Computer Science and Information Systems 12(2):857–872 DOI: 10.2298/CSIS141029030K

On the Security Enhancement of Integrated Electronic Patient Records Information Systems

Muhammad Khurram Khan[1], Ankita Chaturvedi[2], Dheerendra Mishra[3], and Saru Kumari[4]

[1] Center of Excellence in Information Assurance
King Saud University, Riyadh, Kingdom of Saudi Arabia
mkhurram@ksu.edu.sa
[2] Department of Mathematics, Indian Institute of Technology, Kharagpur, India
ankita@maths.iitkgp.ernet.in
[3] Department of Mathematics, LNM Institute of Information Technology, Jaipur, India.
dheerendra.mishra@lnmiit.ac.in
[4] Department of Mathematics, Ch. Charan Singh University, Meerut, India
saryusiirohi@gmail.com

Abstract. Electronic patient records (EPR) information systems maintain the patients' medical information on the web servers, and remain available to the medical institutions, practitioners, and the academia. The transmission of data is being done over the public network, which increases the privacy and security risk. However, authentication mechanism tries to ensure secure and authorized communication over insecure public network. In recent years, several authentication protocols have been proposed, but most of them fail to satisfy desirable security attributes. In this paper, we discuss the failure of two authentication protocols for EPR information systems. To overcome the flows, we present improved scheme for the integrated EPR information systems. The correctness of proposed protocol is proved using BAN logic. Moreover, the protocol performs is comparable and security is efficient than the existing schemes.

Keywords: remote user authentication, smart card, password, electronic patient records information systems.

1. Introduction

The advances in network technology have connected the world, where users can access the stored information from the remote servers at any time and from anywhere. This leads to tremendous useful implications in different types of online services, such as e-commerce, e-medicine, e-voting, e-government, e-cash. These services are highly effective and useful in social, consumer, political, civil, business and administrative areas. It has great impact on every aspect of life that drives new innovations to provide convenient on demand services. User and service provider are gradually appreciating the importance and impact of network technology. Now the services can be easy access via electronic devices, such as mobile phones, computer, tablet, etc. Ubiquitous and easy access of network technology also present a scalable platform for e-medicine services. By adopting this technology, most of the medical institutes are developing electronic patient records (EPR) information systems. It is one of the most popular developments in e-medicine which is trying to replace the traditional culture of written and storing medical record of the patient. It is the most useful and important data for a doctor during consultations [24].

One of the important issue in electronic health records is the patient's privacy [3, 21]. Thus, only authorized user should allow to access the servers. Moreover, medical records is being stored and access via public channel. The health care information is being shared and exchanged between clinicians of all disciplines, across all sectors of health care, different countries and different models of health-care [26]. In addition, the integrated EPR information systems provides the patient records to the doctors, hospitals, medical institute and academia to enhance their decision. It is a tool that will impact the devolvement of doctors and nursing system [9]. Therefore, it is necessary to understand how the patient's records are being controlled and used. Otherwise, EPR information systems without security measure provides full opportunity to an attacker to capture the medical record of all time. Additionally, many entities wish to access this system in a user-friendly way. Thus, the electronic patient record (EPR) systems should ensure user-friendly and authorized access of services.

To protect the medical records, only authorized users should be allowed to access the EPR information systems [10, 12, 16]. In recent time, many smart card based authentication schemes using password have been designed [1, 7, 11, 12, 14–17, 19, 20]. A user is allowed to select his password of his choice to protect security parameters in the smart card. Recently, Wu et al. [28] presented a password based remote user authentication scheme using smart card for integrated EPR information systems. They claimed that their scheme is efficient and secure against various attacks. Although Lee et al. [13] demonstrated that Wu et al.'s scheme is vulnerable against stolen smart card attack and stolen verifier attacks. They also proposed an improved scheme and claimed that their scheme is secure and efficient for integrated EPR information systems.

In this article, we revisit Wu et al.'s scheme and find out that their scheme is vulnerable to some more attacks other than the demonstrated by Lee et al. We point out how inefficient password change phase in Wu et al.'s scheme causes denial of service attack. Then, we briefly review Lee et al.'s scheme and demonstrate its weaknesses to insider attack. Moreover, both the schemes do not protect anonymity, where privacy protection measures increase consumer faith in the system [5].

2. Review of Wu et al.'s authentication scheme

Wu et al. [28] scheme has registration, login, verification, and password change phase. The notations used in the scheme is discussed in Table 1.

2.1. Notations

Table 1. Meaning of symbols used throughout the paper

Notation	Descryption
U	The user
S	The integrated EPR information system
A	Attacker
ID	U's identity
PW	U's password
K	Secret value (master key) of S
$h(\cdot)$	A one-way hash function
$\oplus$	XOR
$\otimes$	NOR
$\vert\vert$	String concatenation operation

2.2. Registration phase

By registering to the EPR information system, a user achieves personalized smart card.

