Multimedia Tools and Applications (2021) 80:16907–16931
https://doi.org/10.1007/s11042-020-09078-z

A secure and improved multi server authentication protocol using fuzzy commitment

Hafeez Ur Rehman[1] · Anwar Ghani[1] · Shehzad Ashraf Chaudhry[2] · Mohammed H. Alsharif[3] · Narjes Nabipour[4]

Received: 22 October 2019 / Revised: 4 May 2020 / Accepted: 15 May 2020 /
Published online: 1 July 2020

Abstract

The advancement in communication and computation technologies has paved a way for connecting large number of heterogeneous devices to offer specified services. Still, the advantages of this advancement are not realized completely due to inherent security issues. Most of the existing authentication mechanisms ensure the legitimacy of requesting user thorough single server leading towards multiple registrations and corresponding credentials storage on user side. Intelligent multimedia networks (IMN) may encompass wide range of networks and applications. However, the privacy and security of IMN cannot be apprehended through traditional multi sign on/single server authentication systems. The multi-server authentication systems can enable a user to acquire services from multiple servers using single registration and with single set of credentials (i.e.Password/smart card etc.) and can be accomplish IMN security and privacy needs. In 2018, Barman et al. proposed a multi-server authentication protocol using fuzzy commitment. The authors claimed that their protocol provides anonymity while resisting all known attacks. In this paper, we analyze that Barman et al.'s protocol is still vulnerable to anonymity violation attack and impersonation based on stolen smart card attack; moreover, it has incomplete login request and is prone to scalability issues. We then propose an enhanced protocol to overcome the security weaknesses of Barman et al.'s scheme. The security of the proposed protocol is verified using BAN logic and widely accepted automated AVISPA tool. The BAN logic and automated AVISPA along with the informal analysis ensure the robustness of the scheme against all known attacks.

Keywords Multi-server · Authentication · Fuzzy commitment · Security · BAN logic · AVISPA

✉ Anwar Ghani
anwar.ghani@iiu.edu.pk

Extended author information available on the last page of the article.

1 Introduction

The multi-server environment provides convenient and suitable online services. Unlike conventional single server authentication, the multi-server environment provides single sign-on without registering with multiple servers and keeping the multiple secrets of passwords and identities. The multi-server architecture works using the centralized trusted registration authority, responsible for registering the servers and users, in return it enables both the servers and users to get hassle free communication with each other. The users keeps only one secret password and one identity. The common use of a multi-server environment requires an efficient and robust user authentication protocol to establish a secure connection between both the requesting user and service providers. In 1981, Lamport [27] presented the first authentication protocol based on a server database containing the passwords of each registered user. Due to storage of the verifier in server database Lamport's protocol is subjected to the stolen verifier attack. Over time, many researchers proposed their protocols to resolve the issues of stolen verifier attack [4, 22].Wu et al.'s [48] presented a smart card-based authentication protocol; later He et al. [14] noticed that the protocol of Wu is vulnerable to insider attack and impersonation attack. Wu et al.'s [48] then presented an improved and enhanced protocol based on He et al.'s protocol. later Zhu et al. [49] found that the protocol of He et al. still has some weaknesses like offline password guessing attack. Anticipating the failure and/or unsuitability of two factor authentication protocols, many researchers proposed fingerprint-based three factor authentication protocols to enhance the security [20, 21, 28, 29, 37]. Lee et al. [28] presented fingerprint-based authentication. Lee et al. enhanced the security using three factors including: 1)smart card, 2)fingerprint minutiae, and 3)user Password. Later Lin et al.'s [29] claimed that Lee et al.'s protocol has weaknesses against spoofing and masquerade attacks. So they proposed an enhanced protocol based on Lee et al.'s protocol. Regretfully, Mitchell et al. [37] noticed that Lin et al.'s protocol still has some weaknesses. Mir and Nikooghadam [35] presented an enhanced biometrics-based authentication protocol and claimed their protocol provides security against well-known attacks like (user anonymity and untraceability, impersonation attacks, Online Password Guessing attacks, etc.) Later Chaudhry et al. [10] noticed that Mir and Nikooghadam [35] suffers from user anonymity attack as well as stolen smart attack. Unfortunately, Qi et al. [40] claimed Chaudhry et al.'s [10] protocol still has some weaknesses including non-resilience against denial of service attack; moreover, protocol in [10] is lacking perfect forward secrecy. In 2016, Wang et al. [47] proposed another biometric-based multi-server authentication and key agreement protocol based on Mishra et al.'s protocol. Wang et al. claimed their protocol provides various security features along-with user revocation/re-registration and biometric information protection. Soon, Reddy et al. [44] showed that Wang et al.'s [47] protocol is vulnerable to server impersonation, user impersonation and insider attacks, as their protocol share user credential to the server. Qi et al.'s [39] proposed yet another key-exchange authentication protocol and claimed it to provide security against well-known attacks. later Reddy et al.'s [43] noticed some vulnerabilities like session key leakage attack, user impersonation attack, insider attack, and user anonymity in the protocol of Qi et al. Some other developments were also proved either incorrect or insecure in [16, 19, 30, 33, 38, 42].

In 2018, Barman et al. [6] proposed a multi-server authentication protocol using fuzzy commitment. The authors in [6] claimed that their protocol provides various security

features like confidentiality of user identity/biometric data, mutual authentication and session key establishment between user and servers, besides this authors also claimed their protocol to provide security against the known attacks. However, the in-depth analysis in this article shows that the protocol of Barman et al. is facing some serious security threats. It is to show that the protocol proposed by Barman et al. is vulnerable to anonymity violation attack and impersonation attack based on stolen smart-card. Moreover, their protocol is not practicable owing to the scalability Issues. Then we propose an enhanced protocol to overcome the security weaknesses of Barman et al.'s protocol. We analyze the security of our proposed protocol through formal and informal analysis. In the formal analysis, we use BAN Logic and widely accepted AVISPA tool (a well known and widely accepted automated tool for security analysis). The informal security features analysis also shows the robustness of the proposed protocol.

2 Preliminaries

A brief review of the basics relating to fuzzy commitment technique, one-way hash function, error correction coding, and revocable template generation, is solicited in following subsections:

2.1 Fuzzy commitment

The fuzzy commitment as proposed by Juels and Wattenberg [23] is a method to hide the secrets under the witness and then release the conceal secrets later in the presence of a witness. In the Registration/enrollment phase a randomly generated key K_c is cipher with codeword $C_w = \aleph_{enc}(K_c)$. $\aleph_{enc}$ is an error correction technique and it helps in a noisy channel to recover equivalent match. When a user imprints his biometric then the binary string is generated against the biometric, C_{T_u} is used to conceal the key with binary string through XOR operation $[C_{T_u} \oplus C_w = H_{public}]$. The system contain only H_{public} and the hash of key $(h(K_c))$. In the authentication phase this H_{public} is available, so every legitimate user imprints his/her biometric to unlock C_w.

2.2 Hash function

Hash function $h : X \rightarrow Y$ is deterministic mapping set $X = \{0, 1\}^*$ of strings having variable length to another set $Y = \{0, 1\}^t$ of strings of fixed length, properties include:

- The input value say, $a \in X$ it is easy to computes $h(a)$, in polynomial times; moreover, $h(.)$ function is deterministic in nature.
- The small change in input value $a \in X$ results in a completely uncorrelated with $h(a)$.
- $One - way\ property$: It is difficult to find the actual message a given the message digest $h(a)$ of $a \in X$.
- $Weak - Collision\ resistant\ property$: Any given value input $a \in X$. it is difficult to find another $a^* \in X$ such that $h(a) = h(a^*)$.
- $Strong - Collision\ resistance\ property$: $h(a) = h(a^*)$ for any $a, a^* \in X$ and $a \neq a^*$, this property states that, it is also difficult to find any two inputs $a, a^* \in X$ such that $a \neq a^*$ with $h(a) = h(a^*)$.

2.3 Revocable template generation

A revocable template [41], provides the privacy and revocability of user biometric. By using transformation parameter TP_u and transformation function, $f(\cdot)$, user biometric data is converted into a cancel-able template $CT_u = f(BIO_u, TP_u)$ with following properties:

1. **Collision-free property:** If $CT_u = f(BIO_u, TP_u)$ and $CT_k = f(BIO_k, TP_k)$, then $CT_u \neq CT_k$. for $BIO_u \neq BIO_k$. Moreover, if $CT_n = f(BIO, TP_n)$ and $CT_m = f(BIO, TP_m)$, then $CT_n \neq CT_m$ for $TP_n \neq TP_m$.
2. **Intra-user variability property :** This property states; two different templates $CT_u = f(BIO_u, TP_u)$, $CT'_u = f(BIO'_u, TP_u)$ can be generated form same fingerprint.
3. **Revocation of biometric:** If user biometric is comprised, then new template can be generated by using new transformation parameter TP_u^{new} with same transformation function $f(\cdot)$.
4. **User privacy:** Cancel-able template should protect the confidentiality of user, moreover template should protect the information about original biometric of a user.

2.4 Error correction technique

In the biometric template, the intra-user variation is considered an error. To remove the errors in the user biometric template, error correction technique [17] is used for noisy biometric image. In the time of enrollment/registration $CT_{enrol_u} = f(BIO_{enrol_u}, TP_u)$ is generated, which is match with query template $CT_{query_u} = f(BIO_{query_u}, TP_u)$, at the authentication time. So the difference can be calculated through Hamming distance $e = HamDis(CT_{enrol_u}, CT_{query_u})$.

2.5 Threat model

According to the well known and widely accepted Dolev-Yao threat (DY) model [15], an attacker not only listens to the communication between two participants but also the attacker can change the entire message or delete the message as well on open channel. An attacker can also extract the secret credential of legitimate user form stolen smart card through power analysis attack [25, 34]. Second adversarial model is Canetti and Krawczyk model (CK-model). In authentication and key exchange protocol, it is considered as defacto standard. According to [9], CK-adversary model not only fallows Dolev-Yao threat (DY) model but in CK model the adversary is also able to get the session key and session states as well. Precisely, the adversary with following capabilities [11, 12] is considered:

1. The channel is under full control of Adversary, who can intercept the communicated messages and can replay original message or can modify it. The adversary can also generate and transmit a fake message.
2. User and server identities are public.
3. Adversary can launch power analysis attack and has abilities to steal verifier stored on server/gateway etc.
4. The private keys of all participants are considered as non-compromised.

