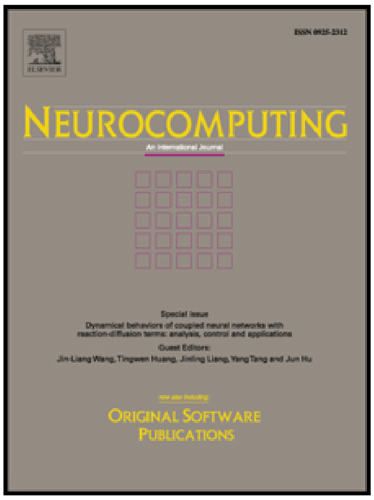
Accepted Manuscript



Security enhancement of an anonymous roaming authentication scheme with two-factor security in smart city

Qi Xie , Lingfeng Hwang

PII: S0925-2312(19)30338-8
DOI: https://doi.org/10.1016/j.neucom.2019.03.020
Reference: NEUCOM 20554

To appear in: *Neurocomputing*

Received date: 19 September 2018
Revised date: 2 February 2019
Accepted date: 19 March 2019

Please cite this article as: Qi Xie , Lingfeng Hwang , Security enhancement of an anonymous roaming authentication scheme with two-factor security in smart city, *Neurocomputing* (2019), doi: https://doi.org/10.1016/j.neucom.2019.03.020

Security enhancement of an anonymous roaming authentication scheme with two-factor security in smart city

Qi Xie*, Lingfeng Hwang

Key Laboratory of Cryptography and Network Security, Hangzhou Normal University, Hangzhou, China

*Corresponding author

E-mail addresses: qixie68@126.com (Q. Xie), lingfhuang888@163.com (L. Hwang).

Abstract: Due to the popularization and application of mobile phones and the Fifth Generation (5G) communication technology in the smart city, it's easy for people to use the internet service at anytime and anywhere. While providing convenience, mobile networks face a series of challenges in security and privacy protection due to the ability of the terminal. Recently, Xiong et al. designed an anonymous authentication scheme based on elliptic curve cryptography (ECC) for roaming in smart city. However, we show that their scheme lacks of two-factor security, and suffers from impersonation attack. To fix these problems, an improved roaming authentication protocol with two-factor security is proposed, which is security by using applied pi calculus based formal validation tool ProVerif, and it has high computational efficiency by comparison with some related schemes.

Keywords: Smart city; Anonymity; Authentication; Two-factor security; ProVerif

1. Introduction

Smart city plays an important role in managing assets and resources efficiently, city operations and services and connect to citizens, etc. Since the birth of wireless communication technology, it has brought greatly convenience to people's life. From First Generation (1G) to Forth Generation (4G) communication technology, mobile communication technology refreshes the limits of data transmission and storage capacity constantly. The Fifth Generation (5G) communication technology [1-4] is a current research hotspot that can satisfy more demands of devices in the next few years. According to [1], it is expect that 5G technology may meet some requirements such as high network capacity and data rates, lower computational and transmission cost, acceptable cost of infrastructures, lower latency, high security, seamless roaming, intelligence, etc. 5G technology has great impetus to smart city. Global mobile network is the basic network of 5G communication technology that provides roaming service for mobile users.

A complete mobile roaming network is modeled as three participants, mobile user (MU), foreign agent (FA) and home agent (HA). To ensure MU's information can be transmitted safely in the wireless network environment, MU and FA must be authenticated each other with the assistance of HA, and establish a session key, which encrypts or decrypts messages in public channels. In recent years, a large number of two-factor (password and smart card) authentication protocols for roaming are proposed. But most of them are proved to be suffered from various attacks (including password guessing attacks, impersonate attacks, replay attacks, verifier-table stolen attacks, and DoS attack, etc), or lack of some security characters (including user anonymity, perfect forward secrecy, fair key agreement, and session key security, etc).

1.1 Related work

In 2004, Zhu et al. [5] proposed a first anonymous authentication scheme for roaming based on one-time symmetric key, smart card and hash function. But Lee et al. [6] demonstrated that Zhu et al.'s scheme exists some security flaws, such as mutual authentication and resist the forgery attack, and then they proposed a new one to remedy these flaws. After that, Wu et al. [7] and Chang et al. [8] independently demonstrated that Lee et al.'s scheme does not achieve user anonymity, and proposed minor improvement of Lee et al.'s scheme. Later some researchers [9-12] independently demonstrated that Wu et al. and Chang et al.'s schemes do also not achieve user anonymity. In 2012, Mun et al. [13] designed a new one to remedy the flaw of [7], which is pointed out to be unable to resist man in the middle attack, offline password guessing attack and lacks of perfect forward secrecy later [14-15]. Karuppiah et al. [16] also demonstrated that Kang et al.'s [17] improvement of [7] does not achieve perfect forward secrecy and user anonymity, and proposed an improvement one. In 2017, Xiong et al. [18] demonstrated that Karuppiah et al.'s scheme [16] lacks of perfect forward secrecy and session key update, suffers from the session key security and faces clock synchronization problem.

In 2011, He et al. [19] designed a new lightweight authentication scheme for roaming. Unfortunately, their scheme can't resist impersonate attack and replay attack, and lacks of user anonymity. Later on, two lightweight anonymous authentication schemes [20-21] are introduced, Xie et al. [22, 23] demonstrated that these schemes can't achieve user anonymity and presented a two-factor roaming authentication scheme. However, He et al. [24] proved that the scheme of [22] is exposed to camouflage server and user attack, and then fixed these flaws.

Based on quadratic residue assumption, He et al. [25] presented a new anonymous roaming authentication scheme. Unfortunately, Jiang et al. [26] declared that their scheme can't resist password guessing attack, and then presented an anonymous user authentication scheme for roaming. Wen et al. [27] pointed out that the scheme of [26] can't resist replay attack and verifer-table stolen attack, and presented an improved one to fix it. Gope et al. [28]

found that the scheme of [27] is also insecure, which can't resist impersonate attack, replay attack, session key disclosure, and does not provide perfect forward and backward secrecy. Farash et al. [29] also demonstrated that Wen et al.'s scheme [27] suffers from session key disclosure attack and known session key attack, and pointed out that Shin et al.'s scheme [30] suffers from impersonation attacks and session key disclosure attack, and does not achieve user's untraceability, so they proposed a new scheme to fix these problems. Unfortunately, Chaudhry et al. [31] showed that Farash et al.'s scheme is vulnerable to session key disclosure attack and impersonation attack, and does not provide mobile user anonymity. In 2017, Xie et al. [32] designed a first roaming authentication scheme based on Chaotic Maps in wireless network.

1.2 Our contribution

In the paper, we declared that the scheme of [18] is insecure, which suffers from impersonation attack, and lacks of two-factor security. Besides, we found an error of their scheme, which may be unworkable. To fix these flaws, we design a security enhancement scheme for roaming in smart city.

