP, which is impossible.

7.5 Resistance to malicious *FA* attack

Proving resistance to malicious user attack is very similar to the previous one. As mentioned earlier, if *FA* is malicious and wants to communicate with *MU* without *HA* authorization, it must calculate a valid D_{10} and to do so ID_{MU} and $b_i \times P_{pub_{HA}}$ are required, which is impossible to find these values in practice. Therefore, even if the *FA* is malicious, there will be no threat to the protocol.

7.6 Resistance to *FA* impersonation attack

As mentioned earlier, in the protocol of Li et al., everyone is able to impersonate *FA* and the reason for this attack is to use the public parameters in D_9. In our proposed protocol, D_9 is set to $h(ID_{FA}\|ID_{HA}\|D_3\|D_5\|D_6\|D_8\|K_{FA} \times D_8)$. Given that it is possible to calculate K_{FA} only for *FA* and *HA*, no one can impersonate *FA*. Besides, computing $K_{FA} \times D_8 = K_{FA} \times c_i \times s \times P$ with the help of $\{P, P_{pub_{FA}} = K_{FA} \times P, P_{pub_{HA}} = s \times P, D_7 = c_i \times P\}$ is impossible because ECDLP and ECDHP are hard problems.

7.7 Resistance to privileged insider attack

In our proposed protocol, during the user registration phase the mobile user sends a request $\{x \times P, y \times P, z \times P, ID_{MU} \oplus x \times P_{pub_{HA}}, HPW_{MU} \oplus y \times P_{pub_{HA}}, R_{MU} \oplus z \times P_{pub_{HA}}\}$ to *HA*. In this case, granting the user request will not pose any threat to the protocol because to calculate $\{ID_{MU}, HPW_{MU}, R_{MU}\}$, $\{x \times P_{pub_{HA}}, y \times P_{pub_{HA}}, z \times P_{pub_{HA}}\}$ is required. Due to the ECDLP and ECDHP, calculating $\{ID_{MU}, HPW_{MU}, R_{MU}\}$ it is not possible for the adversary and the home agent is the only one who can calculate the parameters available in the user request. Therefore, our protocol is proof against the privileged insider attack.

7.8 Forward and backward security

In our protocol, the disclosure of the current secret key has no risk for the keys previously used and the keys that will be used in the future because the secret key is $h(b_i \times c_i \times P)$ and we assumed that the computation of x from $h(x)$ is impossible. Moreover, b_i and c_i are fresh values, generated randomly, and according to ECDLP and ECDHP, it is impossible for adversary to find.

7.9 Secure key distribution

In the registration phase, after registering the *FA*'s, *HA* calculates and delivers the secret keys. The structure of these keys should not be such that foreign

Table 5 Performance comparison of protocols

	[14]	[22]	Ours
MU	$T_{sym} + 5T_h$	$5T_{ecc} + T_p + 10T_h + T_f$	$5T_{ecc} + 10T_h + T_f$
FA	$2T_{sym} + 2T_h$	$3T_{ecc} + T_p + 5T_h$	$5T_{ecc} + 5T_h$
HA	$3T_{sym} + 6T_h$	$2T_{ecc} + 6T_h$	$3T_{ecc} + 7T_h$
Total	$6T_{sym} + 13T_h$	$10T_{ecc} + 2T_p + 21T_h + T_f$	$13T_{ecc} + 22T_h + T_f$

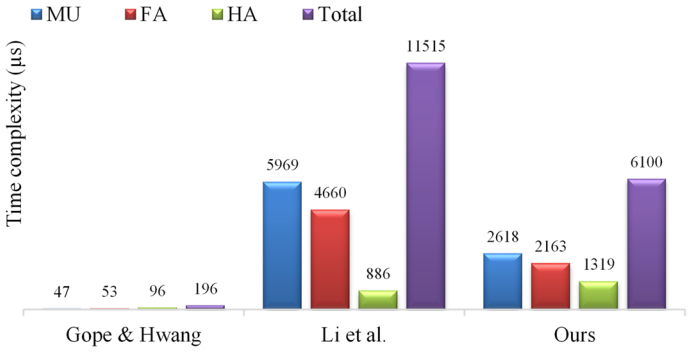

Fig. 6 Time complexity of protocols

agents can find each other's secret keys. We presented Li et al.'s protocol suffers from this vulnerability. In our protocol, the secret key of *FA* is $h(ID_{FA}\|s)$, which is delivered via a secure channel. Because s has been used to generate this key, s is only available to *HA*, and also ID_{FA}'s are distinct values so *FA*'s cannot calculate each other's secret keys.