2.6 The contributions

1. We have cryptanalyzed the recent multi-server authentication protocol proposed by Barman et al. [6] to show its security issues and vulnerabilities.

2. We propose an enhanced authentication protocol using only symmetric cryptography operations and fuzzy commitment.
3. The security of the proposed protocol is checked through BAN logic and widely accepted AVISPA.
4. The security discussion and security features comparison of the proposed protocol with related protocols including Barman et al.'s protocol is explained.
5. We have also provided the comparative computation and communication costs analysis of the proposed protocol with competing related protocols

3 Review of the protocol of Barman et al.

This section briefly reviews Barman et al.'s protocol [6]. The phases of the protocol are detailed in below subsections and the notations used in this paper are provided in Fig. 1.

3.1 Server registration phase

In Barman et al.'s protocol, initially, all the servers $S_k : \{1 \leq k \leq n\}$ gets register with RC. S_k selects its' identity SID_k and dispatches a registration request to the RC. RC computes and sends a secret key $PSK_K = h(SID_k||X_c)$ to each S_k. RC may also consider another n' servers, which may get register with the RC in future. Therefore, the RC chooses identities SID_S for each of the future server and generates the shared keys $PSK_S = h(SID_S||X_c)$ for $n + 1 \leq S \leq n + n'$ The server identities (for $n + n'$ server) along with their corresponding key pairs $(SID_k, PSK_k)|1 \leq k \leq n + n'$ are stored in RC database.

Symbols	Representations
U_u,S_k	user and server
SID_k	identity of server
ID_u, PW_u, BIO_u	identity, Password and biometric of U_u
$CT_u, TP_u, f(.)$	cancel-able template, transformation parameter and transformation function of U_u
RC	trusted registration center
X_c	secret/private key of RC
XR_k	shared keys between S_k and RC
E_{X_c}, D_{X_c}	encryption and decryption using private key of RC
R_{cu}	user's random number
H_u	fuzzy commitment helper data
$SK_{u,_k}$	session key between user U_u, S_k
PSK_k	secret/private key of S_k
$h(.)$	hash function
R_u, r_n, R_s	random number generated by U_u, RC, S_k
$T_1, T_2, T_3,$	time stamped generated by U_u, RC, S_k
T_u	time bound generated by S_k
ΔT	time delay
$\oplus, \|\|$	(XOR) and string concatenation operator
$\aleph_{enc}(.), \aleph_{dec}(.)$	encoding and decoding operator , of the error correction technique
SC_u, A_{adv}	smart card and adversary

Fig. 1 Notations

3.2 User registration phase

The detail steps of the user registration phase are defined below:

1. Initially, U_u registers with the RC to get the services, via a protected channel. U_u selects ID_u, PW_u, and transformation parameter T_{P_u} alongwith a random number Rc_u. U_u also imprints his BIO_u.
2. U_u produces the cancel-able biometric template using transformation function $CT_u = f(BIO_u, TP_u)$ and computes $RPW_u = h(PW_u||CT_u)$, $r_u = h(Rc_u||ID_u||PW_u)$. U_u then generates a random secret k_u and sends the registration request $\langle ID_u, RPW_u \oplus k_u \rangle$ to the RC, via a protected channel.
3. After checking validity of ID_u, RC computes $US_k = h(ID_u||PSK_k)$, $AM_k = US_k \oplus (RPW_u \oplus k_u)$, $SV_k = h(SID_k||PSK_k)$ and $BM_k = SV_k \oplus RPW_u \oplus k_u$ (for all servers). RC Issues a smart card SC_u having $\{(SID_k, AM_k, BM_k)|1 \leq k \leq (n+n')\}$ and sends it to U_u, via a protected channel.
4. Using error correction technique ε, U_u encodes Rc_u produces codeword $R_{cod} = \varepsilon_{enc}(Rc_u)$, computes $H_u = CT_u \oplus R_{cod}$, $R = h(Rc_u)$ and $P = h(r_u)$. U_u then computes $AM_{uk} = (AM_k \oplus k_u) \oplus r_u$ and $BM_{uk} = (BM_k \oplus k_u) \oplus r_u$ (for all servers). U_u then stores $\{(AM_{uk}, BM_{uk})\}|1 \leq k \leq (n+n')$, TPu, H_u, R, P, $h(\cdot)$, $\aleph_{enc}(\cdot)$, $\aleph_{dec}(\cdot)\}$ in smart card SC_u. U_u removes the Rc_u, BIO_u, CTu, r_u, AM_k and BM_k for security reasons.

3.3 Login phase

The detail steps of login request are as under:

1. U_u inserts the smart card into the terminal and provides the credentials ID_u, PW_u and BIO'_u for authentication.
2. The smart card SC_u generates the cancel-able fingerprint $CT'_u = f(BIO'_u, TP_u)$, and extracts $R'_{cod} = H_u \oplus CT'_u$ and then decodes R'_{cod} using error correction technique, $Rc'_u = \aleph_{dec}(R'_{cod})$. SC_u compares both values, $h(Rc'_u)$ with R which is stored in SC_u. If they are equal than proceed further else terminates the session.
3. SC_u computes $r'_u = h(Rc_u||ID_u||PW_u)$ and checks if $h(r'_u) = h(r_u)$, proceeds further; otherwise, SC_u terminates the session.
4. SC_u computes $US_k = AM_{uk} \oplus h(PW_u||CT_u) \oplus r'_u = h(ID_u||PSK_k)$ and $SV_k = BM_{uk} \oplus h(PW_u||CT_u) \oplus r'_u = h(SID_k||PSK_k)$. SC_u selects R_u, generates T_1, and computes $M'_1 = h(ID_u||US_k)$, $M'_2 = ID_u \oplus h(SV_k||T_1)$, $M_3 = M_1 \oplus R_u$, $M_4 = h(ID_u||M'_1||M'_2||T_1||R_u)$.
5. Finally, SC_u sends the request $\langle M'_2, M'_3, M'_4, T_1 \rangle$ to the server S_k.

3.4 Mutual authentication and key agreement phase

The mutual authentication and key agreement consists of the following steps:

1. S_k receives login request $\langle M'_2, M'_3, M'_4, T_1 \rangle$ at time T'_1 and after verifying the allowable time delay, $|T'_1 - T_1|$, S_k computes $M'_5 = M'_2 \oplus h(h(SID_k||\ PSK_k)||T_1)$, $M'_6 = h(M'_5||h(M'_5||PSK_k))$ $M'_7 = M'_3 \oplus M'_6 = R_u$ and $M'_8 = h(M'_5||M'_6||M'_2||T_1||M'_7)$. Check if $M'_8 \neq M'_4$, S_k cancels the login request, else proceeds further.
2. S_k select a random number R_s and generates T_3 then computes $M'_9 = h(h(M'_5||PS_k)||R_u) \oplus R_s$, and session key $SK_{uk} =$

$h(M_5'||h(SID_k||PSK_k)||R_u||R_s||T_1||T_3)$, $M_{10}' = h(h(M_5'||PSK_k)||SK_{uk}||T_3||R_s)$ and sends $\langle M_9', M_{10}', T_3\rangle$ to U_u.

3. The U_u receives $\langle M_9', M_{10}', T_3\rangle$. After checking the delay $|T_3 \leq T_c|$. SC_u computes $R_s' = M_9' \oplus h(US_k||R_u)$, the session key $SK_{uk}' = h(ID_u||SV_k||R_u||R_s||T_1||T_3)$ shared with S_k and $M_{11}' = h(US_k||SK_{uk}'||T_3||R_s')$. SC_u check the condition if $M_{11}' \neq M_{10}'$ terminates the session. Otherwise, the session key SK_{uk} is established between U_u and S_k.

3.5 Password and biometric template update phase

U_u provides the current credentials ID_u, PW_u BIO_u and extracts feature BIO_u' from the BIO_u. SC_u then computes $CT_u' = f(BIO_u', TP_u)$ and $Rc_u' = \aleph_{dec}(H_u \oplus CT_u')$ and then checks if $h(Rc_u') = R$, SC_u further computes $r_u' = h(Rc_u'||ID_u||PW_u)$ check if $h(r_u') = P$ proceeds further; otherwise, terminates the request. SC_u then asks U_u to modify their password and biometric template:

1. To update the password, U_u inputs PW_u^{new}, SC_u computes $r_u^{new} = h(Rc_u'||ID_u||PW_u^{new})$, $AM_{uk}^{new} = AM_{uk} \oplus r_u' \oplus r_u^{new} = h(ID_u||PSK_u) \oplus h(PW_{new}'||CT_u) \oplus h(Rc_u'||ID_u||PW_u^{new})$, $BM_{uk}^{new} = BM_{uk} \oplus r_u' \oplus r_u^{new} = h(SID_k||PSK_k) \oplus h(PW^{new}||CT_u) \oplus h(Rc_u'||ID_u||PW_u^{new})$ for $1 \leq k \leq (n + n')$ and $P^{new} = h(r_u^{new})$.SC_u updates its parameters $\{AM_{uk}, BM_{uk},\}$ with the newly computed values $\{AM_{uk}^{new}, BM_{uk}^{new}, P^{new}\}$ and stored in the SC_u.
2. To update the biometric template, SC_u requests U_u for a new transformation parameter TP_u. SC_u have the old TP_u and then set new $TP_u^{new} = TP_u$ and new cancel-able template $CT_u^{new} = f(BIO_u', TP_u^{new})$ is produced. SC_u also computes $RPW_u^{new} = h(PW_u||CT_u^{new})$, $AM_{uk}^{new} = AM_{uk} \oplus RPW_u \oplus RPW_u^{new} = h(ID_u||PSK_k) \oplus h(PW_u||CT_u^{new})r_u'$, $BM_{uk}^{new} = BM_{uk} \oplus RPW_u \oplus RPW_u^{new} = h(SID_k||PSK_k) \oplus h(PW_u||CT_u^{new}) \oplus r_u'$, and the new helper data $H_u^{new} = CT_u^{new} \oplus \aleph_{enc}(Rc_u')$. Accordingly, the information $\{AM_{uk}, BM_{uk}, H_u\}$ is replaced by $\{AM_{ij}^{new}\ BM_{uk}^{new}, H_u^{new}\}$ stored in the SC_u .

