A Three Factor Remote User Authentication Scheme Using Collision Resist Fuzzy Extractor in Single Server Environment

Debasis Giri and Tanmoy Maitra

1) Department of Computer Science and Engineering, Haldia Institute of Technology, Haldia-721657, India
2) Department of Computer Science and Engineering, Jadavpur University, Kolkata-700032, India

Abstract. Due to rapid growth of online applications, it is needed to provide such a facility by which communicators can get the services by applying the applications in a secure way. As communications are done through an insecure channel like Internet, any adversary can trap and modify the communication messages. Only authentication procedure can overcome the aforementioned problem. Many researchers have proposed so many authentication schemes in this literature. But, this paper has shown that many of them are not usable in real world application scenarios because, the existing schemes cannot resist all the possible attacks. Therefore, this paper has proposed a three factor authentication scheme using hash function and fuzzy extractor. This paper has further analyzed the security of the proposed scheme using random oracle model. The analysis shows that the proposed scheme can resist all the possible attacks. Furthermore, comparison between proposed scheme and related existing schemes shows that the proposed scheme has better trade-off among storage, computational and communication costs.

1 Introduction

Nowadays, online applications like bill payment, banking system, telecare medical system, social networking, e-voting and so on are rapidly used for their easy and efficient access. All the applications are going through a client/server environment and communications are done through public channel like Internet because of availability of public bandwidth. Therefore, all communication messages of the applications are public. As a result, any one can trap and modify the communication messages. For this purpose, in such communication system, authentication scheme is rapidly used by which after verifying the communicators and their messages, a secure communication can be done through public channel. In this regard, smart card and password based user authentication scheme is very much popular for online communication system. However, a suitable smart card based authentication scheme should satisfy the following property:

Low cost: Computational cost, communication cost and smart card storage cost are three basic network parameters to measure performances of an authentication scheme. Therefore, it has to considered that these three parameters are reduced as much possible when an authentication scheme is going to be designed.

Prevention of security attacks: During communication through public channel, it is needed to secure the message from outsider adversary. An authentication scheme needs to be designed such a way by which from the communication messages, any adversary will unable to extract useful information and the scheme can resist security attacks.

Session key agreement: After authentication in both ways, a common secret session key is needed to carry on the communications within the same session after encrypting the paintext messages by the session key.

Mutual authentication: Mutual authentication is a essential property for an authentication scheme by which all the communicators can authenticate or verify each other.

Efficient login procedure: For an authentication scheme, it is needed that smart card checks the wrong inputs before going to send a login message to server. By checking wrong inputs in login phase, extra communication overhead can be avoided. Therefore, it is crucial property of a good authentication scheme.
**Efficient password change procedure:** A valid user can change the password freely and securely without taking any help from server. For this purpose, the smart card should verify the old password in the password change phase, so that an unauthorized user cannot change the authorized user's password even if it gets the valid users' smart card. The afore mentioned property should present in an authentication scheme.

**Traceability:** In an authentication scheme, it is also needed to trace the sender of a message for corresponding receiver [1]. Otherwise, any one can mount denial of service (DoS) attack.

Furthermore, to enhanced the security of password based authentication scheme, biometric feature (i.e., fingerprint, iris, retina etc.) [2] is added with the password. Therefore, this research focus to design a biometric plus password based efficient authentication scheme by considering all the afore mentioned properties.

### 1.1 Literature Survey

A brief survey of existing authentication schemes is described in this section. First Lamport [3] proposed a password-based authentication scheme based on one way hash function. However, Shimizu et al. [4] showed that the Lamport’s scheme [3] suffers from different attacks. After that so many remote user authentication schemes [5–16] have been proposed in this regard which are based on only password. But, the researchers have considered biometric feature [2] with the password to enhance the security label. Therefore, many researchers have proposed biometric and password based authentication schemes in [17–28]. Li and Hwang [17] proposed a biometrics-based remote user authentication scheme in 2010. However, in 2011, Das [18] showed that Li and Hwang’s scheme [17] had flaws in the login phase, authentication phase and password change phase and therefore, Das also proposed an authentication scheme. An [19] showed that Das’s scheme [18] cannot resist the server masquerading attack, user impersonation attack, password guessing attack and insider attack, and so proposed an improved scheme. Li et al. [20] found that An’s scheme [19] suffered from the denial-of-service (DoS) attack, the forgery attack and also did not provide forward secrecy. Furthermore, in 2013, Lee and Hsu [21] pointed out that Das’s scheme [18] is also suffering from privileged insider attack and the off-line password guessing attack. Therefore, Lee and Hsu [21] proposed a biometric based authentication scheme to overcome the weaknesses of Das’s scheme [18]. In 2013, Tan [22] proposed a three-factor authentication scheme. But, Yan et al. [23] pointed out that Tan’s scheme [22] is vulnerable to the Denial-of-Service (DoS) attack. However, recently, Mishra et al. [24] showed that Yan et al.’s scheme [23] suffers from off-line password guessing attack and has inefficient login and password change phases. Huang et al. [29] proposed an authentication scheme based on RSA. But Amin et al. [30] proved that Huang et al.’s scheme [29] unable to protect forgery attack and also introduced an authentication protocol in [30].

### 1.2 Contribution

This paper proposes a three factor authentication scheme using hash function and fuzzy extractor, where three factor means (1) users’ password, (2) users’ biometric and (3) smart card. This paper further analyzes the security of the proposed scheme using random oracle model. The analysis shows that an adversary cannot mount any attacks on the proposed scheme due to hardness of inversion of one-way hash function as well as it has to solve hardness of fuzzy factor. Furthermore, comparison between proposed scheme and related existing schemes shows that the proposed scheme has better trade-off among storage, computational and communication costs. It is a great contribution that the proposed scheme resists all the possible attacks with better trade-off among different costs.