7.10 Space complexity

Given that each of the identities involved in this protocol must process different length variables, this complexity equals the sum of the variables' length. As a result, the space complexity for mobile user is in order of $O(\sum_{i=1}^{20} l_{m_i})$, where l_{m_i} is the bit length of parameter m_i and

$m_i \in \{ID_{MU}, PW_{MU}, B_{MU}, P_{MU}, R_{MU}, B_1, B_2, B_3, D_1, \cdots, D_7, ID_{FA}, ID_{HA}, D_{10}, D_{12}, SK_{MF}\}$.

Similarly, this complexity for foreign and home agents is in the order of $O(\sum_{i=1}^{17} l_{f_i})$ and $O(\sum_{i=1}^{20} l_{h_i})$, respectively. In these cases, we have:

$f_i \in \{K_{FA}, ID_{FA}, ID_{HA}, D_1, \ldots, D_{12}, K_{FA} \times D_8, SK_{FM}\}$,

$h_i \in \{s, D_3, ID_{FA}, ID_{HA}, ID_{MU}, B_2, D_5, \ldots, D_{11}, K_{FA} \times D_8\}$.

Suppose the bit length of the largest parameter is n, therefore $\sum_{i=1}^{20} l_{m_i} \leq 20 \times n$, $\sum_{i=1}^{17} l_{f_i} \leq 17 \times n$, $\sum_{i=1}^{14} l_{h_i} \leq 14 \times n$, and the space complexity of the proposed protocol is in $O(51 \times n) = O(n)$.

Table 6 Communication overhead of protocols

	[14]	[22]	Ours
MU	$l_{sym} + 2l_h + l_{ID}$	$2l_{ecc} + 3l_h + l_{ID}$	$2l_{ecc} + 3l_h + l_{ID}$
FA	$2l_{sym} + 6l_h$	$3l_{ecc} + 5l_h + l_{ID}$	$3l_{ecc} + 5l_h + l_{ID}$
HA	$l_{sym} + 4l_h$	$2l_h$	$2l_h$
Total	$4l_{sym} + 12l_h + l_{ID}$	$5l_{ecc} + 10l_h + 2l_{ID}$	$5l_{ecc} + 10l_h + 2l_{ID}$

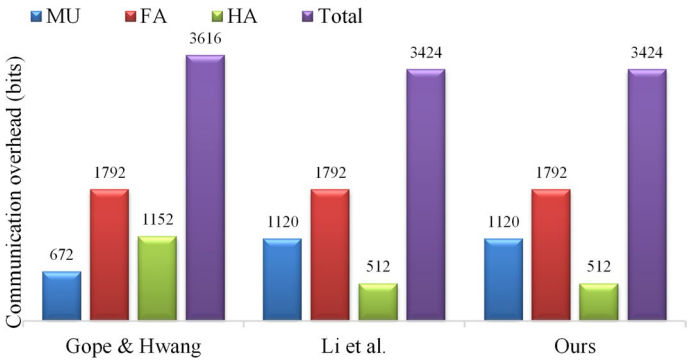

Fig. 7 Practical overhead of protocols

7.11 Performance comparisons

The Li et al.'s scheme was proposed to deal with the weaknesses of Gope and Hwang's protocol [14]. Therefore, we compare our proposed protocol with these two schemes. Table 5 compares the costs of these protocols, where $T_{sym}, T_{ecc}, T_h, T_f$, and T_p are the time complexity of symmetric encryption, elliptic curve cryptography, hash function, fuzzy extractor, and bilinear pairing, respectively. According to [23, 25, 37], $T_{sym} \approx 0.0214835ms, T_{ecc} \approx T_f \approx 0.427576$ ms, $T_h \approx 0.005174$ ms, and $T_p \approx 3.35173$ ms. Using these values, we can compute the numerical time complexity of theses protocols which is depicted in Fig. 6.