3.6 Smart card revocation phase

If the SC_u of a authorized U_u is damaged, lost or stolen, then U_u can get a new SC_u from the RC. U_u provides ID_u and PW_u and to imprints BIO_u, Steps are:

1. U_u computes $CT_u' = f(BIO_u, TP_u)$ and $RPW_u = h(PW_u||CT_u')$, U_u generates a random number k_u', then computes a parameter $RPW_u' = RPW_u \oplus k_u'$ and then sends the request $\langle ID_u, RPW_u'\rangle$ to the RC via a protected channel for a new SC_u^{new}
2. RC computes $AM_k = h(ID_u||PSK_k) \oplus RPW_u'$, $BM_k = h(SID_k||PSK_k) \oplus RPW_u'$ for $k = 1, 2, , , , , (n + n')$ and Issue a new SC_u^{new} containing $\{(SID_k, AM_k, BM_k)|1 \leq k \leq n + n_{\prime}\}$. SC_u^{new} sends to these parameter to U_u via a protected channel.
3. U_u generates a new random number R_u^{new} and computes $r_u = h(R_u^{new}||ID_u||PW_u)$, $H_u^{new} = CT_u' \oplus \aleph_{enc}(R_u^{new})$, $AM_{uk} = (AM_k \oplus k_u') \oplus r_u$, $BM_{uk} = (BM_k \oplus k_u') \oplus r_u$, $R = h(Rc_u^{new})$, $P = h(r_u)$ and stores these values in SC_u^{new}, memory. U_u also stores $\{TP_u, \aleph_{enc}(\cdot), \aleph_{dec}(\cdot), h(\cdot)\}$ in SC_u^{new} memory.

4 Cryptanalysis of the Protocol of Barman et al.

The in depth analysis in following subsections proves that Barman et al.'s protocol [6] entails serious security flaws:

4.1 Incomplete login request

The login message, $\{M_2', M_3', M_4', T_1\}$ sent by user U_u to the server S_k is incomplete, because the identity of server SID_k is not included in the login request, which is the most important parameter for communication [32] and without the server identity, the RC cannot direct the request of U_u to his intended server. This crucial mistake can be treated as typing mistake. The protocol can only work if the login message contains the identity of the server.

4.2 User anonymity violations attack

Here, we show that the protocol of Barman et al. is vulnerable to user anonymity violation attack. Let U_a be a legal but dishonest user of the system and wants to violate user anonymity. In the Mutual Authentication phase of Barman et al.'s protocol user U_u sends the message $\{M_2', M_3', M_4', T_1, SID_k\}$ to the server SID_k on public channel. During the communication, let U_a intercepts the message and using $M_2' = ID_u \oplus h(SV_k \| T_1)$, U_a can easily extract the ID_u of every users. Because all the users connected to the SID_k has SV_k(secret identifier generated by RC for SID_k) stored in the smart card. U_a can extract the identity of user as follows:

Step AV 1: U_u sends the login message to SID_k. During the communication, let user U_a intercepts the message $\{M_2', M_3', M_4', T_1, SID_k\}$.

Step AV 2: U_a using his own smart card, enters his credentials including: ID_a, PW_a and BIO_a. U_a extracts $\{BM_{ak}, AM_{ak}\}$ pair from his own smart card and then computes $CT_a = f(BIO_a, TP_a)$, $R_{cod}' = H_a \oplus CT_a$, $Rc_a' = \aleph_{dec}(R_{cod}')$, $r_a = h(Rc_u||ID_a||PW_a)$, similar to login steps. U_a then computes:

$$US_{k_a} = AM_{ak} \oplus h(PW_a||CT_a) \oplus r_a \tag{1}$$

$$SV_k = BM_{ak} \oplus h(PW_a||CT_a) \oplus r_a' = h(SID_k||PSK_k) \tag{2}$$

$$Z = h(SV_k||T_1) \tag{3}$$

Step AV 3: Based on SV_k, Z and the M_2' from login request, U_a computes:

$$ID_u = M_2' \oplus Z \tag{4}$$

In Eq.4, the ID_u is the real identity of U_u. Therefore, U_a has successfully broken the user anonymity.

4.3 User impersonation attack based on stolen smart-card

Using the stolen smart card of some user say U_u, another legal but dishonest user of the system can launch user impersonation attack in Barman et al.'s protocol. Let U_a be a legal user, gets his card SC_a containing $\{SID_k, AM_{a_k}, BM_{a_k} | 1 \leq k \leq (n + n')\}$ along with $\{TP_a, H_a, P, h(\cdot), \aleph_{enc}, \aleph_{dec}\}$ and steals the smart card SC_u. U_a performs following steps to impersonate on behalf of U_u:

Step ISC 1: U_a enters his credential ID_a, PW_a and biometric BIO_a. U_a now computes US_k, CT_a', r_a', $SV_k = BM_{uk} \oplus h(PW_a||CT_a) \oplus r_a' = h(SID_k||PSK_k)$. As SV_k is common in all smart cards.

Step ISC 2: Extracts $AM_{uk} = US_{uk} \oplus (RPW_u \oplus uk)$ and $BM_{uk} = SV_k \oplus (RPW_u \oplus uk)$ form U_u's stolen smart card SC_u.

Step ISC 3: U_a using SV_k computes:

$$X = AM_{uk} \oplus BM_{uk} = \{US_{uk} \oplus (RPW_u \oplus uk)\} \oplus \{SV_k \oplus (RPW_u \oplus uk)\} \quad (5)$$

$$= US_{uk} \oplus SV_k \quad (6)$$

$$US_{uk} = X \oplus SV_k \quad (7)$$

Step ISC 4: U_a has SV_k and US_{uk} of U_u with ID_u. U_u generates a random number R_u and time stamp T_1 computes:

$$M_1' = h(ID_u||US_k) \quad (8)$$

$$M_2' = ID_u \oplus h(SV_k||T_1) \quad (9)$$

$$M_3' = M_1' \oplus R_u \quad (10)$$

$$M_4' = h(ID_u||M_1'||M_2'||T_1||R_u) \quad (11)$$

Step ISC 5: U_a sends the login request message $\langle M_2', M_3', M_4', T_1, SID_k\rangle$ to the S_k. S_k receives the login request $\langle M_2', M_3', M_4', T_1, SID_k\rangle$ after checking time delay, $|T_1' - TS_1|$, computes following:

$$M_5' = M_2' \oplus h(h(SID_k||PSK_k)||T_1) = (ID_u) \quad (12)$$

$$M_6' = h(M_5'||h(M_5'||PSK_k)) \quad (13)$$

$$M_7' = M_3' \oplus M_6' = R_u \quad (14)$$

$$M_8' = h(M_5'||M_6'||M_2'||T_1||M_7') \quad (15)$$

Step ISC 6: S_k checks if $M_8' = M_4'$, U_a will pass this test because M_8' and M_4' both have same values. S_k selects a nonce R_s, generates current timestamp T_3, and computes:

$$M_9' = h(h(M_5'||PS_k)||R_u) \oplus R_s \quad (16)$$

$$SK_{uk} = h(M_5'||h(SID_k||PSK_k)||R_u||R_s||T_1||T_3) \quad (17)$$

$$M_{10}' = h(h(M_5'||PSK_k)||SK_{uk}||T_3||R_s) \quad (18)$$

Step ISC 7: Then, S_k sends $\langle M_9', M_{10}', T_3\rangle$ to U_a. U_a receives the authentication reply message $\langle M_9', M_{10}', T_3\rangle$ at time T_3' and computes:

$$R_s = M_9' \oplus h(US_k||R_u) \quad (19)$$

$$SK_{uk}' = h(ID_u||SV_k||R_u||R_s||T_1||T_3) \quad (20)$$

$$M_{11}' = h(US_k||SK_{uk}'||T_3||R_s) \quad (21)$$

The session key as computed by U_a in Eq. 20 is same as computed by S_k in Eq.17. Therefore, U_a has succesffuly established a secure connection with S_k by impersonating on behalf of U_a.

4.4 Scalability problems

In the registration phase of Barman et al.'s protocol smart card stores AM_k. As in multi-server environment, there may be several servers and users. So it is inefficient to store (AM_k) against every server within smart card due to its small magnetic chip which has limited storage. This protocol is not practical, suppose we have n servers, so we need to store US_k and SV_k of n servers within the smart card, each of size 160 bits. For large number of servers like 100, the bits stored for US_k and SV_k in the smart card are 32000 bits, which

can be problematic due to its storage restrictions. Moreover, authors did not mention the procedure to update the smart card if some new servers are added, $AM_{uk} = (AM_k \oplus k_u) \oplus r_u$ and $BM_{uk} = (BM_k \oplus k_u) \oplus r_u$ for $1 \leq k \leq (n + n')$.

5 Proposed protocol

This section details the proposed scheme consisting of three entities including, users, servers and the registration center (RC). The details are in following subsections:

5.1 Server registration phase

Every S_k along with its particular identity SID_k must send a registration request to the RC, if they are willing to provide services to the legitimate users U_u. RC computes $X_{RS_k} = h(SID_k||Xc)$ and $M_k = E_{X_c}(X_{RS_k})$ and stores $(SID_k, E_{X_c}(X_{RS_k}))$ in the database of Rc and send the share key to the server (X_{RS_k}).

5.2 User registration phase

U_u chooses ID_u, PW_u, TP_u, then imprints BIO_u and selects random number N_1. U_u computes $CT_u = f(BIO_u, TP_u)$, $A_u = h(N_1||PW_u||ID_u||CT_u)$ and sends A_u, ID_u to the RC. On receiving, RC computes $X_u = h(ID_u||X_c)$ and $Y_u = Xu \oplus A_u$, generates a random number r_o and computes the pseudo identity $PID_u = E_{X_c}(ID_u||r_o) \oplus A_u$. RC then store Y_u, PID_u, $h(.)$ in smart card and sends the smart card to user using some secure channel. On receiving smart card, U_u computes $R_c = \aleph_{enc}(Rc_u)$, $H_u = CT_u \oplus R_{cod}$, $R = h(Rc_u)$, $r_u = (Rc_u||ID_u||PW_u)$, $P = h(r_u)$ and $E_u = N_1 \oplus r_u$. U_u stores $\{TP_u, H_u, R, P, h(.), \aleph_{enc}(\cdot), \aleph_{dec}(\cdot), Y_u, PID_u, E_u\}$ in the smart card. The Server User registration phases are also illustrated in Fig. 2.

