Journal of Ambient Intelligence and Humanized Computing (2019) 10:661–680
https://doi.org/10.1007/s12652-018-0716-4

ORIGINAL RESEARCH

A Privacy Preserving three-factor authenticated key agreement protocol for client–server environment

Alavalapati Goutham Reddy[1] · Ashok Kumar Das[2] · Vanga Odelu[3] · Awais Ahmad[4] · Ji Sun Shin[1]

Received: 15 November 2017 / Accepted: 14 February 2018 / Published online: 22 February 2018

Abstract
Research has proven that accomplishing security properties while improving performance of an authentication protocol is a challenging task. Numerous authentication protocols proposed in the recent times are still behind in achieving the concrete objectives. Qi et al. and Lu et al. recently proposed two-factor authenticated key-agreement protocols for client–server architecture. This paper revisits their protocols and analyzes the shortcomings of such approaches. We also propose an improved authenticated key agreement protocol for client–server environment to defeat mentioned weaknesses of existing protocols that are discussed in related works. The rigorous security analysis using Burrows–Abadi–Needham logic, formal security verification using Real-OR-Random model, simulations using the Automated Validation of Internet Security Protocols and Applications tool, and the informal security analysis shows that the proposed protocol is secure. Additionally, we summarize the results to ensure that the proposed protocol is efficient compared to the existing related protocols.

Keywords Mutual authentication · Key agreement · Client–server · ROR model · BAN logic · AVISPA

1 Introduction

Despite the long rigorous research efforts, designing a perfect authentication protocol for two-party communication remains an interesting topic. The rapid evolution of handheld devices and communication technologies has entailed the installation of different authentication methods. Smartcard with password based authentication is one of the modest and inexpensive methods, which is believed to be safe and sound than passwords alone. On the contrary, studies have shown that two-factor authentication techniques are still vulnerable under several scenarios such as faulty protocol design, and when the passwords are guessed, and the smartcard stored data is leaked out (Kocher et al. 1999; Messerges et al. 2002; Wang and Wang 2015). The ascribed limitations of two-factor authentication methods necessitated additional security called biometrics. The properties of biometric keys (iris, face, finger print, palm print etc.) such as uniqueness, non-transferability and unforgeability makes it robust (Li and Hwang 2010; Mishra et al. 2014).

1.1 Related work

A mobile client–server communication prototype enables multiple clients to avail the services offered by a single server irrespective of geographical location. Authentication of such parties happening over public channels must be secured through a feasible means. Several researchers have followed various approaches during the past three decades, namely, knowledge-based (Lamport 1981; Wu 1995;

✉ Ashok Kumar Das
iitkgp.akdas@gmail.com; ashok.das@iiit.ac.in

Alavalapati Goutham Reddy
goutham.ace@gmail.com

Vanga Odelu
odelu.vanga@gmail.com

Awais Ahmad
aahmad.marwat@gmail.com

Ji Sun Shin
jsshin@sejong.ac.kr

[1] Department of Computer and Information Security, Sejong University, Gwanjin-Gu, Seoul 05006, South Korea

[2] Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500032, India

[3] Department of Computer Convergence Software, Korea University, Sejong 30019, South Korea

[4] Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38541, South Korea

Tzong-Chen and Hung-Sung 1996; Jan and Chey 1998; Tan and Zhu 1999; Chan and Cheng 2000; Chien et al. 2001; Liao et al. 2006), token-based (Wang et al. 2009, 2011, 2015; Xu et al. 2009; Yang and Chang 2009; Yoon and Yoo 2009; Song 2010; Pippa et al. 2010; Sood et al. 2010; Wu and Tseng 2010; Islam and Biswas 2011, 2014; Debiao et al. 2012; He 2012; Hsieh and Leu 2012; Madhusudhan and Mittal 2012; Wen and Li 2012; Chou et al. 2013; Li et al. 2013; Chang et al. 2014; Chen et al. 2014; Farash and Attari 2014; Kumari and Khan 2014; Kumari et al. 2014, 2017; Zhang et al. 2014; Goutham et al. 2015; Jiang et al. 2015; Tu et al. 2015; Farash 2016; Lu et al. 2016; Xie et al. 2016; Luo et al. 2017; Qi and Chen 2017), biometrics-based (Khan et al. 2008, 2014; Fan and Lin 2009; Das 2011; Li et al. 2011, 2014; Chen et al. 2012; An 2012; Yeh et al. 2013; Cao and Ge 2015; Das and Goswami 2015; Wu et al. 2015; Han et al. 2016; Chaturvedi et al. 2017; Xie et al. 2017). This paper's main aim is to study and analyze the two-factor authentication methods, and then to fix the shortcomings by proposing and examining a new three-factor authentication method.

Xu et al. (2009) proposed a smartcard based authentication protocol and stated that their protocol remain safe though the smartcard stored data is extracted. Conversely, Song (2010) proved that Xu et al.'s protocol is vulnerable to user impersonation attacks when the data on the smartcard is gathered. Then they proposed an efficient smartcard with password based protocol, which was later indicated by Pippa et al. (2010) that Song et al.'s protocol cannot afford forward secrecy. In the same year, Sood et al. (2010) found that Xu et al.'s protocol is even prone to forgery attacks when the valid login request is intercepted, and they put forward an improved protocol over Xu et al.'s protocol. Yet Chen et al. (2014) reviewed Xu et al. (2009), Sood et al. (2010) and Song's (2010) protocols, and presented various flaws such as user impersonation attacks, improper mutual authentication and stolen smartcard attacks. Chen et al. proposed an enhanced robust two-factor authentication protocol while considering the merits and demerits of all three aforementioned protocols. However, Li et al. (2013), Jiang et al. (2015), and Xie et al. (2016) have analyzed Chen et al.'s (2014) protocol, independently. They demonstrated that Chen et al.'s protocol is susceptible to password guessing attacks, smartcard stolen attacks and user impersonation attacks. Besides, Chen et al.'s protocol cannot hold perfect forward secrecy, login verification and efficient password changing phase. Nonetheless, Xie et al. (2016) also proposed an authentication protocol based on smartcard with password. However, Lu et al. (2016) asserted that Xie et al.'s protocol consists of several drawbacks such as prone to insider attacks, trace attacks, user impersonation attacks and no login verification. Several other authentication protocols (Irshad et al. 2017a, b, c, d; Gope and Das 2017; Gope 2017; Gope and Hwang 2016a, b; Reddy et al. 2016) have been proposed recently in the literature in order to improve the efficiency and security over the existing protocols.

Lu et al. (2016), and Qi et al. (2017) proposed two-factor authenticated key-agreement protocols for client–server architecture using various approaches. They claimed that their protocol is secure against attacks and provides distinguished properties. Conversely, the cryptanalysis section of this paper shows the drawbacks of Lu et al.'s protocol such as prone to server impersonation attacks, privileged insider attacks, and lack of user anonymity; Qi et al.'s protocol such as prone to user impersonation attacks, session-specific ephemeral secret leakage attacks, insider attacks, and lack of user anonymity.

1.2 Research contributions

- We revisit and provide the cryptanalysis of recently proposed Qi et al. and Lu et al.'s protocols.
- We put forward an anonymous three-factor mutually authenticated key agreement protocol for client–server architecture on elliptic curve cryptography (ECC).
- The formal security of the proposed protocol is verified using ROR model, AVISPA simulation tool, and the mutual authentication using BAN logic.
- The analysis of the paper evident that the proposed protocol performs better compared to its counterparts.

1.3 Threat model

A threat modeling is an imperative module of the designing an authenticated key agreement protocol. The threat modeling is a process for enhancing security by classifying vulnerabilities and objectives, and then defining preventive measures of threats to the system. In this framework, a threat is a potential malicious attack that would be perpetrated by an adversary, say $Æ$ that can cause damage to the assets. The threat model of this paper is built on following assumptions.

- $Æ$ has partial/complete control over the messages that were transmitted over the public channels. This includes to intercept, modify, and delete any communicated message. Under this case, the Dolev-Yao (DY) threat model is followed (Dolev and Yao 1983).
- $Æ$ is capable to extract the parameters that are stored on a smartcard issued to a user.
- $Æ$ can try to obtain sensitive information of users such as password by performing offline/online password guessing attacks.
- $Æ$ may also try to gain access to the authorized system with the stolen smartcard while constantly guessing the credentials. The low-entropy passwords could make $Æ$ job even easier.

- *Æ* can trace the users' activities using the obtained information from the transmitted messages via open channels.
- *Æ* at the server end, called the privileged-insider user, can perform malicious activities using the data that was received during the registration phase.

1.4 Paper organization

Section 2 revisits Qi et al.'s protocol. Section 3 cryptanalyses Qi et al.'s protocol. Section 4 revisits Lu et al.'s protocol. Section 5 cryptanalyses Lu et al.'s protocol. Section 6 portrays the proposed protocol. Section 7 analyzes the security of the proposed protocol. Section 8 affords comparisons with the related protocols. Finally, Sect. 9 concludes the paper.

2 Revisiting Qi et al.'s protocol

This section revisits Qi et al.'s (2017) two-party authenticated key-agreement protocol for mobile architecture. Their scheme comprises of four phases, namely, system initialization phase, user registration phase (over a secure channel), login and mutually authenticated key-agreement phase (over a public channel), and password changing phase (over a secure channel). Here, we present only the first three phases as the drawbacks lies in these phases.

2.1 System initialization phase

In this phase, server S chooses an elliptic curve E/F_p defined over finite field F_p of prime order p, a base point P in E/F_p of order q, a long-term private key k_s, corresponding public key $Q_s = k_sP$, and cryptographic hash functions $h_1()$, $h_2()$.

2.2 User registration phase

Step 1: User U sends chosen credentials $\{ID_u, PW_u\}$ to the server S.

Step 2: S checks the validity of ID_u, and verifies $h_2(ID_u)$ if it exists in the database. S computes secret key $l = h_1(d_s) \oplus h_2(ID_u \| PW_u)$ and delivers it to U, where d_s is server's long-term private key.

S follows two approaches: online and offline to deliver the secret key. Online uses transport layer security channel in the https mode, and offline uses a smartcard stored l in it.

2.3 Login and mutually authenticated key-agreement phase

In this phase, U and S can authenticate each other and make a session key as shown in Fig. 1.

Step 1: U enters his/her ID_u, PW_u then chooses a random number $r_u \in Z_n^*$, and computes $R_u = r_uP$, $R = r_uQ_s$, $CID_u = ID_u \oplus l \oplus h_2(ID_u \| PW_u)$, and $Auth_u = h_2(ID_u \| R \| l \oplus h_2(ID_u \| PW_u))$. U sends $\{Auth_u, CID_u, R_u\}$ to S.

Step 2: S computes $ID_u = CID_u \oplus h_1(d_s)$, $R^* = d_sR_u$, and $Auth_u^* = h_2(ID_u \| R^* \| h_1(d_s))$. S verifies if $Auth_u^* = Auth_u$. If

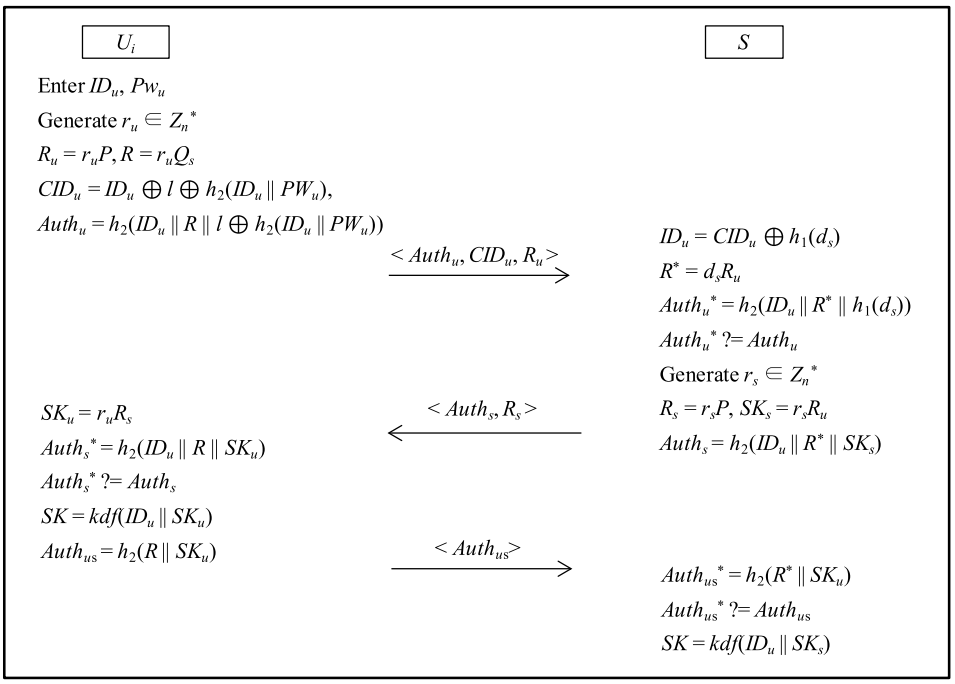

Fig. 1 Login and authenticated key-agreement phase of Qi et al.'s protocol

not, the process aborts. S generates a random number $r_s \in Z_n^*$ and computes $R_s = r_sP$, $SK_s = r_sR_u$, and $Auth_s = h_2(ID_u \| R^* \| SK_s)$. S sends $\{Auth_s, R_s\}$ to U.