As Fig. 6 shows, Li et al.'s and our proposed protocol are in a much worse runtime state than the Gope and Hwang's, which is obvious because the Gope and Hwang's protocol is a symmetric cipher-based protocol, but Li et al. proposed the bilinear pairing to eliminate the vulnerabilities of the previous one.

Unfortunately, bilinear pairing is much slower than symmetric encryption, and according to the implementation presented in [37], symmetric encryption is 156 times faster than bilinear pairing. The use of bilinear pairing in the protocol of Li et al. makes its runtime reach 11.515 ms, which is 58 times slower than the previous protocol. This is also true for our protocol, which is based on elliptic curves.

In this paper, we suggested that these curves be used instead of using bilinear pairing that have a very high execution time. Although such curves have much more execution time than symmetric ciphers, their execution time is much shorter than the bilinear pairing, which reduces the time complexity of our proposed

protocol compared to the Li et al.'s. Figure 6 also confirms this, and it can be easily seen that the time complexity of our protocol is 53% of the time complexity of the previous protocol.

We also compared the communication overhead of our protocol with similar ones in Table 6, where l_{sym}, l_{ID}, l_{ecc}, and l_h are the length of symmetric encryption, *ID*, elliptic curve point, and the hashed value, respectively. We assume that $l_{sym} = 128$ bits, $l_{ID} = 32$ bits, $l_{ecc} = 160$ bits, and $l_h = 256$ bits. These values are based on the results presented in [37]. For example, $T_h \approx 0.005174$ ms was previously considered. This runtime is about implementing a 256-bit hash function so we assumed $l_h = 256$. In Fig. 7, we present the numerical communication overhead of similar protocols. In this figure, four values are provided for each protocol, three of them indicating a communication overhead imposed on each of the parties involved in the protocol. Besides, we report the total communication overhead caused by the implementation of these protocols. In Gope and Hwang's protocol, the lowest overhead is imposed on mobile users, while the foreign agent tolerates the highest communication overhead. Moreover, the home agent tolerates more overhead than mobile users, while in the other two protocols, the home agent imposes much less overhead. Finally, the overhead imposed on a foreign agent of all protocols is 1792 bits.

As Fig. 7 shows, the communication overhead of Li et al.'s protocol is 192 bits less than that of Gope and Hwang's scheme and the overhead of our protocol is exactly the same as that of Li et al.'s.

We finish this section by comparing the security of these protocols which is summarized in Table 7. As a result, our proposed protocol, while removing the security flaws of Li et al.'s protocol, not only does not increase the communications overhead of the previous one but also reduces its time complexity.

Table 7 Total comparison of protocols

	Gope and Hwang [14]	Li et al. [22]	Ours
Security properties			
Resistance to offline guess attack	✓	×	✓
User anonymity	✓	×	✓
Unlinkability	✓	×	✓
Resistance to *HA* impersonation attack	✓	×	✓
Resistance to *FA* impersonation attack	✓	×	✓
Secure key distribution	✓	×	✓
Resistance to malicious *FA* attack	✓	×	✓
Resistance to privileged insider attack	✓	×	✓
Is secure?	No	No	Yes
Its vulnerabilities are analyzed in:	[22]	This paper	-
Runtime (μs)	196	11,515	6100
Communication overhead (bits)	3616	3424	3424

8 Conclusion and future directions

In this paper, we analyzed the issue of secure communications in global mobility network as one of the main infrastructures in smart cities. To do so, we analyzed the Li et al.'s Protocol, published in 2017. We showed that the protocol suffers from offline guess attack. We also revealed that this attack could lead to vulnerability to forgery attacks and is a serious threat to the user anonymity and unlinkability, which are of the essential features of authentication protocols in smart cities. We also proved that the secret key distribution mechanism of the home agent is not secure and foreign agents can compute the secret keys of each other. Moreover, we showed that a malicious foreign agent can be a serious threat to this protocol. We modified the protocol and formal analysis using BAN logic indicated that our suggestion preserves mutual authentication; also, the resistance to various well-known attacks is proved in the paper. We compared our protocol with two recently published ones, and results disclosed that our protocol is completely suitable for smart cities.