5.3 Login and authentication phase

The following steps as shown in Fig. 3, explain the login and authentication phase briefly:

Step AP 1: User need to insert the smart card provides the credentials ID_u, PW_u, BIO'_u and calculates $CT'_u = f(BIO'_u, TP_u)$, $R'_{cod} = H_u \oplus CT'_u$, $Rc'_u = \aleph_{dec}(R'_{cod})$, and check if $h(Rc'_u) \neq R$, terminates the session, otherwise calculates $r'_u = h(Rc'_u||ID_u||PW_u)$, and check again if $h(r'_u) \neq h(r_u)$ terminates the session, else computes $N_1 = (E_u \oplus r_u)$, $A'_u = h(ID_u||PW_u||N_1||CT_u)$, $X_u = (Y_u \oplus A'_u)$, $DID_u = (PID_u \oplus A'_u)$, generates a random no R_u and time stamp T_1, and to get the services of server needs the address SID_k, and computes $G_u = R_u \oplus h(X_u||ID_u||SID_k||T_1)$, $H_u = h(ID_u||G_u||X_u||R_u||T_1||SID_k)$, sends $\{DID_u, H_u, G_u, T_1, SID_k\}$ to the RC on public channel.

Step AP 2: RC receives the login request and checks the time delay $(T_c - T_1 \leq \delta T)$. RC decrypts $(ID_u||r_o) = D_{X_c}(PID_u)$ using X_c and computes $X_u = h(ID_u||X_c)$ $R_u = G_u \oplus h(X_u||ID_u||SID_k||T_1)$ $H'_u = h(ID_u||G_u||X_u||R_u||T_1||SID_k)$. RC then check $H'_u \stackrel{?}{=} H_u$ if not true, terminates the session. Otherwise, RC verifies user successfully. RC then extracts X_{RS_k} from verifier table, generates time stamp T_2, computes $X'_u = h(X_u||ID_u||SID_k||T_1)$, $H_{R_c} =$ and $h(X_{RS_k}||X'_u||ID_u||SID_k||T_2)$. RC now encrypts the parameters $(X'_u, R_u, ID_u, H_{R_c}, SID_k, T_1)$ using share secret key X_{RS_k} and sends $E_{X_{RS_k}}(X'_u\ R_u, ID_u, H_{R_c}, SID_k, T_1), T_2, SID_k$ to the server over public channel.

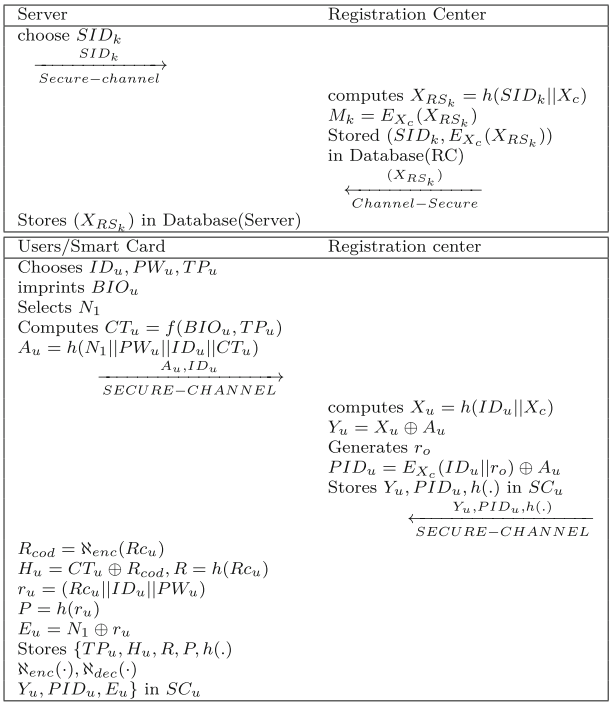

Fig. 2 Registration phase of Sever and User

Step AP 3: On receiving the message, S_k after checking the time delay $(T_c - T_2 \leq \delta T)$, decrypts $D_{X_{RS_k}}(X'_u, R_u, ID_u, H_{R_c}, SID_k, T_1)$ using the shared key X_{RS_k}. S_k then computes $H'_{R_c} = h(X_{RS_k}||\ X'_u||ID_u||SID_k||T_2)$ and checks the equality $H'_{R_c} \stackrel{?}{=} H_{R_c}$ if condition is true, S_k verifies RC successfully. Further S_k generates R_s, T_3 and computes $M_x = R_s \oplus h(ID_u||X'_u||R_u||T_3)$ $H''_{R_c} = h(R_s||M_x||T_u||ID_u||\ T_3)$. S_k further sends $\{M_x, H''_{R_c}, T_3, T_u, \}$ to the RC, which in turn checks $(T_c - T_3 \leq \delta T)$ and on successful verification computes $R_s = M_x \oplus (ID_u||X'_u||R_u||T_3)$ $H'''_{R_c} = h(R_s||M_x||T_u||ID_u||T_3)$. RC then checks $H'''_{R_c} \stackrel{?}{=} H''_{R_c}$ and on successful verification computes new dynamic identity $RID_u = E_{X_c}(ID_u||r_n) \oplus R_s$ for U_u and forwards $\{M_x, H''_{R_c}, T_3, T_u, RID_u\}$ to the legitimate user U_u.

Step AP 4: U_u on receiving the message, checks $T_3 \leq \delta T_c$ and on success, U_u computes $R_s = M_x \oplus (ID_u||X'_u||R_u||T_3)$, $H''''_{R_c} = h(R_s||M_x||T_u||ID_u||T_3)$ and checks whether $H''''_{R_c} \stackrel{?}{=} H''_{R_c}$ if true then session key $SK_{uk} = h(X'_u||ID_u||SID_k||R_s||R_u)$ is established between user and server.

5.4 Password and biometric update phase

In this section, we also proposed the Password change and biometric template update Process of our protocol, the U_u will need to log in successfully to change their current Password and update their biometric template, The detailed steps are described below:

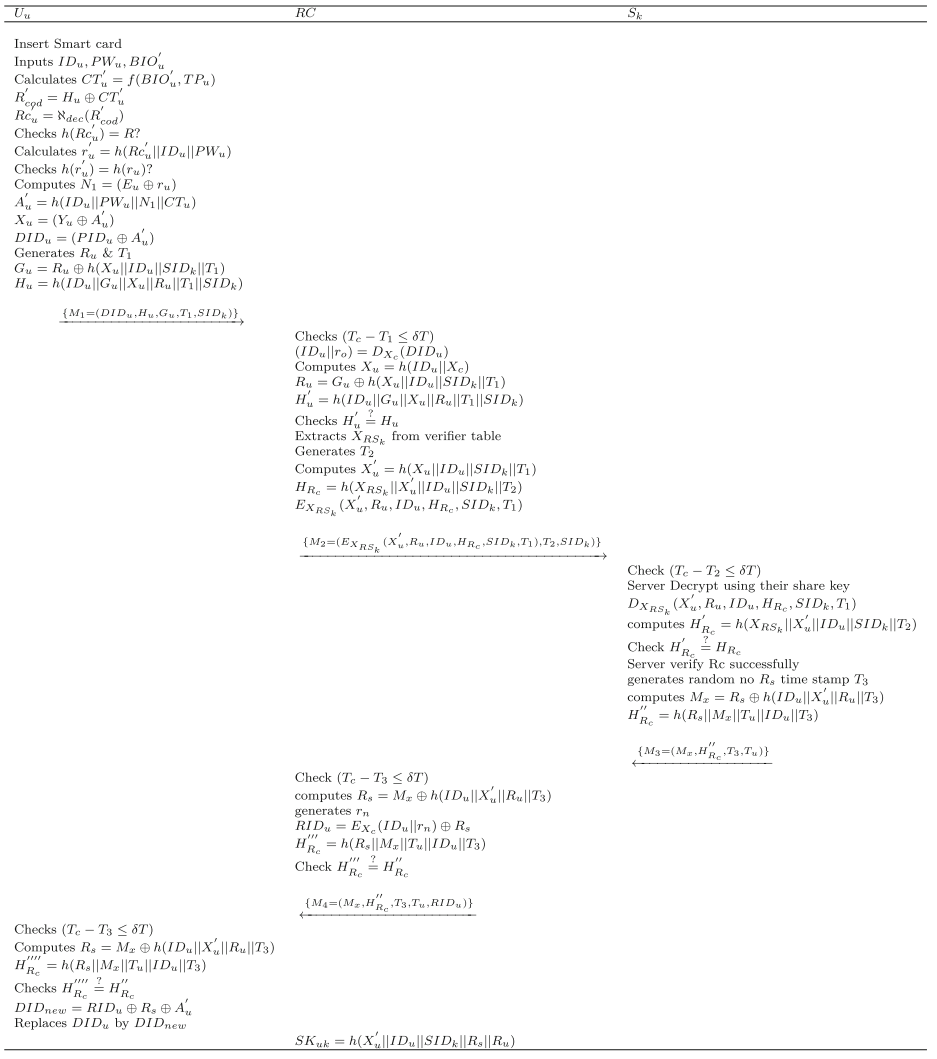

Fig. 3 Login and Authentication Phase

Step CPB 1: U_u provides the credentials ID_u, PW_u, and BIO_u after inserting the smart-card into a card reader to login. $BIO_u^{'}$ is extracted from the captured BIO_u. SC_u then computes $CT_u^{'} = f(BIO_u^{'}, TP_u)$ and $R_{cu}^{'} = \varepsilon_{dec}(H_u \oplus CT_u^{'})$. Checks if $h(R_{cu}^{'}) = R$, then SC_u computes $r_i^{'} = h(R_{cu}^{'}||ID_u||PW_u)$, and check if $h(r_i^{'}) = P$, smart card then asks users U_u to change the password and update the biometric template.