### 1.3 Road Map

This section describes a road map which has been followed throughout this paper. Section 2 briefly introduces some preliminary mathematical concepts for introducing the proposed scheme. Section 3 describes a network model and an adversary model to analyze the proposed scheme. A proposed scheme is described in section 4. Section 5 describes cryptanalysis of the proposed scheme and Section 6 compares the performances of the proposed scheme with previously published schemes. Conclusion of this paper appears in section 7.

### 2 Preliminaries

In this section, a briefly review the basic concepts of cryptographic one-way hash function and collision resist fuzzy extractor are introduced.

**Definition 1.** A collision resistant cryptographic one-way hash function [25, 27] maps a string of arbitrary length to a string of fixed length called the hashed value. It can be symbolized as: \( H : A \rightarrow B \), where \( A \) is a binary string of
arbitrary length and $B$ is a binary string of fixed length $n$. If $Adv_A^{H}(t_1)$ is the advantage to an adversary $\mathcal{A}$ to choose a random pair $(a, b) \in A \times A$ such that $H(a) = H(b)$, where $a \neq b$ for the time duration $t_1$, it can be considered that $Adv_A^{H}(t_1)$ is the probability which is computed over the random choices made by the adversary $\mathcal{A}$ for the time duration $t_1$. Then the cryptographic one-way hash function $H(\cdot)$ is called collision-resistant, if $Adv_A^{H}(t_1) \leq \xi_1$, for any small $\xi_1 > 0$. $Adv_A^{H}(t_1)$ is represented as:

$$Adv_A^{H}(t_1) = \Pr[(a, b) \in R A \times A | (a \neq b) \land H(a) = H(b)],$$

(1)

where $Pr[E]$ denotes the random event $E$.

**Definition 2.** A collision resistant fuzzy extractor [2, 27] can be model as a procedure, known as $Gen$, which takes a binary string say, $B$ of some metric space $M$ as an input, where $M \in \{0, 1\}^k$, for some $k$ bits and produces a random string say, $\phi \in \{0, 1\}^n$, for some $n$ bits and an auxiliary string say, $\theta \in \{0, 1\}^r$, for some $r$ bits, where $r = k$ or $n$ bits. It can be mathematically represented by $Gen : M \to \phi \times \theta$. Another procedure, known as $Rep$, takes a binary string say, $B'$ of the metric space $M \in \{0, 1\}^k$, where $B \neq B'$ and a uniform distribution binary string say, $\theta' \in \{0, 1\}^r$ to produce the random string $\phi' \in \{0, 1\}^n$, symbolized as $Rep : M \times \theta' \to \phi'$. If $Adv_A^{FE}(t_2)$ is the advantage to an adversary $\mathcal{A}$ to choose a pair $(B, B') \in R M \times M$ randomly such that $des(B, B') \leq \delta d$, $Gen(B) = Gen(B')$ and $Rep(B, \theta) = Rep(B', \theta')$, where $\delta d$ is the difference tolerance level and $B \neq B'$ for the time duration $t_2$, it can be considered that $Adv_A^{FE}(t_2)$ is the advantage that is computed over the random choices made by $\mathcal{A}$ for the time duration $t_2$. Then the fuzzy extractor $FE$ is called collision-resistant, if $Adv_A^{FE}(t_2) \leq \xi_2$, for any small $\xi_2 > 0$. $Adv_A^{FE}(t_2)$ is represented as:

$$Adv_A^{FE}(t_2) = \Pr[(B, B') \in R M \times M | (B \neq B') \land des(B, B') \leq \delta d \land Gen(B) = Gen(B') \land Rep(B, \theta) = Rep(B', \theta')],$$

(2)

for all probabilistic polynomial-time algorithms $Gen$ and $Rep$.

**3 Model**

This section will introduce two following models:

**Network Model:** Architecture of the proposed scheme is shown in Figure 1 where, users have to register to a remote server to get their smart card which is known as registration procedure (see Figure 1(a)). Whenever the registered users want to get service from the remote server by accessing their smart card through public channel like Internet, the smart card sends a login request message to the remote server. After verifying the login request message, the remote server sends corresponding reply message to the sender. After receiving the reply message, the corresponding smart card checks the validity of the reply message. Upon receiving correct reply message, both user and the remote server agree for a shared secret session key (See Figure 1(b)).

**Adversary Model:** To analyze the security of the proposed scheme, Dolev-Yao threat model [31] has been considered in which the communicating parties communicate through an insecure channel. Therefore, an adversary $\mathcal{A}$ can trap the transmitted messages over the public or insecure channel, and furthermore he/she can modify, delete or change the contents of the transmitted messages. The adversary $\mathcal{A}$ also obtains the information which are stored
in the user’s smart card by monitoring the power consumption [32, 33]. Generally, identity and password of the user are low entropy in cryptography, that means the adversary can guess the identity and password individually using dictionary attack in polynomial time. But, the adversary cannot guess identity and password simultaneously in on-line/off-line within a polynomial time as pointed out in [34]. According to our adversary model, we consider two following cases:

- **Case 1.** A third party from outside of the system tries to mount various attacks on authentication system as an adversary $A$.
- **Case 2.** A registered user from inside of the system tries to extract secret information of the server by which he/she can mount various attacks on authentication system as an other user or adversary $\tilde{A}$.

