AN ENHANCED BIOMETRIC BASED REMOTE USER AUTHENTICATION SCHEME USING SMART CARD

Trupil Limbasiya1, Nishant Doshi1
1Department of Computer Engineering
Faculty of P.G. Studies & Research in Engineering & Technology
Marwadi Education Foundation - Rajkot, India
limbasiyatrupil@gmail.com, doshinikki2004@gmail.com

Abstract - In remote authentication scheme, a remote user can communicate with server over open networks even though the physical distance is much far. Before interaction, they require to establish common session key by authenticating each other. Recently in 2014, Kumari et al. proposed the efficient scheme for remote user authentication. However in this paper, we show that the Kumari et al.’s scheme is vulnerably susceptible to the Insider Attack, Stolen Verifier Attack, Session Key Disclosure Attack, Password Guessing Attack, Modification Attack, User Impersonation Attack, Replay Attack, Shoulder Surfing Attack and Denial of Service Attack. Afterwards, we have proposed an improved remote user authentication scheme to deal with these attacks and other attacks.

I. INTRODUCTION

A rapid growth in technology placed a tendency to deal with various resources and services through open system networks. This open network requires the authenticity between end users called remote users. To understand this, we have given the simplified diagram to represent the users, servers and adversary scenario in the typical remote user authentication scheme. Consider the typical Bank application (Figure 1) containing some set of account holders (aka users), bank manager having access to secure server and the open/public channel. In this type of scenario, the adversary is sniffing not only on the messages in the public channel but also acts as legal user or manager that are part of the system. In this bank analogy, user required one time registration phase to get the consumer smart card and afterwards do the process with bank ATM machine (called smart card reader) to make many exchanges between user and the bank server. Without loss of generality we assumes that the channel between user and smart card reader is secure and the channel between consumer smart card reader and server is insecure.

Indeed, Leslie Lamport firstly introduced the Remote User Authentication scheme in 1981 [1]. In [2-7], the authors had proposed the different schemes to deal with the remote user authentication schemes. In 2013, Chang et al. [8] proposed dynamic identity based remote user authentication scheme in which Chang et al. recognized a number of issues as regards security attributes of Wang et al.’s scheme [7] and proposed newly scheme. In 2014, Kumari et al. [9] recognized attacks like User Impersonation, Server Masquerade, Insider, Password Guessing etc. in Chang et al.’s [8] scheme and proposed improved scheme to overcome weaknesses of Chang et al.’s scheme.

Figure 1. Simplified example of Smart Card based Remote User Authentication Scheme
Figure 2. Overview of Remote User Authentication Scheme
For the brevity, we have given the Figure 2 that depicts the current state of art in remote user authentication scheme from the scheme of Lamport [1]. In Figure 2, each rectangle shows the respective scheme and the claimed security against various attacks, while the attacks which are as labelled on the arrows shows the weaknesses found in the respective scheme by other scheme.

Paper Organization: In the section 2, we have presented brief review of Kumari et al.’s scheme. In the section 3, we have described Cryptanalysis on Kumari et al.’s scheme Remoute user authentication scheme with key agreement. In the section 4, we explained our proposed scheme in details. In the section 5, we demonstrated security analysis of the proposed scheme. In the section 6, we explained regarding performance analysis. Finally, in the section 7, we conclude with future scope. References are at the end.

Our Contribution: In this paper we have shown that the scheme of [9] is vulnerable to Insider Attack, Stolen Verifier Attack, Session Key Disclosure Attack, Password Guessing Attack, Modification/Replay Attack, User Impersonation Attack, Denial of Service Attack and Shoulder Surfing Attack. Afterwards we have proposed the scheme to be secure against these and other attacks.

II. REVIEW OF KUMARI ET AL.’S SCHEME

In this section, we have discussed Kumari et al.’s scheme in details. Kumari et al.’s scheme contains four different phases viz. registration phase, login phase, authentication phase and password change phase. We have summarized the notations that were used in the Kumari et al.’s scheme and our proposed scheme. We familiarized these notations into Table I, which is as follows:

<table>
<thead>
<tr>
<th>TABLE I NOTATIONS USED IN KUMARI ET AL.’S &amp; PROPOSED SCHEME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notations</td>
</tr>
<tr>
<td>U_i</td>
</tr>
<tr>
<td>ID_i</td>
</tr>
<tr>
<td>PW_i</td>
</tr>
<tr>
<td>BU_i</td>
</tr>
<tr>
<td>b</td>
</tr>
<tr>
<td>SC_i</td>
</tr>
<tr>
<td>S_i</td>
</tr>
<tr>
<td>y_i</td>
</tr>
<tr>
<td>T_0,T_1</td>
</tr>
<tr>
<td>T_0,T_x,T_2,T_3</td>
</tr>
<tr>
<td>x,y</td>
</tr>
<tr>
<td>x_s</td>
</tr>
<tr>
<td>H(·)</td>
</tr>
<tr>
<td>h(·)</td>
</tr>
<tr>
<td>⊕</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>mod</td>
</tr>
</tbody>
</table>

The scheme of Kumari et al. [9] that is between user U_i, consumer Smart Card SC_i and Server S_i is as follows:

A. Registration Phase

Step-1: U_i selects identity ID_i, password PW_i and a random number b. SC_i computes \(RPW_i = h(b||PW_i)\) and sends \{ID_i, RPW_i\} to S_i via secure channel for registration process.

Step-2: After receiving message \{ID_i, RPW_i\} from user side, S_i selects random number \(y_i\) & it is various for different user.

Step-3: S_i calculates \(N_i = h(ID_i||x) \oplus RPW_i, Y_i = h(ID_i||y_i) \oplus y_i, D_i = h(ID_i||y_i||RPW_i)\) & \(E_i = y_i \oplus h(y_i||x)\). S_i stores \{Y_i, D_i, E_i, h(·)\} in SC_i and delivers \{SC_i, N_i\} to U_i via secure channel.

Step-4: Receiving \{ SC_i and N_i \} from server side, SC_i calculates \(A_i = (ID_i||PW_i) \oplus b\) and \(M_i = N_i \oplus b\). After calculation, SC_i inserts A_i and M_i in the Smart Card. SC_i contains \{A_i, M_i, Y_i, D_i, E_i, h(·)\}.

B. Login Phase

Step-1: U_i inserts SC_i, ID_i and PW_i.

Step-2: SC_i retrieves value of \(b = A_i \oplus (ID_i||PW_i)\) and calculates RPW_i = h(b||PW_i). SC_i also retrieves \(h(ID_i||x) = M_i \oplus RPW_i \oplus b, y_i = Y_i \oplus h(ID_i||x), D_i = h(ID_i||y_i||RPW_i)\).

Step-3: SC_i checks that D_i = D_i and D_i are equal or not. If \(D_i = D_i\) then input of ID_i, PW_i is authenticated and SC_i continues process further. SC_i retrieves \(h(y_i||x) = Y_i \oplus E_i, M_i = N_i \oplus b\). Takes current timestamp \(T_i\) and calculates \(CID_i = ID_i \oplus h(N_i||y_i||T_i), N_i' = N_i \oplus h(y_i||T_i), B_i = N_i \oplus RPW_i = h(ID_i||x), C_i = h(N_i||y_i||B_i||T_i)\) & \(F_i = y_i \oplus (h(y_i||x)|T_i)\).
Step-4: \( D_i \neq D_t \) then \( SC_i \) terminates session. If it is repeated thrice then Smart Card gets blocked and \( U_i \) has to enter PUK (Private Unblocking Key) to re-activate his \( SC_i \).

Step-5: After completion of Step-3 successfully, \( SC_i \) sends login message \( \{ CID_i, N_i', C_i, F_i, T_i \} \) to \( S_i \) via open channel.

C. Authentication Phase

Step-1: After receiving \( S_i \) acquires the current timestamp \( T_s \) and authenticates \( (T_s - T_i) \leq \Delta T \) and then checks that there is no other login request within \( (T_s - \Delta T) \) to \( (T_s + \Delta T) \). If both conditions hold then \( S_i \) continues process further. Otherwise login request will be terminated.

Step-2: \( S_i \) retrieves \( y_i = F_i \oplus h(y_i|x)||T_i \), \( N_i = h(y_i||T_i) \oplus N_i' \) and \( ID_i = C I D_i \oplus h(N_i||y_i||T_i) \). \( S_i \) calculates \( B_i^* = h(ID_i||x) \) & \( C_i^* = h(N_i||y_i||B_i^*||T_i) \). \( S_i \) checks \( C_i^* \) and \( C_i \).

Step-3: \( C_i^* \neq C_i \) then \( S_i \) rejects login request. \( C_i^* = C_i \), \( S_i \) continues process further. \( S_i \) gets the current timestamp \( T_{ss} \) and calculates \( a = h(B_i^*||y_i||T_{ss}) \). \( S_i \) sends \( \{ a, T_{ss} \} \) to \( SC_i \).

Step-4: After receiving message \( \{ a, T_{ss} \} \) from \( S_i \), \( SC_i \) checks \( T_{ss} \) is fresh then \( SC_i \) calculates \( a^* = h(B_i||y_i||T_{ss}) \). \( a^* = a \) then only \( U_i \) is linked with legal \( S_i \) else \( U_i \) may or may not be connected with authorized \( S_i \).

Step-5: \( U_i \) and \( S_i \) compute common session key separately \( S_{essk} = h(B_i||y_i||T_i||T_{ss}||h(y_i|x)) \) and \( (S_{essk})^* = h(B_i^*||y_i||T_i||T_{ss}||h(y_i|x)) \) respectively.

D. Password Change Phase

Step-1: \( U_i \) inserts his/her smart card into authorized center then also inserts \( ID_i, PW_i \) and requests to change password.

Step-2: \( SC_i \) retrieves \( b = A_i \oplus (ID_i||PW_i) \) and calculates \( RPW_i = h(b||PW_i) \). \( SC_i \) retrieves \( h(ID_i||x) = M_i \oplus RPW_i \oplus b \) and \( y_i = Y_i \oplus h(ID_i||x) \) then calculates \( D_i^* = h(ID_i||y_i||RPW_i) \).

Step-3: \( D_i^* = D_i \) then \( SC_i \) terminates session. It is repeated thrice then \( SC_i \) gets blocked automatically. Thus, \( U_i \) has necessary to enter PUK to re-activate his/her \( SC_i \).

Step-4: \( D_i^* = D_i \) then inputted \( ID_i \) and \( PW_i \) are verified. \( S_i \) continues process further. \( SC_i \) allows to \( U_i \) to enter \( (PW_i)_{new} \) two times for confirmation. If the entered passwords are not equal then \( SC_i \) asks \( U_i \) to re-enter the \( (PW_i)_{new} \) two times. If the entered passwords are equal, \( SC_i \) computes \( (RPW_i)_{new} = h(b||PW_i)_{new} \) and \( (A_i)_{new} = (ID_i||PW_i)_{new} \oplus b \). \( (M_i)_{new} = RPW_i \oplus M_i \oplus (RPW_i)_{new} \).

Step-5: \( SC_i \) stores \( (A_i)_{new}, (M_i)_{new} \) and \( (D_i)_{new} \) in place of \( A_i, M_i \) and \( D_i \) respectively into \( SC_i \) of \( U_i \).

III. CRYPYTANALYSIS OF KUMARI ET AL.’S SCHEME

We described in this section that how Kumari et al.’s scheme is vulnerable to various attacks. Without loss of generality, we considered some assumption [11] centered as in the earlier remote user authentication schemes and which are listed below:

- Adversary can trace login requests of valid users during communication channel.
- Adversary has ability to extract information which are stored in smart card of consumer \( SC_i \) of \( U_i \) easily.
- Adversary can modify login request of legal users.

We will assume that there are two legal users \( U_i \) and \( U_j \) of network system and consumer have smart card \( SC_i \) and \( SC_j \) respectively.

A. Insider Attack

The scheme is said to be vulnerable to Insider attack if the adversary is one of the system user (including server) and try to masquerade as other user. In other words, if some different user (adversary) of same network can get the information of particular user and he/she can access his/her account illegal.

To our bank analogy example, any valid account holder say \( U_i \) can masquerade as \( U_j \) by proposing Insider attack. In other words \( U_i \) can withdraw the money with only smart card of \( U_j \). \( U_i \) is internal user (and adversary) for \( U_j \) as follows:

\[ RPW_i = h(b||PW_i) \]
\[ h(ID_i||x) = N_i \oplus RPW_i \]
\[ y_i = Y_i \oplus h(ID_i||x) \]
\[ h(y_i|x) = E_i \oplus y_i \]

\( h(y_i|x) \) is shared for all users of network system. Thus, internal legitimate user can compute \( y_j \) of other legitimate user with the help of \( h(y_i|x) \). Once \( y_j \) is derived of any legitimate user (say \( U_j \)) then rest of the scheme becomes vulnerable.
B. Stolen Verifier Attack

The scheme is said to be weak to Stolen Verifier attack if important credentials are recorded by adversary and he has SC_i of U_i. So that adversary can identify ID_i and PW_i.

To our bank analogy example, any authorized bank person can apply this type of attack in bank network system. In the registration phase, ID_j, RPW_j, N_j are captured by adversary. We assume that adversary can extract credentials from SC_j, y_j is secret key for U_j is assigned by S_i. If y_j is known to adversary then ID_j, PW_j are available to adversary. Calculation procedure is as follows:

\[ h(ID_j||x) = N_j \oplus RPW_j \]
\[ y_j = Y_j \oplus h(ID_j||x) \]
\[ b = M_j \oplus N_j \]
\[ (ID_j||PW_j) = A_j \oplus b \]

Here ID_j is known to adversary. Thus, PW_i will be derived successfully from (ID_j||PW_j). U_i is internal user which can be adversary for U_j.

C. Session Key Disclosure Attack

Session Key Disclosure attack can be applied in the scheme if any essential key disclosed to any internal user of network system. To our bank analogy example, this type of attack can happen in ATM network system if any legitimate user does not have knowledge of ATM card procedure.

If h(y_i||x) is successfully derived by any adversary and adversary traced previous communication channels then adversary can derive session key by using stored previous sessions and h(y||x). Adversary may or may not be internal user. Computation process is as follows:

\[ y_j = E_i \oplus h(y||x) \]
\[ h(ID_j||x) = Y_j \oplus y_j \]
\[ B_i = h(ID_i||x) \]
\[ T_i \text{ and } T_{es} \text{ are referred from public communication channel. Then Session key can be computed. } U_i \text{ is internal user which can be adversary for } U_j. \]

\[ S_{esk} = h(B_i||y_j||T_i||T_{es}||h(y_i||x)) \]

D. Offline Password Guessing Attack

Any valid bank consumer can lost his/her smart card in general scenario. In our example, legitimate users can lost or forget their SC_i anywhere and it can be found by any person who can extract important credentials of U_i from SC_i and the open communication Channel.

Normally, Passwords are chosen similar for different accounts. Because users have to recall passwords whenever they want to access various network systems. There is no security like unique biometric in Kumari et al.’s scheme.

\[ b' = A_i \oplus (ID_i||PW_i') \]
\[ RPW_i' = h(b'||PW_i') \]
\[ D_i' = h(ID_i||y_i||RPW_i') \]

Adversary verifies computed D_i' with D_i of SC_i. If it is valid then guessed PW_i is effective else continues with new password.

E. Modification/Replay Attack

A login message \{CID_i, C_i, F_i, T_i\} is sent over public channel from U_i to S_i. Adversary tries to change values in T_i with T_i^* and resends login message to S_i. At the time receiving login message \{CID_i', C_i, F_i, T_i^*\}, S_i checks \( T_s - T_i^* \leq \Delta T \) and S_i retrieves values \( y_i = F_i \oplus (h(y_i||x)||T_i^*) \), \( N_i = N_i' \oplus h(y_i||T_i^*) \) and \( ID_i = ID_i \oplus h(N_i||y_i||T_i^*) \). And S_i sends reply message to U_i. S_i checks timestamp only. So that Kumari et al.’s scheme is vulnerable to Modification/Replay attack.

F. User Impersonation Attack

Adversary trace previous messages of U_i. ID_i, RPW_i, N_i are captured by system administrator or manager during the registration phase of U_i. Then, adversary can compute credentials of forged login request behind of any legitimate user based on previous recorded messages of U_i. CID_{al}, N_{al}, B_{al}, C_{al}, F_{al} are forged credentials which are calculated by adversary. Adversary sends \{ CID_{al}, N_{al}, C_{al}, F_{al}, T_{al} \} to S_i. S_i checks timestamp of received message. If timestamp is valid then it accepts login request of adversary and proceed further. S_i computes parameters and responses to adversary. It means that Kumari et al.’s scheme is vulnerable to User Impersonation attack.
G. Denial of Service Attack

If adversary produces fake login message \( \{CID_i, C_i, F_i, T_i^*\} \) with modification of timestamp only and transmits modified login request to \( S_i \) within valid time duration then at server side, many login requests are available for accessing system. Before completion of one login message \( \{CID_i, N'_i, C_i, F_i, T_i^*\} \) verification, if another more and more login messages are sent to \( S_i \) then \( S_i \) will be blocked to provide services to legitimate user and they are not able to access system when \( U_i \) wants. Hence, Denial of Service attack can be possible into Kumari et al.‘s scheme.

H. Stolen Verifier Attack

Authentication scheme is verified based on alphanumeric identity and password generally. An adversary can capture identity and password through straight surveillance or by the way of recording during individual’s confirmation period while inserting passwords at network system access authorized place.

In bank scenario, any authorized person of bank system can identify secret information of legal users. User inserts identity and text password at the time of either registration or login in Kumari et al.’s scheme. So that any internal person or outside person of network system can observe \( ID_i \) \& \( PW_i \) of users during login phase or registration phase or password change phase. Thus, Kumari et al.’s scheme can be insecure against Shoulder Surfing attack.

IV. PROPOSED SCHEME

Our proposed scheme consist of four mainly phases which are Registration Phase, Login Phase, Authentication Phase and Password Change Phase. Registration Phase is regarding new user related to network system who wants to become legal user of network system so that he/she can access network system later authentically. Login Phase is for existing users of network system who wants to access specific scheme. Authentication Phase is used to verify login requests of legitimate users at server side. Last phase of our proposed scheme is Password Change Phase. Password Change Phase contains two types of different password change like text password change phase and biometric password change phase in which legal users can change their either text password or biometric password. The proposed scheme is between User \( U_i \), Smart Card Reader \( SCR_i \), Smart Card of consumer \( SC_i \) and Server \( S_i \). Overview of our proposed scheme are described in details as below:

A. Registration Phase

Registration phase is one way to become a legitimate user of particular network scheme. So that registered user can access network system through \( ID_i \), \( PW_i \) and \( SC_i \); \( SC_i \) is a smart card of registered consumer which is provided by \( S_i \) to \( U_i \) in private channel. Registration phase is happened on both side \( U_i \) \& \( S_i \).

Step-1: \( U_i \) : Selects \( ID_i \), \( PW_i \) \& \( BU_i \) (Biometric identity of \( U_i \)).

Step-2: \( SCR_i \) : Computes \( RPW_i = H(PW_i||BU_i) \)

\( SCR_i \to S_i : \{ID_i, RPW_i\} \) through secure channel.

Step-3: \( S_i \) : Generates \( y_i \).

\( S_i \) : Computes \( N_i = h(ID_i||x_i) \oplus RPW_i \), \( E_i = y_i \oplus h(ID_i||x_i) \), \( D_i = h(ID_i||y_i||RPW_i), M_i = y_i^{x_i} \mod p \) and stores \( \{N_i, E_i, D_i, M_i, h(\cdot)\} \) into \( SC_i \)

\( S_i \to U_i : [SC_i] \) over secure channel.

B. Login Phase

Login phase can be considered as a starting stage of accessing any network scheme in permitted method for registered valid users. This stage is between of \( U_i \) and \( SCR_i \). It means that login phase process is occurred at user side generally.

Step-1: \( U_i \) : Inserts \( SC_i, ID_i, PW_i \) \& \( BU_i \).

Step-2: \( SCR_i \) : Computes \( RPW_i = H(PW_i||BU_i) \).

\( SCR_i \) : Retrieves \( h(ID_i||x_i) = N_i \oplus RPW_i, y_i = E_i \oplus h(ID_i||x_i) \).

\( SCR_i \) : Computes \( D_i^* = h(ID_i||RPW_i||y_i) \).

\( SCR_i \) : \( D_i^* = D_i \) then session is terminated directly.

\( SCR_i \) : \( D_i^* \neq D_i \) then computes \( UID_i = ID_i \oplus h(N_i||y_i||T_i), N'_i = N_i \oplus h(y_i||T_i), B_i = N_i \oplus RPW_i = h(ID_i||x_i), C_i = h(N_i||y_i||B_i||T_i), F_i = M_i \oplus T_i, P_i = g^{f_i} \).

Step-3: \( SC_i \to S_i : \{UID_i, N'_i, C_i, F_i, P_i, T_i\} \) via public channel.
Figure 3. Proposed Registration Phase

Selects... 
ID_i, PW_i, BU_i

Calculates... 
RPW_i = H(PW_i || BU_i)

Generates y_i
Calculates...
  N_i = h(ID_i || x_2) ⊕ RPW_i
  D_i = h((ID_i || RPW_i) || y_i)
  E_i = y_i ⊕ h(ID_i || x_2)
  M_i = y_i^{s_i} \mod p
  SC_i = (N_i, D_i, E_i, M_i, h(\cdot))

Figure 4. Proposed Login Phase

Inserts... 
SC_i, ID_i, PW_i, BU_i

Calculates... 
RPW_i = H(PW_i || BU_i)

Retrieves...
  h(ID_i || x_2) = N_i ⊕ RPW_i
  y_i = E_i ⊕ h(ID_i || x_2)

Computes...
  D_i* = h((ID_i || RPW_i) || y_i)
  D_i* ≠ D_i

Termination

D_i* ≠ D_i Computes...
  UID_i = ID_i ⊕ h(N_i || y_i || T_i)
  N_i' = N_i ⊕ h(y_i || T_i)
  B_i = N_i ⊕ RPW_i = h(ID_i || x_2)
  C_i = h(N_i || y_i || B_i || T_i)
  F_i = M_i ⊕ T_i

Generates r_i
  P_i = g^{r_i}

{UID_i, N_i', C_i, F_i, P_i, T_i}

Figure 3. Proposed Registration Phase

Figure 4. Proposed Login Phase
Figure 5. Proposed Authentication Phase

Figure 6. Proposed Text Password Change Phase
Authentication Phase

Authentication phase is a critical stage of any user by whom login request is sent over public channel because $S_i$ authenticates received login request from user side. $S_i$ computes some important credentials if and only if received login request is valid as per network system. After computation, $S_i$ sends some parameters through open channel. Authentication phase is proceed between $S_i$ and $S_C R_i$.

\begin{itemize}
  \item \textbf{Step 1:} $S_i$: Verifies $(T_2-T_1) \leq \Delta T$ and $(T_2 - \Delta T)$ to $(T_2 + \Delta T)$. If both conditions satisfies then only retrieves $M_i = F_i \oplus T_1$, $y_i = M_i^{1/x_i} \cdot N_i = N_i \oplus h(y_i||T_1)$. $D_i = UI D_i \oplus h(N_i || y_i || T_1)$ after that computes $B_i = h(ID_i || x_i)$ and $C_i = h(N_i || y_i || B_i || T_1)$.
  \item \textbf{Step 2:} $S_i$: $C_i \neq C_i$ then session is terminated automatically.
  \item \textbf{Step 3:} $S_i$: $C_i \neq C_i$ then generates $r_2$ and computes $Q_i = g^{r_2}, X = h(B_i || C_i || y_i || T_3)$.
  \item \textbf{Step 4:} $S_i \rightarrow SCR_i: \{X, Q_i, T_3\}$ through public channel.
  \item \textbf{Step 5:} $SCR_i: T_3$ is valid then calculates $X = h(B_i || C_i || y_i || T_3)$.
  \item \textbf{Step 6:} $SCR_i: X \neq X$ then session is expired directly.
  \item \textbf{Step 7:} $SCR_i: X = X$ then computes session key from user side ($S_{essk}$) = $h(B_i || C_i || y_i || N_i || Q_i || T_1 || T_3)$ and from server side ($S_{essk}$) = $h(B_i || C_i || y_i || N_i || Q_i || T_1 || T_3)$.
\end{itemize}

Password Change Phase

$U_i$ can change his/her text or biometric password in Password Change Phase if they want. Password change phase can occurred between $U_i$ and $SCR_i$ if and only if inserted credentials by $U_i$ are valid as per network system otherwise session is terminated automatically. Password change phase proceeds between $U_i$, $SCR_i$, and $S_C R_i$ as follow:

\begin{itemize}
  \item \textbf{Step-1:} $S_C R_i$: Checks $y_i = h(ID_i || T_1)$. If both conditions satisfies then only retrieves $M_i = F_i \oplus T_1$, $y_i = M_i^{1/x_i} \cdot N_i = N_i \oplus h(y_i || T_1)$ after that computes $B_i = h(ID_i || x_i)$ and $C_i = h(N_i || y_i || B_i || T_1)$.
  \item \textbf{Step-2:} $S_C R_i$: $C_i \neq C_i$ then session is terminated automatically.
  \item \textbf{Step-3:} $S_C R_i$: $C_i \neq C_i$ then generates $r_2$ and computes $Q_i = g^{r_2}, X = h(B_i || C_i || y_i || T_3)$.
  \item \textbf{Step-4:} $S_C R_i \rightarrow SCR_i: \{X, Q_i, T_3\}$ through public channel.
  \item \textbf{Step-5:} $SCR_i: T_3$ is valid then calculates $X = h(B_i || C_i || y_i || T_3)$.
  \item \textbf{Step-6:} $SCR_i: X \neq X$ then session is expired directly.
  \item \textbf{Step-7:} $SCR_i: X = X$ then computes session key from user side ($S_{essk}$) = $h(B_i || C_i || y_i || N_i || Q_i || T_1 || T_3)$ and from server side ($S_{essk}$) = $h(B_i || C_i || y_i || N_i || Q_i || T_1 || T_3)$.
\end{itemize}
phase is also like an authentication phase because \( U_i \) can change his/her \( PW_i \) or \( BU_i \). Text password change phase and biometric password change phase are two different phases in which \( U_i \) can change either \( PW_i \) or \( BU_i \) respectively.

a) Text Password Change Phase

If any legitimate users needs to change their text password in case of password is identified by others then \( U_i \) has facility to change their \( PW_i \) in Text password change phase because of security concern.

**Step-1**: \( U_i \) : Inserts \( SC_i, ID_i, PW_i, PW_i' \) and \( BU_i \).

**Step-2**: \( \text{SCR}_i \) : Computes \( RPW_i = H(PW_i||BU_i) \) & retrieves \( h(ID_i||x_i) = N_i \oplus RPW_i, y_i = E_i \oplus h(ID_i||x_i) \). Then computes \( D_i^* = h(ID_i||y_i||RPW_i) \).

\( \text{SCR}_i : D_i^* \neq D_i \) then session is expired directly.

\( \text{SCR}_i : D_i^* \neq D_i \), Calculates \( RPW_i' = H(PW_i'||BU_i') \). \((N_i)_{new} = h(ID_i||x_i) \oplus RPW_i', (D_i)_{new} = h(ID_i||RPW_i'||y_i) \).

**Step-3**: \( \text{SCR}_i \) : Updates \((N_i)_{new}\) and \((D_i)_{new}\) as a replacement for \( N_i \) and \( D_i \) respectively.

**Step-4**: \( \text{SCR}_i \rightarrow U_i \) : Updated \( SC_i \).

b) Biometric Password Change Phase

Biometric password change phase is extra feature for legal users in which \( U_i \) can change \( BU_i \) in case of damage or loss of biometric identity.

**Step-1**: \( U_i \) : Inserts \( SC_i, ID_i, PW_i, BU_i \) and \( BU_i' \).

**Step-2**: \( \text{SCR}_i \) : Computes \( RPW_i' = H(PW_i'||BU_i) \) & retrieves \( h(ID_i||x_i) = N_i \oplus RPW_i \) and \( y_i = E_i \oplus h(ID_i||x_i) \). Then computes \( D_i^* = h(ID_i||y_i||RPW_i) \).

\( \text{SCR}_i : D_i^* \neq D_i \) then session is expired directly.

\( \text{SCR}_i : D_i^* \neq D_i \), Calculates \( RPW_i' = H(PW_i'||BU_i') \). \((N_i)_{new} = h(ID_i||x_i) \oplus RPW_i', (D_i)_{new} = h(ID_i||RPW_i'||y_i) \).

**Step-3**: \( \text{SCR}_i \) : Updates \((N_i)_{new}\) and \((D_i)_{new}\) as a replacement for \( N_i \) and \( D_i \) respectively.

**Step-4**: \( \text{SCR}_i \rightarrow U_i \) : Updated \( SC_i \).

V. SECURITY ANALYSIS OF PROPOSED SCHEME

We described in this section that our proposed scheme is protected against various attacks in details. Description of it is as follows:

A. Insider Attack

If specific different user of same network want to retrieve important credentials of individual user at that time he/she has to determine parameters \( ID_i \) of individual user and \( x_i \) at a time. \( y_i \) is derived than also \( ID_i \) and \( x_i \) parameters cannot be derived. Thus, we can say that Insider attack is not possible in our proposed scheme.

B. Stolen Verifier Attack

In the registration phase of our proposed scheme, \( ID_i \) and \( RPW_i \) is sent over secure channel to \( S_i \). \( SC_i \) contains \( \{N_i, D_i, E_i, M_i, h(\cdot) \} \). We used almost unique parameter \( BU_i \) which is biometric identity of \( U_i \) which cannot be generated by anyone. Hence, Stolen Verifier attack cannot be applied on our proposed scheme successfully.

C. Session Key Disclosure Attack

If \( h(y||x) \) is successfully derived by any adversary and adversary traced previous communication channels then also adversary cannot derive session key. \( r_u \) is a random which is generated from user side and random number \( r_s \) is generated from server side. \( r_u \) and \( r_s \) are used during computation of session key. So that two random numbers and user unique key cannot be generated. Thus, our proposed scheme is secure against Session Key Disclosure attack.

D. Offline Password Guessing Attack

If important parameters can be extracted which are stored in \( SC_i \) and public communication Channel. There is no security like unique biometric. \( RPW_i \) is combination of \( PW_i \) & \( BU_i \) and Normally, Passwords are chosen similar for different accounts or common passwords. Because users have to recall passwords. Thus, Offline Password Guessing attack cannot be applied on our proposed scheme.
E. Modification/Replay Attack

A login message \( \{UID_i, N_i', C_i, F_i, P_i, T_i \} \) is sent over public channel from \( SCR_i \) to \( S_I \). Adversary can modify only timestamp and timestamp is verified with two condition at \( S_I \) for freshness of timestamp. Other parameters cannot be modified during public channel. Our proposed scheme is withstand against Modification/Replay attack.

F. User Impersonation Attack

Adversary can estimate \( PW_i \) of \( U_I \) with the help of \( SC_i \) and communication channel. \( h(y|x) \) cannot be derived from important information. Unique key of other user \( y_j \) is not derived with the support. Thus, adversary cannot create other login request and then he/she cannot impersonate to \( U_I \). We can say that our proposed scheme can withstand against User Impersonation attack.

G. Denial of Service Attack

If adversary produces fake login request \( \{UID_i', N_i', C_i, F_i, P_i, T_i \} \) with change of timestamp only and transmits modified login message to \( S_I \). At the time of receiving login request from different users, \( S_I \) verifies freshness of timestamp by two different conditions. So that if any fake login request is sent to \( S_I \) then login request is rejected in the initial stage of authentication. Thus, only valid login requests will be proceed further for authentication. So legal users can login to network system without any interruption.

H. Shoulder Surfing Attack

Our proposed scheme provides to choose biometric password \( BU_i \) to legal users which is unique mostly and users do not have to remember it. \( BU_i \) is biometric identity of \( U_I \) which cannot be captured by direct observation or session recording. Thus, we can say that our proposed scheme can withstand against Shoulder Surfing attack.

I. Server Masquerade Attack

We assume that adversary can extract important credentials from \( SC_i \) illegally and he/she recorded previous public communication between \( U_I \) and \( S_I \). After derivation of values, he/she can compute parameters of login request and sends fake login request to \( S_I \) but not by \( U_I \). If forged login request is accepted at server side then the scheme can be vulnerable to Server Masquerade attack.

We will assume that \( SC_i \) is available to adversary and he/she has information of open channel between \( S_I \) and \( U_I \). We used one way hash function and bio hash function in computation of proposed scheme parameters securely. So that adversary has hard to calculate parameters which are important in computation of login request easily. Our proposed scheme can be secure against Server Masquerade attack.

J. Smart Card Lost/Stolen Smart Card Attack

If any adversary got lost \( SC_i \) of \( U_I \) then also our proposed scheme can withstand against Smart Card Lost/Stolen Smart Card attack. \( SC_i \) contains \( \{N_i, D_i, E_i, M_i, h(\cdot)\} \). If adversary recorded public communication channel then also adversary has parameters like \( \{UID_i, N_i, C_i, F_i, P_i, T_i\} \). We use two different password like text and biometric which cannot be generated or guessed. Thus, our proposed scheme is secure against Stolen Smart Card/Stolen Smart Card lost attack. These types of attack can be possible if stored parameters into \( SC_i \) are useful into computation procedure.

K. Plain Text Attack

Plain Text attack is possible if plain texts are sent over secure communication channel between \( U_I \) & \( S_I \) and it can be available to authority manager of network system. So that important credentials are known to adversary and adversary can be internal authorized person. Plain Text attack can be possible if \( SC_i \) of legal user is available to authorized person of bank and he/she tries to access \( SC_i \) of valid user illegally in bank system.

In our proposed scheme, \( U_I \) send \( \{ID_i, RPW_i\} \) to \( S_I \) through private channel. \( RPW_i \) is computed using one way hash function and concatenation operator with the help of \( PW_i \). It cannot be possible to calculate these parameters. So that we can say that our proposed scheme can withstand against Plain Text attack.

L. Forward Secrecy Attack

Suppose adversary has server secret key and he/she can compute login request to \( S_I \) without any failure then we can say that scheme is insecure against Forward Secrecy attack.

We assumed that sever secret key \( x_o \) is available to adversary. Adversary must have to compute \( UID_i, N_i', C_i, F_i, P_i, T_i \) for valid login request. \( y_j \) is needed to compute to login request which cannot be calculated by any adversary. And other parameters like \( ID_i, N_i, y_j, r_o, x_o \) are also required for generation of session key. Adversary cannot compute these parameters easily. Thus, Forward Secrecy attack cannot be applied in our proposed scheme.
M. Temporary Information Attack

In the example of bank, any valid user $U_j$ knows password, identity of $U_i$ and $U_j$ has $SC_i$ of $U_i$ then $U_j$ has ability to apply Temporary Information attack until $U_i$ does not change password. Temporary Information attack means that any important information is leaked and that information can be useful for particular session only. It means that if credentials of session key are known to adversary then he/she can compute session key and this session key can be valid till the session is in active stage.

In our proposed scheme, session key can be calculated with help of $B_i, C_t, y_t, N_t, T_1, T_3, Q^w_i, P^i$. These credentials cannot be generated without knowledge of $ID_i, N_i, r_u, x_s$. So session key can’t be generated and there is no chance of happening Temporary Information attack.

VI. PERFORMANCE ANALYSIS

We had analyzed various schemes related to remote user authentication scheme. We discussed different attacks which can be available into different systems. We listed different attacks regarding security along with previously proposed schemes in Table II. In the table, √ means that specific scheme can be secure against particular attack and × denotes that specific scheme can be vulnerable to particular attack.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Insider</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>√</td>
</tr>
<tr>
<td>Stolen Verifier</td>
<td>×</td>
<td>√</td>
<td>√</td>
<td>×</td>
<td>√</td>
</tr>
<tr>
<td>Session Key Disclosure</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>√</td>
</tr>
<tr>
<td>Password Guessing</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>√</td>
</tr>
<tr>
<td>Modification /Replay</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>√</td>
</tr>
<tr>
<td>User Impersonation</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>√</td>
</tr>
<tr>
<td>Denial of Service</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>√</td>
</tr>
<tr>
<td>Shoulder Surfing</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>√</td>
</tr>
<tr>
<td>Stolen Smart Card / Smart Card Lost</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>√</td>
</tr>
<tr>
<td>Server Masquerade</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Plain Text</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Forward Secrecy</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Temporary Information</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

We had also discussed various operations regarding to a number of previously proposed schemes in Table 2, which are used in the particular schemes. Table III. shows different number of operations are used in R - Registration phase, L - Login phase, A - Authentication phase and P - Password change phase.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$h(\cdot)$</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>$h(\cdot)$</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>$\oplus$</td>
<td>1</td>
<td>7</td>
<td>7</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>$\perp$</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>7</td>
</tr>
</tbody>
</table>
VII. CONCLUSION

Due to the pluses of smart card like low cost, no overhead, remote usability, etc., it becomes one of the backbone for the remote user authentication schemes. However, due to open channel between user (having smart card) and server, the adversary can take the required information and launch the specified attack. Recently, in Comp. & Elect. Engg, Kumari et al. have proposed an efficient remote user authentications scheme and claimed to be secure against various attacks. However, in this paper we have shown that the scheme of Kumari et al. is vulnerable to Insider Attack, Stolen Verifier Attack, Session Key Disclosure Attack, Password Guessing Attack, Modification/Replay Attack, User Impersonation Attack, Denial of Service Attack and Shoulder Surfing Attack. Thereafter, we have proposed the newly scheme to withstand against the mentioned and other attacks.

REFERENCES