In the future, we aim to integrate our scheme with block-chain and distributed ledger using 5G/Fog technologies and optimize the runtime of the protocol by eliminating the effects of elliptic curves cryptography and using the lightweight functions such as symmetric cryptography which makes our protocol more suitable for resource-constrained devices.

References

1. Aghili SF, Mala H, Shojafar M, Peris-Lopez P (2019) Laco: lightweight three-factor authentication, access control and ownership transfer scheme for e-health systems in IoT. Future Gener Comput Syst 96:410–424
2. Amin R, Islam SH, Biswas G, Khan MK, Leng L, Kumar N (2016) Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks. Comput Netw 101:42–62
3. Chatterjee S, Roy S, Das AK, Chattopadhyay S, Kumar N, Vasilakos AV (2016) Secure biometric-based authentication scheme using chebyshev chaotic map for multi-server environment. IEEE Trans Dependable Secure Comput 15(5):824–839
4. Chen Y, Martínez JF, Castillejo P, López L (2018) A lightweight anonymous client-server authentication scheme for the internet of things scenario: Lauth. Sensors 18(11):3695
5. Cynthia J, Parveen Sultana H, Saroja MN, Senthil J (2019) Security protocols for IoT. In: Jeyanthi N, Abraham A, Mcheick H (eds) Ubiquitous computing and computing security of IoT. Studies in big data. vol 47. Springer, Cham. https://doi.org/10.1007/978-3-030-01566-4_1
6. Dameri RP (2013) Searching for smart city definition: a comprehensive proposal. Int J Comput Technol 11(5):2544–2551
7. Dameri RP (2017) Smart city definition, goals and performance. In: Smart city implementation. Progress in IS. Springer, Cham. https://doi.org/10.1007/978-3-319-45766-6_1
8. Das AK, Wazid M, Kumar N, Vasilakos AV, Rodrigues JJ (2018) Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial internet of things deployment. IEEE Internet Things J 5(6):4900–4913
9. Dodis Y, Reyzin L, Smith A (2004) Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. In: International Conference on the Theory and Applications of Cryptographic Techniques. Springer, pp 523–540