## 4 Proposed Scheme

This section proposes an authentication scheme. A nomenclature is given in Table 1 to introduce the proposed scheme. The proposed scheme consists of five phases namely, 1) initialization phase, 2) registration phase, 3) login phase, 4) authentication and key agreement phase and 5) password update phase. The phases are as follows:

<table>
<thead>
<tr>
<th>Table 1. Nomenclature</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Term</strong></td>
</tr>
<tr>
<td>$U_i$</td>
</tr>
<tr>
<td>$RS$</td>
</tr>
<tr>
<td>$pw_i$</td>
</tr>
<tr>
<td>$ID_i$</td>
</tr>
<tr>
<td>$B_i$</td>
</tr>
<tr>
<td>$r_i$</td>
</tr>
<tr>
<td>$y_i$</td>
</tr>
<tr>
<td>$T$</td>
</tr>
<tr>
<td>des$(\cdot)$</td>
</tr>
<tr>
<td>$\delta d$</td>
</tr>
<tr>
<td>$X'$</td>
</tr>
<tr>
<td>$X^*$</td>
</tr>
<tr>
<td>$\delta T$</td>
</tr>
<tr>
<td>$SK_i$</td>
</tr>
<tr>
<td>$H(\cdot)$</td>
</tr>
<tr>
<td>$s$</td>
</tr>
<tr>
<td>$\parallel$</td>
</tr>
<tr>
<td>$\oplus$</td>
</tr>
</tbody>
</table>

### 4.1 Initialization Phase

A remote server $RS$ runs algorithm $G$ to compute a large prime number $q$. Then it selects a random number $s$ such that $s \in R Z^*_q$. It further chooses a collision resist cryptographic one-way hash function $H : \{0, 1\}^* \rightarrow \{0, 1\}^n$, where $n$ is a fixed length integer number. Finally, $RS$ publishes $H(\cdot)$ as public and keeps $s$ as secret.

### 4.2 Registration Phase

Whenever a new user $U_i$ wants to register to the remote server $RS$, this phase is invoked. This phase is as follows:

1. The user inputs their biometric feature (i.e., finger print) to a sensor. The sensor generates a corresponding biometric information $B_i$ and provides it to $U_i$.

2. The user $U_i$ chooses an identity $ID_i$, password $pw_i$ and generates an unique pair $(\phi_i, \theta_i)$ from $B_i$ by computing $(\phi_i, \theta_i) \leftarrow Gen(B_i)$. $U_i$ then computes $pw_r = H(pw_i || \phi_i)$ and sends $(ID_i, pw_r)$ to $RS$ through a secure channel.
3. After getting a registration request \((ID_i, pur_i)\) from \(U_i\), RS computes \(A_i = H(s||ID_i)\) and \(D_i = H(A_i)\). RS then checks \(D_i\) is present in its database or not. If it is present in its database, RS sends a decline message to \(U_i\) because, \(ID_i\) is already used by another user. Therefore, the user \(U_i\) has to choose another identity until fresh identity is not obtained. If \(D_i\) is not present in its database, RS computes \(C_i = A_i \oplus pur_i\) and stores parameters \((C_i, D_i, des(\cdot))\) into the memory of smart card, where \(des(\cdot)\) is a distance measurement function. Then RS issues the smart card for \(U_i\) and sends it through a secure channel or by person. RS then updates its database by adding \(D_i\) into the list.

4. After getting the smart card, \(U_i\) inserts it into a terminal or card reader and submits their identity \(ID_i\) and password \(pw_i\).

5. The smart card computes \(\overline{B_i} = B_i \oplus H(ID_i||pw_i)\), \(\overline{\theta_i} = \theta_i \oplus H(ID_i||pw_i)\). Finally, the smart card stores \((\overline{B_i}, \overline{\theta_i})\) into its memory. Note that, smart card stored parameters are \((C_i, D_i, \overline{B_i}, \overline{\theta_i}, des(\cdot))\).

Figure 2 shows the registration phase of the proposed scheme.

### 4.3 Login Phase

Whenever a registered user \(U_i\) wants to access the remote server, this phase is invoked. \(U_i\) inserts their smart card into a card reader or terminal and provides their biometric information \(B'_i\) through sensor, identity \(ID_i\) and password \(pw_i\) to the smart card. The smart card then executes following steps:

1. The smart card computes \(B'_i = \overline{B_i} \oplus H(ID_i||pw_i)\) and checks \(des(B'_i, B'_i) \leq \delta d\). If it does not hold, the smart card rejects \(U_i\); otherwise, it follows next step.

2. The smart card computes \(\theta'_i = \overline{\theta_i} \oplus H(ID_i||pw_i)\), \(\phi'_i \leftarrow \text{Rep}(B'_i, \theta'_i)\), \(pur'_i = H(pw_i||\phi'_i)\), \(A'_i = C_i \oplus pur'_i\), \(D'_i = H(A'_i)\) and checks computed \(D'_i\) and stored \(D_i\) are equal or not. If equality does not hold, the smart card rejects \(U_i\); otherwise, it follows next step.

3. The smart card chooses a random number \(r_i \in \mathbb{Z}_p^\times\) and further computes \(E_i = r_i \oplus A'_i\), \(G_i = T_i^1 \oplus A'_i \oplus H(r_i)\) and \(F_i = H(r_i||A'_i||T_i^1)\), where \(T_i^1\) is the current login timestamp of \(U_i\). The user \(U_i\) then sends a login request message \((ID_i, G_i, F_i, E_i)\) to the registration server RS through a public channel.

Figure 3 shows the login phase of the proposed scheme.

### 4.4 Authentication and Key Agreement Phase

After receiving the login request message \((ID_i, G_i, F_i, E_i)\) from the user \(U_i\) at timestamp \(T_i\), the remote server RS computes following steps:

1. RS computes \(A'_i = H(s||ID_i)\), \(r'_i = A'_i \oplus E_i\), \(T_i^{1'} = A'_i \oplus G_i \oplus H(r'_i)\) and checks \((T_s - T_i^{1'}) \leq \delta T\). If it does not hold, RS rejects \(U_i\); otherwise, executes the next step.
After receiving the reply message $U_i$ whenever a user $B_i$ executes the following steps to verify the reply message of $U_i$:

2. $RS$ computes $F_i^r = H(r_i^*||A_i^r||T_i^1)$ and checks computed $F_i^r$ and received $F_i$ are equal or not. If it does not hold, $RS$ rejects login request message of $U_i$; otherwise, follows the next step.