Step 3: U computes $SK_u = r_uR_s$, and $Auth_s^* = h_2(ID_u \| R \| SK_u)$. U verifies if $Auth_s^* = Auth_s$. If not, the process aborts. U further computes the session key $SK = kdf(ID_u \| SK_u)$ and $Auth_{us} = h_2(R \| SK_u)$, and sends $Auth_{us}$ to S, where kdf is key derivation function.

Step 4: S computes $Auth_{us}^* = h_2(R^* \| SK_u)$ and verifies if $Auth_{us}^* = Auth_{us}$. If yes, S computes the session key $SK = kdf(ID_u \| SK_s)$ and allows the further communication.

3 Cryptanalysis of Qi et al.'s protocol

This section demonstrates the security drawbacks of Qi et al.'s protocol (2017) such as prone to user impersonation attacks, insider attacks, session-specific ephemeral secret leakage attacks, and lack of user anonymity.

3.1 Lack of anonymity

Step 1: Assume an adversary $Æ$ is a registered user with valid credentials $l = h_1(d_s) \oplus h_2(ID_Æ \| PW_Æ)$. $Æ$ can retrieve $h_1(d_s) = l \oplus h_2(ID_Æ \| PW_Æ)$.

Step 2: Let $Æ$ intercept the message $\{Auth_u, CID_u, R_u\}$ of U to S. Now $Æ$ can compute the identity of U using $h_1(d_s) = l \oplus h_2(ID_u \| PW_u)$ and $CID_u = ID_u \oplus l \oplus h_2(ID_u \| PW_u)$ as $ID_u = CID_u \oplus h_1(d_s)$.

Thus, it violates user anonymity and does not prevent traceability.

3.2 Prone to user impersonation attacks

Assume $Æ$ knows the identity ID_u of a legitimate U, then $Æ$ can impersonate U as follows:

Step 1: $Æ$ chooses a random number $r_a \in Z_n^*$, and computes $R_a = r_aP$, $R = r_aQ_s$, $CID_a = ID_u \oplus h_1(d_s)$, and $Auth_a = h_2(ID_u \| R \| h_1(d_s))$. $Æ$ sends $M_1 = \{Auth_a, CID_a, R_a\}$ to S.

Step 2: S computes $ID_u^* = CID_a \oplus h_1(d_s)$, $R^* = d_sR_a$, and $Auth_a^* = h_2(ID_u^* \| R^* \| h_1(d_s))$. S verifies if $Auth_a^* = Auth_a$. If not, the process aborts. S generates a random number $r_s \in Z_n^*$ and computes $R_s = r_sP$, $SK_s = r_sR_a$, and $Auth_s = h_2(ID_u \| R^* \| SK_s)$. S sends $M_2 = \{Auth_s, R_s\}$ to $Æ$.

Step 3: $Æ$ computes $SK_a = r_aR_s$, and $Auth_s^* = h_2(ID_u \| R \| SK_a)$. U verifies if $Auth_s^* = Auth_s$. If not, the process aborts. $Æ$ further computes the session key $SK = kdf(ID_u \| SK_a)$ and $Auth_{as} = h_2(R \| SK_a)$, and sends $M_3 = \{Auth_{as}\}$ to S.

Step 4: S computes $Auth_{as}^* = h_2(R^* \| SK_a)$ and verifies if $Auth_{as}^* = Auth_{as}$. If yes, S computes the session key $SK = kdf(ID_u \| SK_s)$ and allows the further communication.

Since no password required to compute the session key and confirmation message, $Æ$ can successfully compute the subsequent valid response messages and establish a session on behalf of U as elucidated above.

3.3 Prone to session-specific ephemeral secret leakage attacks

Assume that $Æ$ intercepts the massages $M_1 = \{Auth_u, CID_u, R_u\}$, $M_2 = \{Auth_s, R_s\}$, and $M_3 = \{Auth_{us}\}$ transmitted between U and S. Let session ephemeral secret r_u' of U is revealed to $Æ$. $Æ$ now can launch offline identity attacks using these parameters as follows:

Step 1: $Æ$ computes $R_a = r_aP$, $R = r_aQ_s$.

Step 2: $Æ$ guesses ID_u^* and then compare $Auth_s$?= $h_2(ID_u^* \| R \| SK_u)$. Repeat the step until correct match is found.

Remark 1 Once identity of U is known to $Æ$ using the above steps, $Æ$ can also compute the secret parameter $h_1(d_s) = CID_u \oplus ID_u^*$, which is static and commonly shared to all registered users. Consequently, $Æ$ can impersonate all the users. $Æ$ then establish a session with S without U's password as follows:

Step 1: $Æ$ computes $R_u' = r_u'P$ and then compare R_u' with R_u in intercepted message M_1. If both matches, $Æ$ confirms that r_u' is corresponding to M_1 and $R_u' = R_u$.

Step 2: $Æ$ computes $R_a = r_aP$, and replays $M_1 = \{Auth_u' CID_u' R_u\}$ to S. Note that $Æ$does not know the credentials of U.

Step 3: S computes $ID_u^* = CID_u \oplus h_1(d_s)$, $R^* = d_sR_u$, and $Auth_u^* = h_2(ID_u^* \| R^* \| h_1(d_s))$. S verifies if $Auth_u^* = Auth_u$. If not, the process aborts. S generates a random number $r_s \in Z_n^*$ and computes $R_s = r_sP$, $SK_s = r_sR_u$, and $Auth_s = h_2(ID_u^* \| R^* \| SK_s)$. S sends $M_2 = \{Auth_s, R_s\}$ to $Æ$.

Step 4: $Æ$ computes $SK_a = r_uR_s$, and $Auth_s^* = h_2(ID_u^* \| R \| SK_a)$. U verifies if $Auth_s^* = Auth_s$. If not, the process aborts. $Æ$ further computes the session key $SK = kdf(ID_u^* \| SK_a)$ and $Auth_{as} = h_2(R \| SK_a)$, and sends $M_3 = \{Auth_{as}\}$ to S.

Step 5: S computes $Auth_{as}^* = h_2(R^* \| SK_s)$ and verifies if $Auth_{as}^* = Auth_{as}$. If yes, S computes the session key $SK = kdf(ID_u^* \| SK_s)$ and allows the further communication.

3.4 Prone to insider attacks

Qi et al.'s protocol is susceptible to the insider attacks as the plain credentials of the users $\{ID_u, PW_u\}$ are shared with the server during the registration phase. In addition, their scheme does not offer user revocation and re-registration phases.

4 Revisiting Lu et al.'s protocol

This section revisits Lu et al.'s (2016) two-factor authenticated key-agreement protocol for mobile client–server architecture. Their scheme comprises of three phases, namely, user registration phase (over a secure channel), login and mutually authenticated key-agreement phase (over a public channel), and password changing phase (over a secure channel). Here, we discuss only the first two phases as the drawbacks lies in these phases.

4.1 User registration phase

Step 1: User U_i chooses an identity ID_a, password PW_a and computes $PWD = h(PW_a \| r_a)$, where $h(.)$ is a one-way hash function. Then U_i sends a registration request $< ID_a, PWD >$ to the server S.

Step 2: S computes $W = h((ID_a)^{rs} \bmod q) \oplus PWD$, $U = x \oplus r_s$, $X = ID_a \oplus r_s$, where r_s is a random nonce. Then S stores the parameters $\{W, U, X, p, q, h(.)\}$ on a smartcard SC that will be delivered to U_i.

Step 3: U_i computes $L = h(h(ID_a \| h(PW_a \| r_a)) \bmod n)$, and adds L and r_a to the SC. Thus, the SC contains $\{W, L, r_a, U, X, p, q, h(.)\}$.

4.2 Login and mutually authenticated key-agreement phase

In this phase, U_i and S can authenticate each other and make a session key as shown in Fig. 2.

Step 1: U_i inserts the SC and enters his/her ID_a, PW_a and checks the login condition $L = h(h(ID_a \| PWD) \bmod n)$. If it generates positive result, SC computes $M = (g^{\alpha} \, mod \, q) \oplus W \oplus h(PW_a \| r_a)$, $C = h((g^{\alpha} \, mod \, q) \| ID_a \| T_a)$ and transmits login request $< M, C, U, X, T_a >$ to S, where α is a random nonce and T_a is current timestamp generated by U_i.

Step 2: S checks the received T_a if it is valid. S then computes $r_s = x \oplus U$, $ID_a = X \oplus r_s$ and $g^{\alpha} = M \oplus h((ID_a)^{rs} \bmod q)$ and verifies the condition $C = h((g^{\alpha} \, mod \, q) \| ID_a \| T_a)$. If it holds, S computes $sk = (g^{\alpha})^{\beta}$, $N = h((ID_a)^{rs} \bmod q) \oplus g^{\beta}$, $D = h(sk \| ID_a \| T_s \| h((ID_a)^{rs} \bmod q))$, where β is a random number. S sends the response $< N, D, T_s >$ to U_i.

Step 3: U_i checks the validity of T_s and derives $g^{\beta} = W \oplus N \oplus h(PW_a \| r_a)$. U_i computes the session key $sk = (g^{\beta})^{\alpha}$ and verifies the condition $D = h(sk \| ID_a \| T_s \| h((ID_a)^{rs} \bmod q))$. If it holds, U_i starts communication using the computed session key $sk = (g^{\alpha})^{\beta} = (g^{\beta})^{\alpha}$.

5 Cryptanalysis of Lu et al.'s protocol

This section demonstrates the security drawbacks of Lu et al.'s protocol (2016) such as prone to insider attacks, server impersonation attacks and lack of user anonymity.

5.1 Prone to server impersonation attack

In Lu et al.'s protocol, another legitimate user U_i' with x value turned as an adversary $Æ$ can mimic a legitimate S as exposed here. During the login and session key establishment phase, U_i transmits the login message $< M, C, U, X, T_a >$. Assume $Æ$ captures this message, then he/she can impersonate the S as follows.

Step 1: During the registration phase, all users (U_i') receives the values $\{W, U, X, p, q, h(.)\}$ on a SC. U_i' can extract server's secret key x using his/her ID_a, $U = x \oplus r_s$ and $X = ID_a \oplus r_s$ values as shown below:

$$U \oplus X \oplus ID_a = x \oplus r_s \oplus ID_a \oplus r_s \oplus ID_a = x.$$

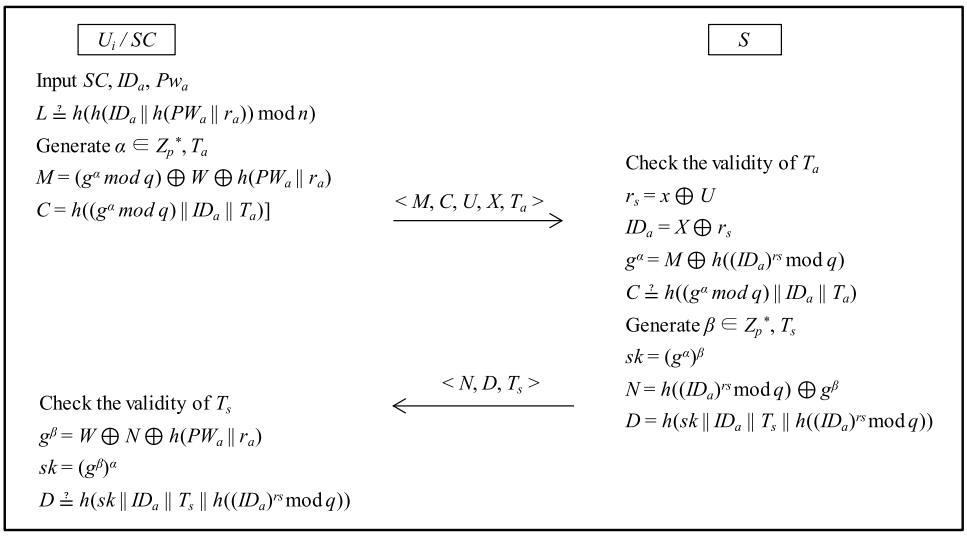

Fig. 2 Login and authenticated key-agreement phase of Lu et al.'s protocol

Step 2: If U_i' turns as an adversary *Æ*, *Æ* can derive r_s, ID_a, and g^{α} values from $<M, C, U, X, T_a>$ by computing $r_s = x \oplus U$, $ID_a = X \oplus r_s$ and $g^{\alpha} = h((ID_a)^{rs} \bmod q)$.