10. Dolev D, Yao A (1983) On the security of public key protocols. IEEE Trans Inf Theory 29(2):198–208
11. Dua A, Kumar N, Das AK, Susilo W (2017) Secure message communication protocol among vehicles in smart city. IEEE Trans Veh Technol 67(5):4359–4373
12. Farash MS, Turkanović M, Kumari S, Hölbl M (2016) An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the internet of things environment. Ad Hoc Netw 36:152–176
13. Gope P, Das AK (2017) Robust anonymous mutual authentication scheme for n-times ubiquitous mobile cloud computing services. IEEE Internet Things J 4(5):1764–1772
14. Gope P, Hwang T (2016) An efficient mutual authentication and key agreement scheme preserving strong anonymity of the mobile user in global mobility networks. J Netw Comput Appl 62:1–8
15. Gunasinghe H, Bertino E (2017) Privbiomtauth: privacy preserving biometrics-based and user centric protocol for user authentication from mobile phones. IEEE Trans Inf Forensics Secur 13(4):1042–1057
16. He D, Kumar N, Chilamkurti N (2015) A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks. Inf Sci 321:263–277
17. He D, Ma M, Zhang Y, Chen C, Bu J (2011) A strong user authentication scheme with smart cards for wireless communications. Comput Commun 34(3):367–374
18. Jannati H, Bahrak B (2017) An improved authentication protocol for distributed mobile cloud computing services. Int J Crit Infrastruct Prot 19:59–67
19. Jiang Q, Ma J, Wei F, Tian Y, Shen J, Yang Y (2016) An untraceable temporal-credential-based two-factor authentication scheme using ecc for wireless sensor networks. J Netw Comput Appl 76:37–48
20. Jung J, Kim J, Choi Y, Won D (2016) An anonymous user authentication and key agreement scheme based on a symmetric cryptosystem in wireless sensor networks. Sensors 16(8):1299
21. Kim Th, Ramos C, Mohammed S (2017) Smart city and IoT. Future Gener Comput Syst 76:159–162
22. Li X, Niu J, Kumari S, Wu F, Choo KKR (2018) A robust biometrics based three-factor authentication scheme for global mobility networks in smart city. Future Gener Comput Syst 83:607–618
23. Li X, Niu J, Kumari S, Wu F, Sangaiah AK, Choo KKR (2018) A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments. J Netw Comput Appl 103:194–204
24. Li X, Peng J, Niu J, Wu F, Liao J, Choo KKR (2017) A robust and energy efficient authentication protocol for industrial internet of things. IEEE Internet Things J 5(3):1606–1615
25. Li X, Wu F, Kumari S, Xu L, Sangaiah AK, Choo KKR (2019) A provably secure and anonymous message authentication scheme for smart grids. J Parallel Distrib Comput 132:242–249. https://doi.org/10.1016/j.jpdc.2017.11.008
26. Menezes A (2009) An introduction to pairing-based cryptography. Recent Trends Cryptogr 477:47–65
27. Neshenko N, Bou-Harb E, Crichigno J, Kaddoum G, Ghani N (2019) Demystifying IoT security: an exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations. IEEE Commun Surve Tutor 21(3):2702–2733. https://doi.org/10.1109/COMST.2019.2910750
28. Park K, Park Y, Park Y, Das AK (2018) 2pakep: Provably secure and efficient two-party authenticated key exchange protocol for mobile environment. IEEE Access 6:30225–30241
29. Roy S, Chatterjee S, Das AK, Chattopadhyay S, Kumari S, Jo M (2017) Chaotic map-based anonymous user authentication scheme with user biometrics and fuzzy extractor for crowdsourcing internet of things. IEEE Internet Things J 5(4):2884–2895
30. Saeed MES, Liu QY, Tian G, Gao B, Li F (2018) Remote authentication schemes for wireless body area networks based on the internet of things. IEEE Internet Things J 5(6):4926–4944
31. Shen J, Zhou T, Wei F, Sun X, Xiang Y (2017) Privacy-preserving and lightweight key agreement protocol for v2g in the social internet of things. IEEE Internet Things J 5(4):2526–2536
32. Song T, Li R, Mei B, Yu J, Xing X, Cheng X (2017) A privacy preserving communication protocol for iot applications in smart homes. IEEE Internet Things J 4(6):1844–1852
33. Tsai JL, Lo NW (2015) A privacy-aware authentication scheme for distributed mobile cloud computing services. IEEE Syst J 9(3):805–815
34. Washington LC (2008) Elliptic curves: number theory and cryptography. Chapman and Hall/CRC, New York
35. Wazid M, Das AK, Odelu V, Kumar N, Conti M, Jo M (2017) Design of secure user authenticated key management protocol for generic IoT networks. IEEE Internet Things J 5(1):269–282

36. Wu CC, Lee WB, Tsaur WJ (2008) A secure authentication scheme with anonymity for wireless communications. IEEE Commun Lett 12(10):722–723
37. Wu F, Xu L, Kumari S, Li X, Das AK, Khan MK, Karuppiah M, Baliyan R (2016) A novel and provably secure authentication and key agreement scheme with user anonymity for global mobility networks. Secur Commun Netw 9(16):3527–3542
38. Xie Q, Wong DS, Wang G, Tan X, Chen K, Fang L (2017) Provably secure dynamic ID-based anonymous two-factor authenticated key exchange protocol with extended security model. IEEE Trans Inf Forensics Secur 12(6):1382–1392
39. Xu G, Liu J, Lu Y, Zeng X, Zhang Y, Li X (2018) A novel efficient MAKA protocol with desynchronization for anonymous roaming service in global mobility networks. J Netw Comput Appl 107:83–92