3. $RS$ chooses a random number $y_i \in \mathbb{Z}_q^*$ and further computes $Q_i = A_i^r \oplus y_i$, $K_i = T_i \oplus A_i^r \oplus H(y_i)$, $S_{K_i} = H(y_i||r_i^*)$, $L_i = H(T_i||S_{K_i}||A_i^r)$ and sends a reply message $(Q_i, L_i, K_i)$ to $U_i$ through a public channel. $RS$ accepts $S_{K_i}$ as a shared secret session key.

After receiving the reply message $(Q_i, L_i, K_i)$ from $RS$ at timestamp $T_i^2$, the smart card of the user $U_i$ further executes the following steps to verify the reply message of $RS$:

1. The smart card computes $y_i' = A_i^r \oplus Q_i$, $T_i' = A_i^r \oplus K_i \oplus H(y_i)$ and checks $(T_i^2 - T_i') \leq \delta T$. If it does not hold, the smart card rejects the reply message; otherwise, executes the next step.

2. The smart card computes $S_{K_i}^* = H(y_i'||r_i^*)$, $L_i' = H(T_i^2||S_{K_i}^*||A_i^r)$ and checks computed $L_i'$ and received $L_i$ are equal or not. If they are equal, the user $U_i$ agrees upon the shared secret key $S_{K_i}$; otherwise, rejects the reply message.

Figure 4 shows the authentication and key agreement phase of the proposed scheme.

4.5 Password Update Phase

Whenever a user $U_i$ wants to change their password, this phase is invoked. $U_i$ inserts their smart card into a card reader or terminal and provides their biometric information $B_i^r$ through sensor, identity $ID_i$ and password $pw_i$ to the smart card. The smart card then executes following steps:

1. The smart card computes $B_i' = \tilde{B}_i \oplus H(ID_i||pw_i)$ and checks $des(B_i'; B_i') \leq \delta d$. If it does not hold, the smart card rejects $U_i$; otherwise, it follows next step.
4.5 Password Update Phase

Figure 5. Password update phase of the proposed scheme

2. The smart card computes \( \theta'_i = \theta_i \oplus H(ID_i || pw_i) \), \( \phi'_i \leftarrow \text{Rep}(B'_i, \theta'_i) \), \( pw'_i = H(pw_i || \phi'_i) \), \( A'_i = C_i \oplus pw'_i \), \( D'_i = H(A'_i) \) and checks computed \( D'_i \) and stored \( D_i \) are equal or not. If equality does not hold, the smart card rejects \( U_i \); otherwise, gives permission to enter their new password.

3. The user \( U_i \) selects their new password \( pw'_i \) and proves it to the smart card. The smart card then further proceeds to next step.

4. The smart card computes \( pw'_i = H(pw_i || \phi'_i) \), \( C'_i = A'_i \oplus pw'_i \), \( B'_i = B'_i \oplus H(ID_i || pw'_i) \) and \( \theta'_i = \theta'_i \oplus H(ID_i || pw'_i) \). The smart card then stores \( C'_i \), \( B'_i \) and \( \theta'_i \) respectively into the memory of smart card.

Figure 5 shows the password update phase of the proposed scheme.

5 Security Analysis of Proposed Scheme

The formal security analysis of the proposed scheme under the random oracle model is presented in this section. This security analysis uses the formal security analysis under the generic group model of cryptography. In the following, this work defines random oracles for the formal security analysis of the proposed scheme:

- **Oracle \( H \)** is a random oracle which maintains a tuple \( \langle x, y \rangle \) such that \( y = H(x) \). It returns \( x \) from \( y \) upon receiving a query \( \langle q, y \rangle \) if \( \langle x, y \rangle \) is present in the tuple; otherwise returns a random number \( r_1 \). Then it stores a new entry \( \langle r_1, y \rangle \) into its tuple.

- **Oracle \( FE \)** is a random oracle which contains two parts:

  1. **Oracle \( FE_{\text{Gen}} \)** unconditionally outputs the pair \( \langle \phi, \theta \rangle \) from the corresponding tuple \( \langle B, \phi, \theta \rangle \) upon receiving a query \( \langle q, \text{Gen}, B \rangle \) such that \( \langle \phi, \theta \rangle \leftarrow \text{Gen}(B) \) if \( \langle B, \phi, \theta \rangle \) is present in its tuple; otherwise returns two random numbers \( r_2 \) and \( r_3 \). Then it stores new entry \( \langle B, r_2, r_3 \rangle \) into its tuple.

  2. **Oracle \( FE_{\text{Rep}} \)** unconditionally outputs \( \phi \) from the corresponding tuple \( \langle B', \phi, \theta \rangle \) upon receiving a query \( \langle q, \text{Rep}, B', \theta \rangle \) such that \( \phi \leftarrow \text{Rep}(B', \theta) \) if \( \langle B', \phi, \theta \rangle \) is present in its tuple; otherwise returns random number \( r_4 \). Then it stores new entry \( \langle B', r_4, \theta \rangle \) into its tuple.

**Theorem 1.** Under the assumption that a cryptographic one-way hash function \( H(\cdot) \) and fuzzy extractor \( FE \) act as random oracles, the proposed scheme is provably secure against an adversary \( A \) for deriving the password \( pw_i \) and biometric parameter \( B_i \) of a user \( U_i \) even if the adversary \( A \) gets parameters that are stored into the memory of \( U_i \)'s smart card and traps the communication messages between \( U_i \) and the remote server \( RS \).

**Proof 1.** This research construct an adversary \( A \) who has the ability to derive the password \( pw_i \) and biometric parameter \( B_i \) of a user \( U_i \). For this purpose, this research assumes that the smart card of a user \( U_i \) is lost or stolen. Thus, the adversary \( A \) can extract the stored parameters \( \langle C_i, D_i, B_i, \theta_i \rangle \) from the memory of the smart card.
of the user $U_i$ by power monitoring [32][33]. The adversary $A$ also traps login request message $(ID_i, G_i, F_i, E_i)$ and a reply message $(Q_i, L_i, K_i)$. The adversary $A$ runs the experiment, $EXP_{\text{oracle}}^{\text{A, TFUAS}}$ for our three factor user authentication scheme (TFUAS) to derive the password $pw_i$ and biometric parameter $B_i$ of the user $U_i$ as given in the Algorithm 1.

