Computing (2014) 96:793–816
DOI 10.1007/s00607-013-0308-2

More efficient key-hash based fingerprint remote authentication scheme using mobile device

Muhammad Khurram Khan · Saru Kumari · Mridul K. Gupta

Received: 19 November 2012 / Accepted: 5 February 2013 / Published online: 15 March 2013

Abstract Today, the world is taking large leaps of progress in technology. The technology is turning the vision of achieving transparency, speed, accuracy, authenticity, friendliness and security in various services and access control mechanisms, into reality. Consequently, new and newer ideas are coming forth by researchers throughout the world. Khan et al. (Chaos Solitons Fractals 35(3):519–524, 2008) proposed remote user authentication scheme with mobile device, using hash-function and fingerprint biometric. In 2012, Chen et al. pointed out forged login attack through loss of mobile device on Khan et al.'s scheme and subsequently proposed a scheme to improve on this drawback. Truong et al. (Proceedings of 26th IEEE International Conference on Advanced Information Networking and Applications, pp 678–685, 2012) demonstrated that in Chen et al.'s scheme, an adversary can successfully replay an intercepted login request. They also showed that how an adversary can make fool of both the participants of Chen et al.'s protocol by taking advantage of the fact that the user is not anonymous in scheme. Further, they proposed an improvement to Chen et al.'s scheme to cut off its problems. Through this paper, we show that Chen et al.'s scheme has some

M. K. Khan (✉)
Center of Excellence in Information Assurance, King Saud University,
Riyadh, Kingdom of Saudi Arabia
e-mail: mkhurram@ksu.edu.sa

S. Kumari
Department of Mathematics, Agra College, Dr. B.R.A.University,
Agra, Uttar Pradesh, India
e-mail: saryusiirohi@gmail.com

M. K. Gupta
Department of Mathematics, Chaudhary Charan Singh University,
Meerut, Uttar Pradesh, India
e-mail: mkgupta2002@hotmail.com

other drawbacks too and the improvement proposed by Truong et al. is still insecure and vulnerable. We also propose an improved scheme which overcomes the flaws and inherits the goodness of both the schemes, Chen et al.'s scheme and Truong et al.'s scheme.

Keywords Mobile device · Fingerprint · Random nonce · Authentication · Attacks and drawbacks

Mathematics Subject Classification 97Pxx · 68-xx

1 Introduction

In this era of rapidly developing and changing variety of communication technologies, remote authentication schemes have been important tools to communicate between entities. Since 1981, when Lamport [1] proposed the first remote authentication scheme using one-time password, many new proposals and improvements have been proposed [2–6] in this field. In 2000, Hwang and Li [7] proposed the first remote authentication scheme with smart card. After this many schemes [8–15] were proposed employing smart cards. Recently, biometric characteristics (fingerprint, face, voice, etc.) have joined the family of authentication factors (password, smart card, etc.), and are playing crucial role in enhancing the security of authentication mechanisms [16–25]. With rapidly growing mobile technology, mobile devices such as mobile phones, PDAs, personal navigation device, etc, are being widely used in day-to-day life. In 2008, Khan et al. [16] proposed biometric based remote user authentication scheme making use of mobile device. In 2010, Chen et al. [17] identified that Khan et al.'s scheme is attackable by an adversary by extracting secret information from mobile device; they also proposed a scheme to mend this shortcoming. In 2012, Truong et al. [26] demonstrated that Chen et al.'s scheme fails to withstand replay attack, server and user spoofing attack and lacks user's anonymity; they also proposed an enhancement of Chen et al.'s scheme to get rid of these attacks. We identified [27] that Chen et al.'s scheme [17] has some drawbacks other than those pointed out by Truong et al. In [27] we continued Truong et al.'s 'user and server spoofing attack' to apply password guessing attack and described how three-factor security falls ineffective in Chen et al.'s scheme [17].

With this paper, we display that both the schemes [17,26] suffer from pitfalls. We reveal that Chen et al.'s scheme is susceptible to password guessing attack if an attacker obtains the mobile device of a user. Afterward, we crypt analyze the improvement of Chen et al.'s scheme proposed by Truong et al. We exhibit, how an adversary can impersonate user and server, using information extracted from mobile device of user and an intercepted login request. Moreover, we explain that these impersonation attacks are feasible with only an intercepted login request in hand. We show that the improved scheme still cannot resist password guessing attack. Besides, the anonymity provided to user is imperfect in the sense that any two (or more) login requests or a login request and the corresponding mobile device belonging to the same user can be traced in Truong et al.'s scheme. Further, we explain that the improved scheme by Truong et al. suffers from the same drawbacks which Chen et al.'s scheme has, like violation of three factor security; server's secret key is at risk etc. Accordingly, we propose a scheme to get rid of security pitfalls of both the schemes [17,26].

The remainder of the paper is organized as follows: in the next two sections, we review Chen et al.'s scheme and Truong et al.'s scheme, respectively. In Sect. 4, we show the security weaknesses of Chen et al.'s scheme and Truong et al.'s scheme. It is Sect. 5, where we propose our protocol. In Sect. 6, the proposed scheme is analyzed for its security and usability aspects. Section 7 is about performance analysis of the proposed scheme. We conclude the paper with Sect. 8.

2 Review of Chen et al.'s scheme

The scheme consists of four phases: the registration phase, the login phase, the authentication phase and the password change phase. Each of these phases is described as follows:

2.1 Registration phase

This phase is meant to register the user with server S. It is conducted over a secure channel indicated by '$\Rightarrow$'. The description of the phase is as follows:

1. U_i chooses its identity $ID_{i,}$ password PW, and a random nonce N.
2. Then computes $h(PW \oplus N)$ and gives the imprint of his fingerprint on the sensor.
3. $U_i \Rightarrow S$: registration request = $\{ID_i, h(PW \oplus N), F_i)\}$, here F_i is the fingerprint template of U_i.

On receiving the registration request of U_i, the server performs the following steps using his secret key x, cryptographic hash function $h(.)$ and cryptographic keyed-hash function $h_k(.)$ with a secret key k.

4. Computes $hpw = h(PW \oplus N) \oplus F_i$, $R_i = h(ID_i \oplus x) \oplus hpw$ and $V_i = h_{h(IDi \oplus x)}(F_i)$.
5. $S \Rightarrow U_i$: secret information = $\{R_i, V_i, h(.), h_k(.)\}$.

On receiving this secret information from server, U_i performs as follows:

6. U_i stores $\{R_i, V_i, h(.), h_k(.)\}$ and N into his mobile device.

2.2 Login phase

This phase facilitates user to login to the server S. For this, U_i inserts his identity $ID_{i,}$ password PW and imprints his fingerprint F_i on the sensor. Then the mobile device of U_i performs the following steps.

1. Computes $hpw = h(PW \oplus N) \oplus F_i$ and $A_i = R_i \oplus hpw$.
2. Verifies if $h_{Ai}(F_i) = V_i$. If so, then the mobile device proceeds to next step and stores the fingerprint template F_i until the end of session; otherwise, it lapse the session.
3. Generates a random nonce N_U, and calculates $C_1 = N_U \oplus A_i$ and $C_2 = h_{Ai}(N_U)$ to challenge S.
4. $U_i \rightarrow S$: login request = $\{ID_i, C_1, C_2\}$.

2.3 Authentication phase

In this phase, both, the server and the user, verify the authenticity of each other. Description of the steps performed is as follows:

1. On receiving the login request of U_i, S checks the validity of ID_i format.
2. If ID_i format is correct, then S computes $h(ID_i \oplus x)$ and retrieves $B_i = C_1 \oplus h(ID_i \oplus x))$, which must be N_U indeed.
3. Next, S verifies if $h_{h(IDi \oplus x)}(B_i) = C_2$. If not so, rejects the login request; otherwise, accepts login request and temporarily stores ID_i till the end of the session.
4. $S \rightarrow U_i : \{S_1\}$. The server computes $S_1 = h(h(ID_i \oplus x)||B_i)$ and sends S_1 to U_i.

On receiving $\{S_1\}$ from S, the user performs the following steps to authenticate the server:

5. Verifies whether $h(A_i||N_U) = S_1$. If so, then the legitimacy of server gets confirmed.
6. Chooses a new random nonce N^*, computes $hpw^* = h(PW \oplus N^*) \oplus F_i$ and $R_i^* = R_i \oplus hpw \oplus hpw^*$
7. Replaces N and R_i with N^* and R_i^* respectively.

2.4 Password change phase

The password change phase facilitates the user to change his password PW to a new one, say PW^*. Following are the steps performed by the user and its mobile device:

1. U_i inserts his identity ID_i, password PW and imprints his fingerprint F_i into its mobile device.
2. The mobile device computes $hpw = h(PW \oplus N) \oplus F_i$, and checks if $h_{(Ri \oplus hpw)}(F_i) = V_i$. If not so, then the mobile device stops further action. Otherwise, allows U_i to insert a new password PW^*.
3. The mobile device computes $hpw^* = h(PW^* \oplus N) \oplus F_i$, $R_i^* = R_i \oplus hpw \oplus hpw^*$ and replaceR_i with R_i^*.

3 Review of Truong et al.'s scheme

The scheme consists of four phases: the registration phase, the login phase, the mutual authentication and session key agreement phase, and the password change phase. Each of the phases is described as follows:

3.1 Registration phase

This phase is meant to register the user with server S. It is conducted over a secure channel. The description of the phase is as follows:

1. U_i chooses its identity $ID_{i,}$ password PW, and a random nonce N.
2. Then computes $h(PW \oplus N)$ and gives the imprint of his fingerprint on the sensor.
3. $U_i \Rightarrow S$: Registration request = $\{ID_i, h(PW \oplus N), F_i)\}$, here F_i is the fingerprint template of U_i.

On receiving the registration request of U_i, the server performs the following steps:

4. Generates a random value e. Computes $hpw = h(PW \oplus N) \oplus F_i$, $E_i = hpw \oplus h(x \,||\, e)$, $R_i = h(ID_i \oplus h(x \,||\, e)) \oplus hpw$ and $V_i = h_{h(IDi \oplus h(x \,||\, e))}(F_i)$.
5. $S \Rightarrow U_i$: secret information = $\{R_i, V_i, E_i, e, h(.), h_k(.)\}$.

On receiving this secret information from server, U_i performs as follows:

6. Stores $\{R_i, V_i, E_i, e, h(.), h_k(.)\}$ and N into his mobile device.

3.2 Login phase

This phase facilitates user to login to the server S. For this, U_i inserts his identity $ID_{i,}$ password PW and imprints his fingerprint F_i on the sensor. Then the mobile device of U_i performs the following steps:

1. Computes $hpw = h(PW \oplus N) \oplus F_i$ and $A_i = R_i \oplus hpw$.
2. Verifies if $h_{Ai}(F_i) = V_i$. If so, then the mobile device proceeds to next step; otherwise, it lapse the session.
3. Generates a random nonce N_U and calculates $C_1 = N_U \oplus E_i \oplus hpw, C_2 = h_{Ai}(N_U)$ and $CID = ID_i \oplus N_U$.
4. $U_i \rightarrow S$: login request = $\{CID,\ e,\ C_1,\ C_2\}$, where '→' denotes a public channel.

3.3 Mutual authentication and session key agreement phase

In this phase, both, the server and the user, verify the authenticity of each other. Description of the steps performed by user and server is as follows:

On receiving the login request of $U_{i,}$ first S performs the following steps:

1. Retrieves $B_i(= N_U) = h(x \,||\, e) \oplus C_1$, $ID_i = CID \oplus N_U$ and checks the validity of ID_i format.
2. Check if $C_2 = h_{h(IDi \oplus h(x \,||\, e))}(N_U)$. If not so, rejects the login request; otherwise, accepts login request and temporarily stores ID_i till the end of the session.
3. Generates a random nonce N_S and computes $S_1 = h(h(ID_i \oplus h(x \,||\, e)) \,||\, N_S \,||\, B_i)$.
4. $S \rightarrow U_i : \{N_S, S_1\}$.

On receiving $\{N_S, S_1\}$, from S, the user performs the following steps to authenticate the server:

5. Verifies if $S_1 = h(A_i \,||\, N_S \,||\, N_U)$. If so, then the legitimacy of server gets confirmed; otherwise lapse the session.
6. $U_i \rightarrow S : \{S_2\}$. Computes $S_2 = h((E_i \oplus hpw) \,||\, N_S)$.

On receiving $\{S_2\}$ from U_i, the server S performs the following:

7. Verifies whether $S_2 = h(h(x \,||\, e) \,||\, N_S)$. If so, authenticity of U_i is confirmed; otherwise this session is lapsed.

At the end of the session both U_i and S independently compute the session key. S computes the session key as $SK = h(h(ID_i \oplus h(x \,||\, e)) \,||\, h(x \,||\, e) \,||\, N_S \,||\ \ B_i)$ and U as $SK = h(A_i \,||\, (E_i \oplus hpw) \,||\, N_S \,||\, N_U)$.

3.4 Password change phase

The password change phase facilitates the user to change his password PW to a new one, say PW^*. Following are the steps performed by the user and its mobile device:

1. U_i inserts his identity $ID_{i,}$ password PW and imprints his fingerprint F_i into its mobile device.
2. The mobile device computes $hpw = h(PW \oplus N) \oplus F_i$, and checks if $h_{(Ri \oplus hpw)}$ $(F_i) = V_i$. If not so, then the mobile device stops further action. Otherwise, allows the user to insert a new password PW^*.
3. The mobile device computes $hpw^* = h(PW^* \oplus N) \oplus F_i$, $R_i^* = R_i \oplus hpw \oplus hpw^*$, $E_i^* = E_i \oplus hpw \oplus hpw^*$ and replace R_i and E_i with R_i^* and E_i^* respectively.

4 Cryptanalysis of Chen et al.'s scheme and Truong et al.'s scheme

According to Rhee et al. [28], the mobile devices such as PCs, mobile phones, USBs, etc, are not perfectly tamper-resistant. In addition, some literature [29–31] indicates that the stored information of a mobile device may not be secure. Thus, we can assume that an attacker U_A can extract the information stored inside a mobile device.

4.1 Cryptanalysis of Chen et al.'s scheme

First of all we describe an attack by Truong et al. on Chen et al.'s scheme. It is given as follows:

4.1.1 User and server spoofing attack by Truong et al.

In Chen et al.'s scheme, using ID_i of U_i, an attacker U_A can re-register to S by sending $\{ID_i, h(PW_{new} \oplus N_{new}), F_{inew})$. Then, S sends back the information $= \{R_i, V_i, h(.), h_k(.)\}$. On obtaining R_i, the attacker U_A can easily obtain $h(IDi \oplus x)$ by computing $R_i \oplus h(PW_{new} \oplus N_{new}) \oplus F_{inew}$. With values $= \{ID_i$ and $h(ID_i \oplus x)\}$ corresponding to U_i, the attacker U_A can successfully login S as U_i. For this, U_A computes $C_1 = N_A \oplus h(ID_i \oplus x)$, $C_2 = h_{h(IDi \oplus_x)}(N_A)$, here N_A is a random nonce selected by U_A. Then, U_A sends $\{ID_i, C_1, C_2\}$ to S, to impersonate U_i. Moreover, having $h(ID_i \oplus x)$, the attacker U_A can successfully imitate S to deceive U_i. When U_i sends the login request $= \{ID_i, C_1, C_2\}$ to S, the attacker U_A blocks this package and computes $N_U = C_1 \oplus h(ID_i \oplus x)$, $S_1 = h(h(ID_i \oplus x) \,||\, N_U)$, and sends $\{S_1\}$ to U_i. On receiving $\{S_1\}$ from $U_{A,}$ the user U_i compares $h(h(ID_i \oplus x) \,||\, N_U)$ with S_1; obviously, both of these are equal. In this way, U_A successfully makes fool of U_i.

Now, further cryptanalysis of Chen et al.'s scheme by us is presented below:

4.1.2 Password guessing attack

This attack is an extension of 'user and server spoofing attack' proposed by Truong et al. [26] on Chen et al.'s scheme. As explained by Truong et al., an attacker U_A can obtain the value $h(ID_i \oplus x)$ by registering to S using identity ID_i of U_i transmitted in login request. We consider the situation when this attacker gets the mobile device corresponding to the identity ID_i, and he extracts the information $\{R_i, V_i, h(.), h_k(.), N\}$ stored inside the mobile device. Then, U_A can guess password PW and can obtain imprint of fingerprint F_i of U as explained below:

1. U_A computes $R_i \oplus h(ID_i \oplus x)$ which needs to be $hpw = h(PW \oplus N) \oplus F_i$.
2. Guesses a password PW^*, computes $h(PW^* \oplus N)$ using N and obtains $F_i^* = hpw \oplus h(PW^* \oplus N)$.
3. Checks if $h_{h(IDi \oplus x)}(F_i^*) = V_i$. If so, then $U_A$now possesses PW as well as the imprint of fingerprint F_i of U_i.

U_A may use these values of U_i to access other servers on behalf of U_i as it is convenient for a user to keep same password for different servers.

4.1.3 Scheme lacks three-factor authentication

Three-factor security is about employing three independent factors (what someone knows—like password, what someone possesses—like token or card or a small device, what someone is—like fingerprint or voice pattern) to test legitimacy and provide the entitled services to a login user. In Chen et al.'s scheme if an entity obtains ID_i and $A_i = h(ID_i \oplus x)$, then there is no need of password, fingerprint and mobile device. Without knowing password of U_i, without having mobile device of U_i, and being other than the legitimate user, that is, without the fingerprint of legitimate user, one can easily pass the mutual authentication. Therefore, the entire exercise of employing three factors to secure the scheme goes in vain.

4.1.4 Server's secret key is at risk

After obtaining $h(ID_i \oplus x)$ as described by Truong, an attacker U_A may try to guess the secret key x of S, as he knows ID_i of U. A malicious user U_K can also try to guess x as he knows his ID_i and can easily extract $A_i = h(ID_i \oplus x)$ from his mobile device.

4.2 Cryptanalysis of Truong et al.'s scheme

4.2.1 User and server impersonation attack using only an intercepted login request via ID guessing

Suppose U_A intercepts a login request $\{CID, e, C_1, C_2\}$ of U_i. Now, U_A performs the following steps:

1. Computes $CID \oplus C_1$, which needs to be $ID_i \oplus h(x \,||\, e)$.
2. Computes the hash of $ID_i \oplus h(x \,||\, e)$ and thus obtains $A_i = h(ID_i \oplus h(x \,||\, e))$, which is a secret shared between U_i and S. The value A_i acts as key for keyed hash function $h_k(.)$.

3. Guesses ID_i^* as identity of U_i, computes $h^*(x \,||\, e) = ID_i^* \oplus [ID_i \oplus h(x \,||\, e)]$ and obtains $N^* = C_1 \oplus h^*(x \,||\, e)$.
4. Computes $C_2^* = h_{Ai}(N^*)$ and checks if $C_2 = C_2^*$. If so, it implies that he has correctly guessed the identity of U_i and has correctly obtained the value $h(x \,||\, e)$.

Now, U_A possess $A_i = h(ID_i \oplus h(x \,||\, e))$ and $h(x \,||\, e)$. Thus, he can easily mount user and server impersonation attack on the scheme in the following manner:

1. U_A generates a random nonce N_A and computes $C_{A1} = N_A \oplus h(x \,||\, e)$, $C_{A2} = h_{Ai}(N_A)$ and $CID_A = ID_i \oplus N_A$.
2. $U_A \rightarrow S$: login request $\{CID_A, e, C_{A1}, C_{A2}\}$.

On receiving $\{CID_A, e, C_{A1}, C_{A2}\}$, the server S performs the following steps:

3. Obtains $N_A = C_{A1} \oplus h(x \,||\, e)$, $ID_i = CID_A \oplus N_A$, and checks ID_i, which is obviously valid.
4. Checks if $h_{h(IDi \oplus h(x \,||\, e))}(N_A) = C_{A2}$, which obviously holds. S generates a random nonce N_{AS}, and computes $S_{A1} = h(h(ID_i \oplus h(x||e)) \,||\, N_{AS} \,||\, N_A)$ and sends $\{N_{AS}, S_{A1}\}$.

On receiving $\{N_{AS}, S_{A1}\}$, the attacker U_A performs as follows:

5. Computes $S_{A2} = h(h(x \,||\, e)) \,||\, N_{AS})$ and sends it to S.

On receiving S_{A2}, the server S, performs the following

6. Computes $h(h(x \,||\, e)) \,||\, N_{AS})$ and checks if it is equal to the received S_{A2}, which obviously holds. At last, U_A computes $SK = h(A_i \,||\, h(x \,||\, e) \,||\, N_{AS} \,||\, N_A)$, which is exactly the same session key that the server calculates.

In this way, U_A can successfully impersonate U_i and S without knowing password of U_i, without having mobile device of U_i, and without the imprint of fingerprint F_i of U_i.

4.2.2 *User and server impersonation attack using information extracted from mobile device of U_i and an intercepted login request*

In Truong et al.'s scheme, it is very easy to relate a mobile device with its corresponding login request $\{CID, e, C_1, C_2\}$ because the random value e is common in both. Consider the situation when an attacker U_A successfully extracts the information $\{R_i, V_i, E_i, e, h(.), h_k(.), N\}$ from mobile device of U_i. Now, using the values extracted from mobile device and one of its corresponding intercepted login requests, U_A can obtain secret values shared between U_i and S in the following manner:

1. Obtains $I = h(ID_i \oplus h(x \,||\, e)) \oplus h(x \,||\, e) = E_i \oplus R_i$, using values from mobile device.
2. Obtains $L = ID_i \oplus h(x \,||\, e) = C_1 \oplus CID$, using values from login request.
3. Computes $A_i = h(L) = h(ID_i \oplus h(x \,||\, e))$, which is a secret shared between U_i and S. The value A_i acts as key for keyed hash function $h_k(.)$.
4. Computes $h(x \,||\, e) = I \oplus A_i = [h(ID_i \oplus h(x \,||\, e)) \oplus h(x \,||\, e)] \oplus A_i$, which is another secret value embedded in the mobile device of U_i. The value $h(x \,||\, e)$ has key role in the entire login-authentication phase and legally only S can compute it.

Now, U_A possess $A_i = h(ID_i \oplus h(x \,||\, e))$ and $h(x \,||\, e)$. Thus, he can easily mount user and server impersonation attack on the scheme in similar way as demonstrated in previous subsection.

4.2.3 Password guessing attack

We further extend the previous scenario to password guessing attack. U_A having the values $\{R_i, V_i, e, h(.), h_k(.), N\}$ extracted from the mobile device and possessing $A_i = h(ID_i \oplus h(x \,||\, e))$ and $h(x \,||\, e)$ can easily guess the password of U_i as explained below:

1. Computes $h(x \,||\, e) \oplus E_i$ which needs to be $hpw = h(PW \oplus N) \oplus F_i$.
2. Guesses a password PW^*, computes $h(PW^* \oplus N)$ using N and obtains $F_i^* = hpw \oplus h(PW^* \oplus N)$.
3. Checks if $h_{h(IDi \oplus h(x \,||\, e))}(F_i^*) = V_i$. If so, then U_A now possesses PW as well as the imprint of fingerprint F_i of U_i.

U_A may use these values of U_i to access other servers on behalf of U_i as it is convenient for a user to keep same password for different servers.

4.2.4 Three-factor security and three way challenge response in scheme is at risk

In Truong et al.'s scheme, we observe that once an attacker U_A intercepts a login request, then he does not need password of U_i, mobile device of U_i, and even imprint of fingerprint F_i of U_i; he can successfully pass the entire login-authentication phase to enjoy the privileges meant for U_i and makes fool of S. Besides, he can also calculate the current session key and even any previous session key (using the corresponding intercepted login-authentication messages). With the session key obtained (previous or current), he can know about the secret communication held between U_i and S and may take further advantage. In this way, three-factor security and three-way challenge-response handshake technique employed in the scheme proves to be useless.

4.2.5 Server's secret key is at risk

A malicious user U_K can easily extract values $\{R_i, V_i, E_i, e, h(.), h_k(.), N\}$ from his mobile device. Then, he can guess the secret key x of S in the following manner:

1. Computes $hpw = h(PW \oplus N) \oplus F_i$, and obtains $h(x \,||\, e) = E_i \oplus hpw$.
2. Guesses x^*, computes $h(x^* \,||\, e)$ and checks if $h(x \,||\, e) = h(x^* \,||\, e)$. If so, it implies his success.

Apart from a malicious user U_K, an adversary U_A can also guess the secret key of S. As described in Sects. 4.2.1 and 4.2.2, U_A can obtain $h(x \,||\, e)$ and he has e from intercepted login request; so he can guess the secret key x of S in similar way as in step-2 by U_K.

5 The proposed scheme

In this section we propose an improvement of Chen et al.'s scheme and Truong et al.'s scheme. Our proposed scheme remedies the flaws and maintains the advantages of both the original versions. The scheme is divided into the four phases of registration, login, mutual authentication and key agreement phase, and password change phase.

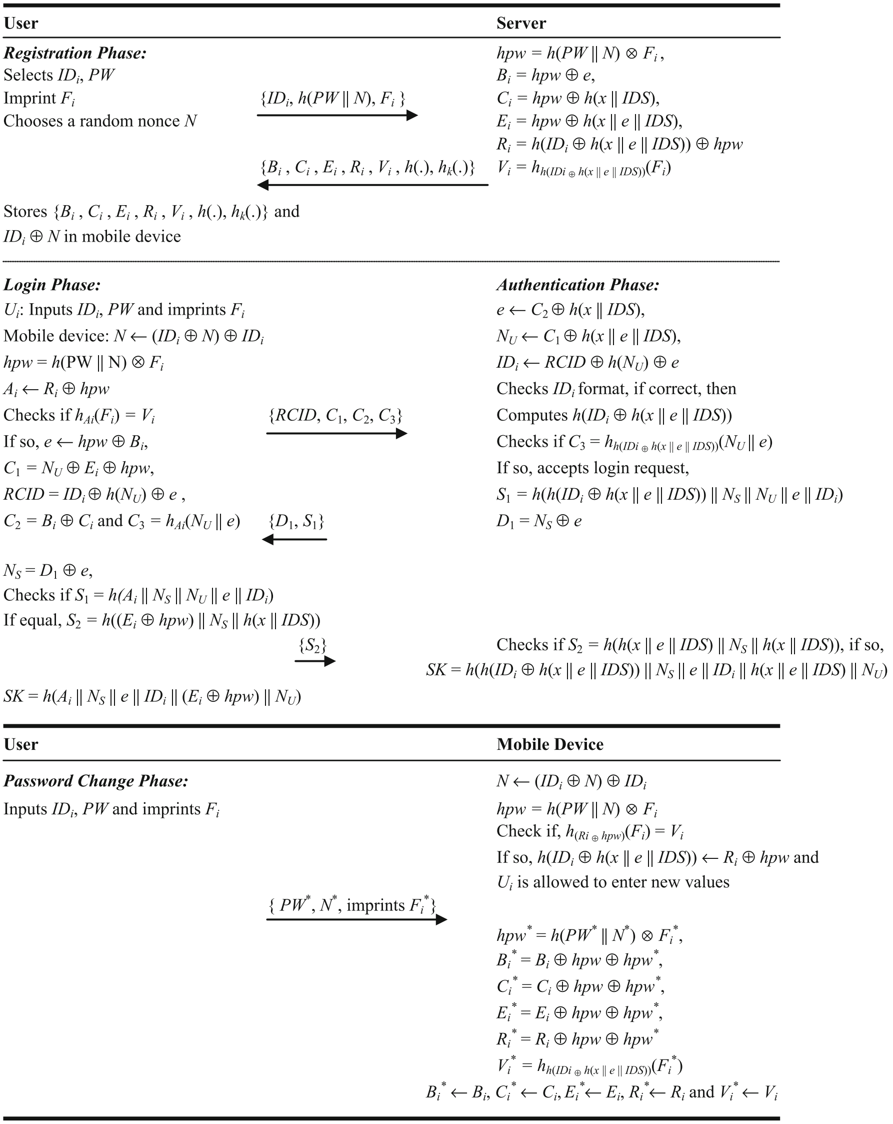
General idea of the proposed scheme

- Instead of $h(x \,||\, e)$ used in Truong et al.'s scheme, we make use of $h(x \,||\, e \,||\, IDS)$, where *IDS* is the secret identity of server S known only to S. Unlike $h(x \,||\, e)$, the value $h(x \,||\, e \,||\, IDS)$ contains two unknown values $\{x, IDS\}$. This resists anyone from guessing the secret key and the secret identity of server S from $h(x \,||\, e \,||\, IDS)$.
- Unlike Truong et al.'s scheme, in the proposed scheme, U_i does not stores random nonce N in plaintext inside its mobile device. This resists the password guessing attack in case of mobile device loss.
- While computing $hpw = h(PW \,||\, N) \otimes F_{i,}$ we make use of bitwise NOR operator $\otimes$ instead of exclusive OR operator $\oplus$. This restricts any adversary from extracting the values $\{h(PW \,||\, N), F_i\}$ out of *hpw* from a lost mobile device. With this little modification, guessing attempt is not possible in the proposed scheme by the way similar to password guessing attempt in Chen et al.'s scheme as shown in Sect. 4.1.2.
- Unlike Truong et al.'s scheme, in the proposed scheme, the random value e (generated by S for each user) is neither stored in plaintext inside U_i's mobile device nor sent in plaintext through each login request. This resists anyone from tracing two (or more) login requests belonging to a particular user. Moreover, it prohibits anyone from relating a lost mobile device with its corresponding login request. This very feature safeguards our scheme from various attacks that can be mounted on a dynamic ID-based scheme. A few of such attacks are as follows:
- Attacks using two login requests belonging to the same user as shown in [32] on Gao-Tu's scheme [33].
- Attacks using lost mobile device/smart card and its corresponding login request as we confirmed mobile device loss attack and offline password guessing attack on Truong et al.'s scheme.

Besides, it imparts strong user anonymity which some schemes [26,33] lack even after using dynamic ID for each new login attempt.

- We modify the value $CID = ID \oplus N_U$ of Truong et al.'s scheme to $RCID = ID \oplus h(N_U) \oplus e$ in the proposed scheme. This modification resists guessing of user's *ID* and subsequent impersonation attacks using only an intercepted login request as in [26].

Now, we describe the proposed scheme with its four phases along with Fig. 1, depicting the entire protocol in a sequence.



User — **Server**

Registration Phase:

Selects ID_i, PW

Imprint F_i

Chooses a random nonce N

$\xrightarrow{\{ID_i,\ h(PW \| N),\ F_i\}}$

$hpw = h(PW \| N) \otimes F_i$,

$B_i = hpw \oplus e$,

$C_i = hpw \oplus h(x \| IDS)$,

$E_i = hpw \oplus h(x \| e \| IDS)$,

$R_i = h(ID_i \oplus h(x \| e \| IDS)) \oplus hpw$

$V_i = h_{h(IDi \oplus h(x \| e \| IDS))}(F_i)$

$\xleftarrow{\{B_i,\ C_i,\ E_i,\ R_i,\ V_i,\ h(.),\ h_k(.)\}}$

Stores $\{B_i, C_i, E_i, R_i, V_i, h(.), h_k(.)\}$ and $ID_i \oplus N$ in mobile device

Login Phase:

U_i: Inputs ID_i, PW and imprints F_i

Mobile device: $N \leftarrow (ID_i \oplus N) \oplus ID_i$

$hpw = h(\mathrm{PW} \| \mathrm{N}) \otimes F_i$

$A_i \leftarrow R_i \oplus hpw$

Checks if $h_{Ai}(F_i) = V_i$

If so, $e \leftarrow hpw \oplus B_i$,

$C_1 = N_U \oplus E_i \oplus hpw$,

$RCID = ID_i \oplus h(N_U) \oplus e$,

$C_2 = B_i \oplus C_i$ and $C_3 = h_{Ai}(N_U \| e)$

$\xrightarrow{\{RCID,\ C_1,\ C_2,\ C_3\}}$

Authentication Phase:

$e \leftarrow C_2 \oplus h(x \| IDS)$,

$N_U \leftarrow C_1 \oplus h(x \| e \| IDS)$,

$ID_i \leftarrow RCID \oplus h(N_U) \oplus e$

Checks ID_i format, if correct, then

Computes $h(ID_i \oplus h(x \| e \| IDS))$

Checks if $C_3 = h_{h(IDi \oplus h(x \| e \| IDS))}(N_U \| e)$

If so, accepts login request,

$S_1 = h(h(ID_i \oplus h(x \| e \| IDS)) \| N_S \| N_U \| e \| ID_i)$

$D_1 = N_S \oplus e$

$\xleftarrow{\{D_1,\ S_1\}}$

$N_S = D_1 \oplus e$,

Checks if $S_1 = h(A_i \| N_S \| N_U \| e \| ID_i)$

If equal, $S_2 = h((E_i \oplus hpw) \| N_S \| h(x \| IDS))$

$\xrightarrow{\{S_2\}}$

Checks if $S_2 = h(h(x \| e \| IDS) \| N_S \| h(x \| IDS))$, if so,

$SK = h(h(ID_i \oplus h(x \| e \| IDS)) \| N_S \| e \| ID_i \| h(x \| e \| IDS) \| N_U)$

$SK = h(A_i \| N_S \| e \| ID_i \| (E_i \oplus hpw) \| N_U)$

User — **Mobile Device**

Password Change Phase:

Inputs ID_i, PW and imprints F_i

$N \leftarrow (ID_i \oplus N) \oplus ID_i$

$hpw = h(PW \| N) \otimes F_i$

Check if, $h_{(Ri \oplus hpw)}(F_i) = V_i$

If so, $h(ID_i \oplus h(x \| e \| IDS)) \leftarrow R_i \oplus hpw$ and

U_i is allowed to enter new values

$\xrightarrow{\{\ PW^*,\ N^*,\ \text{imprints } F_i^*\}}$

$hpw^* = h(PW^* \| N^*) \otimes F_i^*$,

$B_i^* = B_i \oplus hpw \oplus hpw^*$,

$C_i^* = C_i \oplus hpw \oplus hpw^*$,

$E_i^* = E_i \oplus hpw \oplus hpw^*$,

$R_i^* = R_i \oplus hpw \oplus hpw^*$

$V_i^* = h_{h(IDi \oplus h(x \| e \| IDS))}(F_i^*)$

$B_i^* \leftarrow B_i$, $C_i^* \leftarrow C_i$, $E_i^* \leftarrow E_i$, $R_i^* \leftarrow R_i$ and $V_i^* \leftarrow V_i$

Fig. 1 The proposed scheme

5.1 Registration phase

When a user U_i wants to register to the server S, he has to submit his identity ID_i, $h(PW||N)$ and his fingerprint F_i by imprinting on the sensor to S, where PW is U_i's password and N is a nonce chosen by U_i. The details are described as follows:

1. S generates a random value e. Computes $hpw = h(PW||N) \otimes F_i$, $B_i = hpw \oplus e$, $C_i = hpw \oplus h(x||IDS)$, $E_i = hpw \oplus h(x||e||IDS)$, $R_i = h(ID_i \oplus$

$h(x\,||\,e\,||\,IDS)) \oplus hpw$ and $V_i = h_{h(IDi \oplus h(x\,||\,e\,||\,IDS))}(F_i)$; here IDS is the secret identity of S and $\otimes$ is bitwise NOR operator.
2. $S \Rightarrow U_i$: $\{B_i, C_i, E_i, R_i, V_i, h(.), h_k(.)\}$ to the user's mobile device through a secure channel.
3. Upon receiving the message from server, U_i stores it and enters $ID_i \oplus N$ into his mobile device.

5.2 Login phase

The user U_i types his identity ID_i and password PW, and imprints his fingerprint F_i on sensor, to login S. Then the mobile device performs the following steps:

1. Extracts $N = (ID_i \oplus N) \oplus ID_i$ and computes $hpw = h(PW\,||\,N) \otimes F_i$. Extracts $A_i = R_i \oplus hpw$ and checks if $h_{Ai}(F_i) = V_i$. If they are not equal, the mobile device terminates session; otherwise, it allows user to go to the next step.
2. Retrieves $e = hpw \oplus B_i$ and generates a random nonce N_U. Computes $C_1 = N_U \oplus E_i \oplus hpw$, $RCID = ID_i \oplus h(N_U) \oplus e$, $C_2 = B_i \oplus C_i$ and $C_3 = h_{Ai}(N_U\,||\,e)$.
3. $U_i \rightarrow S$: login request = $\{RCID, C_1, C_2, C_3\}$, where '→' denotes a public channel.

5.3 Mutual authentication and session key agreement phase

When the server S receives the login request $\{RCID,\ C_1,\ C_2,\ C_3\}$ from U_i, then the following steps are performed by U_i and S to achieve mutual authentication.

1. S retrieves $e = C_2 \oplus h(x\,||\,IDS), N_U = C_1 \oplus h(x\,||\,e\,||\,IDS), ID_i = RCID \oplus h(N_U) \oplus e$ and checks the validity of ID_i. Then, S computes $h(ID_i \oplus h(x\,||\,e\,||\,IDS))$ and checks if $C_3 = h_{h(IDi \oplus h(x\,||\,e\,||\,IDS))}(N_U\,||\,e)$. If they are equal, S accepts U_i's request, otherwise rejects. Then, S generates a random nonce N_S, computes $S_1 = h(h(ID_i \oplus h(x\,||\,e\,||\,IDS))\,||\,N_S\,||\,N_U\,||\,e\,||\,ID_i)$ and $D_1 = N_S \oplus e$. S sends $\{D_1, S_1\}$ to U_i.
2. On receiving the message (D_1, S_1), the user U_i retrieves $N_S = D_1 \oplus e$ checks if $S_1 = h(A_i\,||\,N_S\,||\,N_U\,||\,e\,||\,ID_i)$. If they are not equal, mobile device terminates session; otherwise, it computes $S_2 = h((E_i \oplus hpw)\,||\,N_S\,||\,h(x\,||\,IDS))$. U_i sends $\{S_2\}$ to S.
3. On receiving the message $\{S_2\}$, the server S checks if $S_2 = h(h(x\,||\,e\,||\,IDS)\,||N_S||\,h(x\,||\,IDS))$. If they are not equal, S terminates session; otherwise, it computes session key $SK = h(h(ID_i \oplus h(x\,||\,e\,||\,IDS))\,||\,N_S\,||\,e\,||\,ID_i\,||\,h(x\,||\,e\,||\,IDS)\,||\,N_U)$. Similarly, U_i also computes $SK = h(A_i\,||\,N_S\,||\,e\,||\,ID_i\,||\,(E_i \oplus hpw)\,||\,N_U)$.

5.4 Password change phase

This phase is mainly invoked whenever the user wants to change his password PW to the new password PW^*. With this phase U_i can not only change his password PW to the new password PW^*, he can also change the random nonce N to a new random nonce N^* and his fingerprint F_i to some other fingerprint F_i^* (that is, some other finger can be used to imprint a different fingerprint). Following are the detailed steps for this phase:

1. U_i enters his identity $ID_{i,}$ password *PW* and imprints his fingerprint F_i into the mobile device and request to change password (and random nonce and fingerprint).
2. U_i's mobile device extracts $N = (ID_i \oplus N) \oplus ID_i$, computes $hpw = h(PW \,||\, N) \otimes F_i$, and continues to check if $h_{(Ri \oplus hpw)}(F_i) = V_i$. If they are not equal, then U_i 's mobile device rejects the request and terminates the operation. Otherwise, extracts $h(ID_i \oplus h(x \,||\, e \,||\, IDS)) = R_i \oplus hpw$ and U_i is allowed to enter new values. U_i chooses new password (PW^*), new random nonce (N^*) and decides another fingerprint (F_i^*). Then U_i submits $\{PW^*, N^*\}$ and imprints F_i^* on the sensor.
3. U_i's mobile device computes $hpw^* = h(PW^* \,||\, N^*) \otimes F_i^*$, $B_i^* = B_i \oplus hpw \oplus hpw^*$, $C_i^* = C_i \oplus hpw \oplus hpw^*$, $E_i^* = E_i \oplus hpw \oplus hpw^*$, $R_i^* = R_i \oplus hpw \oplus hpw^*$, $V_i^* = h_{h(IDi \oplus h(x \,||\, e \,||\, IDS))}(F_i^*)$; and stores B_i^*, C_i^*, E_i^*, R_i^* and V_i^* to replace B_i, C_i, E_i, R_i and V_i respectively.

6 Security analysis of the proposede scheme

The proposed scheme is an improvement of both schemes: Chen et al.'s scheme and Truong et al.'s scheme. Thus, we analyze the security of proposed scheme mainly under the following categories:

- Resistance to attacks pointed out by Truong et al. on Chen et al.'s scheme.
- Resistance to attacks pointed out by us on Chen et al.'s scheme.
- Resistance to attacks pointed out by us on Truong et al.'s scheme.
- Resistance to some other attacks.

At the end of this section, we would also describe a few important usable features accomplished by the proposed scheme.

- Usability

6.1 Resistance to attacks pointed out by Truong et al. on Chen et al.'s scheme

We have inherited the basic structure of the proposed scheme from Truong et al.'s scheme. Our scheme uses *RCID* instead of plaintext identity in login request, provides very strong user anonymity and employs three-way challenge response handshake technique. Truong et al.'s scheme is an improvement of Chen et al.'s scheme so as to resist the attacks verified by Truong et al. Therefore, attacks demonstrated by Truong et al. on Chen et al.'s scheme are not applicable on the proposed scheme.

6.2 Resistance to attacks pointed out by us on Chen et al.'s scheme

Attacks that we have pointed out on Chen et al.'s scheme are due to static identity of U_i. Unlike Chen et al.'s scheme, in the proposed scheme, identity ID_i of U_i is not available in plaintext form through an intercepted login request; also neither an adversary U_A nor a malicious user U_K can obtain sufficient values to calculate a valid login request and other subsequent authentication messages so as to pass the login-authentication phase. For this reason, password guessing attack, server's secret key guessing attack by an adversary U_A, etc, cannot be mounted on the proposed scheme in way similar to that is employed for Chen et al.'s scheme.

6.3 Resistance to attacks pointed out by us on Truong et al.'s scheme

6.3.1 Resists user and server impersonation attack using an intercepted login request via IDi guessing

Unlike Truong et al.'s scheme, in the proposed scheme, U_A cannot obtain $ID_i \oplus h(x \,||\, e \,||\, IDS)$ from an intercepted login request by performing $C_1 \oplus RCID (= N_U \oplus h(x \,||\, e \,||\, IDS) \oplus ID_i \oplus h(N_U) \oplus e)$. Thus, here guessing ID_i^*, retrieving N_U^* and verifying the guess using C_3 is not possible. Consequently, U_A cannot obtain the secret values $A_i = h(ID_i \oplus h(x \,||\, e \,||\, IDS))$ and $h(x \,||\, e \,||\, IDS)$. In addition, U_A cannot calculate a valid login request without knowing $h(x \,||\, IDS)$. Therefore, impersonation attacks using an intercepted login request via ID_i guessing are not feasible in the proposed scheme.

6.3.2 Resistance to mobile device loss attack

Suppose an attacker steals/finds mobile device of U_i and somehow [29–31] extracts all information stored inside it. But U_A can neither make any guess nor can he obtain any secret value shared between user and server. U_A cannot obtain *hpw* from any of the values $\{B_i, C_i, E_i, R_i\}$ without knowing the values $\{e, h(x \,||\, IDS), h(x \,||\, e \,||\, IDS), h(ID_i \oplus h(x \,||\, e \,||\, IDS))\}$. Even if he somehow obtains *hpw*, he cannot guess the password *PW* from $hpw = h(PW \,||\, N) \otimes F_i$ for the following reasons:

- Bitwise NOR operator $\otimes$ is used in *hpw* instead of bitwise XOR operator $\oplus$. It defends the values $\{h(PW \,||\, N), F_i\}$ from extraction out of *hpw*.
- U_A cannot extract random nonce N from $ID_i \oplus N$ without knowing ID_i.
- U_A cannot imprint the fingerprint F_i of U_i.

Furthermore, unlike Truong et al.'s scheme, here it's not possible to relate a lost mobile device $= \{B_i, C_i, E_i, R_i, V_i, ID_i \oplus N, h(.), h_k(.)\}$ with its corresponding login request $= \{RCID, C_1, C_2, C_3\}$, because no value is common between them. We recall that in Truong et al.'s scheme, the random value e is common between mobile device $= \{R_i, V_i, E_i, e, N, h(.), h_k(.)\}$ and login request $= \{CID, e, C_1, C_2\}$.

Let us consider the situation, when somehow U_A manages to obtain the login request corresponding to a lost mobile device. Still he cannot be successful in obtaining any value helpful to break through the security of the proposed scheme. With values extracted from mobile device he can obtain $h(x \,||\, e \,||\, IDS) \oplus h(ID_i \oplus h(x \,||\, e \,||\, IDS))$ by performing $E_i \quad \oplus \quad R_i$. But, unlike Truong et al.'s scheme, U_A cannot obtain $ID_i \oplus h(x \,||\, e \,||\, IDS)$ using values from corresponding intercepted login request. So, he cannot continue to calculate $A_i = h(ID_i \oplus h(x \,||\, e \,||\, IDS))$ and hence not

$h(x \,||\, e \,||\, IDS)$. Without these values, U_A cannot compute a valid login request to impersonate U_i, as is obvious from the construction of $C_1 = N_U \oplus E_i \oplus hpw (= N_U \oplus h(x \,||\, e \,||\, IDS))$ and $C_3 \quad = h_{Ai}(N_U \,||\, e)$. Besides, it is not possible to obtain the values $\{A_i = h(ID_i \oplus h(x \,||\, e \,||\, IDS)), h(x \,||\, e \,||\, IDS), e, ID_i\}$ in some other way: by combining values either from mobile device or from the intercepted login request or from both. Therefore, we can say that the proposed scheme is secure even if all the values from the mobile device are extracted.

6.3.3 Resistance to password guessing attack

We explained in previous sub-section that it is far from being possible for U_A to obtain the value *hpw* either from the mobile device alone or simultaneously employing the corresponding intercepted login request. Undoubtedly, U_A can perform $E_i \oplus R_i$ to obtain $h(x \,||\, e \,||\, IDS) \oplus h(ID_i \quad \oplus \quad h(x \,||\, e \,||\, IDS))$. But, without having master key x of S, secret identity *IDS* of S, random value e of U_i, and the identity ID_i of U_i, an attacker U_A cannot compute any such value with which *PW* can be guessed. Besides, password of U_i is very well protected by the random nonce N and imprint of fingerprint F_i of U_i. Unlike Truong et al.'s scheme, in the proposed scheme the random nonce N is not stored in plaintext inside the mobile device; rather N is stored as $ID_i \quad \oplus \quad N$. Also, identity ID_i of U_i is not transmitted in plaintext in login request. In this way, unavailability of N makes it impossible for U_A to guess *PW*. Further, due to use of bitwise NOR operator $\otimes$ instead of bitwise XOR operator $\oplus$ in *hpw*, correctness of any guess for *PW* cannot be verified using V_i.

6.3.4 Server's secret key is not at risk

Even if a malicious legal user U_K extracts all information from his mobile device, he cannot guess the secret key of S. Certainly, U_K can obtain $h(x \,||\, IDS)$ and $h(x \,||\, e \,||\, IDS)$ from C_i and E_i by performing $C_i \oplus hpw$ and $E_i \oplus hpw$ respectively. Although U_K knows its random value e, yet it is not possible to guess two values $\{x$ and $IDS\}$ simultaneously in real polynomial time. Here, identity *IDS* of S is secret and known to S only.

Also, U_A cannot guess the secret key x of S, either using a stolen mobile device alone or an intercepted login request alone or both. This is due to the fact that whatever value or combination of values U_A tries to guess x, each of these values (like $\{h(x \,||\, e \,||\, IDS) \oplus h(ID_i \oplus h(x \,||\, e \,||\, IDS))\}$ by performing $\{E_i \oplus R_i\}$ or $\{N_U \oplus h(x \,||\, e \,||\, IDS) \oplus ID_i \oplus h(N_U) \oplus e\}$ by performing $\{C_1 \oplus RCID\}$ or $\{N_U \oplus h(x \,||\, e \,||\, IDS) \oplus hpw \oplus h(x \,||\, IDS)\}$ by performing $\{C_1 \oplus C_i\}$, etc) require at least two unknown values to guess simultaneously; which is not possible in real polynomial time. For instance, U_A cannot guess the secret key x from $h(x \,||\, e \,||\, IDS) \oplus h(ID_i \oplus h(x \,||\, e \,||\, IDS))$ without knowing the values $\{e, IDS, ID_i\}$.

6.4 Resistance to some other attacks

This category also involves security analysis concerning the attacks which Truong et al.'s scheme withstands.

6.4.1 Resistance to malicious user attack

A legitimate but malicious user U_K can obtain the values $\{e, h(x||IDS), h(x||e||IDS), h(ID_i \oplus h(x\,||\,e\,||\,IDS))\}$ from his mobile device. Among these values only $h(x\,||\,IDS)$ is common for all users; and rest of the values are user-specific due to identity ID_i and random value e (which are different for each user). However, to calculate a valid login request on behalf of some other legitimate user U_L, malicious user U_K must know the values $\{e_L, h(x\,||\,e_L\,||\,IDS), h(ID_L \oplus h(x\,||\,e_L\,||\,IDS))\}$. Besides, if U_K tries to calculate an arbitrary but valid login request on behalf of an arbitrary identity ID_f and an arbitrary random value e_f, then he cannot achieve success in such effort without separately knowing the secret key x and secret identity *IDS* of S. Subsequently, U_K cannot make fool of S by acting as a legitimate user U_L, or an arbitrary user U_f. For similar reasons, U_K cannot make fool of U_i by acting as S.

6.4.2 Resistance to replay attack

In this attack, an attacker replays previously intercepted messages of the communication participants of a scheme. Like Truong et al., we also make use of random nonce (N_U is used by the user and N_S is used by the server) and three-way challenge response handshake technique to withstand replay attacks. Suppose, U_A replays the login request $= \{RCID, C_1, C_2, C_3\}$ to S, then S will send the response message $\{D_1, S_1\}$ to U_A. Without knowing $\{e, h(x\,||\,IDS), h(x\,||\,e\,||\,IDS)\}$, here e is needed to retrieve the random nonce N_S from D_1, the attacker U_A cannot compute $S_2 = h(h(x\,||\,e\,||\,IDS)\,||\,N_S\,||\,h(x\,||\,IDS))$ to respond to S. Consequently, S will pick out this forged login attempt when it will not receive any response to the message $\{D_1, S_1\}$ and will terminate the session. Hence, in the proposed scheme U_A cannot login successfully by replaying current or an old intercepted login request.

6.4.3 Resistance to known-key attack

Resistance to known-key attack ensures that compromise of a past session key will not result in deriving any further session key. In the proposed scheme, none of the values involved in computing the session key $SK = h(A_i\,||\,N_S\,||\,e\,||\,ID_i\,||\,h(x\,||\,e\,||\,IDS)\,||)$ (N_U) is available in plaintext. If U_A somehow obtains a previous/past session key, he cannot gain $\{A_i, e, ID_i, h(x\,||\,e\,||\,IDS)\}$ due to the one-way property of hash function. Moreover, the random nonce $\{N_U, N_S\}$ impart dynamic nature to the session key; and an attacker cannot predict what random nonce to be used in a future session key. Thus, U_A cannot derive any future session key.

6.4.4 Resistance to denial-of-service attack

In this attack, an attacker can update false verification information of a valid user, then this legal user cannot login S successfully anymore. Like Truong et al.'s scheme, mounting this attack through password change phase is not possible in the proposed scheme because mobile device can authenticate the legitimacy of user by verifying $h_{(Ri \oplus hpw)}(F_i) = V_i$, where $hpw = h(PW || N) \otimes F_i$. Undoubtedly, U_A cannot change or update user's information inside a lost mobile device, without having correct password and without being able to imprint the corresponding fingerprint F_i. Besides, the scheme caries the legacy of no verification table or database maintained at S, from its parent versions. So, the proposed scheme is free from denial-of-service attack.

6.4.5 Resistance to stolen verifier attack

Like Chen et al.'s scheme and Truong et al.'s scheme, in the proposed scheme S does not need to store anything with it. During registration phase, the server securely sends secret information $= \{B_i, C_i, E_i, R_i, V_i, h(.), h_k(.)\}$ to U_i, which the user saves in its mobile device. On receiving the login request $= \{RCID, C_1, C_2, C_3\}$, S uses its secret key x and secret identity *IDS* to retrieve values needed to verify the legitimacy of U_i. As S does not maintains any verification table or database with it, so the stolen verifier attack is not applicable on the proposed scheme.

6.4.6 Resistance to man-in-the-middle attack, parallel session attack and reflection attack

Man-in-the-middle attack is a form of attack in which an attacker acts as a mid-man between user and server so that he can masquerade both of them by intercepting and modifying the communicated messages [34]. In this attack, user and server believe that they are communicating with each other; however, each of them communicates with the man-in-the-middle, that is, the attacker. In parallel session attack, an adversary can complete an authentication session with server by initiating one or more sessions simultaneously; illustrations of parallel session attack are available in [35–37]. Reflection attack is applicable on authentication schemes employing challenge-response technique for mutual authentication when same challenge-response protocol is used by each entity to authenticate the other entity. With this attack the targeted entity is tricked to provide response to its own challenge. To get rid of these attacks, our improved scheme inherits the remedy from its parent schemes, that is, S stores identity ID_i of U_i, until the end of the session. When S receives the login request $= \{RCID, C_1, C_2, C_3\}$ of U_i, it authenticates the legitimacy of U_i. As soon as U_i is authenticated, S stores the identity ID_i of U_i, and keeps it until the end of the session to recognize the same identity login. U_A can immediately start a parallel session just after U_i sends its login request $= \{RCID, C_1, C_2, C_3\}$ to S or U_A can send a currently intercepted login request immediately after observing the response message $\{D_1, S_1\}$ from S as in [10,11]. But in both the cases, the request from U_A, will not be entertained by S because such requests will reach S before the end of the legal user U_i's session. Therefore, on obtaining a value of ID_i similar to one which is stored, S will identify it as a false login attempt by some fraud entity and will terminate the corresponding session.

6.5 Usability

6.5.1 Strong user anonymity

We can see that in Truong et al.'s scheme, the random value e is common in all login requests of a particular user; and e is also stored in plaintext inside the mobile device of user. In the proposed scheme, U_i sends the login request $\{RCID, C_1, C_2, C_3\}$ to S. Here, U_A cannot know about the user of an intercepted login request as it does not contain the identity ID_i of U_i in plaintext. In addition, each value of login request is new for a new login. Unlike Truong et al.'s scheme, no one can identify that any two login requests belong to the same user. Besides, no value is common between mobile device and login request generated from it. For this reason, it is near impossible to co-relate a lost mobile device and its corresponding login request. As a result, we can say that the proposed scheme provides strong user anonymity as compared to that provided in Truong et al.'s scheme.

6.5.2 Mutual authentication

In the proposed scheme, U_i receives $h(x \,||\, e \,||\, IDS)$ and random value e, embedded within E_i and B_i respectively, during registration phase. With the key $h(x \,||\, e \,||\, IDS)$, user U_i can compute C_1 and C_3; and with e he can compute *RCID*. Server S with secret key x, secret identity *IDS* can retrieve random value e. Then using $\{x, IDS, e\}$, S can retrieve random nonce N_U to compute $h_{h(IDi \oplus h(x \,||\, e \,||\, IDS))}(N_U \,||\, e)$. With this value, S can authenticate the legitimacy of U. Then S generates a random nonce N_S to send the response message $\{D_1, S_1\}$ to U_i. Only a legal user U_i can have values $\{h(x \,\|\, IDS), h(x \,||\, e \,||\, IDS)\}$ to suitably respond to S with correct S_2. In this way, like Truong et al.'s scheme, we also use random nonce and three-way challenge response handshake technique to achieve mutual authentication.

6.5.3 Session-key agreement

Like Truong et al.'s scheme, in the proposed scheme U_i and S independently calculate a common session key *SK* to ensure confidentiality of subsequent messages. The values $\{h(ID_i \oplus h(x \,||\, e \,||\, IDS)), N_{S,}e, ID_i, h(x \,||\, e \,||\, IDS), N_U\}$ included in session key $SK = h(h(ID_i \oplus h(x \,||\, e \,||\, IDS)) \,||\, N_S \,||\, e \,||\, ID_i \,||\, h(x \,||\, e \,||\, IDS) \,||\, N_U)$ are such that only legal partners (user and server) can calculate it.

7 Performance analysis VIA comparison

Now, we judge the proposed scheme for ease and efficiency by conducting a comparative analysis with Chen et al.'s scheme and Truong et al.'s scheme. Comparisons within the respective fields are conducted below along with corresponding table depicting the results in a nutshell.

7.1 Comparison of computational complexity, memory space required and communication cost

We compile Table 1 for comparison of computational complexity, memory space required and communication cost. Each of the three schemes use hash function $h(.)$, bitwise XOR operator $\oplus$, and string concatenation operator $||$. We make use of an additional operator, that is, bitwise NOR operator $\otimes$, in our scheme. Except hash function $h(.)$, all other operators require only few computations, so we neglect them and consider only hash function $h(.)$ for comparison of computational complexity. Findings of Table 1 are as follows:

- **Extra computational load on mobile device is nominal:** During registration phase and authentication phase, same number of hash functions is used in all the three schemes. The proposed protocol uses an extra hash function only during the login phase. Thus, we put nominal burden of only one hash function on mobile device. Consequently, our scheme is suitable for low computation power mobile device.
- **Extra computational load on remote server is quite less:** On comparing our scheme with Truong et al.'s scheme, we find that the number of hash functions used during registration phase is same. We also observe that during the authentication phase, increment in the number of hash functions from Chen et al.'s scheme to Truong et al.'s scheme and from Truong et al.'s scheme to the proposed scheme are same, which is two. We use only two more hash functions than the Truong et al.'s scheme. If we look-up the extra load at server side, then it is clear that Truong et al.'s scheme needs three hash functions more than those in Chen et al.'s scheme; but our scheme needs two hash functions more than those in Truong et al.'s scheme.
- **Total extra computational load is low:** In totality, Chen et al.'s scheme uses $12h(.)$, Truong et al.'s scheme uses $15h(.)$ and the proposed scheme uses $18h(.)$. Thus, in all three extra hash functions are required to achieve the aimed security and friendly features.

To compare the proposed scheme for memory space required and communication cost, we assume all the values $\{ID, PW, x, e\}$ and output of hash function to be 128-bit long. In the proposed scheme, the parameters $\{B, C, E, R, V\}$ are required to be stored inside the mobile device, accordingly the required memory space is 5*128 = 640 bits. Further messages transmitted during login-authentication phase are $(\{RCID, C_1, C_2, C_3\}, \{D_1, S_1\}, \{S_1\})$, so that the communication cost comes out to be 7*128 = 896 bits. It is clear from Table 1 that when compared to Truong et al.'s scheme, the proposed scheme requires no extra memory space and the communication cost is also same.

7.2 Comparison regarding susceptibility to different attacks

We compile Table 2 for comparison regarding susceptibility to different attacks. Table 2 showcases two main aspects of attack analysis:

Table 1 Comparison of computational complexity, memory space required and communication cost

Schemes	Computational complexity					Memory required and communication cost	
		Phase			Total	Memory required by mobile device	Communication cost in login-authentication phase
		Registration phase	Login phase	Authentication phase			
The proposed scheme	Server	4h(.)		7h(.)	**11h(.)**	5*128 = 640 bits	7*128 = 896 bits
	Mobile device	1h(.)	4h(.)	2h(.)	**7h(.)**		
	Sum	**5h(.)**	**4h(.)**	**9h(.)**	**18h(.)**		
Truong et al.'s scheme	Server	4h(.)		5h(.)	**9h(.)**	5*128 = 640 bits	7*128 = 896 bits
	Mobile device	1h(.)	3h(.)	2h(.)	**6h(.)**		
	Sum	**5h(.)**	**3h(.)**	**7h(.)**	**15h(.)**		
Chen et al.'s scheme	Server	3h(.)		3h(.)	**6h(.)**	3*128 = 384 bits	4*128 = 896 bits
	Mobile device	1h(.)	3h(.)	2h(.)	**6h(.)**		
	Sum	**4h(.)**	**3h(.)**	**5h(.)**	**12h(.)**		

Bold values indicate the total number of one-way hash operations used during a particular phase of the proposed protocol or by a particular entity participating in the protocol

Table 2 Comparison for susceptibility to different attacks

Attacks	Schemes		
	Chen et al.'s	Truong et al.'s	Proposed
Attacks due to ID theft	Yes	No	No
Replay attack	Yes	No	No
User impersonation attack	Yes	Yes	No
Server impersonation attack	Yes	Yes	No
Attacks using login request	No	Yes	No
Mobile device loss attack	No	Yes	No
Password guessing attack	Yes	Yes	No
Attack on server's secret key	Yes	Yes	No
Malicious user attack	No	No	No
Known-key attack	No	No	No
Denial of service attack	No	No	No
Stolen verifier attack	No	No	No
Man-in-the-middle attack	No	No	No
Parallel session attack	No	No	No
Reflection attack	No	No	No

- Undoubtedly, Truong et al.'s scheme is an improvement over Chen et al.'s scheme as it withstands two more attacks—ID-theft and replay attack; but it falls weaker than Chen et al.'s scheme for attacks via intercepted login request and via lost mobile device. Thus, Truong et al.'s scheme moves forward by two steps and at the same time retreats back again by two steps, thereby strengthening on one end but weakening on the other.
- The proposed scheme is resistant to larger number of attacks as compared to the other two schemes under consideration. It mends the security pitfalls of Chen et al.'s scheme as well as Truong et al.'s scheme.

7.3 Comparison regarding admired friendly features

We compile Table 3 to compare three schemes for the number of friendly features they achieve. Table 3 shows a remarkable increase in usable features as we move horizontally from Chen et al.'s scheme to the proposed scheme through Truong et al.'s scheme. In addition to user anonymity, our scheme also provides user un-traceability without which the anonymity of user may be at risk as in Truong et al.'s scheme. If login requests or messages pertaining to a particular user can be traced and further it is possible to relate them with the corresponding mobile device then it becomes quite easy to trace the legal user. As a result, scheme becomes vulnerable to the attacks, which were mended by avoiding the static identity. Furthermore, the proposed scheme provides three-factor security at which the other two schemes are weak.

Table 3 Comparison for providing admired friendly features

Attacks	Schemes		
	Chen et al.'s	Truong et al.'s	Proposed
User anonymity	No	Yes	Yes
User un-traceability	No	No	Yes
Mutual authentication	Yes	Yes	Yes
Session key establishment	No	Yes	Yes
No verification table at S	Yes	Yes	Yes
Quick wrong password detection	Yes	Yes	Yes
Freedom to change password	Yes	Yes	Yes
Three-factor security	No	No	Yes

From the entire comparison, we observe that the minor addition of three hash functions is worth achieving added security features and admired usable features. Thus, the proposed scheme is more advanced in all ways- cost, complexity, resistance to attacks and usability.

8 Conclusion

Through this paper, we highlight the susceptibility of Chen et al.'s scheme and Truong et al.'s scheme to different attacks. We carried on Truong et al.'s spoofing attack to mount password guessing attack on Chen et al.'s scheme; and also elucidated some more of its flaws. Though the authors patched up the flaws of Chen et al.'s scheme in terms of an improved version, we showed that the improved protocol by Truong et al. is still feeble to defy impersonation attacks and password guessing attack. We explain that both the schemes fail to take advantage of employed three-factor security. Thus, we show that the improvement of Chen et al.'s scheme by Truong et al. does not successfully eradicate the loopholes of the original scheme; and both the schemes are insecure with the arguments what authors had considered. To dispose of the vulnerabilities of both the schemes, we further proposed an improved scheme. We profoundly analyzed the security of the proposed protocol to ensure its resistance to various attacks including those to which its original versions are open. We have also conducted a comparative study of the three schemes to analyze the strength of our proposed scheme over Chen et al.'s scheme and Truong et al.'s scheme. Through comparison, we have shown that the proposed scheme is more robust and yet maintains the simplicity of design regarding cost and complexity.

References

1. Lamport L (1981) Password authentication with insecure communication. Commun ACM 24:770–772
2. Horng G (1995) Password authentication without using password table. Inf Process Lett 55:247–250
3. Jan JK, Chen YY (1998) Paramita wisdom' password authentication scheme without verification tables. J Syst Softw 42:45–57

4. Haller NM (1995) The S/KEY one-time password, system, RFC1760
5. Mitchell CJ, Chen l (1996) Comments on the S/KEY user authentication scheme. ACMOSR 30:12–16
6. Shimizu A (1990) A dynamic password authentication method by one-way function. IEICE Trans Inf Syst 73–D–I:630–636
7. Hwang MS, Li LH (2000) A new remote user authentication scheme using smart cards. IEEE Trans Consum Electron 46(1):28–30
8. Sun HM (2000) An efficient remote user authentication scheme using smart cards. IEEE Trans Consum Electron 46(4):958–961
9. Chein HY, Jan JK, Tseng YM (2002) An efficient and practical solution to remote authentication: smart card. Comput Secur 21(4):372–375
10. Hsu CL (2004) Security of Chein et al.'s remote user authentication scheme using smart cards. Comput Stand Interfaces 26(3):167–169
11. Ku WC, Chen SM (2004) Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards. IEEE Trans Consum Electron 50(1):204–207
12. Liao IE, Lee CC, Hwang MS (2006) A password authentication scheme over insecure networks. J Comput Syst Sci 72(4):727–740
13. Xiang T, Wong KW, Liao X (2008) Cryptanalysis of a password authentication scheme over insecure networks. J Comput Syst Sci 74(5):657–661
14. Wang XM, Zhang WF, Zhang JS, Khan MK (2007) Cryptanalysis and improvement on two efficient remote user authentication scheme using smart cards. Comput Stand Interfaces 29(5):507–512
15. Khan MK, Kim SK, Alghathbar K (2010) Cryptanalysis and security enhancement of a 'more efficient & secure dynamic ID-based remote user authentication scheme. Comput Commun 34(3):305–309
16. Khan MK, Zhang J, Wang X (2008) Chaotic hash based fingerprint biometric remote user authentication scheme on mobile devices. Chaos, Solitons & Fractals 35(3):519–524
17. Chen CL, Lee CC, Hsu CY (2012) Mobile device integration of a fingerprint biometric remote authentication scheme. Int J Commun Syst 25:585–597. doi:10.1002/dac.1277
18. Lee JK, Ryu SR, Yoo KY (2002) Fingerprint based remote user authentication scheme using smart cards. Electron Lett 38:554–555
19. Lin CH, Lai YY (2004) A flexible biometrics remote user authentication scheme. Comput Stand Interfaces 27(1):19–23
20. Khan MK, Zhang J (2007) Improving the security of 'a flexible biometrics remote user authentication scheme'. Comput Stand Interfaces 29:82–85
21. Yuan J, Jiang C, Jiang Z (2010) A biometric-based user authentication for wireless sensor networks. Wuhan Univ J Nat Sci 15:272–276. doi:10.1007/s11859-010-0318-2
22. Saru K, Gupta MK, Kumar M (2012) Cryptanalysis and security enhancement of Chen et al.'s remote user authentication scheme using smart card. Cent Eur J Comput Sci 2(1):60–75
23. Xu J, Zhu WT, Feng DG (2008) Improvement of a fingerprint-based remote user authentication scheme. Int J Secur Appl 2(3):73–80
24. An Y (2012) Security weaknesses of a biometric-based remote user authentication scheme using smart cards. Int J Biosci Biotechnol 4(3):21–28
25. Wang D, Li J (2011) A novel mutual authentication scheme based on fingerprint biometric and nonce using smart cards. Int J Secur Appl 5(4):1–12
26. Truong TT, Tran MT, Duong AD (2012) Robust mobile device integration of a fingerprint biometric remote authentication scheme. In: Proceedings of 26th IEEE International Conference on Advanced Information Networking and Applications, pp 678–685
27. Khan MK, Kumari S, Gupta MK (2012) Further cryptanalysis of 'a remote authentication scheme using mobile device'. In: Fourth International Conference on Computational Aspects of Social Networks (CASoN), pp 234–237
28. Rhee HS, Kwon JO, Lee DH (2009) A remote user authentication scheme without using smart cards. Comput Stand Interfaces 31(1):6–13
29. Kocher P, Jaffe J, Jun B (1999) Differential power analysis. In: Proceedings of Advances in Cryptology, Santa Barbara, pp 388–397
30. Messerges TS, Dabbish EA, Sloan RH (2002) Examining smart-card security under the threat of power analysis attacks. IEEE Trans Comput 51(5):541–552
31. Yen SM, Joye M (2002) Checking before output may not be enough against fault-based cryptanalysis. IEEE Trans Comput 49(9):967–970

32. Kumar M, Gupta MK, Saru K (2011) Cryptanalysis of enhancements of a password authentication scheme over insecure networks. In: Proceedings of 4th International Conference on Contemporary Computing (IC3) (JIIT Noida), vol 168, pp 524–532
33. Gao ZX, Tu YQ (2008) An Improvement of a dynamic ID-based remote user authentication scheme with smart card. In: Proceedings of the 7th World Congress on Intelligent Control and Automation, pp 4562–4567
34. Sun DZ, Huai JP, Sun JZ, Li JX (2009) Cryptanalysis of a mutual authentication scheme based on nonce and smart cards. Comput Commun 32(6):1015–1017
35. Lowe G (1995) An attack on the Needham–Schroeder public key authentication protocol. Inf Process Lett 56(3):131–136
36. Lowe G (1996) Some new attacks upon security protocols. In: Proceedings of Computer Security Foundations Workshop VIII, IEEE Computer Society Press, Los Alamitos
37. Nam J, Kim S, Park S, Won D (2007) Security analysis of a nonce-based user authentication scheme using smart cards. IEICE Trans Fundam 90(1):299–302