Step 3: *Æ* chooses a random number β and computes $sk = (g^{\alpha})^{\beta}$, $N = h((ID_a)^{rs} \bmod q) \oplus g^{\beta}$, $D = h(sk \| ID_a \| T_s \| h((ID_a)^{rs} \bmod q))$. *Æ* sends $<N, D, T_s>$ to U_i.

Step 4: *A* checks the validity of T_s and derives $g^{\beta} = W \oplus N \oplus h(PW_a \| r_a)$. Then, U_i computes the session key $sk = (g^{\beta})^{\alpha}$ and verifies the condition $D \stackrel{?}{=} h(sk \| ID_a \| T_s \| h((ID_a)^{rs} \bmod q))$. U_i treats *Æ* as legitimate *S* since the condition *D* holds.

5.2 Lack of user anonymity

During the login phase, U_i transmits $<M, C, U, X, T_a>$ to *S* via a public channel. The parameters $U = x \oplus r_s$ and $X = ID_a \oplus r_s$ in the message $<M, C, U, X, T_a>$ are unique and static during all logins for each user. This results in failure of providing perfect user anonymity that could eventually lead to trace attacks.

5.3 Prone to privileged insider attack

Consider a consequence where an insider of the server *S* acts as an adversary and knows the registration information ID_a and *PWD* of a valid U_i. Further, assume that the same adversary possesses the extracted information $\{W, r_a, U, X, L, p, q, h(.)\}$ from the stolen smartcard of U_i. Now, the adversary *Æ* applies the offline-password guessing attack to guess correctly the user's low-entropy PW_a as follows.

Step 1: *Æ* guesses a password PW_a' and computes $PWD' = h(PW_a \| r_a)$.

Step 2: *Æ* checks if $PWD' = PWD$. If it is valid, *Æ* is successful to guess the correct password PW_a' $(= PW_a)$. Otherwise, *Æ* repeats from Step 1 until he/she gets success.

Thus, a privileged-insider being an adversary at the server *S* can guess the correct PW_a of a legal U_i.

Table 1 Notations of the protocols

U_i	An ith user
SC	Smartcard
S	Server
ID_U	Identity of U_i
PW_U	Password of U_i
BIO_U	Biometrics of U_i
r, r_u, α	Random numbers chosen by U_i
ID_S	Identity of *S*
k	Secret key of *S*
n, n', β	Random numbers chosen by *S*
T_u, T_u', T_s	Timestamps generated by U_i and *S*
T	Maximum transmission delay
SK	Session key
Æ	An adversary
P	Base point on elliptic curve
$Gen(BIO_U)$	Generation function of biometric keys
$Rep(BIO_U)$	Reproduction function of biometric keys
$h(.)$	A secure one-way hash function
$\|\|$	The concatenation operation
$\oplus$	An exclusive-OR operation

Table 2 Notations of the BAN logic

$Q \mid\equiv X$	Principal *Q* believes the statement *X*
$Q \mid X$	Principal *Q* has jurisdiction over the statement *X*
$\#(X)$	Formula *X* is fresh
$Q \mid\sim X$	Principal *Q* once said the statement *X*
$Q \lhd X$	Principal *Q* sees the statement *X*
$\langle P \rangle_Q$	Formula *P* combined with the formula *Q*
$Q \overset{K}{\longleftrightarrow} R$	Principal *Q* and *R* may use the shared key *K*
(X, Y)	Formula *X* or *Y* is one part of the formula (X, Y)

6 The proposed protocol

This section puts forward a three-factor mutually authenticated key agreement protocol for client–server architecture based on elliptic curve cryptography (ECC). The proposed protocol comprises two participants and four phases. The notations used in the proposed protocol are listed in Table 1. To provide strong biometric verification locally, we apply the fuzzy extractor method (Dodis et al. 2004), which is composed of the following two functions:

- *Gen*: It is a probabilistic generation function in nature, which takes user biometrics BIO_U as inputs and then results a pair (σ_U, θ_U) as biometric secret key and public reproduction parameter, that is, $(\sigma_U, \theta_U) = Gen(BIO_U)$.
- *Rep*: It is deterministic function in nature, which has the ability to reproduce the original biometric key σ_U from the user biometrics BIO_U' and θ_U, that is, $\sigma_U = \mathrm{Rep}(BIO_U', \theta_U)$ provided that the Hamming distance between BIO_U' and BIO_U is less than or equal to a predefined error tolerance threshold value (Dodis et al. 2004).

To protect the replay attack against an adversary, we apply the current timestamp along with the random nonce. For this issue, we assume that the entities are synchronized with their clocks as it is a reasonable assumption used in designing other authentication protocols (Wazid et al. 2017; Das et al. 2017; Roy et al. 2017a, b, 2016; Chang and Le 2016).

6.1 User registration phase

A new user U_i registers at the server S and obtains a smart-card as shown in Fig. 3.

Step 1: U_i chooses ID_U, PW_U, r, r_u and scans BIO_U, and then computes $RPW = h(PW_U \| r_u) \oplus r$. U_i sends a registration request message $< ID_U, RPW >$ to S via a secure channel.

Step 2: S computes $MID_U = E_k(ID_U \| n)$, $M = h(ID_U \| ID_S \| k)$, and $N = M \oplus RPW$.

Step 3: U_i receives a SC with the parameters $\{MID_U, N, P, h(.)\}$ from S through a secure channel.

Step 4: U_i computes $(\sigma_U, \theta_U) = Gen(BIO_U)$, $E = r_u \oplus h(\sigma_U)$, $N' = N \oplus r$, $G = h(ID_U \| h(PW_U \| r_u))$ and then stores $\{E, N', G\}$ on the received SC after deleting N from the SC. SC now holds the parameters $\{MID_U, E, N', G, P, \theta_U, h(.)\}$.

6.2 Login and authenticated key agreement phase

This phase allows U_i and S to mutually authenticate and make a session key for subsequent communication through public channel as illustrated in Fig. 4.

Step 1: $SC \rightarrow S$: $\text{Msg}_1 = < MID_U, \alpha P, C, T_u >$. SC computes $\sigma_U = Rep(BIO_U, \theta_U)$, $r_u = h(\sigma_U) \oplus E$ and verifies the condition $G_U \stackrel{?}{=} h(ID_U \| h(PW_U \| r_u))$. If it produces positive result, the SC generates α, T_u and calculates $M = N' \oplus h(PW_U \| r_u)$, αP, $C = h(MID_U \| \alpha P \| M \| T_u)$. SC sends the login request message $< MID_U, \alpha P, C, T_u >$ to S via public channel.

Step 2: Upon receiving the login request, S first verifies the received T_u by the condition $T_u^* - T_u < \Delta T$, where T_u^* is time when the message is received by S. If it is valid, S then decrypts MID_U and obtains ID_U of U_i. S computes $M = h(ID_U \| ID_S \| k)$ and verifies the condition $C = h(MID_U \| \alpha P \| M \|$

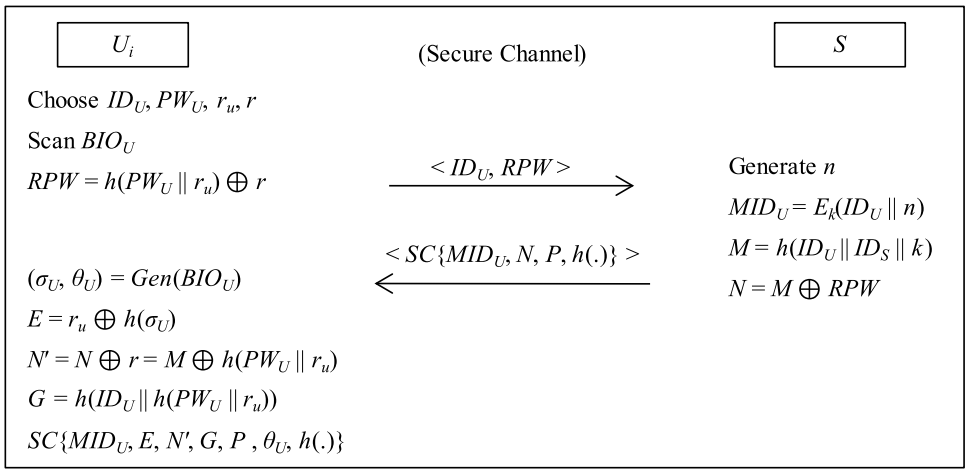

Fig. 3 Summary of user registration phase

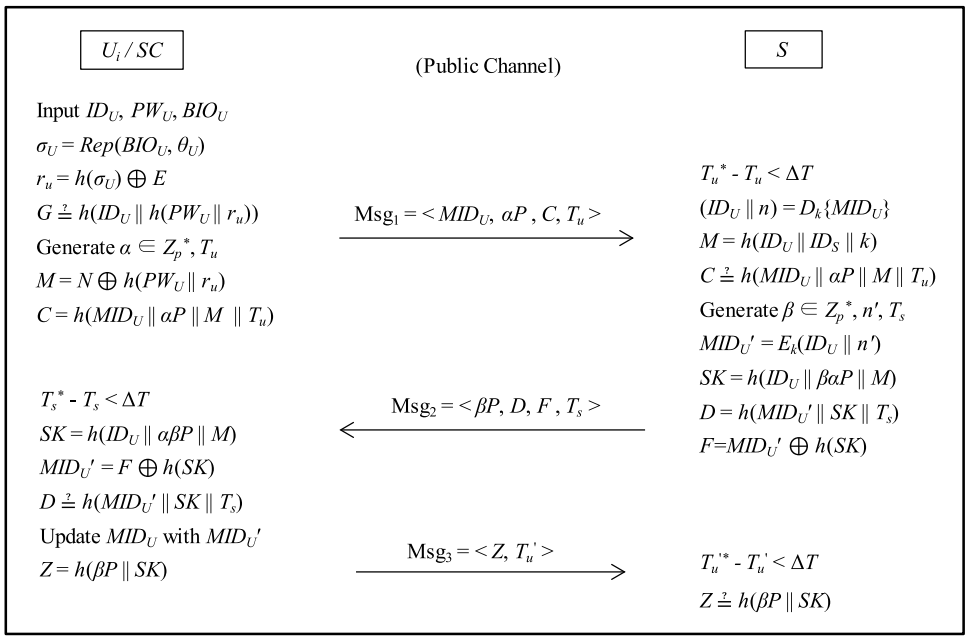

Fig. 4 Summary of mutual authentication and key-agreement phase

T_u). S authenticates U_i only if the condition holds. Else, the process can be terminated.

Step 3: $S \rightarrow SC$: $\text{Msg}_2 = <\beta P, D, F, T_s>$. S further generates β, T_s and computes $MID_U' = E_k(ID_U \| n')$, $SK = h(ID_U \| \beta\alpha P \| M)$, $D = h(MID_U' \| SK \| T_s)$, $F = MID_U' \oplus h(SK)$. S sends $<\beta P, D, F, T_s>$ to SC via public channel.

Step 4: SC first verifies the received T_s by the condition $T_s^* - T_s < \Delta T$, where T_s^* is time when the message is received by U_i. If it holds, SC computes $SK = h(ID_U \| \alpha\beta P \| M)$, $MID_U' = F \oplus h(SK)$and verifies the condition $D = h(MID_U' \| SK \| T_s)$. If the condition holds, U_i authenticates S and updates MID_U with MID_U'; else, the process can be dropped.

Step 5: $SC \rightarrow S$: $\text{Msg}_3 = <Z, T_u'>$. SC generates the current timestamp T_u', computes $Z = h(\beta P \| SK)$, and sends the message $<Z, T_u'>$ to S. S first verifies the received $T_u'$by the condition $T_u'^* - T_u' < \Delta T$, where $T_u'^*$is time when the message is received by S. If it is valid, S then verifies $Z = h(\beta P \| SK)$ and reconfirms the authenticity of U_i, if the condition holds.

6.3 Password and biometrics update phase

In this phase, U_i can update his/her existing PW_U and BIO_U without involvement of S as follows.

Step 1: U_i inserts SC and passes ID_U, PW_U and BIO_U. SC computes $\sigma_U = Rep(BIO_U, \theta_U)$, $r_u = h(\sigma_U) \oplus E$ and verifies $G \stackrel{?}{=} h(ID_U \| h(PW_U \| r_u))$. If the condition holds, U_i can change existing PW_U and BIO_U; otherwise, the request can be rejected.