Algorithm 1 $EXP_{\text{oracle}}^{\text{A, TFUAS}}$

**Input:** $C_i, D_i, B_i, \bar{\theta}, ID_i, G_i, F_i, E_i, Q_i, L_i, K_i$

**Output:** 0 or 1

1: Calls $OracleH$ on the input $D_i$ to retrieve the information $A_i = H(s||ID_i)$ as $(A_i^*) \leftarrow OracleH(D_i)$

2: Calls $OracleH$ on the input $F_i$ to retrieve the information $A_i, r_i$ and $T_i^{+}$ as $(r_i^* || A_i^* || T_i^{+}) \leftarrow OracleH(F_i)$

3: Calls $OracleH$ on the input $L_i$ to retrieve the information $S K_i, T_i$ and $A_i$ as $(T_i || S K_i || A_i^{**}) \leftarrow OracleH(L_i)$

4: if $(A_i^{***} == A_i^{**} == A_i^*)$ then

5: Computes $r_i^{**} = A_i^* \oplus E_i$

6: if $(r_i^{*} == r_i^{**})$ then

7: Computes $y_i^* = A_i^* \oplus Q_i, T_i^{**} = A_i^* \oplus G_i \oplus H(r_i^*)$ and $T_i^{*} = A_i^* \oplus K_i \oplus H(Y_i^*)$

8: if $(T_i^{*} == T_i^{**})$ then

9: Computes $pw_i^* = C_i \oplus A_i^*$

10: else

11: Return 0 (Failure)

12: end if

13: else

14: Return 0 (Failure)

15: end if

else

17: Return 0 (Failure)

18: end if

19: repeat

20: Chooses a password $pw_i^{\text{[guess]}}$

21: Computes $B_i^{\text{[guess]}} = B_i^* \oplus H(ID_i||pw_i^{\text{[guess]}})$ and $\theta_i^{\text{[guess]}} = \bar{\theta} \oplus H(ID_i||pw_i^{\text{[guess]}})$

22: Calls $OracleFE_{\text{Rep}}$ on the input $B_i^{\text{[guess]}}$ and $\theta_i^{\text{[guess]}}$ to retrieve the information $\phi_i$, as $(\phi_i^{*}) \leftarrow OracleFE_{\text{Rep}}(B_i^{\text{[guess]}}, \theta_i^{\text{[guess]}})$

23: Computes $pw_i^{\text{[guess]}} = H(pw_i^{\text{[guess]}}||\phi_i^{*})$

24: until $(pw_i^{\text{[guess]}} == pw_i^*)$

25: if $(pw_i^{\text{[guess]}} == pw_i^*)$ then

26: Return 1 (Success)

27: else

28: Return 0 (Failure)

29: end if

We define the success probability for $EXP_{\text{oracle}}^{\text{A, TFUAS}}$ as $\text{Succ}_{\text{oracle}}^{\text{A, TFUAS}} = Pr[EXP_{\text{oracle}}^{\text{A, TFUAS}} = 1]$. Then the advantage of $EXP_{\text{oracle}}^{\text{A, TFUAS}}$ is given by $Adv_{\text{oracle}}^{\text{A, TFUAS}}(t, qH, qFE) = \max_A[\text{Succ}_{\text{oracle}}^{\text{A, TFUAS}}]$, where the maximum is taken over all $A$ with the execution time $t$, the number of queries $qH$ made to the $OracleH$ oracle and the number of queries $qFE$ made to the $OracleFE$. Our proposed scheme is said to be provably secure against the adversary $A$ for deriving the password $pw_i$ and biometric parameter $B_i$ of a user $U_i$, if $Adv_{\text{oracle}}^{\text{A, TFUAS}}(t, qH, qFE) \leq \xi$, for any small $\xi > 0$. According to algorithm $EXP_{\text{oracle}}^{\text{A, TFUAS}}$ (see Algorithm 1), if the adversary $A$ gets success to compute inversion of the cryptographic one-way hash function $H(\cdot)$ and also gets success to solve hardness of fuzzy extractor, he/she can successfully derive the password $pw_i$ and biometric parameter $B_i$ of the user $U_i$ by using of the $OracleH$ random oracle and $OracleFE$ random oracle, and wins the game. But, according to Definition 1 and Definition 2, we know that $Adv_{\text{oracle}}^{\text{A, TFUAS}}(t, qH, qFE) \leq \xi$, for any small $\xi > 0$ and $Adv_{\text{oracle}}^{\text{A, TFUAS}}(t, qH, qFE) \leq \xi$, for any small $\xi > 0$ because, the proposed scheme depends on both $Adv_{\text{oracle}}^{\text{A, TFUAS}}(t, qH, qFE)$ and $Adv_{\text{oracle}}^{\text{A, TFUAS}}(t, qH, qFE)$. Thus, our proposed scheme is secure against the adversary $A$ for deriving the password $pw_i$ and biometric parameter $B_i$ of the user $U_i$. 


Theorem 2. Under the assumption that a cryptographic one-way hash function $H(\cdot)$ acts as a random oracle, the proposed scheme is provably secure against an adversary $\mathcal{A}$ for deriving the secret key $s$ of the remote server $RS$ even if the adversary $\mathcal{A}$ gets parameters that are stored into the memory of $U_i$’s smart card and traps the communication messages between a user $U_i$ and the remote server $RS$.

Proof 2. This research construct an adversary $\mathcal{A}$ who has the ability to derive the secret key $s$ of the remote server $RS$. For this purpose, this research considers same assumptions as discussed in Theorem 1. The adversary $\mathcal{A}$ runs the experiment, EXP$_{\mathcal{A},TFUAS}$ for our three factor user authentication scheme (TFUAS) to derive the secret key $s$ of the remote server $RS$ as given in the Algorithm 2.