Step 1. U submits the registration request with ID and PW to S.
Step 2. S checks ID. If verification succeeds, then S computes $v = h(K \oplus ID)$.
Step 3. S selects a value N and calculates $v \cdot PW + N = H$. Then, S computes $s = h(PW \parallel K)$.
Step 4. S embeds the parameters $\{h(\cdot), N, s, PW\}$ into SC. Via secure channel, S issues SC to U.

2.3. Login phase

To start the login session, U inputs ID and PW, SC computes the message as.

Step 1 Select random number r_1 and calculate $C_1 = h(s \parallel r_1)$ and $C_2 = r_1 \cdot PW$.
Step 2. Retrieve the saved value N, then send $< N, ID, C_1, C_2) >$ to S.

2.4. Verification phase

Step 1. S verifies ID. On success of verification, accepts the user's request and computes $v = h(K \oplus ID)$, $PW = (H - N) \cdot v^{-1}$, $r_1' = PW^{-1} \cdot C_2 = PW^{-1} \cdot PW \cdot r_1$, and $s' = h(PW \parallel K)$.
Step 2. If $h(s' \parallel r_1') = C_1$, randomly selects a number r_2, and then the message pair (a, b) where $a = r_2 \oplus h(s')$, $b = h(PW \parallel r_2 \parallel r_1')$. Finally, it sends (a, b) to U.
Step 4. On receiving the message, U restores r_2' through $r_2' = a \oplus h(s)$ and verifies $b = h(PW \parallel r_2' \parallel r_1)$. If verification succeeds, U confirms that S is valid and sends c to S where $c = h(PW \parallel r_1 \parallel r_2')$.
Step 5. On receiving the message c from U, S verifies $c = h(PW \parallel r_1' \parallel r_2)$. If verification succeeds, U is authenticated. Finally, U and S can achieves the session key $sk = h(r_1' \parallel r_2) = h(r_1 \parallel r_2')$.

2.5. Password change phase

A legal user can change the password of the smart card with the help of server as follows:

Step 1. U submits the parameters (ID, PW, PW_{new}) to S through a secure channel.
Step 2. S computes $v = h(K \oplus ID)$ and selects new appropriate N^* such that $H = v \cdot PW_{new} + N^*$. Then, S computes $s = h(PW_{new} \parallel K)$, and securely sends s with N^* to U.

3. Cryptanalysis of Wu et al.'s authentication scheme

Lee et al. [13] demonstrated that Wu et al.'s [28] scheme vulnerable to stolen smart card attack and stolen verifier attacks. In this section, we present some more weakness of Wu et al.'s, which are not discussed in [13].

3.1. Insider attack

In Wu et al.'s scheme, user submits his original password to the server, which enable an malicious insider to access user other accounts protected with same password.

3.2. User anonymity

During login phase, user sends a login message to server over the public channel including ID. Thus, an attacker can identify the source of message and can track user's activities [25].

3.3. Known session-specific temporary information attack

In this scenario, compromise of short-term keys should not result the compromise of session key. However, in Wu et al.'s scheme using achieve short-term keys r_1 and r_2, then it can compute session key sk because $sk = h(r_1 \parallel r_2)$.

3.4. Unfriendly and inefficient password change phase

The user should be able to change his password independently without serve assistance [2, 18, 22]. However, user cannot change his password independently in Wu et al.'s scheme. This provides the opportunity to the attacker to change server's database as follows:

- A can acquire U's identity from the public channel as U transmits the message via public channel.
- A generates PW' and PW_{new}, then A submits $< ID, PW', PW_{new} >$ to S.
- Without verifying the correctness of PW', S computes $v = h(K \oplus ID)$ and selects new appropriate N^* such that $H = v \cdot PW_{new} + N^*$. Then, S computes $s = h(PW_{new} \parallel K)$, and sends s with N^* to A.
- When legal user computes $C_1 = h(s \parallel r_1)$ using password PW and submits message $< N, ID, C_1 >$ to S.

- On receiving the request, S computes $v = h(K \oplus ID)$ and $PW'' = (H - N)v^{-1}$ but $H = v \cdot PW + N$ is replaced by $v \cdot PW_{new} + N^*$. Therefore, computed output $r_1' = PW''^{-1} \cdot C_2 \neq r_1$, which results $h(s' \parallel r_1') \neq C_1$, and thus verification fails.

The above facts conclude that U can never establish a session with S using N and PW as an attacker can change server's data using password change mechanism.