The reminder of this article is presented as follows. In Section 2, we briefly review the scheme of [18]. The cryptanalysis of [18] and our scheme are given in Sections 3 and 4. Next, the formal proof and security analysis of our scheme are presented in Sections 5 and 6. Section 7 is the comparisons of efficiency and security. Finally is the paper's conclusion.

2. Review of Xiong et al.'s scheme

Xiong et al's scheme consists of five phases: initialization, registration, authentication and key agreement, session key update and password update, here we only give the first three phases. Some notations are defined in Table 1.

Table 1 Notations

Notations	Description
HA,FA, MU,	Home agent, Foreign agent, Mobile user,
T_H, T_E, T_H	Identities of MU , FA , HA
T_E	MU's password
T_H	The shared key between FA and HA in advance
T_M, T_{SE}	Symmetric encryption and decryption functions with the key *k*
T_{EXP}	A secure one-way hash function
T_H	An additive group defined over a finite field T_{SE}
T_{EXP}	A generator on T_H with large order T_M

T_H, T_M	The private key and public key of HA

2.1 Initialization

HA generates the public parameters { E, p, G, $h()$, $E_k()$, $D_k()$ }, selects a random number $x_{HA} \in Z_p^*$ as a private key and calculates the public key $X = x_{HA}P$. Then HA saves x_{HA} and publishes the public parameters. In addition, HA and FA have already established a shared secret key K_{FH} in advance by other key agreement protocol.

2.2 Registration

MU can register to HA by the following steps.

Step 1: MU selects his identity ID_{MU} and password PW_{MU}, then generates a random nonce b and computes $RPW_{MU} = h(PW_{MU} \| b)$, $A_{MU} = h(ID_{MU} \| PW_{MU} \| b)$. Later MU submits the registration message { ID_{MU}, RPW_{MU}, A_{MU} } to HA securely.

Step 2: HA chooses a random nonce r_{MU} and calculates $B_{MU} = h(ID_{MU} \| x_{HA})$, $C_{MU} = B_{MU} \oplus RPW_{MU} \oplus r_{MU}$, $D_{MU} = B_{MU} \oplus A_{MU}$ and $E_{MU} = h(ID_{MU} \| RPW_{MU} \| r_{MU})$. Then HA stores { C_{MU}, E_{MU}, ID_{HA}, X } into a smart card (SC), and submits it with D_{MU} to MU via a secret channel, where X is public key of HA.

Step 3: MU calculates $F_{MU} = D_{MU} \oplus b$, $G_{MU} = h(ID_{MU} \| PW_{MU}) \oplus b$, and stores { F_{MU}, G_{MU} } into his SC. That is, the SC includes { C_{MU}, E_{MU}, ID_{HA}, F_{MU}, G_{MU}, X }.

2.3 Authentication and key agreement

If MU wants to get roaming service from FA, MU and FA must be authenticated each other and established the session key with the assistance of HA by the following steps.

Step 1: MU inputs his identity ID_{MU} and password PW_{MU}. The mobile device computes $b' = G_{MU} \oplus h(ID_{MU} \| PW_{MU})$, $A_{MU}' = h(ID_{MU} \| PW_{MU} \| b)$, $B_{MU}' = F_{MU} \oplus A_{MU}' \oplus b$, $RPW_{MU}' = h(PW_{MU} \| b')$, $r_{MU}' = C_{MU} \oplus B_{MU}' \oplus RPW_{MU}'$, and checks if $E_{MU} =? h(ID_{MU} \| RPW_{MU}' \| r_{MU}')$ is correct. If the equation is not correct, reject it. Otherwise, MU generates a random number $\alpha \in Z_p^*$ and calculates

$E_1 = \alpha P$,

$E_2 = \alpha X$,

$E_3 = ID_{MU} \oplus E_2$,

$E_4 = h(ID_{MU} \| B_{MU}' \| E_1 \| ID_{FA} \| ID_{HA})$,

Then MU submits login message M_1={ ID_{HA}, E_1, E_3, E_4 } to FA.

Step 2: FA generates a random nonce β, and computes $E_5 = \beta P$, $E_6 = h(K_{FH} \| ID_{FA} \| E_1 \| E_3 \| E_4 \| E_5)$, Then FA sends M_2={ $ID_{FA}, E_1, E_3, E_4, E_5, E_6$ } to HA.

Step 3: HA computes $E_2' = x_{HA} E_1 = x_{HA} \alpha P$, $ID_{MU}' = E_3 \oplus E_2'$, $B_{MU}'' = h(ID_{MU}' \| x_{HA})$ and checks if $E_4 = ?h(ID_{MU}' \| B_{MU}'' \| E_1 \| ID_{FA} \| ID_{HA} \|)$. If the equation is correct, HA checks if $E_6 = ?h(K_{FH} \| ID_{FA} \| E_1 \| E_3 \| E_4 \| E_5)$. If yes, HA authenticates FA and continues to calculate $E_7 = h(ID_{FA} \| K_{FH} \| ID_{HA} \| E_1 \| E_3)$, $E_8 = h(ID_{MU}' \| B_{MU}'' \| E_2' \| E_3)$. Then HA returns the message M_3={ E_7, E_8 } to FA.

Step 4: FA compares whether $E_7 = ?h(ID_{FA} \| K_{FH} \| ID_{HA} \| E_1 \| E_3)$ is true. If it is true, FA believes that massages from HA and MU are effective. Later FA calculates $SK_{FM} = h(\beta E_1 \| E_1 \| E_5)$, $E_9 = h(SK_{FM} \| E_8 \| E_3)$, and sends M_4={ E_5, E_8, E_9 } to MU.

Step 5: MU first verifies $E_8 = ?h(ID_{MU} \| B_{MU}' \| E_2 \| E_3)$. If it's true, MU calculates the session key $SK_{MF} = ?h(\alpha E_5 \| E_1 \| E_5)$, and verifies if $E_9 = ?h(SK_{MF} \| E_8 \| E_3)$ to authenticate FA. If so, MU and HA share the same secret key $SK_{MF} = h(\alpha E_5 \| E_1 \| E_5)$.

3. Cryptanalysis of Xiong et al.'s scheme

The next analysis to show that Xiong et al.'s scheme is insecure.