Algorithm 2 EXP$_{\mathcal{A},TFUAS}$

\begin{itemize}
  \item Input: $D_i, ID_i, G_i, F_i, E_i, Q_i, L_i, K_i$
  \item Output: 0 or 1
  \item 1: Calls Oracle$H$ on the input $D_i$ to retrieve the information $A_i = H(s||ID_i)$ as $(A_i^*) \leftarrow$ Oracle$H(D_i)$
  \item 2: Calls Oracle$H$ on the input $F_i$ to retrieve the information $A_i, r_i$ and $T_i^1 = (r_i^* || A_i^* || T_i^1) \leftarrow$ Oracle$H(F_i)$
  \item 3: Calls Oracle$H$ on the input $L_i$ to retrieve the information $S K_i, T_i$, and $A_i$ as $(T_i||S K_i||A_i^*) \leftarrow$ Oracle$H(L_i)$
  \item 4: If $(A_i^* = A_i^*)$ then
  \item 5: Computes $r_i^* = A_i^* \oplus E_i$.
  \item 6: If $(r_i^* = r_i^*)$ then
  \item 7: Computes $y_i^* = Q_i \oplus A_i^*, T_i^{1*} = A_i^* \oplus G_i \oplus H(r_i^*)$ and $T_i^{s*} = A_i^* \oplus K_i \oplus H(y_i^*)$.
  \item 8: If $(T_i^{s*} = T_i^{s*}) \& \& (T_i^{1*} = T_i^{1*})$ then
  \item 9: Calls Oracle$H$ on the input $A_i^*$ to retrieve the information $s$ and $ID_i$ as $(s^*||ID_i^*) \leftarrow$ Oracle$H(A_i^*)$
  \item 10: If $(ID_i = ID_i^*)$ then
  \item 11: Accepts $s^*$ as secret key of RS
  \item 12: Return 1 (Success)
  \item else
  \item 13: Return 0 (Failure)
  \item end if
  \item end else
  \item 14: Return 0 (Failure)
  \item end if
  \item end else
  \item 16: Return 0 (Failure)
  \item end if
  \item 17: Return 0 (Failure)
  \item end if
  \item 18: Return 0 (Failure)
  \item end if
  \item 19: Return 0 (Failure)
  \item end if
  \item 20: Return 0 (Failure)
  \item end if
  \item 21: Return 0 (Failure)
  \item end else
  \item 22: Return 0 (Failure)
  \item end if
  \item 23: Return 0 (Failure)
  \item end if
\end{itemize}

We define the success probability for EXP$_{\mathcal{A},TFUAS}$ as $SUCC_{\mathcal{A},TFUAS} = Pr[EXP_{\mathcal{A},TFUAS} = 1]$. Then the advantage of EXP$_{\mathcal{A},TFUAS}$ is given by $Adv_{\mathcal{A},TFUAS}(t, qH) = max_s[SUCC_{\mathcal{A},TFUAS} - \frac{1}{2}]$, where the maximum is taken over all $\mathcal{A}$ with the execution time $t$, the number of queries $qH$ made to the Oracle$H$ oracle. The proposed scheme is said to be provably secure against the adversary $\mathcal{A}$ for deriving the secret key $s$ of the remote server $RS$, if $Adv_{\mathcal{A},TFUAS}(t, qH) \leq \xi$, for any small $\xi > 0$. According to algorithm EXP$_{\mathcal{A},TFUAS}$ (see Algorithm 2), if the adversary $\mathcal{A}$ gets success to compute inversion of the cryptographic one-way hash function $H(\cdot)$, he/she can successfully derive the secret key $s$ of the remote server $RS$ by using of the Oracle$H$ random oracle and wins the game. But, according to Definition 1, we know that $Adv_{\mathcal{A},H}(t) \leq \xi_1$, for any small $\xi_1 > 0$. Since, we get $Adv_{\mathcal{A},TFUAS}(t, qH) \leq \xi$, for any small $\xi > 0$ because, the proposed scheme depends on $Adv_{\mathcal{A},H}(t)$. Thus, our proposed scheme is secure against the adversary $\mathcal{A}$ for deriving the secret key $s$ of the remote server $RS$.

Theorem 3. Under the assumption that a cryptographic one-way hash function $H(\cdot)$ acts as a random oracle, the proposed scheme is provably secure against an adversary $\mathcal{A}$ for deriving a shared secret session key $SK_i$ between a user $U_i$ and the remote server $RS$ even if the adversary $\mathcal{A}$ gets parameters that are stored into the memory of $U_i$’s smart card and traps the communication messages between $U_i$ and the remote server $RS$.

Proof 3. This research construct an adversary $\mathcal{A}$ who has the ability to derive the session key $SK_i$ between a user $U_i$ and the remote server $RS$. For this purpose, this research considers same assumptions as discussed
in Theorem 1. The adversary $\mathcal{A}$ runs the experiment, $\text{EXP}_3^{\text{oracle}}_{\text{T FUAS}}$ for our three factor user authentication scheme (TFUAS) to derive the session key $SK_i$ between the user $U_i$ and the remote server $RS$ as given in the Algorithm 3.