3.5. Inefficient login phase:

Wu et al.'s scheme does not support smart card pre-authentication. Thus, mistake in login phase cannot be identified. The justification is given below:
Case 1. If U enters incorrect password (PW^*) by mistake. Then,

- Smart card chooses a random number r_1 and computes $C_1 = h(s \parallel r_1)$ and $C_2^* = r_1 \cdot PW^*$. Then, it sends $< N, ID, C_1, C_2^*) >$ to S.
- S verifies the validity of ID. On success of verification, S accepts the user's login request. It computes $v = h(K \oplus ID)$, $PW = (H - N) \cdot v^{-1}$, $r_1' = PW^{-1} \cdot C_2^* = PW^{-1} \cdot PW^* \cdot r_1$, which is not equal to r_1 as $PW \neq PW^*$.
- S computes $s' = h(PW \parallel K)$ and verifies $h(s' \parallel r_1') = C_1$. The verification fails, since $r_1' \neq r_1$, then S denies the request.

Case 2. If U enters incorrect identity (ID^*) by mistake. Then,

- Smart card chooses a random number r_1 to compute $C_1 = h(s \parallel r1)$ and $C_2 = r_1 \cdot PW$. Then, it sends $< N, ID^*, C_1, C_2) >$ to S.
- S verifies the validity of ID^*. If verification fails, then S denies the request.

Case 3. When a legitimate user U sends the message $< N, ID, C_1, C_2) >$ to S. An attacker intercepts the message and generates a random number r_A and computes $C_2^* = C_2 \oplus r_A$. Then, it sends a new message $< N, ID, C_1, C_2^*) >$ instead of $< N, ID, C_1, C_2) >$ to S. Then,

- S verifies the validity of ID. On success of verification, accepts the user's request. It computes $v = h(K \oplus ID)$, $PW = (H - N) \cdot v^{-1}$, $r_1' = PW^{-1} \cdot C_2^* = PW^{-1} \cdot (PW \cdot r_1 \oplus r_A)$, which is not equal to r_1.
- S computes $s' = h(PW \parallel K)$ and verifies $h(s' \parallel r_1') = C_1$. The verification fails, since $r_1' \neq r_1$. Then, S denies the request.

4. Review of Lee et al.'s authentication scheme

In 2013, Lee et al. [13] proposed scheme comprises four phases, namely, registration, login, verification and password change.

4.1. Registration phase

A user complete his registration as follows: following steps:

Step 1. U submits ID and PW to S via secure channel.
Step 2. S checks the validity of ID. If user is valid, S computes $v = h(K \oplus ID)$, $s_1 = h(PW \parallel K), s_2 = h(h(PW \parallel s_1))$ and $N = v \oplus s_2 \oplus H$, where H is a constant secret value. S personalizes U's smart card SC by embedding $\{ID, h(.), N, s_1\}$. S sends the card to U via secure channel.

4.2. Login phase

U inserts SC and inputs ID and PW. Then, SC chooses a random number r_1, and computes $s_2 = h(h(PW \parallel s_1))$ and $C_1 = r_1 \oplus s_2$. SC sends (N, ID, C_1) to S.

4.3. Verification phase

The verification phase executes as follows:

Step 1. S verifies the validity of ID. if verification succeeds accepts the request.
Step 2. S computes $v = h(K \oplus ID)$ and $s'_2 = H \oplus N \oplus v$. It also computes $r'_1 = s'_2 \oplus C_1 = s'_2 \oplus (s_2 \oplus r_1)$, then generates a random number r_2 and computes the message pair (a, b) where $a = r_2 \oplus h(r'_1 \parallel s'_2), b = h(s'_2 \parallel r_2 \parallel r'_1)$. Finally, it sends (a, b) to U.
Step 3. Upon receiving (a, b) from S, U computes $h(r_1 \parallel s_2)$ and $r'_2 = a \oplus h(r_1 \parallel s_2)$ and verifies $b = h(s_2 \parallel r'_2 \parallel r_1)$. If verification fail, U denies the request. Otherwise, U confirms the validity of S, then computes $C_2 = h(r'_2 \parallel s_2) \oplus h(PW \parallel s_1)$ and sends C_2 to S.
Step 4. Upon receiving C_2 from U, S computes $u = h(r_2 \parallel s'_2) \oplus C_2 = h(r_2 \parallel s'_2) \oplus h(PW \parallel s_1)) \oplus h(r'_2 || s_2)$. S verifies $s'_2 = h(u)$. If verification succeeds, U is authenticated.
Step 5. U and S can generate a common session key sk by $sk = h(r'_1 \parallel r_2) = h(r_1 \parallel r'_2)$.

4.4. Password change phase

Any legal user U can change the password by using the following steps.

Step 1. U sends the parameters (ID, PW, PW_{new}) to S through a secure channel.
Step 2. S computes $v = h(K \oplus ID), s^*_1 = h(PW_{new} \parallel K)$, $s^*_2 = h(PW \parallel s^*_1)$ and $N^* = v \oplus s^*_2 \oplus H$. Then, S sends (s^*_1, N^*) to U through the secure channel. Finally, U updates his smart card with $\{ID, h(\cdot), N^*, s^*_1\}$.