Step CPB 2: For Password change, SC_u asks U_u for a new Password. U_u inputs the new Password PW_u^{new}. SC_u computes $r_u^{new} = h(R_{cu}^{'}||ID_u||PW_u^{new})$, $E_u^{new} = N_1 \oplus r_u^{new}$ and $P^{new} = h(r_i^{new})$. SC_u updates its parameters stored $\{TP_u, H_u, R, P^{new}, h(\cdot), \varepsilon_{enc}(\cdot), \varepsilon_{dec}(\cdot), Y_u, PID_u, E_u^{new}\}$ in smart card.

Step CPB 3: To update the biometric template, SC_u asks U_u for a new transformation parameter TP_i^{new}. The new cancel-able template is generated as $CT_i^{new} =$

$f(BIO_u, TP_i^{new})$, along-with helper data $H_i^{new} = CT_i^{new} \oplus \varepsilon_{enc}(R'_{ci})$. Then $CT_i^{new} = f(BIO_u, TP_i^{new})$ and $H_i^{new} = CT_i^{new} \oplus \varepsilon_{enc}(R'_{ci})$ are stored in memory of SC_u.

5.5 Smart card revocation procedure

If SC_u of the legitimate user U_u is damaged, lost or stolen, then RC will Issue the new smart card. For this Process, the user provides their credential ID_u, PW_u, BIO_u. The following steps are esential to complete this procedure:

Step SCR 1: U_u computes $CT'_i = f(BIO_i, TP_i)$ and generates a 160-bit secret N'_1. Then U_u computes $A'_u = h(N'_1||PW_u||ID_u||\ CT'_u)$, and transmits the request message $\{A'_u, ID_u\}$ to the RC via a protected channel for SC_u^{new}.

Step SCR 2: RC computes $X_u = h(ID_u||Xc)$, $Y'_u = X_u \oplus A'_u$, generates random r'_o and computes $PID'_u = E_{X_c}(IDu||r'_o) \oplus A'_u$ store Y'_u, PID'_u, $h(.)$ in SC_u, then Issue a SC_i^{new} containing the credentials , Y_u, PID'_u, $h(.)$. SC_i^{new} is then sent to U_u via some protected channel.

Step SCR 3: U_u computes $r'_u = h(Rc_i^{new}||ID_u||PW_u)$, $H_{new}^u = CT'_u \oplus \varepsilon_{enc}(Rc_u^{new})$, , $R = h(Rc_u^{new})$, $P = h(r_u)$ and stores these values in SC_i^{new} memory.

6 Security analysis

This section provides the formal and informal security analysis of the proposed scheme. Moreover, automated formal security proof using popular tool AVISPA is also provided in this section:

6.1 Formal analysis using BAN logic

For formal analysis, Burrows-Abadi-Needham (BAN) logic [8] is applied in this subsection to verify the mutual authentication between user U_u and server S_k with the help of RC. Fig. 4 presents the notation guide for BAN logic.

Notations	Description
$M\| \equiv N$	M believes N
$M \lhd N$	M sees N
$M\| \sim N$	M said N once
$M \Rightarrow N$	M has jurisdiction on N
$\#(A)$	A is fresh
(A, B)	A or B are piece of principle (A,B)
$< A >_B$	The A rule is joined with B
$\{A\}_K$	This show that formula A is encoded with key K
$(A)_K$	This show that A value hashed with the key K
$M \overset{K}{\Longleftrightarrow} N$	M and N are shared Private key K
$M \overset{K}{\longrightarrow} N$	M have a public key K
SK	SK session Key

Fig. 4 Notations and Concepts in BAN-Logic

6.2 Rules of BAN-Logic

Rule 1: Message Meaning $\frac{P|\equiv P \overset{K}{\longleftrightarrow} Q.P \lhd <X>_K}{P|\equiv Q|\sim X}$ It shows that if P obtain the X encoded with Key K and P deems K is fine key to communicate with Q, and then P believes Q said X.
Rule 2: Nonce Verification $\frac{P|\equiv \#(X), P|\equiv Q|\sim X}{P|\equiv Q|\equiv X}$ When a principal P trusted that X is new/fresh also then principal Q only once time sends X after that Principal after that P believe Q held X.
Rule 3: Jurisdiction $\frac{P|\equiv Q \Rightarrow X, P|\equiv Q|\equiv X}{P|\equiv X}$ Principal P believes that Q have control/jurisdiction on X also P believes that Q believes X, after that P trusted that X is right.
Rule 4: Acceptance Conjuncatenation $\frac{P|\equiv X, P|\equiv Y}{P|\equiv (X,Y)}$ If a principal P is believes X as well as Y, subsequently then principal P also believes on (X, Y).
Rule 5: Freshness Conjuncatenation $\frac{P|\equiv \#(X)}{P|\equiv \#(X,Y)}$ If a principal P confident that X is a fresh, after that a principal P also believes newness / freshness of (X, Y).
Rule 6: Session Key $\frac{P|\equiv \#(X), P|\equiv Q \equiv X}{P|\equiv P \overset{K}{\longleftrightarrow} Q}$ If a principal P believe the fresh session key also then principal P as well 'Q' also believes on X which is the essential constraint of a session key, next principal P also believes that he/she share a session key 'K' with Q.

6.3 Assumptions

We assume that the following holds at the beginning of every run of our protocol.

- A1: $U_u| \equiv \#(R_u, T_1)$
- A2: $RC| \equiv \#(T_2, r_n)$
- A3: $S_k \equiv \#(R_s, T3)$
- A4: $U_u| \equiv (U_u \overset{SK_{uk}}{\longleftrightarrow} S_k)$
- A5: $RC| \equiv U_u| \equiv (U_u \overset{SK_{uk}}{\longleftrightarrow} S_k)$
- A6: $S_k| \equiv (U_u \overset{SK_{uk}}{\longleftrightarrow} S_k)$
- A7: $RC| \equiv S_k| \equiv (U_u \overset{SK_{uk}}{\longleftrightarrow} S_k)$
- A8: $U_u| \Rightarrow R_u$
- A9: $RC| \Rightarrow r_n$
- A10: $S_k| \Rightarrow R_s$

6.4 Goals

- G1: $S_k| \equiv (U_u \overset{SK_{uk}}{\longleftrightarrow} S_k)$
- G2: $S_k| \equiv U_u| \equiv (U_u \overset{SK_{uk}}{\longleftrightarrow} S_k)$
- G3: $U_u| \equiv (U_u \overset{SK_{uk}}{\longleftrightarrow} S_k)$
- G4: $U_u| \equiv S_k| \equiv (U_u \overset{SK_{uk}}{\longleftrightarrow} S_k)$

The protocol's generic form is illustrated as under:

- **Messages(1)** $U_u \rightarrow RC$:$\{DID_u, H_u, G_u, T_1, SID_k\}$
- **Messages(2)** $RC \rightarrow S_k$:$\{E_{X_{RS_k}}(X'_u, R_u, ID_u, H_{R_c}, SID_k, T_1), T_2, SID_k\}$
- **Messages(3)** $S_k \leftarrow RC$:$\{M_x, H''_{R_c}, T_3, T_u\}$
- **Messages(4)** $RC \leftarrow U_u$:$\{M_x, H''_{R_c}, T_3, T_u, RID_u\}$

The idealized forms of the protocol are designed as follows:

– Considering the message 1 and applying seeing rule,

$$S_1 : RC \lhd \{(PID_u)_{A_u}, (ID_u, G_u, R_u, T_1, SID_k, X_u), (X_u, ID_u, SID_k, T_1)_{R_u}, T_1, SID_k\} \quad (22)$$

– Considering the message 2 and applying the seeing rule,

$$S_2 : S_k \lhd \{X'_u, R_u, ID_u, H_{Rc}, SID_k, T_1)_{X_{RS_k}}, T_2, SID_k\} \quad (23)$$

– Considering the message 3 and applying the seeing rule,

$$S_3 : RC \lhd \{(ID_u, X_u, R_u, T_3)_{R_s}, (R_s, M_x, T_u, ID_u, T_3), T_3, T_u\} \quad (24)$$

– Considering the message 4 and applying seeing rule,

$$S_4 : U_u \lhd \{(ID_u, X_u, R_u, T_3)_{R_s}, (R_s, M_x, T_u, ID_u, T_3), T_3, T_u, (ID_u, r_n)_{X_c}\} \quad (25)$$

6.5 Protocol analysis

The main security proofs are consist of the following steps:

– According to (S_1, $A5$) and message meaning rule,

$$BN1 : RC| \equiv \{(PID_u)_{A_u}, (ID_u, G_u, R_u, T_1, SID_k, X_u), (X_u, ID_u, SID_k, T_1)_{R_u}, T_1, SID_k\} \quad (26)$$

– According to ($BN1$, $A1$), freshness conjuncatenation and nonce verification rule,

$$BN2 : RC| \equiv U_u| \equiv \{(PID_u)_{A_u}, (ID_u, G_u, R_u, T_1, SID_k, X_u), (X_u, ID_u, SID_k, T_1)_{R_u}, T_1, SID_k\} \quad (27)$$

– According to ($A8$, $BN1$, $BN2$) and jurisdiction rule,

$$BN3 : RC| \equiv \{(PID_u)_{A_u}, (ID_u, G_u, R_u, T_1, SID_k, X_u), (X_u, ID_u, SID_k, T_1)_{R_u}, T_1, SID_k\} \quad (28)$$

– According to (S_2, $A5$) and message meaning rule,

$$BN4 : S_k| \equiv \{(X'_u, R_u, ID_u, H_{Rc}, SID_k, T_1)_{X_{RS_j}}, T_2, SID_k\} \quad (29)$$

– According to ($A2$, $BN4$), freshness conjuncatenation and nonce Verification rule,

$$BN5 : S_k| \equiv RC| \equiv \{(X'_u, R_u, ID_u, H_{Rc}, SID_k, T_1)_{X_{RS_j}}, T_2, SID_k\} \quad (30)$$

– According to ($BN4$, $BN5$) and jurisdiction rule,

$$BN6 : S_k| \equiv \{(X'_u, R_u, ID_u, H_{Rc}, SID_k, T_1)_{X_{RS_j}}, T_2, SID_k\} \quad (31)$$

– According to ($A4$, $BN5$, $BN6$) and session key rule,

$$BN7 : S_k| \equiv U_u| \equiv (U_u \overset{SK_{uk}}{\longleftrightarrow} S_k) \ \textbf{Goal 2} \quad (32)$$