Algorithm 3 $\text{EXP}_3^{\text{oracle}}_{\text{T FUAS}}$

**Input:** $D_i, ID_i, G_i, F_i, E_i, Q_i, L_i, K_i$

**Output:** 0 or 1

1. Calls $\text{Oracle}H$ on the input $D_i$ to retrieve the information $A_i = H(s||ID_i)$ as $(A_i) \leftarrow \text{Oracle}H(D_i)$
2. Calls $\text{Oracle}H$ on the input $F_i$ to retrieve the information $A_i, r_i$ and $T_i$ as $(r_i^* || A_i^* || T_i^*) \leftarrow \text{Oracle}H(F_i)$
3. Calls $\text{Oracle}H$ on the input $L_i$ to retrieve the information $SK_i, T_i$ and $A_i$ as $(T_i^* || SK_i || A_i^*) \leftarrow \text{Oracle}H(L_i)$
4. if $(A_i^{**} == A_i^* == A_i')$ then
5. Computes $r_i^* = A_i^* \oplus E_i$
6. if $(r_i^* == r_i')$ then
7. Computes $T_i^{**} = A_i^* \oplus G_i \oplus H(r_i^*), y_i^* = Q_i \oplus A_i^*$ and $T_i^{***} = A_i^* \oplus K_i \oplus y_i^*$
8. if $(T_i^{**} == T_i^* \rangle \& \&( T_i^{***} == T_i^*)$ then
9. Calls $\text{Oracle}H$ on the input $SK_i^*$ to retrieve the information $y_i$ and $r_i$ as $(y_i^* || r_i^{**}) \leftarrow \text{Oracle}H(SK_i^*)$
10. if $(r_i^{**} == r_i')\&\&( y_i^* == y_i^{**})$ then
11. Accepts $SK_i^*$ as the shared secret session key
12. Return 1 (Success)
13. else
14. Return 0 (Failure)
15. end if
16. else
17. Return 0 (Failure)
18. end if
19. else
20. Return 0 (Failure)
21. end if
22. else
23. Return 0 (Failure)
24. end if

We define the success probability for $\text{EXP}_3^{\text{oracle}}_{\text{T FUAS}}$ as $\text{Succ}_3^{\text{oracle}}_{\text{T FUAS}} = \Pr[\text{EXP}_3^{\text{oracle}}_{\text{T FUAS}} = 1]$. Then the advantage of $\text{EXP}_3^{\text{oracle}}_{\text{T FUAS}}$ is given by $\text{Adv}_{\text{oracle}}^{\text{T FUAS}}(t,qH) = \max_A\{\text{Succ}_3^{\text{oracle}}_{\text{T FUAS}} \}$, where the maximum is taken over all $\mathcal{A}$ with the execution time $t$, the number of queries $qH$ made to the $\text{Oracle}H$ oracle. The proposed scheme is said to be provably secure against the adversary $\mathcal{A}$ for deriving the session key $SK_i$ between the user $U_i$ and the remote server $RS$, if $\text{Adv}_{\text{oracle}}^{\text{T FUAS}}(t,qH) \leq \xi$, for any small $\xi > 0$. According to algorithm $\text{EXP}_3^{\text{oracle}}_{\text{T FUAS}}$ (see Algorithm 3), if the adversary $\mathcal{A}$ gets success to compute inversion of the cryptographic one-way hash function $H()$, he/she can successfully derive the session key $SK_i$ between the user $U_i$ and the remote server $RS$ by using of the $\text{Oracle}H$ random oracle and wins the game. But, according to Definition 1, we know that $\text{Adv}_{\text{oracle}}^{\text{T FUAS}}(t,qH) \leq \xi$, for any small $\xi > 0$ because, the proposed scheme depends on $\text{Adv}_{\text{oracle}}^{\text{T FUAS}}(t)$. Thus, our proposed scheme is secure against the adversary $\mathcal{A}$ for deriving the session key $SK_i$ between the user $U_i$ and the remote server $RS$.

Theorem 1 demonstrated that the proposed scheme is secure against the off-line password guessing attack. Theorem 3 demonstrates that the proposed scheme is secure against the session key recovery attack because, without knowing random numbers $\{r_i, y_i\}$ then $\mathcal{A}$ cannot compute the session key $SK_i$. In the proposed scheme, the communicating messages depend on random numbers and the timestamp. Therefore, the communication messages are guaranteed to be different for every session. Thus, $\mathcal{A}$ cannot mount a replay attack on this proposed scheme. In this proposed scheme, $\mathcal{A}$ cannot mount a forgery attack without knowing secret password $pw_i$ and biometric parameter $B_i$ of a user $U_i$ and the secret key $s$ of the remote server $RS$. Theorems 1 and 2 show that the secret information of the remote server and the user are secure from $\mathcal{A}$. Thus, it is infeasible to mount a forgery attack on this proposed scheme.

A valid user say, $U_\bar{X}$ as an adversary $\bar{A}$ cannot login into the proposed authentication scheme as an another user say, $U_i$ because, to login into the system, $\bar{A}$ has to know the secret key $s$ of the remote server $RS$. As, $\bar{A}$ is a valid user, it knows their identity $ID_\bar{X}$, password $pw_\bar{X}$ and biometric information $\phi_\bar{X}$. Therefore, $\bar{A}$ can compute...
the proposed scheme depends on $A_d$ and the remote server $RS$ oracle. Therefore, $A$ cannot extract or computed from known parameters of $\hat{A}$. From $H(s||ID_A)$, $\hat{A}$ cannot extract $s$ due to hardness of inversion of one-way hash function. Furthermore, Theorem 2 shows that $s$ cannot be extracted or computed from known parameters of $\hat{A}$. Therefore, $\hat{A}$ unable to mount any attacks on the proposed scheme.

### 6 Comparison

In this section, the performances of the proposed scheme with the existing authentication schemes namely, Li and Hwang’s scheme [17], Das’s scheme [18], An’s scheme [19], Tan’s scheme [22], Yan et al.’s scheme [23], Mishra et al.’s scheme [24] and He et al.’s scheme [26] are compared. However, the compared schemes in [17–19] and [22–24] and [26] are not suitable for practical use because, the schemes cannot resist the possible attacks as shown in Table 2. In the introduction part of this paper, it has been described that is insecure against security attacks. Moreover, security analysis of the proposed scheme (see Section 5) shows that the proposed scheme can resist all the possible attacks. Thus, the proposed scheme is more secure than other schemes.