5. Cryptanalysis of Lee et al.'s authentication scheme

Lee et al.'s scheme also faces some kind of attacks as we discuss for Wu et al.'s scheme.

5.1. Insider attack

In Wu et al.'s scheme, user submits his original password to the server, which enable an malicious insider to access user other accounts protected with same password.

5.2. No user anonymity

In Lee et al.'s scheme, an adversary can achieve the user's identity.

5.3. Unfriendly and inefficient password change phase

User can not change his password independently, which makes the mechanism unfriendly [23].

- A can acquire ID when U performs authentication with S, as U uses its original identity during communication with server over the public channel.
- A generates PW' and PW_{new}, then A submits $< ID, PW', PW_{new} >$ to S.
- Without verifying the correctness of PW', S computes $v = h(K \oplus ID), s_1^* = h(PW_{new} \parallel K)$, $s_2^* = h(PW' \parallel s_1^*)$ and $N^* = v \oplus s_2^* \oplus H$, i.e., $H = v \oplus s_2^* \oplus N^*$. Then, S sends (s_1^*, N^*) to A.
- When legal user compute C_1 with password PW and submits his login request $< N, ID, C_1 >$ to S.
- On receiving the request, S computes $v = h(K \oplus ID)$ and $s_2' = H \oplus N \oplus v$ but $H = v \oplus s_2^* \oplus N^*$, therefore, it gets $s_2 = v \oplus s_2^* \oplus N^* \oplus N \oplus v \neq s_2$ since neither $N = N^*$ nor $s_2 = s_2^*$ as server has changed N and s_2.
- S computes $r_1' = s_2' \oplus C_1^* = s_2' \oplus s_2 \oplus r_1 \neq r_1$ as $s_2 \neq s_2'$.
- S generate r_2 and computes $a = r_2 \oplus h(r_1' \parallel s_2'), b = h(s_2' \parallel r_2 \parallel r_1')$. Then, it sends (a, b) to U.
- U computes $h(r_1 \parallel s_2), r_2' = a^* \oplus h(r_1 \parallel s_2) = r_2 \oplus h(r_1' \parallel s_2') \oplus h(r_1 \parallel s_2)$
- U also computes $b' = h(s_2 \parallel r_2' \parallel r_1)$ and verifies $b' =? b^*$, which will fail as neither $s_2 \neq s_2'$ nor $r_2 \neq r_2'$.

The above facts conclude that A can perform denial of service attack such that U can never establish a session with S using N and PW.

5.4. Fails to achieve strong login and verification phase:

Lee et al.'s scheme fails to provide strong login phase, which is clear from the following cases:

Case 1. In their assumption, user U always enters his correct password PW and does not verify the password in login phase. However, it may not be true in general. U may also enter a wrong password. If U enters his wrong password PW^*, in that case also login and verification phase execute as follows:

- Smart card generates r_1 and computes $s_2^* = h(h(PW^* \parallel s_1))$ and $C_1^* = r_1 \oplus s_2^*$, then sends (N, ID, C_1^*) to S.
- S verifies the U's identity, if verification succeeds then computes $v = h(K \oplus ID), s_2 = H \oplus N \oplus v$.

- S computes $r_1^* = s_2 \oplus C_1^* = s_2 \oplus s_2^* \oplus r_1$ where $s_2 \neq s_2^*$ as $PW \neq PW^*$.
- S generates r_2 and compute $a^* = r_2 \oplus h(r_1^* \parallel s_2), b^* = h(s_2 \parallel r_2 \parallel r_1^*)$. Then, it sends (a^*, b^*) to U.
- U computes $h(r_1 \parallel s_2^*)$ and then $r_2^* = a^* \oplus h(r_1 \parallel s_2^*) = r_2 \oplus h(r_1^* \parallel s_2) \oplus h(r_1 \parallel s_2^*)$
- Computes $b' = h(s_2^* \parallel r_2^* \parallel r_1)$ and verifies $b' =?\ b^*$, which will fail as $s_2^* \neq s_2$ and $r_2 \neq r_2^*$

Case 2. An attacker can also impersonate the login message, which is justified as follows:

- Let the user enters the correct password PW and sends the legal login request (N, ID, C_1) to S. A intercepts the message, generates a random number r_A and computes $C_1^* = C_1 \oplus r_A = r_1 \oplus s_2 \oplus r_A = r_1^* \oplus s_2$, where $r_1^* = r_1 \oplus r_A$, since $\oplus$ operation is commutative. Then, it sends (N, ID, C_1^*) to S.
- On receiving the message, S achieves $r_1^* = s_2 \oplus C_1^* = r_1 \oplus r_A$ and computes $a^* = r_2 \oplus h(r_1^* \parallel s_2) = r_2 \oplus h(r_1 \oplus r_A \parallel s_2), b^* = h(s_2 \parallel r_2 \parallel r_1^*) = h(s_2 \parallel r_2 \parallel r_1 \oplus r_A)$. Then, It sends (a^*, b^*) to U.
- On receiving the message, U computes $h(r_1 \parallel s_2)$ and achieve r_2^* as $r_2^* = a^* \oplus h(r_1 \parallel s_2) = r_2 \oplus h(r_1^* \parallel s_2) \oplus h(r_1 \parallel s_2)$ and computes $b' = h(s_2 \parallel r_2^* \parallel r_1)$, then U verifies $b' =?b^*$. The verification fails as $r_2 \neq r_2^*$ since $r_1 = r_1' \oplus r$

In both the cases, the message authentication fails in the same steps. Therefore, a user may not identify that it was because of impersonation attack or it's his mistake of inputting wrong password.

6. Proposed authentication protocol

The proposed scheme is designed to present secure and efficient mechanism for EPR information system. The brief review of protocol is given in figure 1.

6.1. Registration phase

A new user completes his registration as:

Step 1. U chooses a random number u and computes $I_U = h(ID \parallel u)$ and $P_U = h(PW \parallel u)$, Then U submits (ID, I_U, P_U) to S via secure channel.

Step 2. On receiving the registration request, S verifies the validity of user identity ID. If ID is invalid, then S denies the request. Otherwise, it generates random values s_U and N for U, then computes $v = h(K \oplus I_U), s = h(s_U \oplus K), B_1 = v \oplus P_U, B_2 = s \oplus P_U$ and $B_3 = N \oplus P_U$, then provides the smart card to U through a secure channel where the user's smart card includes parameters $\{B_1, B_2, B_3, h(.)\}$. S also computes $H = v \oplus s \oplus N$ and stores (H, T_U) corresponding to I_U in its secure database, where T_U is the time when smart card is issued.

Step 2. On receiving the smart card, user computes $B_2 \oplus ID$, $B = ID \oplus PW \oplus u$ and $V = ID \otimes PW \otimes u$, then replace B_2 with $B_2 \oplus ID$ and stores B and V into his smart card.

6.2. Login phase

Step 1. U inputs ID and PW into smart card, then smart card computes $u = ID \oplus PW \oplus B$ and verifies $V =? ID \otimes PW \otimes u$. If verification does not hold, it stops the session. Otherwise, SC compute $I_U = h(ID \parallel u)$ and $P_U = h(PW \parallel u)$.

Step 2. The smart card also computes $v = B_1 \oplus P_U, s = B_2 \oplus P_U \oplus ID, N = B_3 \oplus P_U$, and h(v). SC generates a random value r_1 and computes $C_1 = h(v) \oplus r_1$ and $C_2 = h(v) \oplus N$, then sends $< M_1 >=< I_U, C_1, C_2, T'_U, \text{mac} >$ to S, where $\text{mac} = h(I_U||C_1||C_2||s||T'_U)$ and T'_U is the current timestamp.

6.3. Verification phase

On receiving the login request, this phase executes, where the user and server mutually authenticate each other.

Step 1. Upon receiving the message $< M_1 >$, S retrieves its database and achieves (I_U, H, T_U). Then verifies the freshness of timestamp T'_U. If T'_U is fresher than T_U. S computes $v = h(I_U \oplus K)$ and $h(v)$, then achieves $r_1 = C_1 \oplus h(v)$ and $N = C_2 \oplus h(v)$. It also computes $s = H \oplus N \oplus v$ and $\text{mac}^* = h(I_U||C_1||C_2||s||T'_U)$ and verifies $\text{mac} =? \text{mac}^*$.

Step 2. S selects a random number r_2 and computes $sk = h(I_U||r_1||r_2||s||v)$, $C_3 = h(v) \oplus r_2$, $\text{mac}_1 = h(I_U||C_3||sk||T'_U)$, and sends the message $< M_2 >=< C_3, \text{mac}_1 >$ to U. Moreover S replaces (I_U, H, T_U) with (I_U, H, T'_U).

Step 3. SC computes $r_2^* = C_3 \oplus v$ and $sk^* = h(I_U||r_1||r_2^*||s||v)$. SC also computes $\text{mac}_1^* = h(I_U||C_3||sk^*||T'_U)$ and verifies $\text{mac}_1 =? \text{mac}_1^*$. If verification succeeds, then S is authenticated by U, and U also considers sk^* as the session key. Once S is authenticated, smart card sends the session key confirmation message $< M_3 >=< \text{mac}_2 >$ to S, where $\text{mac}_2 = h(\text{mac}_1^* \oplus r_2^*)$.

Step 4. Upon receiving the message $< M_3 >$, S computes $\text{mac}_2^* = h(\text{mac}_1 \oplus r_2)$ and verifies $\text{mac}_2 =? \text{mac}_2^*$. If verification succeeds, U is authenticated by S and consider sk as the secret session key.