– According to ($A8$, $BN7$) and jurisdiction rule,

$$BN8 : S_k| \equiv (U_u \overset{SK_{uk}}{\longleftrightarrow} S_k) \ \textbf{Goal 1} \quad (33)$$

– According to (S_3, $A7$) and message meaning rule,

$$BN9 : RC| \equiv \{(ID_u, X_u, R_u, T_3)_{R_s}, (R_s, M_x, T_u, ID_u, T_3), T_3, T_u\} \quad (34)$$

– According to ($A3$, $BN9$) freshness conjuncatenation and nonce verification rule,

$$BN10 : RC| \equiv S_k| \equiv \{(ID_u, X_u, R_u, T_3)_{R_s}, (R_s, M_x, T_u, ID_u, T_3), T_3, T_u\} \quad (35)$$

– According yo ($A10$, $BN9$, $BN10$) and jurisdiction rule,

$$BN11 : RC| \equiv \{(ID_u, X_u, R_u, T_3)_{R_s}, (R_s, M_x, T_u, ID_u, T_3), T_3, T_u\} \quad (36)$$

– According to (S_4, $A7$) and message meaning rule,

$$BN12 : U_u| \equiv \{(ID_u, X_u, R_u, T_3)_{R_s}, (R_s, M_x, T_u, ID_u, T_3), T_3, T_u, (ID_u, r_n)_{X_c}\} \quad (37)$$

– According to ($A2$, $BN12$), freshness conjuncatenation and nonce verification rule,

$$BN13 : U_u| \equiv RC| \equiv \{(ID_u, X_u, R_u, T_3)_{R_s}, (R_s, M_x, T_u, ID_u, T_3), T_3, T_u, (ID_u, r_n)_{X_c}\} \quad (38)$$

– According to ($A9$, $BN12$, $BN13$) and jurisdiction rule,

$$BN14 : U_u| \equiv \{(ID_u, X_u, R_u, T_3)_{R_s}, (R_s, M_x, T_u, ID_u, T_3), T_3, T_u, (ID_u, r_n)_{X_c}\} \quad (39)$$

– According to ($A6$, $BN13$, $BN14$) and session key rule,

$$BN15 : U_u| \equiv S_k| \equiv (U_u \overset{SK_{uk}}{\longleftrightarrow} S_k) \textbf{ Goal 4} \quad (40)$$

– According to ($A9$, $BN15$) and jurisdiction rule,

$$BN16 : U_u| \equiv (U_u \overset{SK_{uk}}{\longleftrightarrow} S_k) \textbf{ Goal 3} \quad (41)$$

6.6 Discusion on functional security

Following subsection solicit brief discussions on several security features and resistance to known attacks provided by the proposed scheme.

6.6.1 Anonymity and untraceability

In the authentication protocol, user anonymity and untraceability are substantial aspects and if anonymity is broken, an adversary A_{adv} can easily recover sensitive information of the legitimate user like his current location, moving tracks, a personal record and social circle, etc. In the registration phase RC encrypt the identity with random number $E_{X_c}(ID_u||r_o)$ by using his own secret key X_c. SC_u does not store this pseudo identity directly, as it is hidden by PID_u, So even if the smart card was stolen by A_{adv} he will still be incapable to get the identity of the user. Moreover, after each successful authentication request, this pseudo-identity is dynamically changed. Therefore, the proposed protocol provides anonymity and untreceability.

6.6.2 Impersonation attacks

To act as RC an A_{adv} required the secret key X_c of RC, which is hash with user identity $h(ID_u||X_c)$, to computes the session key $SK = h(X'_u||ID_u||SID_k||R_s||R_u)$ an A_{adv}

also requires to first computes $X_u = h(ID_u||X_c)$. In addition X_u is also used in the construction of RC signature that is, $X'_u = h(X_u||ID_u||SID_k||T_1)$. So without secret key X_c an A_{adv} does not impersonate themselves as RC. Similarly to act as legitimate user an A_{adv} will required a valid login request that is,$\{DID_u, H_u, G_u, T_1, SID_k\}$. To get all these values an A_{adv} needs the user credential like Password PW_u as well as biometric BIO_u.

6.6.3 Replay attack

Our protocol combat replay attack against all the login and authentication Messages. Suppose an A_{adv} replays a past message that is $\{DID_u, H_u, G_u, T_1, SID_k\}$. then on receiving side RC will always check the time-stamp T_1, as T_1 is outdated, RC will considered as replay, they neglect the message request.

6.6.4 Stolen verifier attack

Our protocol is fully secured against stolen verifier attack. RC encrypt shared key $E_{X_c}(X_{RS_k})$ using their own secret key X_c to handle stored verifier table, so adversary does not extract anything without knowing the X_c.

6.6.5 Privileged insider attack

The proposed protocol successfully prevents a privilege insider attack. In the registration phase ID_u and $A_u = h(N_1||PW_u||ID_u||CT_u)$ are sent to RC, where Password PW_u identity ID_u a random number N_1 and cancel able template CT_u are protected by one way hash function. So it is impossible for an insider to guess these value.

6.6.6 Password guessing attacks

The proposed protocol is fully secured against the Password Guessing attack. Suppose RC take the screen shot of the user sensitive parameters like $\{TP_u, H_u, R, P, h(.)\aleph_{enc}(\cdot), \aleph_{dec}(\cdot)\ Y_u, PID_u, E_u\}$ which is stored on user smart card. Then they still requires the cancel-able transformation parameter CT_u along with N_1. Moreover, an A_{adv} still needs to guess identity ID_u and Password PW_u of user, if they unfortunately gets the N_1 and CT_u.

6.6.7 Denial of services attack

Our protocol is fully protected against the denial of services. SC_u checks the validity of identity ID_u, Password PW_u and template CT_u. If A_{adv} or legitimate user try to enter the incorrect values, then the SC_u just simply cancel the request.

6.6.8 Perfect forward secrecy

The proposed protocol poses the prefect forward secrecy. The shared session key $SK_{uk} = h(X'_u||ID_u||SID_k||R_s||R_u)$ incorporate a random number R_u used by the user. Suppose if RC signature X'_c is exposed to some A_{adv} he will not be able to computes previously shared session keys.

6.6.9 Resolve the scalability issues

In previous protocol the smart card store the $AM_{uk} = (AM_k \oplus k'_u) \oplus r_u$, $BM_{uk} = (BM_k \oplus k'_u) \oplus r_u$ for every server $1 \leq k \leq (n+n')$, which is insufficient to store (AM_k) within smart card due to its small magnetic chip which has limited storage. In the proposed protocol there is no such parameter which stored the information of a server.

6.7 AVISPA based security simulation

In this section, we analyze proposed protocol security using formal simulation tool AVISPA [3]. AVISPA is used for security verification.

AVISPA implements the HLPSL language which is then translated into the intermediate format (IF) with the help of translator known as "hlpsl2if". Four back ends are used by IF, to check security goals, is satisfied or disrupt. The output shows safe, unsafe or unsatisfactory. Details are mentioned in [3]. We define the three basic role i.e. role of user U_u, role of registration center RC and role of server S_k along with the session (between these participant), environment role and goals Fig. 5, 6, 7 and 8 are stated in HLPSL. The results of AVISPA are shown in Fig. 9 which tells that proposed protocol is secure against man in the middle attack as well as replay attack. The OFMC back end shows the parse time: 0.00

```
role role_USERS(USERS,RC,SERVER:agent,XRSJ,XRS,XUR,XC
:symmetric_key,H:hash_func,SND,RCV:channel(dy))
played_by USERS

def=

local
State:nat,PID,RU,A,N1,CT,PW,NR,XU,ID,SIDJ,T1,%USERS
GU,XUN,T2,XRN,%RC RS,T3,MX,TU:text,%SERVER F:hash_func
init
State := 0
transition

3. State=0 /\ RCV(start) =|> State':=1 /\ SND({F(N1.
PW.ID.CT).ID}_XUR) 4. State=1 /\ RCV({xor(XU,A).{ID.
NR'}_XC'}_XUR) =|> State':=2 /\ secret(RU',sec_2,{
USERS,RC,SERVER}) /\ GU':=new() /\ PID':=new() /\
RU':=new() /\secret(PID',sec_1,{RC}) /\ SND({PID'}_
XC'.xor(RU',F(XU.ID.SIDJ.T1)).F(ID.GU'.XU.RU'.T1.SIDJ
).T1.SIDJ) 8.State=2 /\ RCV(xor(RS',F(ID.XUN'.RU'.T3'
)).F(RS'.MX'.TU'.ID.T3').T3'.TU'.T1)=|> State':=3 /\
witness(USERS,SERVER,auth_6,RS') /\ secret(RS',sec_3,
{SERVER,RC,USERS}) /\ secret(RU',sec_2,{USERS,RC,SERVER
}) /\ SND(F(XUN'.ID.SIDJ.RS'.RU))

end role
```

Fig. 5 Role specification of user

```
role role_SERVER(USERS,RC,SERVER:agent,XRSJ,XRS,XUR,XC
:symmetric_key,H:hash_func,SND,RCV:channel(dy))
played_by SERVER

def=

local
State:nat,PID,RU,A,N1,CT,PW,NR,XU,ID,SIDJ,T1,%USERS GU,
XUN,T2,XRN, %RC RS,T3,MX,TU:text, %SERVER F:hash_func
init
State := 0
transition

1. State=0 /\RCV(start) =|> State':=1 /\SND({SIDJ}_XRS)
2. State=1 /\RCV({F(SIDJ.NR')}_XRS)=|> State':=2
6. State=2 /\RCV({F(XU'.ID'.SIDJ.T1').RU'.ID'.F(XRN'.
XUN'.ID'.SIDJ.T2').SIDJ.T1'.T2'}_XRSJ)=|> State':=3
/\witness(SERVER,RC,auth_5,XRN') /\secret(RU',sec_2,{
USERS,RC,SERVER}) /\MX':=new() /\RS':=new() /\secret(RS
',sec_3,{SERVER,RC,USERS}) /\SND(xor(RS',F(ID'.XUN'.RU'
.T3)).F(RS'.MX'.TU.ID'.T3).T3.TU.T1') 9.State=3 /\RCV
(F(XUN.ID.SIDJ.RS'.RU'))=|> State':=4 /\secret(RU',
sec_2,{USERS,RC,SERVER}) /\secret(RS',
sec_3,{SERVER,RC,USERS})

end role
```

Fig. 6 Role specification of server

seconds, the search time: 42.16 seconds, the number of visited nodes is 3344 and the depth 12 plies. whereas ATSE analyzes 8 states, the translation time is 0.98 seconds. Hence, form this results it is shown our protocol provides better security against Barman et al.'s protocol [6]. The search and translation time is slightly high compared to Barman et al.'s protocol, because the number of visited nodes depth of proposed protocol is greater than the previous protocol.