Table 3 shows the computational, storage cost and communication overhead comparison of schemes in [17–19], [22–24], and [26] with our proposed scheme. For this purpose, only login and authentications phases have been considered due to maximum use. $T_{\text{enc}}, T_{\text{dec}}$ are the time required for hash operation, symmetric key encryption and decryption respectively. The proposed scheme takes time for 10 hashing operations in two phases which is the lower among related and compared schemes in [18, 22–24, 26]. It can be reasonably assumed that the length of $ID_A$ and $pw_i$ are 64 bits each. Large prime number $q$, cryptographic one-way hash function $H(\cdot)$ like SHA-1, symmetric key encryption/decryption like AES with key size 128 bits\(^2\), random numbers, symmetric key encryption/decryption and timestamp returns 128 bits each. The communication cost of proposed scheme for login message is $(128+128+128+64) = 448$ bits and message generated in authentication phase is $(128+128+128)$

Table 2. Security Functionality comparison of the proposed scheme with related schemes

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>A2</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>A3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>×</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>A4</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>A5</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>A6</td>
<td>-</td>
<td>-</td>
<td>×</td>
<td>✓</td>
<td>-</td>
<td>×</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>EPC</td>
<td>×</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>ELP</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>


Table 3. Computation, communication and storage cost comparison of the proposed scheme with related schemes

<table>
<thead>
<tr>
<th>Schemes</th>
<th>Storage Cost (in bits)</th>
<th>Communication Cost (in bits)</th>
<th>Computation cost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Login + Authentication</td>
<td>Login</td>
</tr>
<tr>
<td>Ref. [17]</td>
<td>448</td>
<td>576</td>
<td>$2T_H$</td>
</tr>
<tr>
<td>Ref. [18]</td>
<td>576</td>
<td>832</td>
<td>$3T_H$</td>
</tr>
<tr>
<td>Ref. [19]</td>
<td>576</td>
<td>704</td>
<td>$3T_H$</td>
</tr>
<tr>
<td>Ref. [22]</td>
<td>384</td>
<td>576</td>
<td>$4T_H+1T_{\text{enc}}$</td>
</tr>
<tr>
<td>Ref. [23]</td>
<td>640</td>
<td>960</td>
<td>$3T_H$</td>
</tr>
<tr>
<td>Ref. [24]</td>
<td>800</td>
<td>1120</td>
<td>$4T_H$</td>
</tr>
<tr>
<td>Ref. [26]</td>
<td>384</td>
<td>1216</td>
<td>$4T_H$</td>
</tr>
<tr>
<td>Proposed scheme</td>
<td>512</td>
<td>832</td>
<td>$4T_H$</td>
</tr>
</tbody>
</table>

$H(s||ID_A)$ by computing $C_A \oplus H(pw_i||\phi_A)$, where $C_A$ is the smart card store parameter of $\hat{A}$. From $H(s||ID_A)$, $\hat{A}$ cannot extract $s$ due to hardness of inversion of one-way hash function. Furthermore, Theorem 2 shows that $s$ cannot be extracted or computed from known parameters of $\hat{A}$. Therefore, $\hat{A}$ unable to mount any attacks on the proposed scheme.


\[^2\]http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf
= 384 bits. Therefore, total communication cost is (448+384) = 832 bits which is lower than existing and compared schemes in [23, 24, 26]. The storage cost of our proposed scheme is (128 + 128 + 128 + 128) = 512 bits which is also lower than related schemes in [18, 19, 23, 24].

After resisting all possible attacks as shown in Section 5, the proposed scheme provides better trade-off among storage, computational and communication costs than other related existing schemes. Hence, it can be claimed that the proposed scheme is more efficient and secure than other related existing schemes and also it is applicable for practical applications.

- **Advantages of proposed scheme**

In the following, the advantages of the proposed scheme have been discussed.

**Efficient login phase:** If a user \( U_i \) enters faulty password and faulty identity by some means in login phase of the proposed scheme, the smart easily can detect the wrong inputs before going to generate a login request message. For this purpose, smart card computes \( B'_i = B_i \oplus H(ID_i) || pw_i ) \) and checks \( \text{des}(B'_i, B'_i) \leq \delta d \). If it does not hold, the smart card rejects \( U_i \); otherwise, it computes \( \theta'_i = \theta_i \oplus H(ID_i) || pw_i ) \), \( \phi'_i \leftarrow \text{Rept}(B'_i, \theta'_i) \), \( pw'_i = H(pw_i || \phi'_i) \), \( A'_i = C_i \oplus pw'_i \), \( D'_i = H(A'_i) \) and checks computed \( D'_i \) and stored \( D_i \) are equal or not. If equality does not hold, the smart card rejects \( U_i \); otherwise, accepts the password \( pw_i \) and identity \( ID_i \) as correct inputs. Therefore, extra communication overhead due to wrong inputs can be avoided in the proposed scheme.

**Efficient password change phase:** If a user \( U_i \) enters faulty password and faulty identity by some means in password phase of the proposed scheme, the smart easily can detect the wrong inputs before going to give permission to the user to submit their new password. For this purpose, smart card computes executes the same steps as mentioned above to check the submitted inputs. If provided inputs are correct, then only the smart card give permission to enter new password to \( U_i \). Furthermore, to change password of a user, smart card does not need to communicate with the remote server. Therefore, communication overhead is also reduced in proposed scheme with efficient wrong input detection.

**Mutual authentication:** In the proposed scheme, the remote server \( RS \) computes and accepts a secret session key \( SK_i \) after verifying legitimacy of a user \( U_i \) through login request message and then, \( RS \) sends a reply message to the user \( U_i \). The user \( U_i \) agrees upon the same secret session key \( SK_i \) with \( RS \) after verifying legitimacy of \( RS \) through reply message. Therefore, both way authentication has been done in the proposed scheme. Furthermore, the proposed scheme can resist all the possible attacks (see, Section 5). Hence, the proposed scheme achieves mutual authentication.

7 Conclusion

This paper have proposed an authentication scheme. After analyzing the proposed scheme it can be stated that the proposed scheme can overcome the all possible attacks and has better trade-off among computational, storage and communication costs. Therefore, the proposed scheme is suitable for real world online applications.

References