6.4. Password change phase

When a user wishes to change his password, he enters his login identity ID, password PW and new password PW_{new} into smart card. Then, to change the password, smart card works as follows:

Step 1. Execute the operations and achieve u by $u = B \oplus PW \oplus ID$, then verifies $V =? ID \otimes PW \otimes u$. If verification succeeds, then computes $I_U = h(ID \parallel u)$ and $P_U = h(PW \parallel u)$, and gets v, s and N as: $v = B_1 \oplus P_U, s = B_2 \oplus P_U \oplus ID, N = B_3 \oplus P_U$.

Step 2. Compute $P_U^* = h(PW_{new} \oplus u)$, then $B_1^* = v \oplus P_U^*, B_2^* = s \oplus P_U^*, B_2^* \oplus ID, B_3^* = N \oplus P_U^*, B^* = ID \oplus PW_{new} \oplus u$.

Step 3. Replace $B_1, B_2 \oplus ID, B_3$, and B by $B_1^*, B_2^* \oplus ID, B_3^*$, and B^* respectively. Moreover, it computes $V^* = ID \otimes PW_{new} \otimes u$ and replace V by V^*.

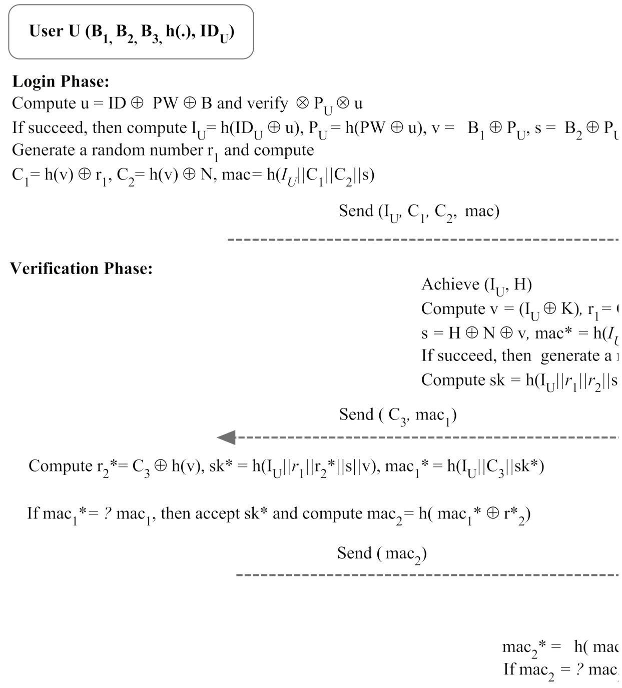
User U (B_1, B_2, B_3, h(.), ID_U)
Login Phase:
Compute u = ID ⊕ PW ⊕ B and verify ⊗ P_U ⊗ u
If succeed, then compute I_U= h(ID_U ⊕ u), P_U = h(PW ⊕ u), v = B_1 ⊕ P_U, s = B_2 ⊕ P_U
Generate a random number r_1 and compute
C_1= h(v) ⊕ r_1, C_2= h(v) ⊕ N, mac= h(I_U||C_1||C_2||s)
Send (I_U, C_1, C_2, mac)
Verification Phase:
Achieve (I_U, H)
Compute v = (I_U ⊕ K), r_1=
s = H ⊕ N ⊕ v, mac* = h(I_U
If succeed, then generate a
Compute sk = h(I_U||r_1||r_2||s
Send (C_3, mac_1)
Compute r_2*= C_3 ⊕ h(v), sk* = h(I_U||r_1||r_2*||s||v), mac_1* = h(I_U||C_3||sk*)
If mac_1*= ? mac_1, then accept sk* and compute mac_2= h(mac_1* ⊕ $r*_2$)
Send (mac_2)
mac_2* = h(mac
If mac_2 = ? mac

Fig. 1. An overview of login & verification phase

7. Analysis

7.1. Security Analysis

These assumption are used in security analysis:

- The one way hash function $h(.)$ is hard to revert.
- A specific value can not be achieved from XORed output without knowing the other.

Proposition 1. The proposed scheme protects anonymity.
Proof. In the proposed scheme, user's dynamic ID is used during communication. The user's original identity is first XORed with random value then hashed with one way hash function, which is hard to revert. This dynamic identity mechanism helps to protect user anonymity. Moreover, an attacker cannot identify with which server a user is communicating, as server identity and public keys are also not associated with the message.

Proposition 2. The proposed scheme resists stolen smart card attack.
Proof. An attacker can retrieve the information $< B_1, B_2 \oplus ID, B_3, B, V >$ from the smart card. In smart card the values v, s, N are XORed with $P_U = h(PW \parallel u)$. Therefore, to achieve v, s and N, an adversary has to achieve PW and u. However, PW and u can not be uniquely retrieve from $B = ID \oplus PW \oplus u$, as ID is secret. Moreover, no value can be retrieved uniquely from the NAND $(\times)$ output. This shows that an adversary

cannot achieve the secret values using the stolen smart card.

Proposition 3. The proposed scheme withstands password guessing attack.
Proof. In general, existing systems suffer two kinds of password guessing attacks, one is online password guessing and other is offline password guessing attacks.

An active adversary may try to execute online password by continuously tries to login to the server by guessing the possible passwords until the success. As to generates a login message v is needed, where $v = B_1 \oplus P_U$, $i.e$, online password guessing is equivalent to compute P_U using guessed password. This will not work as follows:

- Guess a password PW^*.
- Try to retrieve u from $B = ID \oplus PW \oplus u$. Since, ID is secret, only password guessing will not work.
- Try to retrieve u from $V = ID \times PW \times u$. Since, no value can be uniquely retrieved from the NAND $(\times)$ output, an adversary cannot retrieve u from V.

This shows that the proposed scheme resist online password guessing attack. An adversary may try to verify the guessed password using the off-line password guessing attack as follows:

- Guess a password PW^*.
- Try to verify guessed password with mac $= h(I_U||C_1||C_2||s||T'_U)$, for that an adversary has to compute $P^*_U = h(PW^*||u)$ with the guessed password, as $s = B_2 \oplus ID \oplus P_U$.
- Try to retrieve u and ID from $B = ID \oplus PW \oplus u$, $i.e$, compute $B \oplus PW^* = ID^* \oplus u^*$. However, to verify guessed password with $V = ID \times PW \times u$, ID and u is needed not $ID^* \oplus u^*$.

The discussion shows that an adversary cannot verify the guessed value using the password guessing attack.

Proposition 4. The proposed scheme is efficient to resist stolen verifier attacks.
Proof. In stolen verifier attack, some malicious insiders can steal user's related information from the the server's database. In the proposed scheme, the server stores the value H corresponding to I_U in its secure database. An malicious insider can steal a copy of the verifier $\{H, h(.), I_U\}$ from S's database and try to make communication vulnerable to attack between user and server. However, the stolen value H will not provide any information to the adversary, as $H = v \oplus s \oplus N$, and v, s and N are unknown to an adversary.

Proposition 5. The proposed scheme presents efficient login phase.
Proof. The proposed scheme is efficient to identify incorrect login attempt:

Case 1. On receiving wrong identity $ID*$ and right password PW.

- SC calculates $u^* = ID^* \oplus PW \oplus B \neq u$ since $ID* \neq ID$.
- SC verifies $V =? \ ID^* \otimes PW \otimes u^*$, which fails since $V = ID \otimes PW \otimes u$, $u* \neq u$ and $ID* \neq ID$.

Case 2. On receiving ID and incorrect password PW^*.

- SC computes $u'^{*} = ID \oplus PW^{*} \oplus B \neq u$, as $PW* \neq PW$.
- SC verifies $V =? ID \otimes PW^{*} \otimes u'^{*}$, which fails since $V = ID \otimes PW \otimes u$, $u'^{*} \neq u$ and $PW^{*} \neq PW$.

Case 3. On incorrect identity ID^{*} and password PW^{*}.

- Compute $u''^{*} = ID^{*} \oplus PW^{*} \oplus B \neq u$, as $PW* \neq PW$ & $ID* \neq ID$.
- Verify $V =? ID^{*} \otimes PW^{*} \otimes u''^{*}$, which fails since $V = ID \otimes PW \otimes u$, $PW^{*} \neq PW$, $ID* \neq ID$ and $u''^{*} \neq u$.

The above discussion shows that smart card can identify the incorrect input.

Proposition 6. The proposed scheme presents user-friendly and efficient password changes phase. **Proof.** In the proposed scheme, the user can change his password freely without server assistance. Moreover, the smart card verifies the correctness of inputs with the condition $V =? ID \otimes PW \otimes u$ in the similar way as demonstrated in login phase $i.e.$, efficiency of password change phase is equivalent to the efficiency of the login phase in incorrect input detection. Since, the login phase can correctly verifies the correctness of input, the password change phase is also efficient.

Proposition 7. The proposed scheme withstands replay attack.
Proof. The common countermeasures for replay attack are random number and timestamp. We adopt timestamp as a counter measure. Each session usages a fresh timestamp and each transmitted login message includes timestamp. Moreover, to modify the login message according to the new timestamp T_E, an adversary has to calculate mac $= h(I_U||C_1||C_2||s||T_E)$, which requires the knowledge of s. Since, s is protected with password and password is unknown to the adversary, the adversary can not modify previously transmitted message. This shows that the proposed scheme resists replay attack.