7 Comparisons

In this section, we show the performance and security comparisons of the proposed protocol with some related multi-server authentication protocols [1, 2, 6, 13, 18, 31, 36, 46]. attacks.

7.1 Security and functionality comparisons

The security and functionality comparison of proposed scheme with related schemes is solicited in Table 1 under the DY and CK adversarial model as described in subsection 2.5. The security comparisons show that only proposed scheme provides resistance to all known attacks and fulfills related security features; whereas, all the competing schemes either lacks one or more security features or vulnerable to some security attacks.

```
role role_RC(USERS,RC,SERVER:agent,XRSJ,XRS,XUR,XC:
symmetric_key,H:hash_func,SND,RCV:channel(dy))
played_by RC
def=
local

State:nat,PID,RU,A,N1,CT,PW,NR,XU,ID,SIDJ,T1,%USERS
GU,XUN,T2,XRN, %RC RS,T3,MX,TU:text,%SERVER F:hash
_func

init
State := 0
transition

1.State=0 /\RCV({SIDJ}_XRS)=|>State':=1/\NR':=new()
/\SND({F(SIDJ.NR')}_XRS)3.State=1/\RCV({F(N1'.PW'.ID
.CT').ID}_XUR)=|>State':=2/\A':=new()/\XU':=new()/\
SND({xor(XU',A').{ID.NR}_XC}_XUR)5.State=2/\RCV({PID
'}_XC.xor(RU',F(XU.ID.SIDJ.T1')).F(ID.GU'.XU.RU'.T1'
.SIDJ).T1'.SIDJ)=|>State':=3/\secret(RU',sec_2,{USERS
,RC,SERVER})/\secret(PID',sec_1,{RC})/\XUN':=new()/\
SND({F(XU.ID.SIDJ.T1').RU'.ID.F(XRN.XUN'.ID.SIDJ.T2).
SIDJ.T1'.T2}_XRSJ)7.State=3/\RCV(xor(RS',F(ID.XUN.RU'
.T3')).F(RS'.MX'.TU'.ID.T3').T3'.TU'.T1)=|> State':=4
/\secret(RS',sec_3,{SERVER,RC,USERS})/\secret(RU',
sec_2,{USERS,RC,SERVER})/\SND(xor(RS',F(ID.XUN.RU.T3')
).F(RS'.MX'.TU'.ID.T3').T3'.TU'.T1)
end role
```

Fig. 7 Role specification of Rc

7.2 Computation cost

In this subsection, we compare our protocol with the existing multi-server authentication protocols considering the computation cost of login and authentication phases. The following notation used for computation cost describe below:

- RT_h: one-way cryptographic hash cost
- RT_{bh}: bio-hashing cost
- RT_{fe}: fuzzy extractor cost
- RT_{fcs}: fuzzy commitment cost
- RT_{ecm}: ecc point multiplication cost
- RT_{asm}: asymmetric key encryption/decryption cost
- RT_{sed}: cost of block cipher encryption

As per the experimental results disclosed in [24], RT_h = 0.0023 ms, RT_{sed} = 0.0046 ms, RT_{ecm} = 2.226 ms and RT_{asm} = 0.0046 ms. Furthermore, $RT_{fe} = RT_{ecm}$, we also assume $RT_{bh} = RT_{ecm}$ and $RT_{fcs} = RT_{ecm}$. Although our protocol has slightly high computation cost compared to Barman et al. [6], but the security level of our protocol is high. The comparisons are briefly shown in Table 2.

```
role session(USERS,RC,SERVER:agent,XRSJ,XRS,XUR,XC:symm
etric_key, H:hash_func)
def=
local
SND1,RCV1,SND2,RCV2,SND3,RCV3:channel(dy)
composition
role_USERS(USERS,RC,SERVER,XRSJ,XRS,XUR,XC,H,SND1,RCV1)
/\role_RC(RC,USERS,SERVER,XRSJ,XUR,XRS,XC,H,SND2,RCV2)
/\role_SERVER(SERVER,USERS,RC,XRSJ,XUR,XRS,XC,H,SND3,
RCV3)
end role
role environment()
def=
const x,xrs,xur,xl:symmetric_key,user,rc,server:agent,h
,f:hash_func,t1,t2,t3,tu:text,sec_1,sec_2,sec_3,auth_4,
auth_5,auth_6:protocol_id
intruder_knowledge = {user,rc,server,h,t1,t2,t3,tu}
composition
session1(user,rc,server,x,xrs,xur,xl,h)/\session2(i,rc,
server,x,xrs,xur,xl,h)/\session3(user,i,server,x,xrs,
xur,xl,h)/\session4(user,rc,i,x,xrs,xur,xl,h)
end role
goal
secrecy_of sec_1 secrecy_of sec_2 secrecy_of sec_3
authentication_on auth_4 authentication_on auth_5
authentication_on auth_6
end goal
environment()
```

Fig. 8 Role specification of session/Goal

OFMC	ATSE
————Output of OFMC————	————Output of ATSE————
% OFMC	
% Version of 2006/02/13	
SUMMARY	SUMMARY
SAFE	SAFE
DETAILS	DETAILS
BOUNDED_NUMBER_OF_SESSIONS	BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL	TYPED_MODEL
/home/span/span/testsuite/results/todaynew.if	/home/span/span/testsuite/results/todaynew.if
GOAL	GOAL
as_specified	As Specified
BACKEND	BACKEND
OFMC	CL-AtSe
COMMENTS	
STATISTICS	STATISTICS
parse time: 0.00 seconds	Analysed : 8 states
search time: 42.16 seconds	Reachable : 0 states
visitedNodes: 3344 nodes	Translation: 0.98 seconds
depth: 12 plies	Computation: 0.00 seconds

Fig. 9 Results of OFMC and CL-AtSe backends

Table 1 Security and functionality features comparison

Property/Feature	Our	[6]	[13]	[2]	[46]	[36]	[18]	[31]	[1]
FUN_1	✓	χ	χ	✓	✓	χ	χ	χ	χ
FUN_2	✓	✓	✓	✓	χ	✓	✓	✓	✓
FUN_3	✓	✓	✓	χ	✓	✓	✓	✓	✓
FUN_4	✓	χ	✓	✓	χ	✓	✓	✓	✓
FUN_5	✓	χ	✓	✓	χ	✓	✓	✓	✓
FUN_6	✓	✓	✓	✓	χ	✓	✓	✓	✓
FUN_7	✓	χ	✓	✓	✓	χ	✓	✓	✓
FUN_8	✓	✓	✓	✓	✓	✓	✓	✓	✓
FUN_9	✓	✓	✓	χ	χ	✓	✓	✓	✓
FUN_{10}	✓	✓	✓	✓	✓	✓	χ	χ	✓
FUN_{11}	✓	✓	✓	χ	χ	✓	✓	χ	χ

FUN_1: user anonymity violation and untraceability; FUN_2: three-factor security feature; FUN_3: error detection mechanism; FUN_4: participant having mutual authentication; FUN_5: exchange of session key; FUN_6: Password update security; FUN_7: resistance against stolen smart card attack; FUN_8: resistance against offline Password Guessing; FUN_9: resistance against replay attack; FUN_{10}:resistance against forgery attack; FUN_{11}: resistance against privileged-insider attack.

✓: a protocol safeguard the security functionality feature; χ: a protocol is lack of the security functionality feature.

7.3 Communication cost

In this subsection, we evaluate and compare the communication cost of proposed with existing protocols. During the login and authentication phases, the communication cost is computed by the total number of bits which is transmitted to other parties in the network, over a protected channel. We are assuming the "SHA-1" hash function is used, which has the cost of 160 bits [7], in the symmetric key encryption/decryption, has the cost of 256 bits of length [26], time stamp is 32 bits of length, an elliptic curve point $P = (P_a, P_b)$ is 160 length of bits, where P_a and P_b is x and y coordinate of P point. Furthermore the security of RSA [45] public key cryptosystem is 1024-bit which is comparable to ECC (elliptic

Table 2 Computation costs comparison

Protocol	Bits	Computation cost	Time(ms)
Chuang-Chen [13]	1024	$17RT_h$	0.0391
Amin-Biswas[2]	1920	$RT_{bh} + 18RT_h$	2.2674
Sood [46]	2112	$31RT_h$	0.0713
Mishra [36]	1280	$18RT_h$	0.0414
He-Wang [18]	3520	$21RT_h + 8RT_{ecm}$	17.856
Lu [31]	1226	$RT_{bh} + 15RT_h$	2.2605
Ali-Pal [1]	1664	$13RT_h + RT_{bh} + 2RT_{asm}$	2.2651
Barman [6]	896	$RT_{fcs} + 17RT_h$	2.2651
Our	1804	$RT_{fcs} + 19RT_h + 3RT_{sed}$	2.2789

curve cryptography) of 160-bits of length [5]. In the proposed protocol, the communication cost for the login request message $\{DID_u, H_u, G_u, T_1, SID_k\}$, which is transmitted from a user U_u to theRC has cost of (160+160+160+32+32) = 544 bits of length and the message $\{E_{X_{RS_k}}(X'_u, R_u, ID_u, H_{R_c}, SID_k, T_1), SID_k, T_2\}$ transmitted to server S_k from RC is (256+32+32) = 332 bits and the message transmitted to RC from server S_k is $\{M_x, H''_{R_c}, T_3, T_u, \}$ (160+160+32+32) = 384 bits and message transmitted to U_u from RC is $\{M_x, H''_{R_c}, T_3, T_u, RID_u\}$ (160+160+32+32+160) = 544 bits hence, the total number of bits for communication is (544+332+384+544) = 1804 bits. The comparison results are shown in Table 2. The high communication cost as compared with Barman et al. is due to the communication of dynamic identity from server to user in each authentication request inorder to provide user anonymity.