3.1 Suffer from impersonate attack

In Xiong et al.'s scheme, when HA sends a response message M_3={ E_7, E_8 } to FA, the adversary intercepts it, and generates a random nonce $g \in Z_p^*$, calculates $E_5^* = gP$, $SK^* = h(gE_1 \| E_1 \| E_5^*)$, $E_9^* = h(SK^* \| E_8 \| E_3)$. After that, he sends forgery message M_4^*={ E_5^*, E_8, E_9^* } to MU. When MU receives message M_4^*, he can verify the correction of $R_8 = h(ID_{MU}' \| B_{MU}' \| E_2' \| E_3)$, and compute the session key

$SK^* = h(\alpha E_5^* \| E_1 \| E_5^*)$ and can also verify the correction of $E_9^* = h(SK^* \| E_8 \| E_3)$. That is, the adversary and MU can share a session key $SK^* = h(\alpha E_5^* \| E_1 \| E_5^*)$.

The reason why Xiong et al.'s scheme suffers from impersonate attack is that MU does not verify whether $E_5 = \beta P$ is generated by the FA or not, which is an important parameter for generating the session key.

3.2 Lack of two-factor security

Two-factor security means that the scheme is secure, if either all data stored in SC including user's identity or user's password is compromised [33]. Why Xiong et al.'s scheme lacks of Two-factor security is that they designed the protocol without considering the two-factor security model, the details are as follows.

Assume that an adversary can extract all data { C_{MU}, E_{MU}, F_{MU}, G_{MU}, ID_{HA}, X } stored in SC and the mobile user's identity ID_{MU}, then he can launch offline password guessing attack. Because the adversary can know M_1={ ID_{HA}, E_1, E_3, E_4 }from public channel, he can select password PW_{MU}^* and calculate $b^* = G_{MU} \oplus h(ID_{MU} \| PW_{MU}^*)$, $A_{MU}^* = h(ID_{MU} \| PW_{MU}^* \| b^*)$, $B_{MU}^* = F_{MU} \oplus A_{MU}^* \oplus b^*$, $RPW_{MU}^* = h(PW_{MU}^* \| b^*)$, $r_{MU}^* = C_{MU} \oplus B_{MU}^* \oplus RPW_{MU}^*$, then the adversary can verify if $E_{MU} = ?h(ID_{MU} \| RPW_{MU}^* \| r_{MU}^*)$ and $E_4 = ?\,h(ID_{MU} \| B_{MU}^* \| E_1 \| ID_{FA} \| ID_{HA})$ are correct? If so, PW_{MU}^* is the correct password, and the adversary can know $B_{MU} = h(ID_{MU} \| x_{HA})$. Thus, the adversary can impersonate the user MU to login onto the FA and obtain the services. Otherwise the adversary selects another password PW_{MU}^* and continues to execute above process.

3.3 An error

In authentication and key agreement of Xiong et al.'s scheme, MU computes $E_2 = \alpha X$ and $E_3 = ID_{MU} \oplus E_2$, where X is public key of HA and is a point of elliptic curve, and ID_{MU} is an integer number. Therefore, it may unworkable. To fix this problem, we can correct it as $E_3 = ID_{MU} \oplus h(E_2)$.

4. The improved scheme

In this section, we present an improved scheme to fix the flaws of Xiong et al.'s scheme.

4.1 Initialization

This phase is the same as that of Xiong et al.'s scheme.

4.2 Registration

MU can register to HA by the following steps.

Step 1: MU selects ID_{MU} and PW_{MU}, and a random nonce y, calculates $C_1 = h(ID_{MU} \| PW_{MU} \| y)$ and sends { ID_{MU}, C_1 } to HA.

Step 2: HA computes $C_2 = C_1 \oplus h(ID_{MU} \| x_{HA})$, where x_{HA} is a secret key of HA. Then HA stores { $C_2, h(.), P, X = x_{HA}P$ } into a SC and sends it to MU.

Step 3: MU calculates $C_3 = C_2 \oplus y$ and $C_4 = h(ID_{MU} \| PW_{MU}) \oplus y$. After that, MU stores { $C_3, C_4, h(.), P, X = x_{HA}P$ } into the SC.

4.3 Login and authentication

If MU wants to get roaming service from FA, MU and FA must be authenticated each other and establish the session key with the assistance of HA. The process is illustrated in algorithm 1.

Step 1: MU inputs ID_{MU} and PW_{MU}, the device terminal calculates

$y' = h(ID_{MU} \| PW_{MU}) \oplus C_4$,

$h(ID_{MU} \| x_{HA}) = C_3 \oplus y' \oplus h(ID_{MU} \| PW_{MU} \| y')$,

then chooses a random nonce $d_1 \in Z_p^*$ and computes

$E_1 = d_1 P$,

$E_2 = d_1 X$,

$E_3 = ID_{MU} \oplus h(E_2)$,

$E_4 = h(h(ID_{MU} \| x_{HA}) \| ID_{MU} \| ID_{HA} \| ID_{FA} \| E_1 \| E_2 \| E_3)$.

Then MU submits M_1={ $ID_{HA}, ID_{FA}, E_1, E_3, E_4$ } to FA.

Step 2: After receiving M_1 from MU, FA generates a random nonce d_2 and calculates

$E_5 = d_2 P$,

$E_6 = h(K_{FH} \| ID_{FA} \| ID_{HA} \| M_1 \| E_5)$,

where K_{FH} is a shared key between FA and HA. Then FA sends M_2={ M_1, E_5, E_6 } to HA.

Step 3: When getting message M_2 from FA, HA verifies if $E_6 = h(K_{FH} \| ID_{FA} \| ID_{HA} \| M_1 \| E_5)$ is correct? If not, reject. Otherwise, HA calculates

$$E_2' = x_{HA} E_1 = x_{HA} d_1 P,$$

$$ID_{MU}' = E_3 \oplus h(E_2'),$$

and verifies if $E_4 = h(h(ID_{MU}' \| x_{HA}) \| ID_{MU} \| ID_{HA} \| ID_{FA} \| E_1 \| E_2' \| E_3)$ is correct? If not, reject. Otherwise, HA calculates

$$E_7 = h(h(ID_{MU}' \| x_{HA}) \| ID_{MU}' \| ID_{FA} \| ID_{HA} \| E_2' \| E_5),$$

$$E_8 = h(ID_{FA} \| ID_{HA} \| K_{FH} \| E_1 \| E_5 \| E_7),$$

Then, HA returns M_3={ E_7, E_8 } to FA.

Step 4: After obtaining the message M_3, FA verifies if $E_8 = h(ID_{FA} \| ID_{HA} \| K_{FH} \| E_1 \| E_5 \| E_7)$ is correct? If not, reject. Otherwise, FA calculates the session key $SK_{FM} = h(d_2 E_1 \| E_1 \| E_5 \| ID_{FA} \| ID_{HA})$, $E_9 = h(SK_{FM} \| E_7 \| E_3)$. Then FA sends message M_4={ E_5, E_7, E_9 } to MU.

Step 5: After receiving M_4 from FA, MU checks validity of $E_7 = ?\, h(h(ID_{MU} \| x_{HA}) \| ID_{MU} \| ID_{FA} \| ID_{HA} \| E_2 \| E_5)$. If not, reject. Otherwise, MU calculates $SK_{MF} = h(d_1 E_5 \| E_1 \| E_5 \| ID_{FA} \| ID_{HA})$, and verifies the validity of $E_9 = ?\, h(SK_{FM} \| E_7 \| E_3)$. If yes, MU and FA share a session key SK_{FM} / SK_{MF}.

4.4 Password update

If MU needs to update his password, he inserts SC into the mobile device and enter his ID_{MU}, PW_{MU} and new password PW_{MU}^{new}, then the device calculates $y = C_4 \oplus h(ID_{MU} \| PW_{MU})$, $C_3^{new} = C_3 \oplus h(ID_{MU} \| PW_{MU} \| y) \oplus h(ID_{MU} \| PW_{MU}^{new} \| y)$, $C_4^{new} = C_4 \oplus h(ID_{MU} \| PW_{MU}) \oplus h(ID_{MU} \| PW_{MU}^{new})$. Finally, { C_3^{new}, C_4^{new} } is stored into the SC instead of { C_3, C_4 }.

MU	$\xleftrightarrow{\text{public channels}}$	FA	$\xleftrightarrow{\text{public channels}}$	HA

MU:

Inputs ID_{MU}, PW_{MU}

generates a nonce $d_1 \in Z_p^*$

computes

$y = h(ID_{MU} \| PW_{MU}) \oplus C_4$

$h(ID_{MU} \| x_{HA}) = C_3 \oplus y \oplus h(ID_{MU} \| PW_{MU} \| y)$

$E_1 = d_1 P$

$E_2 = d_1 X$

$E_3 = ID_{MU} \oplus h(E_2)$

$E_4 = h(h(ID_{MU} \| x_{HA}) \| ID_{MU} \| ID_{HA} \| ID_{FA} \| E_1 \| E_2 \| E_3)$

$\xrightarrow{M_1=\{ID_{HA}, ID_{FA}, E_1, E_3, E_4\}}$

FA:

Generates a nonce $d_2 \in Z_p^*$

$E_5 = d_2 P$

$E_6 = h(K_{FH} \| ID_{FA} \| ID_{HA} \| M_1 \| E_5)$

$\xrightarrow{M_2=\{M_1, E_5, E_6\}}$

HA:

Verifies $E_6 = ?\, h(K_{FH} \| ID_{FA} \| ID_{HA} \| M_1 \| E_5)$

$E_2' = x_{HA} E_1 = x_{HA} d_1 P$

$ID_{MU}' = E_3 \oplus h(E_2')$

$E_4 = ?\, h(h(ID_{MU}' \| x_{HA}) \| ID_{MU} \| ID_{HA} \| ID_{FA} \| E_1 \| E_2' \| E_3)$

$E_7 = h(h(ID_{MU}' \| x_{HA}) \| ID_{MU}' \| ID_{FA} \| ID_{HA} \| E_2' \| E_5)$

$E_8 = h(ID_{FA} \| ID_{HA} \| K_{FH} \| E_1 \| E_5 \| E_7)$

$\xleftarrow{M_3=\{E_7 , E_8\}}$

FA:

Verifies $E_8 = ?\, h(ID_{FA} \| ID_{HA} \| K_{FH} \| E_1 \| E_5 \| E_7)$

$SK_{FM} = h(d_2 E_1 \| E_1 \| E_5 \| ID_{FA} \| ID_{HA})$

$E_9 = h(SK_{FM} \| E_7 \| E_3)$

$\xleftarrow{M_4=\{E_5, E_7, E_9\}}$

MU:

Verifies $E_7 = ?\, h(h(ID_{MU} \| x_{HA}) \| ID_{MU} \| ID_{FA} \| ID_{HA} \| E_2 \| E_5)$

$SK_{MF} = ?\, h(d_1 E_5 \| E_1 \| E_5 \| ID_{FA} \| ID_{HA})$

verifies $E_9 = ?\, h(SK_{MF} \| E_7 \| E_3)$

The session key is $SK_{FM} = h(d_1 d_2 P \| d_1 P \| d_2 P \| ID_{FA} \| ID_{HA})$

Algorithm 1. Login and authentication of the our scheme

5. Formal verification

We apply formal verification tool ProVerif [34] which based on applied pi calculus [35] to verify authentication and security of our improved protocol. The ProVerif code is divided into three prats.

First is the declaration part that gives all definitions such as variables, constants, functions, equations, events and transmission channels, etc. Channel sch is used as a private channel between HA and MU in the registration, fmch and fhch are used as public communication channels between MU and FA, HA and FA, respectively, cch is the channel that all parameters are published by HA:

free sch:channel [private].

free cch:channel.

free fmch:channel.

free fhch:channel.

According to our protocol, P is a constant generator of elliptic curve. All participants MU, FA, HA need their free names IDMU, IDFA and IDHA. PWMU, xHA are defined as MU's password and HA's secret key, respectively. KFH is used as a pre-shared key between HA and FA. The most important free name SKFM, SKMF are used as a goal of session key security verification.

const P:bitstring.

free IDMU:bitstring [private].

free IDFA:bitstring.

free IDHA:bitstring.

free PWMU:bitstring [private].

free xHA:bitstring [private].

free KFH:bitstring [private].

free SKFM:bitstring [private].

free SKMF:bitstring [private].

The function h() represents a secure one-way hash function. The function concat() represents the bit-concatenation function. The functions xor(), mult() are modeled as the xor operation, the multiplication operation in elliptic curve cryptography, respectively.

fun h(bitstring):bitstring.

fun concat(bitstring):bitstring.

fun xor(bitstring,bitstring):bitstring.

fun mult(bitstring,bitstring):bitstring.

equation forall a:bitstring,b:bitstring;xor(xor(a,b),b)=a.

equation forall a:bitstring,b:bitstring;mult(a,mult(b,P))=mult(b,mult(a,P)).

In the second part, all actions of every participant process are structured as follows.

Registration phase:

Message 1: MU-->HA:{ ID_{MU}, C_1 }

Message 2: HA-->MU:{ C_2, X }

Login and authentication phase:

Message 3: MU-->FA:{ $ID_{HA}, ID_{FA}, E_1, E_3, E_4$ }

Message 4: FA-->HA:{ M_1, E_5, E_6 }

Message 5: HA-->FA:{ E_7, E_8 }

Message 6: FA-->MU:{ E_5 , E_7 , E_9 }

The process of MU consists of two different parts. In the registration process, MU submits his registration message 1 to HA and accepts message 2 from HA. The process codes are executed over secure channel sch. In the authentication, MU sends login message 3 to FA and waits for authentication message 6 from FA. Later MU performs the process of calculating and verifying the session key SKMF. The above processes are executed over public channel fmch. The ProVerif codes MUProcess are designed as:

```
let MUProcess=
        new y:bitstring;
        let C1=h(concat((IDMU,PWMU,y))) in
        out(sch,(IDMU,C1));
        in(sch,(xC2:bitstring,xX:bitstring));
        let C3=xor(xC2,y) in
        let C4=xor(y,h(concat((IDMU,PWMU)))) in

        !(let y'=xor(C4,h(concat((IDMU,PWMU)))) in
        let sv=xor(xor(C3,y'),h(concat((IDMU,PWMU,y')))) in
        new d1:bitstring;
        let E1=mult(d1,P) in
        let E2=mult(d1,xX) in
        let E3=xor(IDMU,h(E2)) in
        let E4=h(concat((sv,IDMU,IDHA,IDFA,E1,E2,E3))) in
        let M1=concat((IDHA,IDFA,E1,E3,E4)) in
        out(fmch,M1);
        event MUstartHA(E2);

        in(fmch,xM4:bitstring);
        let (xE5:bitstring,xxE7:bitstring,xE9:bitstring)=xM4 in
        if xxE7=h(concat((sv,IDMU,IDFA,IDHA,E2,xE5))) then
                event HAendMU(xxE7);
                let SKMF=h(concat((mult(d1,xE5),E1,xE5,IDFA,IDHA))) in
                if xE9=h(concat((SKMF,xxE7,E3))) then
                        event FAendMU(xE9)).
```

The process of HA has only authentication part. In this process, FA accepts request message 3 from MU and sends message 4 to HA. After getting and authenticating message 5 from HA, FA calculates the new session key SKFM and its related verification message. Then FA sends message 6 to MU. The above process is performed via public channels fmch and fhch. The codes are showed as below.

```
let FAProcess=
```

```
        in(fmch,xM1:bitstring);
        new d2:bitstring;
        let E5=mult(d2,P) in
        let E6=h(concat((KFH,IDFA,IDHA,xM1,E5))) in
        let M2=concat((xM1,E5,E6)) in
        out(fhch,M2);
        event FAstartHA(E6);

        in(fhch,xM3:bitstring);
        let
(xxIDHA:bitstring,xxIDFA:bitstring,xxE1:bitstring,xxE3:bitstring,xxE4:bitstring)=xM1 in
        let (xE7:bitstring,xE8:bitstring)=xM3 in
        if xE8=h(concat((IDFA,xxIDHA,KFH,xxE1,E5,xE7))) then
                event HAendFA(xE8);
                let SKFM=h(concat((mult(d2,xxE1),xxE1,E5,IDFA,xxIDHA))) in
                let E9=h(concat((SKFM,xE7,xxE3))) in
                let M4=concat((E5,xE7,E9)) in
                out(fmch,M4);
                event FAstartMU(E9).
```

The process of HA consists of two different parts. In the registration process, HA waits for registration message 1 from MU and responds message 2 to MU via a secure channel sch. In the authentication process, HA accepts request message 4 from FA and returns message 5 to FA via a public channel fhch.

```
let HAProcess=
        let X=mult(xHA,P) in
        out(cch,X);
        in(sch,(xIDMU:bitstring,xC1:bitstring));
        let C2=xor(h(concat((xIDMU,xHA))),xC1) in
        out(sch,(C2,X));

        in(fhch,(xM2:bitstring));
        let (xxM1:bitstring,xE5:bitstring,xE6:bitstring)=xM2 in
        let
(xIDHA:bitstring,xIDFA:bitstring,xE1:bitstring,xE3:bitstring,xE4:bitstring)=xxM1 in
        if xE6=h(concat((KFH,xIDFA,IDHA,xxM1,xE5))) then
                event FAendHA(xE5);
                let E2'=mult(xHA,xE1) in
                let IDMU'=xor(xE3,h(E2')) in
```

if xE4=h(concat((h(concat((IDMU',xHA))),IDMU',IDHA,xIDFA,xE1,E2',xE3))) then
event MUendHA(E2');
let E7=h(concat((h(concat((IDMU',xHA))),IDMU',xIDFA,IDHA,E2',xE5))) in
let E8=h(concat((xIDFA,IDHA,KFH,xE1,xE5,E7))) in
event HAstartFA(E8);
event HAstartMU(E7);
let M3=concat((E7,E8)) in
out(fhch,M3).

The main process is modeled as parallel executions of multiple participants so the exclamation(!) point is placed in front of each subprocess.

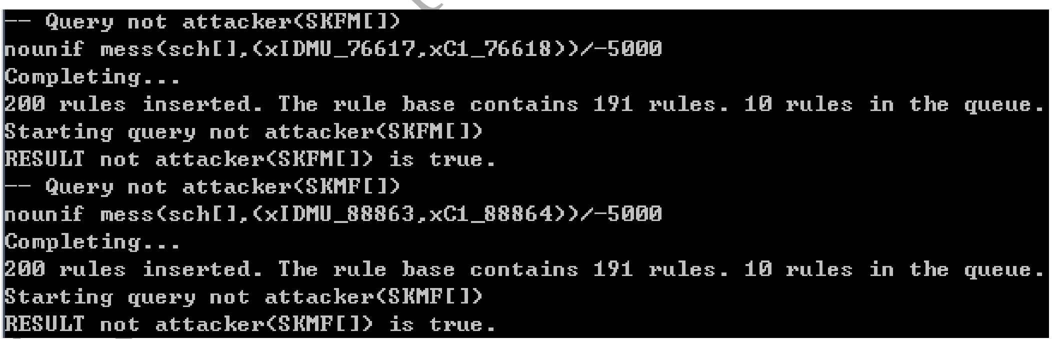
process MUProcess|!FAProcess|!HAProcess

The third part is security property that defines all queries and authentication attributes. We check the secrecy of session key by attacker's queries. The ProVerif query codes are defined as follow.

query attacker(SKMF).

query attacker(SKFM).

Figure 1 demonstrates that attacker(SKMF)/attacker(SKFM) is not true in the results of attacker query. It deeply reveals that the session key is secure, and the attacker is unable to compute it by any method.



```
-- Query not attacker(SKFM[])
nounif mess(sch[],(xIDMU_76617,xC1_76618))/-5000
Completing...
200 rules inserted. The rule base contains 191 rules. 10 rules in the queue.
Starting query not attacker(SKFM[])
RESULT not attacker(SKFM[]) is true.
-- Query not attacker(SKMF[])
nounif mess(sch[],(xIDMU_88863,xC1_88864))/-5000
Completing...
200 rules inserted. The rule base contains 191 rules. 10 rules in the queue.
Starting query not attacker(SKMF[])
RESULT not attacker(SKMF[]) is true.
```

Figure 1. The results of attacker's queries.

We defined ten events to evaluate the reachability of authentication in the model.

event MUstartHA(bitstring).
event MUendHA(bitstring).
event FAstartHA(bitstring).
event FAendHA(bitstring).
event HAstartFA(bitstring).
event HAendFA(bitstring).
event HAstartMU(bitstring).
event HAendMU(bitstring).

event FAstartMU(bitstring).

event FAendMU(bitstring).

We use correspondence assertions to verify authentication properties of three participants. In the formal proof, we construct five authentication correlations. The event MUstartHA(bitstring) presents the beginning of the record that MU has already performed the authentication process with HA. The event MUendHA(bitstring) presents the end of the record that HA terminates the authentication process with MU. The other events are similar to the two events. Reachability of all events is verified by the following ProVerif queries.

query id:bitstring;inj-event(MUendHA(id))==>inj-event(MUstartHA(id)).

query id:bitstring;inj-event(FAendHA(id))==>inj-event(FAstartHA(id)).

query id:bitstring;inj-event(HAendFA(id))==>inj-event(HAstartFA(id)).

query id:bitstring;inj-event(HAendMU(id))==>inj-event(HAstartMU(id)).

query id:bitstring;inj-event(FAendMU(id))==>inj-event(FAstartMU(id)).

Figures 2 demonstrate five correspondence query results are true, that is, the proposed protocol satisfies all authentication requirements.

```
-- Query inj-event(FAendMU(id)) ==> inj-event(FAstartMU(id))
nounif mess(sch[],(xIDMU_10234,xC1_10235))/-5000
Completing...
200 rules inserted. The rule base contains 198 rules. 60 rules in the queue.
400 rules inserted. The rule base contains 396 rules. 46 rules in the queue.
600 rules inserted. The rule base contains 536 rules. 67 rules in the queue.
800 rules inserted. The rule base contains 708 rules. 50 rules in the queue.
1000 rules inserted. The rule base contains 866 rules. 25 rules in the queue.
Starting query inj-event(FAendMU(id)) ==> inj-event(FAstartMU(id))
RESULT inj-event(FAendMU(id)) ==> inj-event(FAstartMU(id)) is true.
-- Query inj-event(HAendMU(id_15569)) ==> inj-event(HAstartMU(id_15569))
nounif mess(sch[],(xIDMU_25558,xC1_25559))/-5000
Completing...
200 rules inserted. The rule base contains 198 rules. 43 rules in the queue.
400 rules inserted. The rule base contains 371 rules. 12 rules in the queue.
Starting query inj-event(HAendMU(id_15569)) ==> inj-event(HAstartMU(id_15569))
RESULT inj-event(HAendMU(id_15569)) ==> inj-event(HAstartMU(id_15569)) is true.
-- Query inj-event(HAendFA(id_29088)) ==> inj-event(HAstartFA(id_29088))
nounif mess(sch[],(xIDMU_38984,xC1_38985))/-5000
Completing...
200 rules inserted. The rule base contains 194 rules. 10 rules in the queue.
Starting query inj-event(HAendFA(id_29088)) ==> inj-event(HAstartFA(id_29088))
RESULT inj-event(HAendFA(id_29088)) ==> inj-event(HAstartFA(id_29088)) is true.
-- Query inj-event(FAendHA(id_41885)) ==> inj-event(FAstartHA(id_41885))
nounif mess(sch[],(xIDMU_51774,xC1_51775))/-5000
Completing...
200 rules inserted. The rule base contains 192 rules. 10 rules in the queue.
Starting query inj-event(FAendHA(id_41885)) ==> inj-event(FAstartHA(id_41885))
RESULT inj-event(FAendHA(id_41885)) ==> inj-event(FAstartHA(id_41885)) is true.
-- Query inj-event(MUendHA(id_54219)) ==> inj-event(MUstartHA(id_54219))
nounif mess(sch[],(xIDMU_64108,xC1_64109))/-5000
Completing...
200 rules inserted. The rule base contains 196 rules. 14 rules in the queue.
Starting query inj-event(MUendHA(id_54219)) ==> inj-event(MUstartHA(id_54219))
RESULT inj-event(MUendHA(id_54219)) ==> inj-event(MUstartHA(id_54219)) is true.
```

Figure 2. The results of correspondence queries.

6. Security analysis

We will show that our scheme can resist a variety of attacks and possess some good security properties.

6.1 Anonymity and unlinkability

User anonymity and unlinkability are important properties of privacy protection. If messages { M_1, M_2, M_3, M_4 }and{ C_3 , C_4 , $h(.)$, P , $X = x_{HA}P$ }stored in SC are captured, and the adversary wants to compute the user's identity $ID_{MU} = E_3 \oplus h(E_2)$, he must calculate $E_2 = d_1 x_{HA} P$ or $h(E_2)$, but it's hard even if he knows d_1P and $x_{HA}P$, due to Elliptic Curve Diffie-Hellman problem (ECDHP). On the other hand, d_1 and d_2 are randomly chosen in different session, Therefore, the adversary cannot know the same MU in different session run.

6.2 Impersonation attack and man in the middle attack

Xiong et al.'s scheme suffers from impersonation attack, the reason is that $E_5 = \beta P$ generated by FA, does not embed to E_8. In our scheme, $E_5 = d_2 P$ is generated by FA, and authenticated by HA using $E_6 = h(K_{FH} \| ID_{FA} \| ID_{HA} \| M_1 \| E_5)$. After that, HA embed $E_5 = d_2 P$ to $E_7 = h(h(ID_{MU}' \| x_{HA}) \| ID_{MU}' \| ID_{FA} \| ID_{HA} \| E_2' \| E_5)$, so that MU can believe E_5 is authenticated by HA, and can use E_5 and $E_1 = d_1 P$ to generate the session. FA also believe $E_1 = d_1 P$ is authenticated by HA by verifying $E_8 = h(ID_{FA} \| ID_{HA} \| K_{FH} \| E_1 \| E_5 \| E_7)$. On the other hand, FA and HA are authenticated each other by using a pre-shared key SK_{FH} .

From the above analysis, we can know that our scheme can resist above two attacks.

6.3 Two-factor security and offline password guessing attack

On the one hand, when an adversary knows all data { C_3 , C_4 , $h(.)$, P , X } in smart card, he can perform follow steps to guess MU's password. Assume that an adversary has already intercepted all message from the public channels and MU's identity ID_{MU} . Then he selects a password PW_{MU}' and calculates $y' = h(ID_{MU} \| PW_{MU}') \oplus C_4$, $h(ID_{MU} \| x)' = C_3 \oplus y' \oplus h(ID_{MU} \| PW_{MU}' \| y')$, and tries to verify $E_4 = ?h(h(ID_{MU} \| x)' \| ID_{MU} \| ID_{HA} \| ID_{FA} \| E_1 \| E_2 \| E_3)$. However, it is impossible, because he cannot calculate $E_2 = d_1 X$ from { d_1P , $x_{HA}P$, $h(E_2) = ID_{MU} \oplus E_3$ } due to CDHP and one-way hash function.

On the other hand, if an adversary knows MU's password but not knows $\{C_3, C_4, h(.), P, X\}$ stored in SC, he cannot compute $y' = h(ID_{MU} \| PW_{MU}') \oplus C_4$ and $h(ID_{MU} \| x_{HA}) = C_3 \oplus y' \oplus h(ID_{MU} \| PW_{MU} \| y')$, so he cannot launch impersonation attack.

Therefore, our scheme satisfies two-factor security.

6.4 Replay attack

In each session run of our scheme, d_1 and d_2 are random integer numbers chosen by MU and FA, respectively, they are different in each session, so the replay attack is invalid.

6.5 Perfect forward secrecy

Since the session key is $T_E \approx$, assume that all data stored in SC, MU's T_{EXP}, and the secret key $T_H \approx$ are compromised, an adversary cannot compute $d_1 d_2 P$ due to ECDHP. Thus, the adversary can still not to compute SK_{FM}.

6.6 Known session key security

Since an adversary cannot compute $d_1 d_2 P$, so he cannot compute SK_{FM}. On the other hand, even if the adversary knows one session key, he cannot compute the before and the future session keys, because d_1 and d_2 are random integer numbers and they are different in each session.

6.7 Fair key agreement

Because $SK_{FM} = h(d_1 d_2 P \| d_1 P \| d_2 P \| ID_{FA} \| ID_{HA})$ consists of two secret value $\{d_1, d_2\}$, which chosen by MU and FA independently.

6.8 Verifier stolen attack

Because both HA and FA need not store user's registration information, therefore, our scheme can resist verifier stolen attack.

6.9 Session key unknown to HA

In our scheme, the session key is $T_E \approx$, where d_1 and d_2 are random integer numbers chosen by MU and FA, and SK_{FM} is computed by MU and FA, respectively. Therefore, HA can not know the session key.

7. Security and efficiency comparisons

Since the latest schemes in [17,18] and [29-31] have relatively well computational efficiency and security. Therefore, we only give the comparisons between our scheme and these schemes in terms of security and efficiency, which are given in tables 2 and 3. According to [36], we know some computation costs are as follow.

The unit cost of hash function: $T_H \approx 0.0023$ms;

The unit cost of symmetric encryption or decryption: $T_{SE} \approx 0.0046$ms;

The unit cost of multiplication operation in elliptic curve cryptography: $T_M \approx 2.226$ms;

The unit cost of modular exponentiation: $T_{EXP} \approx 3.85$ ms.

According to tables 2 and 3, our scheme is more secure than others, and has acceptable efficiency.

Table 2. Security comparison

	Farash et al.[29]	Chaudhry et al.[31]	Shin et al.[30]	Kang et al.[17]	Xiong et al.[18]	Our scheme
Anonymity and unlinkability	×	×	×	√	√	√
Session key security	√	√	×	√	√	√
Resist impersonation attack	×	√	×	√	×	√
Resist man-in-the-middle attack	√	√	√	√	√	√
Two-factor security	×	×	×	×	×	√
Resist replay attack	√	√	√	×	√	√
Resist verifier stolen attack	√	√	√	×	√	√
Known session key security	×	×	√	√	√	√
Session key unknown to HA	√	√	√	×	√	√
Perfect forward secrecy	×	×	×	×	√	√
Fair key agreement	√	√	√	√	√	√

Table 3. Efficiency comparison

	Farash et al.[29]	Chaudhry et al.[31]	Shin et al.[30]	Kang et al.[17]	Xiong et al.[18]	Our scheme
Total time	$11T_H+4$	$8T_H+5T_E$	$12T_H+4$	$20T_H+2$	$17T_H+6T_M$	$17T_H+6$

	T_{SE}		$T_{SE}+1T_{EXP}$	$T_{SE}+3T_{EXP}$		T_M
Estimated time(ms)	0.0437	0.0414	3.896	11.6052	13.3951	13.3951

8. Conclusion

In this paper, we point out some security flaws of Xiong et al's scheme. First, an adversary can pretend to be FA and communicate with MU. Second, it lacks of two-factor security because an adversary can obtain MU's password by launching off-line password guessing attack. Third, we found their scheme may be unworkable due to an error, which is easily to be fixed. Then we propose an improve scheme to fix these flaws. And it is proved that the proposed scheme is security and has some good security properties. Therefore, the proposed scheme can be used to the smart city.

Acknowledgements

This research was supported by National Natural Science Foundation of China (Grant No. 61702152), and the National Key R&D Program of China (Grant No.2017YFB0802000).

References

1. M. U. Farooq, M. Waseem, M. T. Qadri, M. Waqar, Understanding 5g wireless cellular network: challenges, emerging research directions and enabling technologies, Wirel. Pers. Commun. 95(2) (2017) 1–25.
2. M. Chen, Y. Zhang, L. Hu, T. Taleb, Z. Sheng, Cloud-based wireless network: virtualized, reconfigurable, smart wireless network to enable 5g technologies, Mobile Netw. Appl. 20(6) (2015) 704–712.
3. P. Lynggaard, K. E. Skouby, Deploying 5g-technologies in smart city and smart home wireless sensor networks with interferences, Wirel. Pers. Commun. 81(4) (2015) 1399–1413.
4. C. I. Badoi, N. Prasad, R. Prasad, Virtualization and scheduling methods for 5G cognitive radio based wireless networks, Wirel. Pers. Commun. 89(2) (2016) 1–21.
5. J. Zhu, J. Ma, A new authentication scheme with anonymity for wireless environments, IEEE Trans. Consum. Electron. 50(1) (2004) 231–235.

6. C. C. Lee, M. S. Hwang, I. E. Liao, Security enhancement on a new authentication scheme with anonymity for wireless environments, IEEE Trans. Ind. Electron. 53(5) (2006) 1683–1687.
7. C. C. Wu, W. B. Lee, W. J. Tsaur, A secure authentication scheme with anonymity for wireless communications, IEEE Commun. Lett. 12(10) (2008) 722–723.
8. C. C. Chang, C. Y. Lee, Y. C. Chiu, Enhanced Authentication scheme with anonymity for roaming service in global networks, Comput. Commun. 32(4) (2009) 611–618.
9. T. Y. Youn, Y. P. Park, J. Lim, Weaknesses in an anonymous authentication scheme for roaming service in global mobility networks, IEEE Commun. Lett. 13(7) (2009) 471–473.
10. D. He, M. Ma, Y. Zhang, C. Chen, J. Bu, A strong user authentication scheme with smart cards for wireless communications, Comput. Commun. 34(3) (2011) 367–374.
11. J. Xu, D. Feng, Security flaws in authentication protocols with anonymity for wireless environments, ETRI Journal 31(4) (2009) 460–462.
12. P. Zeng, Z. Cao, K. K. R. Choo, S. Wang, On the anonymity of some authentication schemes for wireless communications, IEEE Commun. Lett. 13(3) (2009) 170–171.
13. H. Mun, K. Han, Y. S. Lee, C. Y. Yeun, H. H. Choi, Enhanced secure anonymous authentication scheme for roaming service in global mobility networks, Math. Comput. Model. 55(1–2) (2012) 214–222.
14. Q. Xie, X. Tan, D. Wong, G. S. Wong, M. Bao, N. Dong, A practical anonymous authentication protocol for wireless roaming, Secur. Commun. Netw. 7(8) (2014) 1264–1273.
15. D. He, Y. Zhang, J. Chen, Cryptanalysis and improvement of an anonymous authentication protocol for wireless access networks, Wirel. Pers. Commun. 74(2) (2014) 229–243.
16. M. Karuppiah, R. Saravanan, A secure authentication scheme with user anonymity for roaming service in global mobility networks, Wirel. Pers. Commun. 84(3) (2015) 1–24.
17. M. Kang, H. S. Rhee, J. Y. Choi, Improved user authentication scheme with user anonymity for wireless communications, IEICE Trans. Fundamentals 94(2) (2011) 860–864.
18. L. Xiong, A. K. Sangaiah, S. Kumari, W. Fan, S. Jian, M. K. Khan, An efficient authentication and key agreement scheme with user anonymity for roaming service in smart city, Pers. Ubiquit. Comput. 21(12) (2017) 1–15.
19. D. He, M. Ma, Y. Zhang, C. Chen, J. Bu, A strong user authentication scheme with smart cards for wireless communications, Comput. Commun. 34(3) (2011) 367–374.
20. Y. C. Chen, S. C. Chuang, L. Y. Yeh, J. L. Huang, A practical authentication protocol with anonymity for wireless access networks, Wirel. Commun. Mob. Comput. 11(10) (2011) 1366–1375.
21. C. Chen, D. He, S. Chan, J. Bu,Y. Gao, R. Fan, Lightweight and provably secure user authentication with anonymity for the global mobility network, Int. J. Commun. Syst. 24(3) (2011) 347–362.
22. Q. Xie, B. Hu, X. Tan, M. Bao, X. Yu, Robust anonymous two-factor authentication scheme for roaming service in global mobility network, Wirel. Pers. Commun. 74(2) (2014) 601–614.

23. Q. Xie, D. Hong, M. Bao, N. Dong, D. S. Wong, Privacy-preserving mobile roaming authentication with security proof in global mobility networks, International Journal of Distributed Sensor Networks 2014(1) (2014) 1–7.

24. D. He, N. Kumar, M. Khan, J. H. Lee, Anonymous two-factor authentication for consumer roaming service in global mobility networks, IEEE Trans. Consum. Electron. 59(4) (2013) 811–817.

25. D. He, S. Chan, C. Chen, J. Bu, R. Fan, Design and validation of an efficient authentication scheme with anonymity for roaming service in global mobility networks, Wirel. Pers. Commun. 61(2) (2011) 465–476.

26. Q. Jiang, J. Ma, G. Li, L. Yang, An enhanced authentication scheme with privacy preservation for roaming services in global mobility networks, Wirel. Pers. Commun. 68(4) (2013) 1477–1491.

27. F. Wen, W. Susilo, G. Yang, A secure and effective anonymous user authentication scheme for roaming service in global mobility networks, Wirel. Pers. Commun. 73(3) (2013) 993–1004.

28. P. Gope, T. Hwang, Enhanced secure mutual authentication and key agreement scheme preserving user anonymity in global mobile networks, Wirel. Pers. Commun. 82(4) (2015) 2231–2245.

29. M.S. Farash, S.A. Chaudhry, M. Heydari, S. Sadough, S. Mohammad, S. Kumari, M.K. Khan, A lightweight anonymous authentication scheme for consumer roaming in ubiquitous networks with provable security, Int. J. Commun. Syst. 30(4) (2017) e3019.

30. S. Shin, H. Yeh, K. Kim, An efficient secure authentication scheme with user anonymity for roaming user in ubiquitous networks, Peer-to-Peer Netw. Appl. 8(4) (2015) 674–683.

31. S.A. Chaudhry, A. Albeshri, N. Xiong, C. Lee, T. Shon, A privacy preserving authentication scheme for roaming in ubiquitous networks, Clust. Comput. 20(2) (2017) 1223–1236.

32. Q. Xie, B. Hu, X. Tan, D. S. Wong, Chaotic maps-based strong anonymous authentication scheme for roaming services in global mobility networks, Wirel. Pers. Commun. 96(4) (2017) 1–16.

33. Q. Xie, D. Wong, G. Wang, X. Tan, K. Chen, L. Fang, Provably secure dynamic id-based anonymous two-factor authenticated key exchange protocol with extended security model, IEEE Trans. Inf. Forensics Security 12(6) (2017) 1382–1392.

34. M. Abadi, B. Blanchet, H. C. Lundh, Models and Proofs of Protocol Security: A Progress Report, 21st International Conference on Computer Aided Verification, Grenoble, France, 2009, pp. 35–49.

35. M. Abadi , C. Fournet, Mobile Values, New Names, Secure Communication, Proceedings of the 28th ACM SIGPLAN-SIGACT symposium on Principles of programming languages(POPL'01), ACM New York, 2001, pp. 104–115.

36. H. H. Kilinc, T. Yanik, A survey of sip authentication and key agreement schemes, IEEE Commun. Surv. Tuts. 16(2) (2014) 1005–1023.

Qi Xie received the Ph.D. degree in applied mathematics from Zhejiang University, China, in 2005. He was a Visiting Scholar with the Department of Computer Science, University of Birmingham, U.K., from 2009 to 2010, and a Visiting Scholar with the Department of Computer Science, City University of Hong Kong, in 2012. He is currently a Professor with the Hangzhou Key Laboratory of Cryptography and Network Security, Hangzhou Normal University, China. He has published over 70 research papers in international journals and conferences, and served as a General Co-Chair of the ISPEC2012 and the ACM ASIACCS2013, and a Reviewer for over 20 international journals. His research area is applied cryptography, including digital signatures, authentication and key agreement protocols.

Lingfeng Hwang is currently a M.S. candidate of Hangzhou Normal University, China. His research focus on authentication and key agreement protocols in wireless environment.