Step 2: U_i passes new $BIO_U^{\#}$ and $PW_U^{\#}$, and computes $M = N'' \oplus \text{h}(PW_U \| r_u)$, $N^{\#} = M \oplus h(PW_U^{\#} \| r_u)$, $(\sigma_U^{\#}, \theta_U^{\#}) = Gen(BIO_U^{\#})$, $E^{\#} = r_u \oplus h(\sigma_U^{\#})$,and $G^{\#} = h(ID_U \| h(PW_U^{\#} \| r_u))$.

Step 3: U_i replaces N', E and G_U with $N^{\#}$, $E^{\#}$ and $G_U^{\#}$ on the SC, respectively. SC now holds the updated information $\{MID_U, E^{\#}, N'^{\#}, G^{\#}, P, \theta_U^{\#}, h(.)\}$.

6.4 User revocation/re-registration phase

Users can revoke or re-register when their smartcard is lost by proving their authenticity in following way.

Step 1: U_i chooses revoke/re-register option and sends existing ID_U^{prev} to S via a secure channel.

Step 2: S verifies ID_U^{prev} existence and replies U_i asking the new credentials.

Step 3: U_i chooses ID_U, PW_U, r, r_u and scans BIO_U, and then computes $RPW = h(PW_U \| r_u) \oplus r$. U_i sends a registration request message $<ID_U, RPW>$ to S via a secure channel.

Step 4: S computes $MID_U = E_k(ID_U \| n)$, $M = h(ID_U \| ID_S \| k)$, and $N = M \oplus RPW$.

Step 5: U_i receives a SC with the parameters $\{MID_U, N, P, h(.)\}$ from S through a secure channel.

Step 6: U_i computes $(\sigma_U, \theta_U) = Gen(BIO_U)$, $E = r_u \oplus h(\sigma_U)$, $N' = N \oplus r$, $G = h(ID_U \| h(PW_U \| r_u))$ and then stores $\{E, N', G\}$ on the received SC after deleting N from the SC. SC now holds the parameters $\{MID_U, E, N', G, P, \theta_U, h(.)\}$.

7 Security analysis

This section demonstrates the mutual authentication using BAN logic, formal and informal security analyses of the proposed authentication protocol. The proposed scheme is shown that a user U_i and the server S mutually authenticate among each with the help of the widely-used Burrows–Abadi–Needham logic (BAN logic) (Burrows et al. 1990) (see Sect. 7.2). We then prove the session key security (SK-security) of the proposed scheme under the broadly-accepted Real-Or-Random (ROR) model (Abdalla et al. 2005) (see Sect. 7.1). Moreover, the proposed scheme is shown to be secure against various other known attacks informally (see Sect. 7.3). In addition, the formal security verification using the broadly-accepted AVISPA tool (AVISPA Team 2006) assures that the scheme is secure against replay and man-in-the-middle attacks (see Sect. 7.4). Wang et al. (Wang et al. 2015a, b) observed the following interesting point: the widely used formal methods (for example, random oracle model, BAN logic) cannot always capture some structural mistakes, and therefore, assuring soundness of authentication protocols still remains an open issue. Due to this, we need the security analysis informally as well as formal security verification using AVISPA tool to ensure that the proposed scheme can be made more secure with high probability.

7.1 Formal security analysis using Real-OR-Random (ROR) model

In this section, through the widely-accepted Real-Or-Random (ROR) model (Abdalla et al. 2005), we prove the session key *(SK)* security of the proposed scheme. Recently, the ROR model based formal security analysis has drawn much attention in analyzing the formal security in many authentication protocols (Wazid et al. 2017; Das et al. 2017; Roy et al. 2017a, b, 2016; Chang and Le 2016).

7.1.1 ROR model

Two participants, namely user U_i and the server S are associated with the proposed scheme. We have the following components associated with the ROR model (Abdalla et al. 2005).

- *Participants* The instances t_1 and t_2 of U_i and S are denoted by $\pi_U^{t_1}$ and $\pi_S^{t_2}$, respectively, which are also termed as oracles (Chang and Le 2016).
- *Accepted state* If an instance π^t makes transition to an accept state after receiving the last expected protocol message, it is said to be in accepted state. The session identification (*sid*) of π^t for present session is constituted by the ordered concatenation of all communicated (sent and received) messages by π^t.
- *Partnering* Let π^{t_1} and π^{t_2} are two instances. They are partners to each other when the following three criteria are simultaneously fulfilled: (1) π^{t_1} and π^{t_2} in accepted state; (2) both π^{t_1} and π^{t_2} mutually authenticate each other, and share the same *sid*; and (3) both π^{t_1} and π^{t_2} are mutual partners.
- *Freshness* If the session key *SK* between U_i and S is not divulged to an adversary A with the help of the following defined reveal oracle query Reveal (π^t), $\pi_U^{t_1}$or $\pi_S^{t_2}$ is said to be fresh.
- *Adversary* Under the ROR model, the adversary A cannot only read the transmitted messages, but also can modify, delete or change the message contents during the communication. In other words, A is allowed to have full control over the communication. Moreover, A will have access to the following queries (Chang and Le 2016):

– *Execute* (π^{t_1}, π^{t_2}) With the help of this query, the transmitted messages between the valid parties U_i and S are intercepted by A. It is modeled as an eavesdropping attack.
– *Reveal* (π^t) This query allows A to compromise the present session key SK_{ij} created by π^t (and its partner).
– *Send* (π^t, m) This query helps a participant instance π^t to transmit a message m and also receives a message, which is further modeled as an active attack.
– *CorruptSmartcard* ($\pi_U^{t_1}$) It implements the smart card *SC* lost/stolen attack. With the help of this query, the secret credentials stored in *SC* are revealed to A.
– *Test* (π^t) The semantic security of session key *SK* between U_i and S following the indistinguishability in the ROR model (Abdalla et al. 2005) is implemented under this query. At first, an unbiased coin c is flipped prior to beginning of the game, whose output result is only secret to A. This value is later used to verify whether the output of the *Test* query is consistent. If A executes this query and it is found that the session key *SK* is fresh, π^t delivers *SK* when $c=1$ or a random number when $c=0$; otherwise, it delivers $\perp$ (null).

A restriction for A is imposed here in order to acquire only limited number of *CorruptSmartcard* ($\pi_U^{t_1}$) queries. However, A is permitted to acquire *Test* (π^t) query as many times as he/she can have.

- *Semantic security of session key* The ROR model (Abdalla et al. 2005) demands that the adversary A requires to distinguish between an instance's actual session key and a random secret key. A can make the *Test* queries to either $\pi_U^{t_1}$ or $\pi_S^{t_2}$ and its output is checked for consistency against the random bot c. After the game is completed, A judges a guessed bit c' for winning purpose. A wins the game when $c'=c$. The advantage $Adv_{(A)}^{AKAP}$ of A in breaking the semantic security of the proposed authenticated key agreement protocol, say *AKAP* for deriving the session key *SK* between U_i and S is defined by $Adv_{(A)}^{AKAP} = |2.Pr[Succ] - 1|$, where *Succ* represents an event that A can win the game.
- *Random oracle* The communicating entities U_i and S along with A will have access to a collision resistant one-way cryptographic hash function $h(\cdot)$, which is further modeled by a random oracle, say H.

7.1.2 Security proof

To prove the semantic security of the proposed scheme, we first define collision-resistant one-way hash function $h(.)$ and the Elliptic Curve Decisional Diffie-Hellman Problem (ECDDHP). We then provide the proof in Theorem 1.

Definition 1 (*Collision-resistant one-way hash function*) A collision-resistant one-way hash function h: $\{0,1\}^* \rightarrow \{0,1\}^n$ is a deterministic mathematical function that takes a variable length input string and produces a fixed length output string of n bits. If $Adv_{(A)}^{HASH}(rt)$ denote the advantage of an adversary A in finding a hash collision,

$$Adv_{(A)}^{HASH}(rt) = Pr[(i_1, i_2) \in_R A : i_1 \neq i_2, h(i_1) = h(i_2)],$$

where the probability of a random event X is denoted by $Pr[X]$, and the pair $(i_1, i_2) \in_R A$ means the input strings i_1 and i_2 are generated randomly by A. An (ψ, rt)-adversary A attacking the collision resistance of $h(.)$ means the runtime of A will be at most rt and that $Adv_{(A)}^{HASH}(rt) \leqslant \psi$.

An elliptic curve $y^2 = x^3 + ax + b$ over the finite field $GF(p)$ is the set $E_p(a, b)$, which contains the solutions $(x, y) \in Z_p \times Z_p$ to the congruence $y^2 \equiv x^3 + ax + b \pmod p$, where p being a large prime and a, b $\in Z_p$ are two constants, together with a special point O, called the point at infinity or zero point, and $Z_p = \{0, 1, \ldots, p-1\}$. $E_p(a, b)$ is called non-singular if the condition $4a^3 + 27b^2 \neq 0 \pmod p$ is satisfied. The

elliptic curve decisional Diffie-Hellman problem (ECDDHP) is defined as follows.

Definition 2 (*Elliptic curve decisional Diffie-Hellman problem (ECDDHP)*) Let $P \in E_p(a, b)$ be a point on $E_p(a, b)$. The ECDDHP states that given a quadruple $(P, k_1.P, k_2.P, k_3.P)$, to decide whether $k_3 = k_1k_2$ or a uniform value.

Theorem 1 *Suppose A is an adversary running in polynomial time t against the proposed scheme AKAP in the ROR model. If PD is a uniformly distributed password dictionary, l is the number of bits in the biometrics key* σ_U $Adv_{(A)}^{ECDDHP}(t)$ *is the advantage of breaking the ECDDHP in time t by A, and* q_h, q_{send}, $|Hash|$ *and* $|PD|$ *are respectively the number of H queries, send queries, range space of h(.) and the size of PD, A's advantage in breaking semantic security of the proposed scheme AKAP for deriving the session key SK between* U_i *and S in time t is given by*

$$Adv_A^{AKAP}(t) \leqslant \frac{q_h^2}{|Hash|} + \frac{q_{send}}{2^{l-1}.|PD|} + 2Adv_A^{ECDDHP}(t)$$

Proof We follow the similar proof as executed in other authentication protocols (Wazid et al. 2017; Das et al. 2017; Chang and Le 2016). A sequence of five games, say Gam_i ($i = 0, 1, 2, 3, 4$) are essential in this proof in which Succ_i is the winning probability of A in game Gam_i where A can guess the random bit c correctly. The detailed description of all these games is given below.

- Gam_0 It is considered as an actual attack by A against the proposed scheme AKAP in the ROR model. Since the bit c needs to be chosen at the start of Gam_0, it is clear that

$$Adv_A^{AKAP}(t) = \left|2.\Pr\left[Succ_0\right] - 1\right|. \tag{1}$$

- Gam_1 This game is modeled as an eavesdropping attack in which A intercepts the transmitted messages $Msg_1 = <MID_U, \alpha P, C, T_u>$, $Msg_2 = <\beta P, D, F, T_s>$ and $Msg_3 = <Z, T_u^{'}>$ during the mutual authentication and key agreement phase of AKAP. Under this game, A makes Execute (π^{t_1}, π^{t_2}) query. After that A makes the Test query and its result is checked to verify whether it is the real session key SK or a random number. In AKAP, SK is calculated as $SK = h(ID_U \,||\, \alpha\beta P \,||\, M)$, where $M = N \oplus h(PW_U \,||\, r_u) = E_k(ID_U \,||\, n)$. Therefore, computation of SK clearly demands for the leakage of secret credentials ID_U, α,β *and* M, and these credentials are unknown to A. In a nutshell, A's winning the game Gam_1 by eavesdropping of messages is not increased, and thus, we have,

$$\Pr\left[Succ_1\right] = \Pr\left[Succ_0\right]. \tag{2}$$

- Gam_2 The difference between this game and the previous game Gam_1 is that the simulations of the Send and H queries are included in Gam_2. It is therefore treated as an active attack where A can try to fool a legitimate entity to accept an illegal message. Since all the intercepted messages Msg_1, Msg_2 and Msg_3 are constructed using random secrets α, β and timestamps T_u, T_s and $T_u^{'}$, no hash collision occurs when A makes Send query with the help of H query. The birthday paradox results provide the following result:

$$\left|Pr\left[Succ_2\right] - Pr\left[Succ_1\right]\right| \leqslant \frac{q_h^2}{2|Hash|} \tag{3}$$

- Gam_3 The simulation CorruptSmartcard is added into the Gam_3, which differs from Gam_2.*A* then knows the information $\{MID_U, E, N', G, P, \theta_U, h(.)\}$. stored in the smart card SC of U_i. In $AKAP$, a user U_i uses both password PW_U and personal biometrics BIO_U. Due to use of fuzzy extractor, guessing the biometric key $\sigma_U \in \{0, 1\}^l$ from public reproduction parameter θ_U with the help of $Rep(\cdot)$ function has the probability approximately $\frac{1}{2^l}$ (Odelu et al. 2015). Moreover, A can try to guess low-entropy passwords using the password dictionary attacks. If we impose a restriction on the limited number of wrong password inputs in the system by A to guess correct U_i's password PW_U, it then follows that

$$\left|Pr\left[Succ_3\right] - Pr\left[Succ_2\right]\right| \leqslant \frac{q_{send}}{2^l.|PD|} \tag{4}$$

- Gam_4 This is the final game, where A attempts to derive the correct session key SK shared between U_i and S. It is worth noticing that SK is calculated by both the parties U_i and S as $SK = h(ID_U \,||\, \alpha\beta P \,||\, M)$. Now, computation of $\alpha\beta P$ from the eavesdropped αP in Msg_1 and βP in Msg_2 is equivalent to solving the intractable *ECDDHP* in polynomial time t. In addition, A requires secret credentials ID_U and M. Hence, we have,

$$\left|Pr\left[Succ_4\right] - Pr\left[Succ_3\right]\right| \leqslant Adv_A^{ECDDHP}(t). \tag{5}$$

Since all the queries are made by A, it is left only with guessing the bit c to win the game after the Test query is made by A. It follows that.

$$\left|Pr\left[Succ_4\right]\right| = \frac{1}{2} \tag{6}$$

Equations (1) and (2) give the following:

$$\frac{1}{2}Adv_A^{AKAP}(t) = \left|Pr\left[Succ_0\right] - \frac{1}{2}\right| = \left|Pr\left[Succ_1\right] - \frac{1}{2}\right| \quad (7)$$

Equations (6) and (7) give the following:

$$\frac{1}{2}Adv_A^{AKAP}(t) = \left|Pr\left[Succ_1\right] - \frac{1}{2}\right| = \left|Pr\left[Succ_1\right] - Pr\left[Succ_4\right]\right| \quad (8)$$

The triangular inequality gives the following

$$\begin{aligned}\left|Pr\left[Succ_1\right] - Pr\left[Succ_4\right]\right| &\leqslant \left|Pr\left[Succ_1\right] - Pr\left[Succ_2\right]\right| + \left|Pr\left[Succ_2\right] - Pr\left[Succ_4\right]\right| \\ &\leqslant \left|Pr\left[Succ_1\right] - Pr\left[Succ_2\right]\right| + \left|Pr\left[Succ_2\right] - Pr\left[Succ_3\right]\right| \\ &+ \left|Pr\left[Succ_3\right] - Pr\left[Succ_4\right]\right|.\end{aligned} \quad (9)$$

From Eqs. (3), (4), (5) and (9), we get,

$$\left|Pr\left[Succ_1\right] - Pr\left[Succ_4\right]\right| \leqslant \frac{q_h^2}{2|Hash|} + \frac{q_{send}}{2^l.|PD|} + Adv_A^{ECDDHP}(t) \quad (10)$$

Equations (8) and (10) give the following result:

$$\frac{1}{2}Adv_A^{AKAP}(t) \leqslant \frac{q_h^2}{2|Hash|} + \frac{q_{send}}{2^l.|PD|} + Adv_A^{ECDDHP}(t). \quad (11)$$

Finally, multiplying both sides of Eq. (11) by a factor of 2, we obtain the required result:

$$Adv_A^{AKAP}(t) \leq \frac{q_h^2}{|Hash|} + \frac{q_{send}}{2^{l-1} \cdot |PD|} + 2Adv_A^{ECDDHP}(t).$$

7.2 Mutual authentication verification using BAN logic

Mutual authentication between U_i and S of the proposed protocol is proved using widely-accepted Burrows–Abadi–Needham (BAN) logic (Burrows et al. 1990).

Various notations along with their descriptions are provided in Table 2. Following four rules are used to substantiate the mutual authentication between U_i and S in the proposed protocol:

- *Rule 1* Message-meaning rule: $\frac{R \mid\equiv R \overset{Y}{\leftrightarrow} S,\ \ R \lhd <X>_Y}{R \mid\equiv S \mid\sim X}$
- *Rule 2* Nonce-verification rule: $\frac{R \mid\equiv \#(X), R \mid\equiv S \mid\sim X}{R \mid\equiv S \mid\equiv X}$
- *Rule 3* Jurisdiction rule: $\frac{R \mid\equiv S \mid\Rightarrow X,\ \ R \mid\equiv S \mid\equiv X}{R \mid\equiv X}$
- *Rule 4* Freshness-conjuncatenation rule: $\frac{R \mid\equiv \#(X)}{R \mid\equiv \#(X,Y)}$

The following three goals are expected to be achieved to show the mutual authentication between a node R in the cluster C_i and TM:

- $Goal1\ U_i| \equiv U_i \overset{SK}{\leftrightarrow} S$
- $Goal2\ S| \equiv U_i \overset{SK}{\leftrightarrow} S$
- $Goal3\ U_i| \equiv U_i \overset{MID_U{}'}{\leftrightarrow} S$

Generic form The generic forms of the communicated messages between U_i and S in the proposed protocol are specified below:

M1. $U_i \rightarrow S : C = \langle MID_U, \alpha P, T_u \rangle_M$
M2. $S \rightarrow U_i : D = \left\langle MID'_U,\ SK,\ T_u \right\rangle$, where $SK = \mathrm{h}(ID_U, \alpha\beta P, M)$

We rewrite D as $D = \left\langle MID'_U, ID_U,\ \alpha\beta P,\ T_u \right\rangle_M$

M3. $U_i \rightarrow S : Z = \langle \beta P,\ SK \rangle$, where $SK = \mathrm{h}(ID_U, \alpha\beta P, M)$

We rewrite Z as $Z = \langle \beta P,\ \alpha\beta P \rangle_M$.

Idealized form The idealized forms of the communicated messages between U_i and S in the proposed protocol are as follows:

$$\text{M1. } U_i \rightarrow S : \left\langle MID_U, U_i \overset{\alpha P}{\leftrightarrow} S, T_u \right\rangle_{U_i \overset{M}{\leftrightarrow} S}.$$

$$\text{M2. } S \rightarrow \left\langle U_i : U_i \overset{MID'_U}{\leftrightarrow} S,\ ID_U, U_i \overset{\alpha\beta P}{\leftrightarrow} S, T_s \right\rangle_{U_i \overset{M}{\leftrightarrow} S}.$$

$$\text{M3. } U_i \rightarrow S : \left\langle \beta P, ID_U, U_i \overset{\alpha\beta P}{\leftrightarrow} S \right\rangle_{U_i \overset{M}{\leftrightarrow} S}.$$

Hypotheses The initial assumptions of the proposed protocol are as follows:

H1 : $U_i| \equiv \#(T_u),\ \#(T_s)$

H2 : $S| \equiv \#(\beta P),\ \#(T_u)$

H3 : $U_i| \equiv U_i \overset{M}{\leftrightarrow} S$

$$\text{H4}: S| \equiv U_i \overset{M}{\leftrightarrow} S$$

$$\text{H5}: U_i| \equiv S| \Rightarrow U_i \overset{\alpha\beta P}{\leftrightarrow} S$$

$$\text{H6}: S| \equiv U_i| \Rightarrow U_i \overset{\alpha\beta P}{\leftrightarrow} S$$

$$\text{H7}: U_i| \equiv S| \Rightarrow U_i \overset{MID'_U}{\leftrightarrow} S.$$

Note that the identity of user is assumed to be shared between U_i and S.

The following steps proves that the proposed protocol achieves mutual authentication between U_i and S using above hypotheses and rules

- From M1, we have, S1 : $S \lhd \left\langle MID_U, U_i \overset{\alpha P}{\leftrightarrow} S, T_u \right\rangle_{U_i \overset{M}{\leftrightarrow} S}$
- From S1, H4, and Rule 1, we get, S2 : $S| \equiv U_i| \sim \left\langle MID_U, U_i \overset{\alpha P}{\leftrightarrow} S, T_u \right\rangle$
- From S2, H2, Rule 2, and Rule 4, we get, S3 : $S| \equiv U_i| \equiv U_i \overset{\alpha P}{\leftrightarrow} S$
- From M2, we have, S4: $U_i \lhd U_i \overset{MID'_U}{\leftrightarrow} S, ID_U, U_i \overset{\alpha\beta P}{\leftrightarrow} S, T_{s_{U_i \overset{M}{\leftrightarrow} S}}$
- From S4, H3, and Rule 1, we have, S5: $U_i| \equiv S| \sim \left\langle U_i \overset{MID'_U}{\leftrightarrow} S, ID_U, U_i \overset{\alpha\beta P}{\leftrightarrow} S, T_s \right\rangle$
- From S5, H1, Rule 2, and Rule 4, we get, S6: $U_i| \equiv S| \equiv \left\langle U_i \overset{MID'_U}{\leftrightarrow} S, ID_U, U_i \overset{\alpha\beta P}{\leftrightarrow} S \right\rangle$
- From S6, H7 and Rule 3, we obtain, S7: $U_i| \equiv U_i \overset{MID'_U}{\leftrightarrow} S$ (Goal 3)
- From S6, H5 and Rule 3, we obtain, S8: $U_i| \equiv U_i \overset{\alpha\beta P}{\leftrightarrow} S$
- From S8 and H3, we obtain, S9: $U_i| \equiv U_i \overset{SK}{\leftrightarrow} S$ (Goal 1)
  From M3, we have, S10: $S \lhd \left\langle \beta P, ID_U, U_i \overset{\alpha\beta P}{\leftrightarrow} S \right\rangle_{U_i \overset{M}{\leftrightarrow} S}$
- From S10, H4, and Rule 1, we have, S11: $S| \equiv U_i| \sim \left\langle \beta P, ID_U, U_i \overset{\alpha\beta P}{\leftrightarrow} S \right\rangle$
- From S11, H2, Rule 2, and Rule 4, we get S12: $S| \equiv U_i| \equiv U_i \overset{\alpha\beta P}{\leftrightarrow} S$
- From S12, H6, and Rule 3, we get S13: $S| \equiv U_i \overset{\alpha\beta P}{\leftrightarrow} S$
- Finally, from S13 and H4, we obtain, S14: $S| \equiv U_i \overset{SK}{\leftrightarrow} S$ (Goal 2)

The above goals 1–3 clearly indicate the proposed scheme achieves the mutual authentication between U_i and S.

7.3 Informal security analysis

In this section, the security properties satisfied by the proposed protocol in the following propositions.

Proposition 1 *The proposed protocol provides user anonymity and untraceability properties.*

Proof In the proposed protocol, user's real identity ID_U is enciphered with server's secret key k. In order to retrieve ID_U from $MID_U = E_k(ID_U \| n)$, k is essential which is known only to the server. Thus, the real ID_U value is available only with U_i and S. Though the users' identity is anonymous, the chances of tracing the user still exist when other transmitting parameters are static. During the login phase, U_i sends authentication request message $<MID_U, L, C, T_u>$ to S, where $L = \alpha P \oplus M$, $C = h(MID_U \| \alpha P \| M \| T_u)$. All the parameters in the message are dynamic due to the incorporation of random numbers $\alpha$and n. Hence, the proposed protocol offers user anonymity and untraceability.

Proposition 2 *The proposed protocol withstands replay attack.*

Proof Consider a scenario where *Æ* tries to gain access to the system by mitigating a registered user with the previous transmitted message $<MID_U, L, C, T_u>$. The proposed scheme can identify it as a malicious attempt if *Æ* performs this due to the following reasons.

- *Case* 1: If *Æ* replays the message $<MID_U, L, C, T_u>$, S can classify it as a malicious attempt when it finds the condition $T_u^* - T_u < \Delta T$ is not valid. Similarly, U_i can also recognize replay attacks on $<\beta P, D, F, T_s>$ using the condition $T_s^* - T_s < \Delta T$.
- *Case* 2: In the mutual authentication with key-agreement phase, the server S obtains $(ID_U, \alpha P)$ by computing $(ID_U \| n) = D_k\{MID_U\}$, $\alpha P = M \oplus L$, and stores it in its database. Note that α is a randomly generated number and varies for each session. When *Æ* sends the captured message $<MID_U, L, C, T_u>$, S extracts $(ID_U^{\#}, \alpha P^{\#})$ values and compares with the stored values in the database and subsequently drops the request when it notices $(ID_U^{\#}, \alpha P^{\#}) == (ID_U, \alpha P)$.