8 Conclusion

The single signin/multiserver environments can apprehend the security and privacy needs of intelligent multimedia networks to encompass large number of applications/networks using single credentials. In 2018, Barman et al. proposed such multi-server authentication system. In this article, we proved some security weaknesses of Barman et al.'s protocol. We then proposed a new enhanced authentication scheme for multi-server scenarios. Based on three factors including biometrics, the proposed scheme makes use of fuzzy commitment for correcting errors in imprinted biometrics in noisy environments. Proposed scheme provides anonymity and privacy alongwith other security properties and resists the known attacks. The BAN logic based formal as well as informal security discussion proves the robustness of the proposed scheme. Moreover, the automated AVISPA protocol also validates the security claims. The proposed scheme completes an authentication cycle in just 2.2789 milli seconds.

References

1. Ali R, Pal AK (2017) Three-factor-based confidentiality-preserving remote user authentication scheme in multi-server environment. Arab J Sci Eng 42(8):3655–3672
2. Amin R, Biswas G (2015) A novel user authentication and key agreement protocol for accessing multi-medical server usable in tmis. Journal of medical systems 39(3):33
3. Armando A, Basin D, Cuellar J, Rusinowitch M, Viganò L (2006) Avispa: automated validation of internet security protocols and applications ERCIM News 64(January)
4. Arshad H, Nikooghadam M (2016) An efficient and secure authentication and key agreement scheme for session initiation protocol using ecc. Multimedia Tools and Applications 75(1):181–197
5. Barker E, Barker W, Burr W, Polk W, Smid M (2012) Recommendation for key management part 1: General (revision 3). NIST special publication 800(57):1–147
6. Barman S, Das AK, Samanta D, Chattopadhyay S, Rodrigues JJ, Park Y (2018) Provably secure multi-server authentication protocol using fuzzy commitment. IEEE Access 6(38):578–38,594
7. Burrows J (2015) Secure hash standard. fips pub 180-1, national institute of standards and technology (nist), us department of commerce april 1995
8. Burrows M, Abadi M, Needham RM (1989) A logic of authentication. Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences 426(1871):233–271
9. Canetti R, Krawczyk H (2001) Analysis of key-exchange protocols and their use for building secure channels. In: International conference on the theory and applications of cryptographic techniques, pp 453–474. Springer
10. Chaudhry SA, Naqvi H, Khan MK (2018) An enhanced lightweight anonymous biometric based authentication scheme for tmis. Multimedia Tools and Applications 77(5):5503–5524

11. Chen CM, Wang KH, Yeh KH, Xiang B, Wu TY (2019) Attacks and solutions on a three-party password-based authenticated key exchange protocol for wireless communications. Journal of Ambient Intelligence and Humanized Computing 10(8):3133–3142
12. Chen CM, Xiang B, Liu Y, Wang KH (2019) A secure authentication protocol for internet of vehicles. IEEE Access 7(12):047–12,057
13. Chuang MC, Chen MC (2014) An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics. Expert Systems with Applications 41(4):1411–1418
14. Debiao H, Jianhua C, Rui Z (2012) A more secure authentication scheme for telecare medicine information systems. Journal of medical systems 36(3):1989–1995
15. Dolev D, Yao A (1983) On the security of public key protocols. IEEE Transactions on information theory 29(2):198–208
16. Ghani A, Mansoor K, Mehmood S, Chaudhry SA, Rahman AU, Najmus Saqib M (2019) Security and key management in iot-based wireless sensor networks: an authentication protocol using symmetric key. Int J Commun Syst 32(16):e4139
17. Hao F, Anderson R, Daugman J (2006) Combining crypto with biometrics effectively. IEEE transactions on computers 55(9):1081–1088
18. He D, Wang D (2014) Robust biometrics-based authentication scheme for multiserver environment. IEEE Syst J 9(3):816–823
19. Hussain S, Chaudhry SA (2019) Comments on "biometrics-based privacy-preserving user authentication scheme for cloud-based industrial internet of things deployment". IEEE Internet of Things Journal 6(6):10,936–10, 940
20. Irshad A, Sher M, Chaudhry SA, Xie Q, Kumari S, Wu F (2018) An improved and secure chaotic map based authenticated key agreement in multi-server architecture. Multimedia Tools and Applications 77(1):1167–1204
21. Irshad A, Sher M, Nawaz O, Chaudhry SA, Khan I, Kumari S et al (2017) A secure and provable multi-server authenticated key agreement for tmis based on amin. scheme. Multimedia Tools and Applications 76(15):16,463–16,489
22. Juang WS, Chen ST, Liaw HT (2008) Robust and efficient password-authenticated key agreement using smart cards. IEEE Trans Ind Electron 55(6):2551–2556
23. Juels A, Wattenberg M (1999) A fuzzy commitment scheme. In: Proceedings of the 6th ACM conference on Computer and communications security, pp 28–36. ACM
24. Kilinc HH, Yanik T (2014) A survey of sip authentication and key agreement schemes. Communications Surveys & Tutorials, IEEE 16(2):1005–1023
25. Kocher P, Jaffe J, Jun B (1999) Differential power analysis. In: Annual international cryptology conference, pp 388–397. Springer
26. Kumar V, Ahmad M, Kumari A, Kumari S, Khan M (2019) Sebap: a secure and efficient biometric-assisted authentication protocol using ecc for vehicular cloud computing. Int J Commun Syst, pp e4103. https://doi.org/10.1002/dac.4103
27. Lamport L (1981) Password authentication with insecure communication. Commun ACM 24(11):770–772
28. Lee J, Ryu S, Yoo K (2002) Fingerprint-based remote user authentication scheme using smart cards. Electron Lett 38(12):554–555
29. Lin CH, Lai YY (2004) A flexible biometrics remote user authentication scheme. Computer Standards & Interfaces 27(1):19–23
30. Lin H, Wen F, Du C (2017) An anonymous and secure authentication and key agreement scheme for session initiation protocol. Multimedia Tools and Applications 76(2):2315–2329
31. Lu Y, Li L, Yang X, Yang Y (2015) Robust biometrics based authentication and key agreement scheme for multi-server environments using smart cards. PLoS One 10(5):e0126,323
32. Lwamo NM, Zhu L, Xu C, Sharif K, Liu X, Zhang C (2019) Suaa: a secure user authentication scheme with anonymity for the single & multi-server environments. Information Sciences 477:369–385
33. Mansoor K, Ghani A, Chaudhry SA, Shamshirband S, Ghayyur SAK (2019) Securing iot based RFID systems: a robust authentication protocol using symmetric cryptography. Sensors 19:21. https://doi.org/10.3390/s19214752
34. Messerges TS, Dabbish EA, Sloan RH (2002) Examining smart-card security under the threat of power analysis attacks. IEEE transactions on computers 51(5):541–552
35. Mir O, Nikooghadam M (2015) A secure biometrics based authentication with key agreement scheme in telemedicine networks for e-health services. Wirel Pers Commun 83(4):2439–2461
36. 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

37. Mitchell CJ, Tang Q (2005) Security of the lin-lai smart card based user authentication scheme Technical Report
38. Nguyen NT, Chang CC (2018) A biometric-based authenticated key agreement scheme for session initiation protocol in ip-based multimedia networks. Multimedia Tools and Applications 77(18):23,909–23,947
39. Qi M, Chen J (2017) An efficient two-party authentication key exchange protocol for mobile environment. Int J Commun Syst 30(16):e3341
40. Qi M, Chen J (2018) New robust biometrics-based mutual authentication scheme with key agreement using elliptic curve cryptography. Multimedia Tools and Applications 77(18):23,335–23,351
41. Ratha NK, Chikkerur S, Connell JH, Bolle RM (2007) Generating cancelable fingerprint templates. IEEE Transactions on pattern analysis and machine intelligence 29(4):561–572
42. Ravanbakhsh N, Nazari M (2018) An efficient improvement remote user mutual authentication and session key agreement scheme for e-health care systems. Multimedia Tools and Applications 77(1):55–88
43. Reddy AG, Das AK, Odelu V, Ahmad A, Shin JS (2018) A privacy preserving three-factor authenticated key agreement protocol for client–server environment. Journal of Ambient Intelligence and Humanized Computing 10(2):661–680
44. Reddy AG, Yoon EJ, Das AK, Odelu V, Yoo KY (2017) Design of mutually authenticated key agreement protocol resistant to impersonation attacks for multi-server environment. IEEE access 5:3622–3639
45. Rivest RL, Shamir A, Adleman L (1978) A method for obtaining digital signatures and public-key cryptosystems. Commun ACM 21(2):120–126
46. Sood SK, Sarje AK, Singh K (2011) A secure dynamic identity based authentication protocol for multi-server architecture. J Netw Comput Appl 34(2):609–618
47. Wang C, Zhang X, Zheng Z (2016) Cryptanalysis and improvement of a biometric-based multi-server authentication and key agreement scheme. Plos one 11(2) e0149:173
48. Wu ZY, Lee YC, Lai F, Lee HC, Chung Y (2012) A secure authentication scheme for telecare medicine information systems. Journal of medical systems 36(3):1529–1535
49. Zhu Z (2012) An efficient authentication scheme for telecare medicine information systems. Journal of medical systems 36(6):3833–3838

Affiliations

Hafeez Ur Rehman[1] · Anwar Ghani[1] · Shehzad Ashraf Chaudhry[2] · Mohammed H. Alsharif[3] · Narjes Nabipour[4]

Hafeez Ur Rehman
hafeezkami@gmail.com

Shehzad Ashraf Chaudhry
ashraf.shehzad.ch@gmail.com

Mohammed H. Alsharif
malsharif@sejong.ac.kr

Narjes Nabipour
narjesnabipour@duytan.edu.vn

[1] Department of Computer Science and Software Engineering, International Islamic University, Islamabad, Pakistan

[2] Department of Computer Engineering, Faculty of Engineering and Architecture, Istanbul Gelisim University, 34310 Istanbul, Turkey

[3] Department of Electrical Engineering, College of Electronics and Information Engineering, Sejong University, 209 Neungdong-ro, Gwangjin-gu, Seoul 05006, Korea

[4] Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam