Research Article

An Improved Biometrics-Based Remote User Authentication Scheme with User Anonymity

Muhammad Khurram Khan¹ and Saru Kumari²

¹ King Saud University, P.O. Box 92144, Riyadh 11653, Saudi Arabia
² Department of Mathematics, Agra College, Agra, Dr. B. R. A. University, Agra, Uttar Pradesh 282002, India

Correspondence should be addressed to Muhammad Khurram Khan; mkhurram@ksu.edu.sa

Received 4 August 2013; Accepted 2 September 2013

Academic Editor: Sabah Mohammed

The authors review the biometrics-based user authentication scheme proposed by An in 2012. The authors show that there exist loopholes in the scheme which are detrimental for its security. Therefore the authors propose an improved scheme eradicating the flaws of An’s scheme. Then a detailed security analysis of the proposed scheme is presented followed by its efficiency comparison. The proposed scheme not only withstands security problems found in An’s scheme but also provides some extra features with mere addition of only two hash operations. The proposed scheme allows user to freely change his password and also provides user anonymity with untraceability.

1. Introduction

In the last two decades, digital authentication has originated as a preferred method to authenticate remote users over insecure networks. After the first proposal of user authentication scheme by Lamport [1], considerable amount of research has been conducted in this field of which schemes [1–25] are few examples. In due course of time user authentication schemes underwent many changes. Initial schemes were based only on password [1–4], then schemes were based on smart card and password [5–13], and reliability of biometrics authentication over traditional password-based authentication gave rise to biometrics-based user authentication schemes [14–20].

In 2010, Li and Hwang [19] proposed a biometrics-based user authentication scheme. In 2011, Das [26] examined Li-Hwang’s scheme and observed problems in login and authentication phase, in password change phase, and in biometrics verification mechanism of the scheme. Das depicted that user’s smart card does not validate the inputted password during login phase which leads to useless computations in login and authentication phase. Owing to the same reason, Das further showed that the scheme suffers from incorrect password updating problem. Thus, Das proposed an improvement [26] of Li-Hwang’s scheme and claimed their scheme to be free from problems observed in Li-Hwang’s scheme. According to Das, their scheme [26] also provides mutual authentication. In 2012, An [27] pointed out that Daś’s scheme [26] deviates from the author’s claim since an adversary can mount impersonation attacks and password guessing attack once he gets a chance to extract values from the smart card of the legal user. Thereby An [27] proposed an enhanced scheme to eradicate the flaws of Daś’s scheme.

In this paper, we review An’s biometrics-based user authentication scheme. We show that An’s scheme is vulnerable to the security problems to which Daś’s scheme is susceptible like online and offline password guessing attacks, user and server impersonation attacks, lack of mutual authentication, and lack of user anonymity. Besides, An’s scheme lacks password change facility which is an important part of password-based user authentication schemes. We remove drawbacks from An’s scheme by means of proposing an improved user authentication scheme. In addition, to resist various security threats, the proposed scheme incorporates features of password changing and user anonymity. The rest of this paper is arranged as follows. In Section 2, we review An’s user authentication scheme. Section 3 is about cryptanalysis of An’s scheme. In Section 4, we present our improved scheme. Section 5 is about security analysis of the improved
scheme. In Section 6, we compare the improved scheme with related schemes. Finally, the conclusion is presented in Section 7.

2. Review of An’s Scheme

The notations useful in this paper are summarized along with their description in Table 1. In this section, we review An’s scheme [27] which is an enhanced version of Das’s scheme [26]. It has three phases: registration phase, login phase and authentication phase. Registration phase is carried over a secure channel whereas login phase, and authentication phase are carried over an insecure channel. There are three participants in the scheme, the user (Ci), the server (Si), and the registration centre (R), where R is assumed to be a trusted party. Details of each phase are given in the following subsections.

2.1. Registration Phase. In the beginning of scheme, the registration centre R and the user Ci carry out this phase involving the following steps.

(1) Ci submits his identity IDi and information (PWi ⊕ Ki) containing password to R via a secure channel. Ci also submits information (Bi ⊕ Ki) containing his biometrics via the specific device to R; here Ki is a random number chosen by Ci.

(2) R computes fi = h(Bi ⊕ Ki) and ri = h(PWi ⊕ Ki) ⊕ fi, and ei = h(IDi || xi) ⊕ ri, where xi is a secret key generated and maintained by Si. Then R stores {IDi, fi, ei, h(·)} in a smart card SCi for user and provides it to Ci via a secure channel.

(3) On receiving SCi = {IDi, fi, ei, h(·)}, the user stores the random number Ki into SCi issued by R so that now SCi = {IDi, fi, ei, h(·)}.

2.2. Login Phase. When the user Ci wishes to login the server Si, the user and his smart card SCi perform the following steps.

(1) Ci inserts his smart card into a card reader and inputs his biometrics information Bi on the specific device. SCi computes h(Bi ⊕ Ki) and verifies if fi = h(Bi ⊕ Ki) or not. If this biometrics information matches, Ci passes the biometrics verification.

(2) Ci inputs his IDi and PWi; then SCi generates a random number Re and computes the following equations:

\[ r_i' = h(PW_i ⊕ K_i) ⊕ f_i, \]
\[ M_1 = e_i ⊕ r_i', \]
\[ M_2 = M_1 ⊕ r_e, \]
\[ M_3 = h(M_1 || R_e). \]  (1)

(3) Ci sends the login request = {IDi, M2, M3} to Si.

2.3. Authentication Phase. On receiving the request login = {IDi, M2, M3} from Ci, the server Si and the user Ci perform the following steps to authenticate each other.

(1) Si first checks the format of IDi. If IDi is valid, Si computes M4 = h(IDi || x_i) and M5 = M2 ⊕ M4.

(2) Si checks if M3 = h(M4 || M5) or not. If both are equal, it generates a random number Re and computes the following equations:

\[ M_6 = M_4 ⊕ R_e, \]
\[ M_7 = h(M_4 || R_e). \]  (2)

Then, Si sends the reply message = {M6, M7} for its authentication to Ci.

(3) On receiving {M6, M7} from Si, the user Ci computes M8 = M6 ⊕ M1 and checks if M7 = h(M1 || M5) or not. If both are equal, Ci computes M9 = h(M1 || R_e || M4) and sends the reply message {M9} for its authentication to Si.

(4) On receiving {M9} from Ci, the server checks if M9 = h(M4 || M5 || R_e) or not. If both are equal, Si accepts the login request = {IDi, M2, M3} of Ci.

3. Cryptanalysis of An’s Scheme

This section is about security problems in An’s scheme. Here we show that an attacker Ua can mount different types of attacks on the scheme. Independent researches by Kocher and Messerges [28, 29] show that it is possible to extract different types of values stored inside a smart card. So we assume that Ua can extract out parameters stored inside a user’s smart card.

Table 1: Notations with their description.

<table>
<thead>
<tr>
<th>Notations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>Trusted registration centre</td>
</tr>
<tr>
<td>Si</td>
<td>Server</td>
</tr>
<tr>
<td>Ci</td>
<td>User</td>
</tr>
<tr>
<td>IDi</td>
<td>Identity of Ci</td>
</tr>
<tr>
<td>PWi</td>
<td>Password of Ci</td>
</tr>
<tr>
<td>Bi</td>
<td>Biometric template of Ci</td>
</tr>
<tr>
<td>SCi</td>
<td>Smart card of Ci</td>
</tr>
<tr>
<td>Ki</td>
<td>Random number generated by Ci</td>
</tr>
<tr>
<td>Re</td>
<td>Random number generated by SCi of Ci</td>
</tr>
<tr>
<td>Ri</td>
<td>Random number generated by Si</td>
</tr>
<tr>
<td>Ua</td>
<td>Attacker</td>
</tr>
<tr>
<td>x, yi</td>
<td>Secret keys maintained by Si</td>
</tr>
<tr>
<td>h(·)</td>
<td>One-way hash function</td>
</tr>
<tr>
<td>⊕</td>
<td>Bitwise XOR operator</td>
</tr>
<tr>
<td>‖</td>
<td>Concatenation operator</td>
</tr>
</tbody>
</table>
3.1. Online Password Guessing Attack. If $U_a$ obtains the smart card $SC_i$ of user $C_i$ and extracts $\{ID_i, f_i, e_i, K_i, h(\cdot)\}$ stored inside it, then he can mount online password guessing attack as explained below.

(1) $U_a$ computes

$$e_i \oplus f_i = [h(ID_i \parallel x_i) \oplus r_i] \oplus f_i$$
$$= [h(ID_i \parallel x_i) \oplus h(PW_i \oplus K_i) \oplus f_i] \oplus f_i$$
$$= [h(ID_i \parallel x_i) \oplus h(PW_i \oplus K_i)]$$

(3) to obtain $[h(ID_i \parallel x_i) \oplus h(PW_i \oplus K_i)]$.

(2) $U_a$ guesses $PW_i$ as user’s possible password and computes $M_{1a} = [e_i \oplus f_i] \oplus h(PW_i \oplus K_i)$. Then $U_a$ computes $M_{2a} = M_{1a} \oplus R_{ca}$ and $M_{3a} = h(M_{1a} \parallel R_{ca})$, where $R_{ca}$ is the random number generated by the system of $U_a$. He sends $\{ID_i, M_{2a}, M_{3a}\}$ as login request to $S_i$.

(3) If $U_a$ does not receive any response from $S_i$ then he repeats step (2) with some other guess for user’s password. But if $U_a$ receives response message from $S_i$, then it implies that his guessed password $PW_i$ is correct.

3.2. Offline Password Guessing Attack. In the scheme, $U_a$ can easily identify the login request corresponding to a smart card since both contain the identity of user. If $U_a$ extracts $\{ID_i, f_i, e_i, K_i, h(\cdot)\}$ from the smart card $SC_i$ of user $C_i$ and intercepts the login request $= \{ID_i, M_2, M_3\}$ from open network, then he can mount offline password guessing attack as explained below.

(1) $U_a$ computes

$$e_i \oplus f_i = [h(ID_i \parallel x_i) \oplus r_i] \oplus f_i$$
$$= [h(ID_i \parallel x_i) \oplus h(PW_i \oplus K_i) \oplus f_i] \oplus f_i$$
$$= [h(ID_i \parallel x_i) \oplus h(PW_i \oplus K_i)]$$

(4) to obtain $[h(ID_i \parallel x_i) \oplus h(PW_i \oplus K_i)]$.

(2) $U_a$ guesses $PW_i$ as user’s possible password and computes $M_{1a} = [e_i \oplus f_i] \oplus h(PW_i \oplus K_i)$.

(3) $U_a$ computes $R_{ca} = M_2 \oplus M_{1a}$ and $M_{3a} = h(M_{1a} \parallel R_{ca})$, and finally compares $M_{3a}$ with $M_3$. For $M_{3a} \neq M_3$, he repeats from step (2) with some other guess for user’s password. But if $M_{3a} = M_3$, then it provides $U_a$ with the exact password $PW_i$ of $C_i$.

3.3. User Impersonation Attack. As just discussed in previous subsections, $U_a$ can guess a user’s password if he obtains the smart card of user. It is noticeable that the successful process of password guessing (online or offline manner) also yields $M_{1a} = h(ID_i \parallel x_i)$. In fact, $h(ID_i \parallel x_i)$ is the key value required to compute a valid login request or valid reply messages. Further, $U_a$ has easy access to user’s identity $ID_i$ from $SC_i = \{ID_i, f_i, e_i, K_i, h(\cdot)\}$ or from the login request $= \{ID_i, M_2, M_3\}$ of $C_i$. Having $h(ID_i \parallel x_i)$ and $ID_i$ in hand, $U_a$ can impersonate the user $C_i$ as explained below.

(1) $U_a$ generates a random number $R_{ca}$ in his system and computes

$$M_{2a} = M_{1a} \oplus R_{ca},$$
$$M_{3a} = h(M_{1a} \parallel R_{ca}).$$

Then $U_a$ sends the login request $= \{ID_i, M_{2a}, M_{3a}\}$ to $S_i$.

(2) On receiving $\{ID_i, M_{2a}, M_{3a}\}$, the server $S_i$ first checks the format of $ID_i$. Clearly, $S_i$ would proceed further because $ID_i$ is the identity of a legitimate registered user and hence it is in valid format.

(3) $S_i$ computes $M_4 = h(ID_i \parallel x_i)$ and $M_5 = M_{2a} \oplus M_4$ and checks if $M_{3a} = h(M_{1a} \parallel M_3)$; clearly it would hold. Therefore $S_i$ believes that the login request $= \{ID_i, M_{2a}, M_{3a}\}$ is from the legitimate user.

(4) $S_i$ generates a random number $R_s$ and computes $M_s = M_4 \oplus R_s$ and $M_6 = h(M_4 \parallel R_s)$. Then $S_i$ transmits the reply message $\{M_6, M_7\}$.

(5) On receiving $\{M_6, M_7\}$ from $S_i$, the attacker $U_a$ first obtains the random number $R_s$ by computing $M_{8a} = M_6 \oplus M_{1a}$. Next, it computes $M_{9a} = h(M_{1a} \| R_{ca} \| M_8)$ and sends $\{M_{9a}\}$ to $S_i$.

(6) On receiving $\{M_9\}$, the server $S_i$ checks if $M_9 = h(M_4 \| M_5 \| R_s)$ or not. Clearly, this would hold, so $S_i$ will accept the login request $= \{ID_i, M_{2a}, M_{3a}\}$.

3.4. Server Impersonation Attack. $U_a$ can easily impersonate the legal server $S_i$ to cheat the user $C_i$ whose information $\{ID_i$ and $M_{1a} = h(ID_i \parallel x_i)\}$ he possesses as described in Section 3.3. To masquerade as $S_i$, the attacker proceeds in the following manner.

(1) $U_a$ can easily recognize the login request $= \{ID_i, M_2, M_3\}$ of $C_i$ transmitted over open channel as he possesses the identity $ID_i$ of $C_i$. So when $C_i$ sends his login request $= \{ID_i, M_2, M_3\}$ to $S_i$, the attacker $U_a$ intercepts and blocks it from reaching $S_i$.

(2) $U_a$ first obtains the random number $R_s$ by computing $M_{8a} = M_2 \oplus M_{1a}$. Next, he generates a random number $R_{ca}$ in his system and computes $M_{9a} = M_{1a} \oplus R_{ca}$ and $M_{10a} = h(M_{1a} \parallel R_{ca})$. Then $U_a$ transmits the reply message $\{M_{9a}, M_{10a}\}$ to $C_i$.

(3) On receiving $\{M_{9a}, M_{10a}\}$, the user $C_i$ first obtains the random number $R_{ca}$ by computing $M_{6a} = M_{9a} \oplus M_1$, where $M_1 = h(ID_i \parallel x_i)$. Next, he checks if $M_{5a} = h(M_1 \parallel M_8)$ or not. Clearly, this equivalence will hold and hence $C_i$ will believe that he is communicating with the intended server. However, it is the clever attacker $U_a$ who is deceiving $C_i$. 

---

BioMed Research International 3
3.5. Lack of Mutual Authentication. Like Das’s scheme [26], the enhanced scheme by An also fails to resist user impersonation attack and server impersonation attack as described in Sections 3.3 and 3.4. In fact, if \( U_n \) extracts values \( \{ID_i, f_i, e_i, K_i, h(\cdot)\} \) from the smart card \( SC_i \) of user \( C_i \) and successfully obtains the secret value \( h(ID_i \parallel x_i) \), then he can easily craft valid login request and reply messages so as to deceive the legal user or the legal server. Therefore, the scheme loses mutual authentication feature.

3.6. Lack of User Anonymity. In An’s scheme, \( C_i \) sends \( \{ID_i, M_2, M_3\} \) as his login request to \( S_i \) through an insecure channel. User’s identity \( ID_i \) is openly available if an attacker \( U_n \) intercepts the login request of \( C_i \) from the open channel. Moreover, identity \( ID_i \) is also stored inside user’s smart card \( SC_i \). Having \( ID_i \) in hand, it is easy for \( U_n \) to craft threats against \( C_i \). To the worst, \( U_n \) may be able to compromise user’s biometrics information which would result in serious consequences. Thus, the scheme does not provide user anonymity.

4. The Proposed Scheme

In this section, we propose a new user authentication scheme which is an improvement of An’s scheme. In addition to resist the security problems found in An’s scheme, it also provides password change phase with which user can change his password at his will. It has four phases: registration phase, login phase, authentication phase and password change phase. Registration phase, and password change phase are carried over a secure channel whereas login phase and authentication phase are carried over an insecure channel. It also consists of three participants, the user \((C_i)\), the server \((S_i)\), and the registration centre \((R)\). In the proposed scheme, the server maintains two secret keys \( x_i \) and \( y_i \). Details of each phase along with Figure 1 are given in the following.

4.1. Registration Phase. Before starting the scheme, the registration centre \( R \) and the user \( C_i \) carry out this phase involving the following steps.

1. \( C_i \) submits his identity \( ID_i \) and information \((PW_i \oplus K_i)\) containing password to \( R \) via a secure channel. \( C_i \) also submits information \((B_i \oplus K_i)\) containing his biometrics via a specific device to \( R \); here \( K_i \) is a random number chosen by \( C_i \).
2. \( R \) computes the following values:

\[
\begin{align*}
    f_i &= h(B_i \oplus K_i), \\
    r_i &= h(PW_i \oplus K_i) \oplus f_i, \\
    c_i &= h(x_i \parallel y_i) \oplus f_i, \\
    e_i &= h(ID_i \parallel x_i) \oplus r_i,
\end{align*}
\]

where \( R \) stores \( \{c_i, e_i, h(\cdot)\} \) in a smart card \( SC_i \) for user. Then \( R \) provides \( SC_i = \{c_i, e_i, h(\cdot)\} \) and \( f_i \) to the user \( C_i \) via a secure channel.

3. On receiving \( SC_i = \{c_i, e_i, h(\cdot)\} \) & \( f_i \), the user computes the following values:

\[
\begin{align*}
    g_i &= (ID_i \parallel PW_i) \oplus f_i, \\
    j_i &= (ID_i \parallel PW_i) \oplus K_i,
\end{align*}
\]

where \( C_i \) inserts \( g_i \) and \( j_i \) into \( SC_i \) issued by \( R \) so that now \( SC_i = \{c_i, e_i, g_i, j_i, h(\cdot)\} \).

4.2. Login Phase. When the user \( C_i \) wishes to login the server \( S_i \), the user and his smart card \( SC_i \) perform the following steps.

1. \( C_i \) inserts his smart card into a card reader, keys in his identity \( ID_i \), and password \( PW_i \) and inputs his biometrics information \( B_i \) on the specific device.
2. \( SC_i \) retrieves \( f_i \leftarrow (ID_i \parallel PW_i) \oplus g_i \) and \( K_i \leftarrow (ID_i \parallel PW_i) \oplus j_i \). It then checks if \( f_i = h(B_i \oplus K_i) \) or not. If this biometrics information matches, \( C_i \) passes the biometrics verification; otherwise \( SC_i \) terminates the session. This process also verifies the correctness of inserted \( ID_i \) and \( PW_i \).
3. \( SC_i \) generates a random number \( R_c \) and computes the following equations:

\[
\begin{align*}
    r_i &= h(PW_i \oplus K_i) \oplus f_i, \\
    M_1 &= c_i \oplus f_i \quad \text{(which is indeed } h(x_i \parallel y_i), \text{)} \\
    M_2 &= e_i \oplus r_i \quad \text{(which is indeed } h(ID_i \parallel x_i), \text{)} \\
    M_3 &= M_1 \oplus R_c \quad \text{(which is indeed } h(x_i \parallel y_i) \oplus R_c, \text{)} \\
    M_4 &= (M_1 \parallel R_c) \oplus ID_i \\
    (\text{which is indeed } [h(x_i \parallel y_i) \parallel R_c] \oplus ID_i), \\
    M_5 &= h(M_2 \parallel R_c), \\
    (\text{which is indeed } h(h(ID_i \parallel x_i) \parallel R_c)).
\end{align*}
\]

4. \( C_i \) sends the login request \( = \{M_3, M_4, M_5\} \) to \( S_i \).

4.3. Authentication Phase. On receiving the request \( login = \{M_3, M_4, M_5\} \) from \( C_i \), the server \( S_i \) and the user \( C_i \) perform the following steps to authenticate each other.

1. \( S_i \) computes the following values:

\[
\begin{align*}
    M_6 &= h(x_i \parallel y_i), \\
    M_7 &= M_3 \oplus M_6 \quad \text{(which is indeed } R_c, \text{)} \\
    ID_i &= M_4 \oplus (M_6 \parallel M_7).
\end{align*}
\]

2. \( S_i \) checks the format of \( ID_i \). If \( ID_i \) is valid, \( S_i \) computes \( M_8 = h(ID_i \parallel x_i) \). It then checks if \( M_5 = h(M_8 \parallel M_7) \).
User ($C_0$) Registration centre ($R$)

**Registration phase**

- Chooses $ID_i$, $PW_i$, & $K_i$
- Computes $c_i = h(x_i || y_i) \oplus f_i$, and $e_i = h(ID_i || x_i) \oplus r_i$
- [SC$_i$] $\left\{ c_i, e_i, h() \right\}$

Computes:

- $g_i = (ID_i || PW_i) \oplus f_i$ and
- $j_i = (ID_i || PW_i) \oplus K_i$

Inserts $g_i$ & $j_i$ into SC$_i$ so that SC$_i$ = $\{ c_i, e_i, g_i, j_i, h() \}$

**User ($C_0$)**

**Login and authentication phase**

- U: inserts $ID_i$, $PW_i$ & $b_i$
- SC: $f_i \leftarrow (ID_i || PW_i) \oplus g_i$, and $K_i \leftarrow (ID_i || PW_i) \oplus j_i$
- For $f_i = h(R_i \oplus K_i)$ computes:
  - $r_i = h(PW_i \oplus K_i) \oplus f_i$
  - $M_1 = c_i \oplus f_i$, $M_2 = c_i \oplus r_i$
  - $M_3 = M_1 \oplus R_i$, $M_4 = (M_3 \oplus R_i) \oplus ID_i$
  - $M_5 = h(M_2 || R_i)$.  

Computes $M_{11} = M_5 \oplus M_2$

For $M_{10} = h(M_5 || M_{11})$

Computes $M_{12} = h(M_4 || R_i || M_{11})$  

- For $M_{12} = h(M_4 || M_2 || R_i)$
  - Accepts login request

**User ($C_0$)**

**Password change phase**

- U: inserts $ID_i$, $PW_i$ & $b_i$
- For $f_i = h(R_i \oplus K_i)$ asks for new password

Computes $(g_i)_{new} = (ID_i || (PW_i)_{new}) \oplus f_i$, $(j_i)_{new} = (ID_i || (PW_i)_{new} \oplus K_i$, and

$e_{i,new} = e_i \oplus h(PW_i \oplus K_i) \oplus h((PW_i)_{new} \oplus K_i)$.

$(g_i)_{new} \leftarrow g_i$, $(j_i)_{new} \leftarrow j_i$

**Figure 1:** The proposed scheme.
If both are equal, $S_i$ generates a random number $R_s$ and computes:

$$M_9 = M_8 \oplus R_s$$

(10)

(\text{which is indeed } h(ID_i \parallel x_i) \oplus R_s)

$$M_{10} = h(M_9 \parallel R_s)$$

(\text{which is indeed } h(h(ID_i \parallel x_i) \oplus R_s)).$$

Then, $S_i$ sends the reply message $\{M_9, M_{10}\}$ for its authentication to $C_i$.

(3) On receiving $\{M_9, M_{10}\}$ from $S_i$, the user $C_i$ computes $M_{11} = M_9 \oplus M_2$ (which is indeed $R_s$). It then checks if $M_{10} = h(M_2 \parallel R_s \parallel M_{11})$ or not. If both are equal, $C_i$ computes $M_{12} = h(M_9 \parallel R_s \parallel M_{11})$ (which is indeed $h[h(ID_i \parallel x_i) \parallel R_s \parallel R_s]$). Then $C_i$ sends the reply message $\{M_{12}\}$ for its authentication to $S_i$.

(4) On receiving $\{M_{12}\}$ from $C_i$, the server checks if $M_{12} = h(M_8 \parallel M_7 \parallel R_s)$ or not. If both are equal, $S_i$ accepts the login request $\{M_3, M_4, M_5\}$ of $C_i$.

4.4. Password Change Phase. When the user wishes to change his old password $PW_i$, he invokes this phase. Details of the steps required to update the smart card $SC_i$ with new password ($PW_i^{\text{new}}$) are as follows.

(1) $C_i$ inserts his smart card into a card reader, keys in his identity $ID_i$, and password $PW_i$ and inputs his biometrics information $B_i$ on the specific device.

(2) $SC_i$ retrieves $f_i \leftarrow (ID_i \parallel PW_i) \oplus g_i$ and $K_j \leftarrow (ID_i \parallel PW_i) \oplus j_i$. It then checks if $f_i = h(B_i \oplus K_j)$ or not. If this biometrics information matches, $C_i$ passes the biometrics verification, otherwise terminates the session. This process also verifies the correctness of inserted $ID_i$ and $PW_i$. Then $SC_i$ allows the user to enter the new password ($PW_i^{\text{new}}$).

(3) $SC_i$ computes the following equations:

$$g_i^{\text{new}} = (ID_i \parallel (PW_i^{\text{new}})) \oplus f_i,$$

$$j_i^{\text{new}} = (ID_i \parallel (PW_i^{\text{new}})) \oplus K_j,$$

$$e_i^{\text{new}} = e_i \oplus h(PW_i \oplus K_j) \oplus h((PW_i^{\text{new}} \oplus K_j)).$$

(11)

(4) $SC_i$ replaces $e_i, g_i,$ and $j_i$ with $(e_i^{\text{new}}, g_i^{\text{new}}, (j_i^{\text{new}})$, respectively.

5. Security Analysis of the Proposed Scheme

In this section, we analyze security of the proposed scheme.

We show that the scheme remains unaffected even if an attacker $U_a$ extracts [28, 29] all the values stored inside a user's smart card.

5.1. Online Password Guessing Attack. On having access to user's smart card $SC_i$, an attacker $U_a$ can extract [28, 29] all values $\{c_i, e_i, g_i, j_i, h()\}$ from it. In order to compute $e_i \oplus f_i$ and obtain $[h(ID_i \parallel x_i) \oplus h(PW_i \oplus K_j)]$, he requires $f_i$. But $U_a$ cannot obtain $f_i$ from $g_i = (ID_i \parallel PW_i) \oplus f_i$ as he does not know about user's identity $ID_i$ and password $PW_i$. The attacker $U_a$ can obtain $f_i \oplus K_j$ by performing $g_i \oplus j_i = [(ID_i \parallel PW_i) \parallel f_i] \oplus [(ID_i \parallel PW_i) \parallel K_j]$. Next, he can compute

$$e_i \oplus (f_i \oplus K_j)$$

$$= [h(ID_i \parallel x_i) \parallel r_i] \oplus (f_i \oplus K_j)$$

$$= [h(ID_i \parallel x_i) \parallel h(PW_i \oplus K_j) \parallel f_i] \oplus (f_i \oplus K_j)$$

$$= h(ID_i \parallel x_i) \oplus h(PW_i \oplus K_j) \oplus f_i.$$

(12)

But $U_a$ cannot compute forged $M_{2a} (= h(ID_i \parallel x_i)) = [e_i \oplus f_i \oplus K_j] \oplus h(PW_a \oplus K_j)$ using a guessed password $PW_a$ because it requires knowledge of $K_j$. It is troublesome for $U_a$ to obtain $K_j$ because $K_j$ is not stored in plaintext inside user's smart card but is stored securely in $j_i = (ID_i \parallel PW_i) \oplus K_j$. Further $U_a$ cannot obtain $K_j$ from $j_i$ without knowing $ID_i$ and password $PW_i$. Besides, $U_a$ cannot compute $M_{2a} (= h(x_i \parallel y_j)) = (e_i \oplus f_j)$ as he does not have access to $f_j$. Moreover, $U_a$ does not have ID$i_i$ of $C_i$ as $ID_i$ is not stored in plaintext inside user's smart card. Thus, $U_a$ cannot compute a login request $\{M_{x1}, M_{x2}, M_{x3}\}$ in a way so as to guess user's password in an online manner. Hence, the proposed scheme withstands online password guessing attack.

5.2. Offline Password Guessing Attack. Suppose $U_a$ obtains the smart card of some user. Though $U_a$ can intercept login message of any user from open channel, he cannot relate a user's smart card with its corresponding login request. This is due to the fact that, unlike An's scheme, in the proposed scheme user's identity in plaintext is neither stored inside user's smart card nor transmitted in login request. As a result, $U_a$ cannot combine values extracted from a user's smart card with values of corresponding login request to guess user's password in an offline manner. If we consider the situation that $U_a$ somehow happens to get the correct combination of user's smart card and login request, we show that still $U_a$ cannot mount offline password guessing attack. To guess password of $C_i$ and then verify the guess, $U_a$ can use $M_5 = h(M_2 \parallel R_s)$ provided that he possesses the values $\{h(ID_i \parallel x_i) \parallel h(PW_i \oplus K_j) \parallel K_j, R_s\}$ in hand. As explained in Section 5.1, $U_a$ can obtain $[h(ID_i \parallel x_i) \parallel h(PW_i \oplus K_j) \parallel K_j]$ using $\{g_i, j_i, e_i\}$ extracted [28, 29] from $SC_i$, but he cannot obtain the random number $K_j$. Besides, $U_a$ cannot obtain the random number $R_s$ using $M_3 = M_1 \parallel R_s$ without having $M_1 (= h(x_i \parallel y_j))$ and $U_a$ fails to obtain $M_1 (= h(x_i \parallel y_j))$ as discussed in Section 5.1. Thus an attacker $U_a$ cannot guess user's password in an offline manner.

5.3. User Impersonation and Server Impersonation Attack. To impersonate a legal user, $U_a$ should possess $M_1 = h(x_i \parallel y_j)$ and $M_2 = h(ID_i \parallel x_i)$; otherwise he cannot compute a valid login request $\{M_{x1}, M_{x2}, M_{x3}\}$ or a valid reply message $M_{12d}$. The value $h(ID_i \parallel x_i)$ is equally important if $U_a$ wishes to masquerade as legal server. Unlike An's scheme, in the proposed scheme $U_a$ is not able to obtain $M_3 (= M_8) = h(ID_i \parallel x_i)$ while making attempts of guessing user's
password. This is due to the fact that password guessing is not feasible as explained in Sections 5.1 and 5.2. Moreover, \( U_a \) cannot obtain \( M_1 = h(x_i || y_i) \) (i) from \( M_3 = M_1 \oplus R_c \) obtained by intercepting the login request of \( C_i \) because of not having random number \( R_c \), and (ii) from \( c_i = h(x_i || y_i) \oplus f_i \) extracted from user's smart card without knowing \( f_i \). Thus, the proposed scheme resists impersonation attacks.

5.4. Supporting Mutual Authentication. The success of mutual authentication in the proposed scheme follows directly from resistance against user impersonation attack and server impersonation attack as described in Section 5.3. In fact, \( U_a \) has many hurdles before him to act as a legal user or a legal server: (i) the secret keys \( x_i \) and \( y_i \) maintained by the server are unknown for \( U_a \) and (ii) \( U_a \) has no access to the identity ID\(_i\) of user \( C_i \). As a result, \( U_a \) cannot compute \( h(x_i || y_i) \) and \( h(ID_i || x_i) \) required to mount impersonation attacks. Besides, \( U_a \) has no method to retrieve these values either from the parameters extracted out of user’s smart card or from the login request or using both. Therefore, the proposed scheme provides proper mutual authentication.

5.5. Providing User Anonymity and User Untraceability. In the proposed scheme, user’s plaintext identity ID\(_i\) is completely out of scene; it is neither stored in user’s smart card SC\(_i\) nor sent in any of the login-authentication messages transmitted over insecure network. If \( U_a \) extracts [28, 29] the values \( \{c_i, e_i, g_i, j_i, h(\cdot)\} \) from SC\(_i\), we explain in the following that he cannot obtain ID\(_i\) of \( C_i \). To guess ID\(_i\), from \( g_i = (ID_i \parallel PW_i) \oplus f_i \) and from \( j_i = (ID_i \parallel PW_i) \oplus K_i \), the attacker must have the knowledge of \( \{PW_i, f_i\} \) and \( \{PW_i, K_i\} \), respectively. \( U_a \) cannot guess out ID\(_i\) from \( c_i = h(ID_i \parallel x_i) \oplus r_i \) without knowing \( r_i \) and \( x_i \). If \( U_a \) intercepts a login request \( \{M_1, M_2, M_3\} \) or the reply message \( \{M_9, M_{10}/M_{12}\} \), he cannot guess out ID\(_i\) using \( \{M_5, M_{10}/M_{12}\} \) without the knowledge of \( \{x_i, R_c \oplus R_i\} \). Besides, it is not feasible for \( U_a \) to retrieve ID\(_i\) out of \( \{e_i, M_5, M_{10}/M_{12}\} \) due to one-way property of hash function. Moreover, each value \( \{M_3, M_4, M_5, M_9, M_{10}, M_{12}\} \) transmitted over insecure network is dynamic in nature by virtue of random numbers \( R_c \) and \( R_i \) which are different for each session. Thus, \( U_a \) can neither obtain user’s identity ID\(_i\) nor can he trace the legal user by means of observing and analyzing some fixed parameter in the login request or the reply messages. Hence, the scheme provides user anonymity as well as user untraceability.

5.6. Providing Password Change Facility. In An’s scheme, once user chooses his password during registration phase, it is fixed forever as user cannot change his password at his will. Probably the author might have opined that in the presence of biometrics verification procedure there is no need of password change facility. Undoubtedly, it is very difficult to forge copy or compromise biometrics, but once compromised then biometrics cannot be changed like passwords. So we opine that if password is employed in user authentication scheme then there should be the provision to facilitate the user to freely change his password. The proposed scheme provides password changing facility with which a user can freely (without interacting with server) change his old password to a new one whenever he feels to do so. Before updating stored values with the new password \( PW_{i_{new}} \), the smart card verifies the correctness of identity ID\(_i\), old password \( PW_i \) along with verifying the biometrics information \( f_i = h(B_i \oplus K_i) \). Thus the proposed scheme provides secure and easy password changing facility.

### Table 2: Comparison of security attributes.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Resist online PW(_i) guessing attack</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Resist offline PW(_i) guessing attack</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Resist user impersonation attack</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Resist server impersonation attack</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Provides mutual authentication</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Provides PW(_i) change facility</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Provides user anonymity</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Table 3: Comparison of computational load in terms of hash functions.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Registration phase</td>
<td>3 h((\cdot))</td>
<td>3 h((\cdot))</td>
<td>3 h((\cdot))</td>
<td>4 h((\cdot))</td>
</tr>
<tr>
<td>Login phase</td>
<td>2 h((\cdot))</td>
<td>2 h((\cdot))</td>
<td>3 h((\cdot))</td>
<td>3 h((\cdot))</td>
</tr>
<tr>
<td>Authentication phase</td>
<td>5 h((\cdot))</td>
<td>8 h((\cdot))</td>
<td>6 h((\cdot))</td>
<td>7 h((\cdot))</td>
</tr>
<tr>
<td>Total</td>
<td>10 h((\cdot))</td>
<td>13 h((\cdot))</td>
<td>12 h((\cdot))</td>
<td>14 h((\cdot))</td>
</tr>
</tbody>
</table>

### 6. Comparison

In this section, we examine the proposed scheme by means of comparing its efficiency with Li-Hwang’s scheme [19], Das’s scheme [26], and An’s scheme [27]. Table 2 displays comparison of security attributes and Table 3 displays comparison of computational load in terms of hash functions. Comparison in Table 2 shows that the proposed scheme resists various attacks possible on schemes [19, 26, 27] and provides additional feature of user anonymity with untraceability. Besides, it also restores password change facility which is provided by original versions [19, 26] but is missing in An’s scheme [27]. As Table 3 shows, the proposed scheme carries only two additional hash operations over its immediate
predecessor scheme [27]. The important aspect about the proposed scheme is minor increase of two hash functions in computational load to achieve higher efficiency as compared to other schemes [19, 26, 27].

7. Conclusion

This paper shows that the recently proposed biometrics-based user authentication scheme by An is susceptible to many threats. Once an attacker obtains the smart card of a legal user, he can guess user’s password and impersonate the user. Further, the attacker can also cheat the user by masquerading as the legal server. Consequently, the scheme fails to provide mutual authentication. Besides, the scheme also suffers from the restriction of static password. We have proposed a new scheme based on the design of An’s scheme so as to fix the problems identified in An’s scheme. In the proposed scheme an attacker cannot figure out the identity of user either from the smart card or by intercepting all login-authentication messages transmitted over insecure network. Analysis and comparison show improved performance of the proposed scheme.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors would like to extend their sincere appreciation to the Deanship of Scientific Research at King Saud University for its funding of this research through the Research Group Project no. RGP-VPP-288.

References


Submit your manuscripts at
http://www.hindawi.com