Proposition 3 *The proposed protocol withstands privileged insider and stolen token attacks.*

Proof Assume that an insider who knows the registration information ID_U and RPW of a valid U_i turns as *Æ*, and reads the stolen smartcard information $\{MID_U, E, N', G, P, \theta_U, h(.)\}$ by using power analysis methods (Kocher et al. 1999; Messerges et al. 2002; Wang and Wang 2015). *Æ* may

now try to obtain some useful information such as credentials of U_i. However, *Æ* cannot succeed due to the reason described here. All the obtained parameters such as MID_U, E, N', G, where $MID_U = E_k(ID_U \| n)$, $(\sigma_U, \theta_U) = Gen(BIO_U)$, $E = r_u \oplus h(\sigma_U)$, $N' = N \oplus r$, $G = h(ID_U \| h(PW_U \| r_u))$ are safeguarded using either one-way has function or symmetric encryption. In order to extract PW_U from $RPW = h(PW_U \| r_u) \oplus r$, *Æ* requires r_u and r values. The r_u and r values are stored on the *SC* in association with biometrics and N values, respectively. It is impossible to obtain r_u and r without passing valid biometrics, moreover, biometrics can neither be stolen nor forged. On the other hand, guessing both r_u and r values simultaneously is impractical. In this way, the proposed protocol can resist privileged insider attacks and stolen smartcard attacks.

Proposition 4 *The proposed protocol withstands password guessing attacks.*

Proof *Online password guessing attacks: Æ* may try to login using the stolen smartcard, while guessing the user's PW_U. In order to perform this, *Æ* requires to satisfy the login condition $G = h(ID_U \| h(PW_U \| r_u))$ which entails r_u and BIO_U. Unless *Æ* passes valid BIO_U; r_u value cannot be extracted from E and consequently leads to failure of satisfying the login condition $G = h(ID_U \| h(PW_U \| r_u))$.

Offline password guessing attacks *Æ* can attempt to guess the password using the extracted parameters $\{MID_U, E, N', G, P, \theta_U, h(.)\}$ from the stolen smartcard. Note that the password PW_U is not stored on the *SC* in the plaintext form, but in $N' = M \oplus h(PW_U \| r_u)$ shielded with one-way hash function. Moreover, *Æ* needs r_u value to verify the guessed password $N' \stackrel{?}{=} M \oplus h(PW_U{}^{Æ} \| r_u)$. However, there are no means to obtain r_u value without valid BIO_U. Thus, the proposed protocol is resistant to offline password guessing attacks.

Proposition 5 *The proposed protocol is resilient against user impersonation attack.*

Proof If *Æ* wants to masquerade U_i, he/she needs to form a login message $< MID_U, L, C >$, where $MID_U = E_k(ID_U \| n)$, $L = \alpha P \oplus M$, $C = h(MID_U \| \alpha P \| M \| T_u)$. Conversely, *Æ* can barely compute $g^{\alpha} \bmod q$, but not L and C due to the unavailability of valid PW_U and BIO_U. In case if *Æ* sends captured message $< MID_U, L, C, T_u >$, then S can easily identify it as a replay attack and would drop the session as elaborated in replay attacks section.

Proposition 6 *The proposed protocol is secure against server impersonation attack.*

Proof If *Æ* wants to masquerade S, he/she needs to construct a valid response message $< \beta P, D, F >$, where $MID_U' = E_k(ID_U \| n')$, $SK = h(ID_U \| \beta\alpha P \| M)$, $F = MID_U' \oplus h(SK)$, $D = h(MID_U' \| SK \| T_s)$. Assume that *Æ* replies the message $< \beta P^{Æ}, D^{Æ}, F^{Æ}, T_s^{Æ} >$, where $MID_U{}'^{Æ} = E_k{}^{Æ}(ID_U \| n')$, $SK^{Æ} = h(ID_U{}^{Æ} \| \alpha^{Æ}\beta^{Æ}P \| M^{Æ})$, $F^{Æ} = MID_U{}'^{Æ} \oplus h(SK^{Æ})$, $D^{Æ} = h(MID_U{}'^{Æ} \| SK^{Æ} \| T_s^{Æ})$. Despite the fact that U_i cannot be able to check the correctness of received $MID_U{}'^{Æ}$ due to unavailability of real server's secret key k, U_i can definitely detect it as a fake response as described here. Upon receiving the message, U_i computes $SK = h(ID_U \| \alpha\beta^{Æ}P \| M)$, $MID_U' = F^{Æ} \oplus h(SK)$, $D = h(MID_U' \| SK \| T_s)$ and verifies whether $D \stackrel{?}{=} D^{Æ}$. It is obvious that the condition cannot hold because *Æ* has computed SK with wrong ID_U and αP, which result in $h(ID_U \| \alpha\beta P \| M) \neq h(ID_U{}^{Æ} \| \alpha\beta^{Æ}P \| M^{Æ})$.

Proposition 7 *The proposed protocol is secure against ephemeral secret leakage (ESL) attack.*

Proof The shared session key between U_i and S during the mutual authentication and key-agreement phase is computed as $SK = h(ID_U \| \beta\alpha P \| M)$, where $M = h(ID_U \| ID_S \| k)$ and k is the secret key of the server S. In the proposed protocol, the session key security (SK-security) depends on the following two cases:

- *Case 1*. Let the ephemeral (short term) secrets alpha and beta are leaked to an adversary A. Even then without having the long-term secrets ID_U, ID_S and k, it is computationally infeasible for *Æ* to calculate SK.
- *Case 2*. Let the long-term secrets ID_U, ID_S and k are revealed to *Æ*. However, without the ephemeral secrets α and β, it is difficult for *Æ* to calculate SK.

In summary, *Æ* can only calculate SK when the ephemeral secrets as well as long-term secrets are known to him/her. It is worth noticing that even if the current SK is revealed to *Æ* in a particular session, all other session keys in previous and future sessions are distinct since both long-term secrets and fresh ephemeral random nonces are applied in the construction of the session keys. Hence, the leakage of a session key will have no effect on the security of other previous and future sessions for secure communications. As a result, the proposed scheme provides forward and backward secrecy, and it also provides the SK-security. Thus, the proposed scheme protects ESL attack.

7.4 Formal security verification using AVISPA tool: simulation study

This section provides a brief overview of the AVISPA tool, the various roles implemented, and the final simulation results of the proposed protocol.

7.4.1 AVISPA overview

AVISPA is a widely-accepted push-button tool for the automated validation of Internet security-sensitive protocols and applications (Armando et al. 2005; AVISPA Team 2006). AVISPA is used to formally verify whether a cryptographic protocol is secure or vulnerable against active and passive attacks including the man-in-the-middle and replay attacks. In AVIPSA, a security protocol is implemented using HLPSL (High Level Protocols Specification Language). HLPSL is translated using HLPSL2IF translator to convert to the intermediate format (IF). IF is fed into one of the four back-ends: Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP), On-the- fly Model-Checker (OFMC), Constraint Logic based Attack Searcher (CL-AtSe), and SAT-based Model-Checker (SATMC). The proposed protocol is simulated under the CL-AtSe and OFMC back-ends using the SPAN (Security Protocol ANimator) for AVISPA (SPAN-Security Protocol Animator for AVISPA

```
role user (Ui, S: agent,
        SKuis: symmetric_key,
        H : hash_func,
        Send, Recv: channel(dy))
played_by Ui
def=
 local State: nat,
      IDu, PWu, Ru, R, RPW, BIO, P, K, IDs: text,
      Alpha, Beta, L, C, Tu, Ts, N, Nn, Z: text,
      F: hash_func
const sk1, sk2, sk3, ui_s_alpha, ui_s_tu,
    s_ui_beta, s_ui_ts: protocol_id
% Initialize state State to 0
init State := 0
transition
% User registration phase
1. State = 0 /\ Recv(start) =|>
% Send registration request to server S securely
 State' := 1 /\ secret({IDu}, sk1, {Ui,S})
   /\ secret({PWu}, sk2, {Ui})
        /\ secret({IDs,K}, sk3, {S})
        /\ Ru' := new() /\ R' := new()
   /\ RPW' := xor(H(PWu.Ru'), R')
   /\ Send({IDu.RPW'}_SKuis)
% Receive smart card from S securely
2. State = 1 /\ Recv({{IDu.N'}_K.xor(H(IDu.IDs.K),
                H(PWu.Ru')).P.H}_SKuis) =|>
% Login phase
State' := 3 /\ Alpha' := new() /\ Tu' := new()
        /\ L' := xor(F(Alpha'.P), H(IDu.IDs.K))
        /\ C' :=
H({IDu.N'}_K.F(Alpha'.P).H(IDu.IDs.K).Tu')
% Send login message to S via public channel
        /\ Send({IDu.N'}_K.L'.C'.Tu')
% Ui has freshly generated the values alpha and Tu for S
        /\ witness (Ui, S, ui_s_alpha, Alpha')
        /\ witness (Ui, S, ui_s_tu, Tu')
% Authentication and key–agreement phase
% Receive authentication message from S via public channel
3. State = 3 /\ Recv(F(Beta'.P).H({IDu.Nn'}_K.H(IDu.
                F(Beta'.F(Alpha'.P)).H(IDu.IDs.K)).Ts').
                xor({IDu.Nn'}_K,H(H(IDu.
                F(Beta'.F(Alpha'.P))))).Ts') =|>
% Send authentication reply message to S via public channel
State' := 5 /\ Z' := H(F(Beta'.P).H(IDu.F(Beta'.F(Alpha'.P)).
                H(IDu.IDs.K)))
        /\ Send(Z')
% Ui's acceptance of the values beta and Ts generated for Ui
by S
        /\ request(S, Ui, s_ui_beta, Beta')
        /\ request(S, Ui, s_ui_ts, Ts')
end role
```

```
role server (Ui, S: agent,
        SKuis: symmetric_key,
        H : hash_func,
        Send, Recv: channel(dy))
played_by S
def=
 local State: nat,
 IDu, PWu, BIO, Ru, R, MIDu, M, N, P, K, IDs: text,
 Alpha, Beta, SK, Tu, Ts, Nn, D, F1, N1: text,
 F: hash_func
const sk1, sk2, sk3, ui_s_alpha, ui_s_tu,
    s_ui_beta, s_ui_ts: protocol_id
% Initialize state State to 0
init State := 0
transition
% User registration phase
% Receive registration request from user Ui securely
1. State = 0 /\ Recv({IDu.xor(H(PWu.Ru'), R')}_SKuis) =|>
 State' := 2 /\ secret({IDu}, sk1, {Ui,S})
   /\ secret({PWu}, sk2, {Ui})
        /\ secret({IDs,K}, sk3, {S})
        /\ N' := new()
        /\ MIDu' := {IDu.N'}_K
        /\ M' := H(IDu.IDs.K)
        /\ N1' := xor(M',H(PWu.Ru'))
% Send smart card to Ui securely
        /\ Send({MIDu'.N1'.P.H}_SKuis)
% Login phase
% Receive login message from Ui via public channel
2. State = 2 /\ Recv({IDu.N'}_K.xor(F(Alpha'.P),
H(IDu.IDs.K)).
            H({IDu.N'}_K.F(Alpha'.P).H(IDu.IDs.K).Tu').Tu')
=|>
% Authentication and key agreement phase
State' := 4 /\ Ts' := new() /\ Beta' := new() /\ Nn' := new()
        /\ SK' := H(IDu.F(Beta'.F(Alpha'.P)).H(IDu.IDs.K))
        /\ D' := H({IDu.Nn'}_K.SK'.Ts')
        /\ F1' := xor({IDu.Nn'}_K,H(SK'))
% Send authentication message to Ui via public channel
        /\ Send(F(Beta'.P).D'.F1'.Ts')
% S has freshly generated the values beta and Ts for Ui
        /\ witness (S, Ui, s_ui_beta, Beta')
        /\ witness (S, Ui, s_ui_ts, Ts')
% Receive authentication reply message from Ui via public
channel
3. State = 4 /\ Recv(H(F(Beta'.P).H(IDu.F(Beta'.F(Alpha'.P)).
                H(IDu.IDs.K)))) =|>
% S's acceptance of alpha and Tu generated for S by Ui
State' := 6 /\ request(Ui, S, ui_s_alpha, Alpha')
        /\ request(Ui, S, ui_s_tu, Tu')
end role
```

Fig. 5 Role specification of *User* and *Server*

```
role session(Ui, S: agent,
SKuis: symmetric_key,
H : hash_func)
def=
local TX1, RX1, TX2, RX2: channel (dy)
composition
user(Ui, S, SKuis, H, TX1, RX1)
/\ server(Ui, S, SKuis, H, TX2, RX2)
end role
role environment()
def=
const ui, s : agent,
skuis: symmetric_key,
h, f : hash_func,
p, tu, ts : text,
sk1, sk2, sk3, ui_s_alpha, ui_s_tu,
s_ui_beta, s_ui_ts: protocol_id
intruder_knowledge = {h, f, tu, ts, p}
composition
session(ui, s, skuis, h)
/\ session(i, s, skuis, h)
/\ session(ui, i, skuis, h)
end role
goal
secrecy_of sk1, sk2, sk3
authentication_on ui_s_alpha, ui_s_tu
authentication_on s_ui_beta, s_ui_ts
end goal
environment()
```

Fig. 6 Role specification of session, goal, and environment

2016). Both back-ends are chosen for an execution test and a bounded number of sessions model checking (Basin et al. 2005).

7.4.2 Various roles implemented in HLPSL

The implementation details of the roles of user, server, session, and goal and environment are performed as evident in the Figs. 5 and 6, and provided the final results in Fig. 7.

The HLPSL specification of the basic role of user U_i is provided in Fig. 5. In this role, after receiving the start signal, it sends the registration request to the server for registration purpose, and updates its state (maintained by the variable State) from 0 to 1. Once U_i receives the smart card SC from the server S, it also changes its state from 1 to 3. During the mutual authentication and key agreement phase, U_i sends the login request to S. U_i declares a witness of freshly generated random number and timestamp T_u by the declarations

```
% OFMC
% Version of 2006/02/13
SUMMARY
SAFE
DETAILS
BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
C:\progra~1\SPAN\testsuite
\results\auth.if
GOAL
as_specified
BACKEND
OFMC
COMMENTS
STATISTICS
parseTime: 0.00s
searchTime: 0.04s
visitedNodes: 16 nodes
depth: 4 piles
```

Fig. 7 The result of the analysis using OFMC backend

witness(Ui, S, ui_s_alpha, Alpha') and witness(Ui, S, ui_s_tu, Tu'). After that U_i receives the authentication request from S and finally, Ui also sends authentication reply to S. At the end, U_i authenticates S based on the random number and timestamp T_s by the declarations *request(S, Ui, s_ui_beta, Beta') and request(S, Ui, s_ui_ts, Ts')*. Similarly, the HLPSL role specification of the sever S is also defined in Fig. 5. The roles of session, goal and environment are mandatory roles in AVISPA. In these roles, the secrecy and authentication goals need to be specified in order to check whether the protocol is safe or unsafe.

7.4.3 Analysis of simulation results

The proposed protocol is verified in following aspects as described in (Lv et al. 2013; Reddy et al. 2016):

- *Executabiltiy check* Here, unexpected modeling errors may sometimes cause the incomplete execution of protocol. Accordingly, the OFMC may not find an attack when the model is not able to reach the attack happening state. Thus, the executability test is indispensable in AVISPA as described in (AVISPA Team 2006).
- *Replay attack check* OFMC back-end offered information regarding some normal sessions between legitimate parties to the intruder. The simulation result furnished in Fig. 7 ensures that the proposed protocol can withstand replay attacks.
- *Dolev-Yao model check* During this check (Dolev and Yao 1983), the possibilities of man-in-the-middle attacks are also verified by the OFMC. As illustrated in simulation results (Fig. 7), the depth of search is 4, and the total

Table 3 Comparison of security properties with two-factor authentication protocols

Security property	Li et al. (2013)	Chen et al. (2014)	Jiang et al. (2015)	Lu et al. (2016)	Qi et al. (2017)	Our
User anonymity and untraceability	No	No	No	No	No	Yes
Provides perfect mutually authenticated key agreement	Yes	Yes	Yes	Yes	Yes	Yes
Provides login verification	Yes	No	No	Yes	Yes	Yes
Prevents replay attack	Yes	Yes	Yes	Yes	Yes	Yes
Prevents stolen token attack	Yes	Yes	Yes	Yes	Yes	Yes
Prevents impersonation attack	Yes	No	No	No	No	Yes
Prevents insider attack	No	Yes	No	No	No	Yes
Prevents password guessing attack	Yes	Yes	Yes	Yes	Yes	Yes
Prevents clock synchronization problem	No	No	No	No	Yes	No
Prevents ephemeral secret leakage attacks	No	Yes	Yes	Yes	No	Yes

Table 4 Comparison of performance with two-factor authentication protocols

Protocol	Computational cost	Communication rounds	Bandwidth (bits)	Formally analyzed	Deployed method
Li et al. (2013)	U_i:$5T_h + 4T_e + 1T_m$ S:$4T_h + 3T_e$	2	864	No	Modular exponentiation
Chen et al. (2014)	U_i:$5T_h + 2T_e + 2T_m$ S:$4T_h + 1T_e + 1T_m$	2	704	No	Modular exponentiation
Jiang et al. (2015)	U_i:$5T_h + 3T_e + 1T_m$ S:$4T_h + 2T_e$	2	704	No	Modular exponentiation
Lu et al. (2016)	U_i:$5T_h + 2T_e$ S:$4T_h + 3T_e$	2	1024	No	Modular exponentiation
Qi et al. (2017)	U_i:$6T_h + 3T_m$ S:$6T_h + 3T_m$	3	1280	No	ECC
Our	U_i:$1T_b + 7T_h + 2T_m$ S:$6T_h + 2T_m + 2T_f$	3	1536	Yes	ECC

Table 5 Comparison of security properties with three-factor authentication protocols

Security property	Yeh et al. (2013)	Li et al. (2014)	Wu et al. (2015)	Han et al. (2016)	Xie et al. (2017)	Our
User anonymity and untraceability	Yes	No	Yes	No	Yes	Yes
Provides perfect mutually authenticated key agreement	No	Yes	Yes	Yes	Yes	Yes
Provides login verification	Yes	Yes	Yes	Yes	Yes	Yes
Prevents replay attack	Yes	Yes	Yes	Yes	Yes	Yes
Prevents stolen token attack	Yes	Yes	Yes	Yes	Yes	Yes
Prevents impersonation attack	No	Yes	No	Yes	Yes	Yes
Prevents insider attack	Yes	Yes	Yes	Yes	Yes	Yes
Prevents password guessing attack	Yes	Yes	Yes	Yes	Yes	Yes
Prevents clock synchronization problem	Yes	Yes	Yes	No	Yes	No
Prevents ephemeral secret leakage attacks	–	No	No	Yes	No	Yes

Table 6 Comparison of performance with three-factor authentication protocols

Protocol	Computational cost	Communication rounds	Bandwidth (bits)	Formally analyzed	Deployed method
Yeh et al. (2013)	U_i:$1T_b + 2T_h + 2T_m + 6T_a$ S:$6T_a + 3T_m$	3	2240	No	ECC
Li et al. (2014)	U_i:$1T_b + 7T_h + 2T_e$ S:$6T_h + 2T_e$	3	960	No	Modularv exponentiation
Wu et al. (2015)	U_i:$1T_b + 5T_h + 2T_m + 2T_f$ S:$6T_h + 2T_m + 2T_f$	2	1856	Yes	ECC
Han et al. (2016)	U_i:$7T_h + 2T_m$ S:$5T_h + 2T_m + 2T_f$	3	1728	Yes	ECC
Xie et al. (2017)	U_i:$1T_b + 8T_h + 2T_m + 2T_f$ S:$6T_h + 2T_m + 2T_f$	2	1856	Yes	ECC
Our	U_i:$1T_b + 7T_h + 2T_m$ S:$6T_h + 2T_m + 2T_f$	3	1536	Yes	ECC

number of searched nodes is 16 which require 0.04 s. Simulation results also substantiate that the proposed protocol attains the design standards and is secure against replay and man-in-the-middle attacks.

8 Performance analysis

This section summarizes the performance of the proposed protocol in terms of security properties, computational cost, and communication cost. The comparison follows between the proposed protocol and some of the recently published two-factor and three-factor protocols as depicted in the Tables 3, 4, 5 and 6.

To analyze the computational cost, few symbolizations are given for the comprised actions in the existing two-factor and three-factor protocols and the proposed protocol in following way: T_m: time complexity of an elliptic curve point division or multiplication operation; T_a: time complexity of an elliptic curve point addition or subtraction operation; T_f: time complexity of a symmetric key encryption or decryption function; T_h: time complexity of a one-way hash function; T_b: time complexity of a biometrics extraction function; T_e: time complexity of a modular exponentiation operation.

To analyze the communication cost, we consider 32-bits for a timestamp, 160-bits for a random number, 80-bits for a human-memorable identity, 160-bits for each co-ordinate of an elliptic curve point, 160-bit for a modular prime number operation, 160-bits for a SHA-1 output, and 128-bits key for an AES algorithm of cipher block chaining (CBC) mode.

8.1 Comparison with the two-factor authentication protocols

The security properties between Li et al. (2013), Chen et al. (2014), Jiang et al. (2015), Lu et al. (2016), Qi et al. (2017) and the proposed protocol are summarized in Table 3. It is evident from Table 3 that the proposed protocol is secure against all the renowned threats and achieves diverse features. As presented in Table 4, the proposed protocol deploys $1T_b + 13T_h + 4T_m + 2T_f$, whereas Li et al., Chen et al., Jiang et al., and Lu et al., Qi et al. uses $9T_h + 7T_e + 1T_m$, $9T_h + 3T_e + 3T_m$, $9T_h + 5T_e + 1T_m$, $9T_h + 5T_e$, and $12T_h + 4T_m$, respectively. The computational cost of the proposed protocol is reasonably equal or slightly higher compared to other protocols. On other hand, the proposed protocol requires more bandwidth compared to the other protocols. But, it is well worth deploying additional computations and communication cost to afford enhanced security level.

8.2 Comparison with the three-factor authentication protocols

The security properties between Yeh et al. (2013), Li et al. (2014), Wu et al. (2015), Han et al. (2016), Xie et al. (2017) and the proposed protocol are summarized in Table 5. It is evident from Table 5 that the proposed protocol is secure against all the renowned threats and achieves diverse features. As presented in Table 6, the proposed protocol deploys $1T_b + 13T_h + 4T_m + 2T_f$, whereas Yeh et al., Li et al., Wu et al., Han et al., Xie et al. uses $1T_b + 2T_h + 5T_m + 12T_a$, $1T_b + 13T_h + 4T_e$, $1T_b + 11T_h + 4T_m + 4T_f$, $12T_h + 4T_m + 2T_f$, and $1T_b + 14T_h + 4T_m + 4T_f$, respectively. The computational cost of the proposed protocol is lesser than the other protocols. On other hand, the proposed protocol requires less bandwidth compared to Yeh et al., Wu et al., Han et al., Xie et al., and more bandwidth compared to Li et al.

9 Conclusion remarks

This paper aimed to study and analyze some of recently proposed two-factor authentication protocols for client–server architecture and proposed a new solution to overcome the existing pitfalls. We believe that two-factor authentication mechanisms are still vulnerable under various phenomena. Thus, we proposed a three-factor authentication mechanism which improves the security with an additional factor known as biometrics. The proposed three-factor authenticated key agreement protocol is not only secure from numerous attacks but also achieve the eminent security properties such as user anonymity, mutual authentication, biometrics deployment, perfect forward secrecy. The comparisons between the proposed protocol and the other related protocols prove that our protocol is robust and efficient.

Acknowledgements This work was supported by the faculty research fund of the Sejong University in 2017. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions that helped us to improve the presentation and quality of the paper.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interests.

References

Abdalla M, Fouque P, Pointcheval D (2005) Password-based authenticated key exchange in the three-party setting. 8th International Workshop on Theory and Practice in Public Key Cryptography (PKC'05), Les Diablerets, Switzerland, pp. 65–84

An Y (2012) Security analysis and enhancements of an effective biometric-based remote user authentication scheme using smart cards. Biomed Res Int. https://doi.org/10.1155/2012/519723

Armando A, Basin D… Mödersheim S (2005) The AVISPA tool for the automated validation of internet security protocols and applications. In: International Conference on Computer Aided Verification, pp. 281–285

AVISPA Team (2006) AVISPA V1.1 User Manual, [Online]. Available: http://www.avispa-project.org/package/user-manual.pdf Accessed Dec 2015

Basin D, Mödersheim S, Vigano L (2005) OFMC: a symbolic model checker for security protocols. Int J Inf Secur 4(3):181–208

Burrows M, Abadi M, Needham R R (1990) A logic of authentication. ACM Trans Comput Syst 8(1):18–36

Cao L, Ge W (2015) Analysis and improvement of a multi-factor biometric authentication scheme. Secur Commun Netw 8(4):617–625

Chan CK, Cheng LM (2000) Cryptanalysis of a remote user authentication scheme using smart cards. IEEE Trans Consum Electron 46(4):992–993

Chang CC, Le HD (2016) A Provably secure, efficient and flexible authentication scheme for ad hoc wireless sensor networks. IEEE Trans Wireless Commun 15(1):357–366

Chang YF, Tai WL, Chang HC (2014) Untraceable dynamic-identity-based remote user authentication scheme with verifiable password update. Int J Commun Syst 27(11):3430–3440

Chaturvedi A, Mishra D, Jangirala S, Mukhopadhyay S (2017) A privacy preserving biometric-based three-factor remote user authenticated key agreement scheme. J Inf Secur Appl 32:15–26

Chen CL, Lee CC, Hsu CY (2012) Mobile device integration of a fingerprint biometric remote authentication scheme. Int J Commun Syst 25(5):585–597

Chen BL, Kuo WC, Wuu LC (2014) Robust smart-card-based remote user password authentication scheme. Int J Commun Syst 27(2):377–389

Chien HY, Jan JK, Tseng YM (2001) A modified remote login authentication scheme based on geometric approach. J Syst Softw 55(3):287–290

Chou CH, Tsai KY, Lu CF (2013) Two ID-based authenticated schemes with key agreement for mobile environments. J Supercomput 66(2):973–988

Das AK (2011) Analysis and improvement on an efficient biometric-based remote user authentication scheme using smart cards. IET Inf Secur 5(3):145–151

Das AK, Goswami A (2015) A robust anonymous biometric-based remote user authentication scheme using smart cards. J King Saud Univ-Comput Inf Sci 27(2):193–210

Das AK, Wazid M, Kumar N, Khan MK, Choo KKR, Park Y (2017) Design of secure and lightweight authentication protocol for wearable devices environment. IEEE J Biomed Health Inform, https://doi.org/10.1109/JBHI.2017.2753464

Debiao H, Jianhua C, Jin H (2012) An ID-based client authentication with key agreement protocol for mobile client–server environment on ECC with provable security. Inf Fusion 13(3):223–230

Dodis Y, Reyzin L, Smith A (2004) Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. Advances in cryptology-eurocrypt 2004. Interlaken, Springer-Verlag, Berlin, Heidelberg, pp 523–540

Dolev D, Yao A (1983) On the security of public key protocols. IEEE Trans Inf Theory 29(2):198–208

Fan CI, Lin YH (2009) Provably secure remote truly three-factor authentication scheme with privacy protection on biometrics. IEEE Trans Inf Forensics Secur 4(4):933–945

Farash MS (2016) Security analysis and enhancements of an improved authentication for session initiation protocol with provable security. Peer-to-Peer Netw Appl 9(1):82–91

Farash MS, Attari MA (2014) A secure and efficient identity-based authenticated key exchange protocol for mobile client–server networks. J Supercomput 69(1):395–411

Gope P (2017) Enhanced secure mutual authentication and key agreement scheme with user anonymity in ubiquitous global mobility networks. J Inf Secur Appl 35:160–167

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

Gope P, Hwang T (2016a) 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

Gope P, Hwang T (2016b) Lightweight and energy-efficient mutual authentication and key agreement scheme with user anonymity for secure communication in global mobility networks. IEEE Syst J 10(4):1370–1379

Goutham RA, Lee GJ, Yoo KY (2015) An anonymous ID-based remote mutual authentication with key agreement protocol on ECC using smart cards. In Proceedings of the 30th Annual ACM Symposium on Applied Computing, pp. 169–174

Han L, Tan X, Wang S, Liang X (2016) An efficient and secure three-factor based authenticated key exchange scheme using elliptic curve cryptosystems. Peer-to-Peer Netw Appl 11(1): 63–73

He D (2012) An efficient remote user authentication and key agreement protocol for mobile client–server environment from pairings. Ad Hoc Netw 10(6):1009–1016

Hsieh WB, Leu JS (2012) Exploiting hash functions to intensify the remote user authentication scheme. Comput Secur 31(6):791–798
Irshad A, Chaudhry SA, Kumari S, Usman M, Mahmood K, Faisal MS (2017a) An improved lightweight multiserver authentication scheme. Int J Commun Syst, 30(17)
Irshad A, Sher M, Nawaz O, Chaudhry SA, Khan I, Kumari S (2017b) A secure and provable multi-server authenticated key agreement for TMIS based on Amin et al. scheme. Multimed Tools Appl 76(15):16463–16489
Irshad A, Sher M, Ashraf MU, Alzahrani BA, Wu F, Xie Q, Kumari S (2017c) An Improved and Secure Chaotic-Map Based Multi-server Authentication Protocol Based on Lu et al. and Tsai and Lo's Scheme. Wireless Pers Commun 95(3):3185–3208
Irshad A, Kumari S, Li X, Wu F, Chaudhry SA, Arshad H (2017d) An improved SIP authentication scheme based on server-oriented biometric verification. Wireless Pers Commun 97(2):2145–2166
Islam SH, Biswas GP (2011) A more efficient and secure ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem. J Syst Softw 84(11):1892–1898
Islam SH, Biswas GP (2014) Dynamic id-based remote user mutual authentication scheme with smartcard using elliptic curve cryptography. J Electron 31(5):473–488
Jan JK, Chen YY (1998) "Paramita wisdom" password authentication scheme without verification tables. J Syst Softw 42(1):45–57
Jiang Q, Ma J, Li G, Li X (2015) Improvement of robust smart-card-based password authentication scheme. Int J Commun Syst 28(2):383–393
Khan MK, Zhang J, Wang X (2008) Chaotic hash-based fingerprint biometric remote user authentication scheme on mobile devices. Chaos Solitons Fractals 35(3):519–524
Khan MK, Kumari S, Gupta MK (2014) More efficient key-hash based fingerprint remote authentication scheme using mobile device. Computing 96(9):793–816
Kocher P, Jaffe J, Jun B (1999) Differential power analysis. Advances in Cryptology—CRYPTO'99, pp 388–397
Kumari S, Khan MK (2014) Cryptanalysis and improvement of 'a robust smart-card-based remote user password authentication scheme'. Int J Commun Syst 27(12):3939–3955
Kumari S, Khan MK, Li X (2014) An improved remote user authentication scheme with key agreement. Comput Electr Eng 40(6):1997–2012
Kumari S, Chaudhry SA, Wu F, Li X, Farash MS, Khan MK (2017) An improved smart card based authentication scheme for session initiation protocol. Peer-to-Peer Netw Appl 10(1):92–105
Lamport L (1981) Password authentication with insecure communication. Commun ACM 24(11):770–772
Li CT, Hwang MS (2010) An efficient biometrics-based remote user authentication scheme using smart cards. J Netw Comput Appl 33(1):1–5
Li X, Niu JW, Ma J, Wang WD, Liu CL (2011) Cryptanalysis and improvement of a biometrics-based remote user authentication scheme using smart cards. J Netw Comput Appl 34(1):73–79
Li X, Niu J, Khan MK, Liao J (2013) An enhanced smart card based remote user password authentication scheme. J Netw Comput Appl 36(5):1365–1371
Li X, Niu J, Wang Z, Chen C (2014) Applying biometrics to design three-factor remote user authentication scheme with key agreement. Secur Commun Netw 7(10):1488–1497
Liao IE, Lee CC, Hwang MS (2006) A password authentication scheme over insecure networks. J Comput Syst Sci 72(4):727–740
Lu Y, Li L, Peng H, Yang Y (2016) Robust anonymous two-factor authenticated key exchange scheme for mobile client-server environment. Secur Commun Netw 9(11):1331–1339
Luo M, Zhang Y, Khan MK, He D (2017) A secure and efficient identity-based mutual authentication scheme with smart card using elliptic curve cryptography. Int J Commun Syst, 30(16)
Lv C, Ma M, Li H, Ma J, Zhang Y (2013) A novel three-party authenticated key exchange protocol using one-time key. J Netw Comput Appl 36(1):498–503
Madhusudhan R, Mittal RC (2012) Dynamic ID-based remote user password authentication schemes using smart cards: a review. J Netw Comput Appl 35(4):1235–1248
Messerges TS, Dabbish EA, Sloan RH (2002) Examining smart-card security under the threat of power analysis attacks. IEEE Trans Comput 51(5):541–552
Mishra D, Das AK, Mukhopadhyay S (2014) A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards. Expert Syst Appl 41(18):8129–8143
Odelu V, Das AK, Goswami A (2015) A secure biometrics-based multi-server authentication protocol using smart cards. IEEE Trans Inf Forensics Secur 10(9):1953–1966
Pippa RS, Jaidhar CD, Tapaswi S (2010) Comments on symmetric key encryption based smart card authentication scheme. In 2nd IEEE International Conference on Computer Technology and Development, pp. 482–484
Qi M, Chen J (2017) An efficient two-party authentication key exchange protocol for mobile environment. Int J Commun Syst, 30(16)
Reddy AG, Das AK, Odelu V, Yoo KY (2016a) An enhanced biometric based authentication with key-agreement protocol for multi-server architecture based on elliptic curve cryptography. PloS one 11(5):e0154308
Reddy AG, Das AK, Yoon EJ, Yoo KY (2016b) A secure anonymous authentication protocol for mobile services on elliptic curve cryptography. IEEE Access 4:4394–4407
Roy S, Chatterjee 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. https://doi.org/10.1109/TDSC.2016.2616876
Roy S, Chatterjee S, Das AK, Chattopadhyay S, Kumar N, Vasilakos AV (2017a) On the design of provably secure lightweight remote user authentication scheme for mobile cloud computing services. IEEE Access 5(1):25808–25825. https://doi.org/10.1109/ACCESS.2017.2764913
Roy S, Chatterjee S, Das AK, Chattopadhyay S, Kumari S, Jo M (2017b) Chaotic map-based anonymous user authentication scheme with user biometrics and fuzzy extractor for crowdsourcing internet of things. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2017.2714179
Song R (2010) Advanced smart card based password authentication protocol. Comput Stand Interfaces 32(5):321–325
Sood SK, Sarje AK, Singh K (2010) An improvement of Xu et al.'s authentication scheme using smart cards. In: Proceedings of the third annual ACM Bangalore conference on communications, pp. 15
SPAN-Security Protocol Animator for AVISPA, [Online]. Available: http://www.irisa.fr/celtique/genet/span/. Accessed Dec 2016
Tan K, Zhu H (1999) Remote password authentication scheme based on cross-product. Comput Commun 22(4):390–393
Tu H, Kumar N, Chilamkurti N, Rho S (2015) An improved authentication protocol for session initiation protocol using smart card. Peer-to-Peer Netw Appl 8(5):903–910
Tzong-Chen W, Hung-Sung S (1996) Authenticating passwords over an insecure channel. Comput Secur 15(5):431–439
Wang D, Wang P (2015) Offline dictionary attack on password authentication schemes using smart cards. In: Desmedt Y (eds)

Information Security. Lecture Notes in Computer Science, vol 7807. Springer, Cham, pp 221–237

Wang YY, Liu JY, Xiao FX, Dan J (2009) A more efficient and secure dynamic ID-based remote user authentication scheme. Comput Commun 32(4):583–585

Wang RC, Juang WS, Lei CL (2011) Robust authentication and key agreement scheme preserving the privacy of secret key. Comput Commun 34(3):274–280

Wang D, Wang N, Wang P, Qing S (2015a) Preserving privacy for free: efficient and provably secure two-factor authentication scheme with user anonymity. Inf Sci 321:162–178

Wang D, He D, Wang P, Chu CH (2015b) Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment. IEEE Trans Dependable Secure Comput 12(4):428–442

Wazid M, Das AK, Odelu V, Kumar N, Susilo W (2017) Secure remote user authenticated key establishment protocol for smart home environment. IEEE Trans Dependable Secure Comput. https://doi.org/10.1109/TDSC.2017.2764083

Wen F, Li X (2012) An improved dynamic ID-based remote user authentication with key agreement scheme. Comput Electr Eng 38(2):381–387

Wu TC (1995) Remote login authentication scheme based on a geometric approach. Comput Commun 18(12):959–963

Wu TY, Tseng YM (2010) An efficient user authentication and key exchange protocol for mobile client–server environment. Comput Netw 54(9):1520–1530

Wu F, Xu L, Kumari S, Li X (2015) A novel and provably secure biometrics-based three-factor remote authentication scheme for mobile client–server networks. Comput Electr Eng 45:274–285

Xie Q, Dong N, Wong DS, Hu B (2016) Cryptanalysis and security enhancement of a robust two-factor authentication and key agreement protocol. Int J Commun Syst 29(3):478–487

Xie Q, Tang Z, Chen K (2017) Cryptanalysis and improvement on anonymous three-factor authentication scheme for mobile networks. Comput Electr Eng 59:218–230

Xu J, Zhu WT, Feng DG (2009) An improved smart card based password authentication scheme with provable security. Comput Stand Interfaces 31(4):723–728

Yang JH, Chang CC (2009) An ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem. Comput Secur 28(3):138–143

Yeh HL, Chen TH, Hu KJ, Shih WK (2013) Robust elliptic curve cryptography-based three factor user authentication providing privacy of biometric data. IET Inf Secur 7(3):247–252

Yoon EJ, Yoo KY (2009) Robust id-based remote mutual authentication with key agreement scheme for mobile devices on ECC. In: IEEE International Conference on Computational Science and Engineering CSE'09, pp 633–640

Zhang L, Tang S, Cai Z (2014) Efficient and flexible password authenticated key agreement for voice over internet protocol session initiation protocol using smart card. Int J Commun Syst 27(11):2691–2702