Proposition 8. The proposed scheme supports mutual authentication.
Proof. To ensure the correctness of the user, the server checks the condition mac $= h(I_U||C_1||C_2||s||T'_U)$. And, to verify the correctness of the server, the user checks the condition $\text{mac}_1 = h(I_U||C_3||sk||T'_U)$. To compute mac and mac_1, secret value s is needed. Since the value s is secret, a legal user and the server can only compute and verify the condition. This shows that the proposed scheme support mutual authentication.

Proposition 9. The proposed scheme supports session key verification.
Proof. User and server both verify the session key $\text{mac}_1 = h(I_U||C_3||sk^{*}||T'_U)$ and $\text{mac}_2 = h(\text{mac}_1 \oplus r_2)$. Moreover, no adversary can forge this value, as to compute $sk = h(I_U||r_1||r_2||s||v)$, secret value s and v are needed. Therefore, both user and server can correctly verify the session key.

Proposition 10. The proposed scheme ensures known key secrecy.
Proof. If an adversary achieves some past session keys then he may try to extract some information from the compromised session key to construct other session keys [8]. Although compromised session key does not provide any information, which can helpful to compute other session keys, as each session key is the hashed output of one way hash

function, which cannot be reverted. Therefore, no information can be extracted from session key. In addition, each session key involves random session keys r_1 and r_2, which are different for different sessions.

Proposition 11. The proposed scheme achieves forward security.
Proof. If the user's long term secret key v compromised. Although an adversary cannot compute the session key with the compromised session key as follows:

- Key $sk = h(I_U||r_1||r_2||s||v)$.
- The attacker can compute r_1 and r_2 from $C_1 = v \oplus r_1$ and $C_3 = v \oplus r_2$ using v.
- The attacker can achieve P_U from $B_1 = v \oplus P_U$ using v.
- The attacher can not achieve s from $s \oplus P_U \oplus ID$, as ID is secret.

Since, an adversary cannot compute the value s, the adversary cannot compute the session key $sk = h(I_U||r_1||r_2||s||v)$ as it is the hashed output of v, r_1 and r_2 along with s.

7.2. Performance Analysis

In table 2, we discussed the security of related schemes with the proposed scheme, where symbol $\times$ demonstrates that the scheme does not prevent the attack and $\surd$ demonstrate that scheme prevents the attack. It is clear from the Table 2 that proposed scheme present efficient and secure solution.

Table 2. Comparison of the schemes in different security scenarios

Security attributes\ Schemes	Wu et al. [28]	Lee et al. [13]	Proposed scheme
Preserving user anonymity	$\times$	$\times$	$\surd$
Resistance against Insider Attack	$\times$	$\times$	$\surd$
Resistance offline password guessing attack	$\times$	$\surd$	$\surd$
Resistance against stolen smart card attack	$\times$	$\surd$	$\surd$
Resistance against known session keys attack	$\surd$	$\surd$	$\surd$
Resistance against impersonation attack	$\surd$	$\times$	$\surd$
Resistance to stolen verifier attack	$\times$	$\surd$	$\surd$
Resistance against Replay attack	$\surd$	$\surd$	$\surd$
Mutual authentication	$\surd$	$\surd$	$\surd$
Establishes session key	$\surd$	$\surd$	$\surd$
Session key verification	$\times$	$\times$	$\surd$
Efficient login phase	$\times$	$\times$	$\surd$
Efficient password change phase	$\times$	$\times$	$\surd$
User-friendly password change phase	$\times$	$\times$	$\surd$

We show the efficiency analysis of proposed schemes with similar schemes based on smart card, namely, Wu et al. [28] and Lee et al. [13] in Table 3. Let, output and input size of $h(.)$, identity ID, password PW, random number 128-bits.

The values stored in smart card are $< B_1, B_2 \oplus ID, B_3, B, V >$, hence the required memory in card is $5 * 128 = 640$ bits. The communication cost is $7 * 128 = 896$ bits as from user's side, transmitted values are I_U, C_1, C_2, mac and mac_2 which capacity is $5 * 128 = 640$, and from server's side, transmitted values are C_3, mac that capacity is $2 * 128 = 256$. In Wu et al.'s scheme total 7 values $< N, ID, C_1, C_2, a, b, C >$ are transmitted. Therefore, communication overhead is $7 * 128 = 896$ bits. In Lee et al.'s scheme the communication cost is $6*128 = 768$ bits, including $< ID, N, C_1, a, b, C_2 >$.

Table 3. Comparison of computation overhead of our scheme with related schemes

Overhead \ Schemes	Wu et al. [28]	Lee et al. [13]	Proposed scheme
Memory needed in smart Card	384 bits	384 bits	640 bits
Communication cost in authentication	896 bits	768 bits	896 bits
Login Phase	1H +1M	2H	3H
Authentication Phase	10H +1M	10H	9H
Total computation cost	11H + 2M	12H	12H

M: multiplication operation; H: hash operation.

8. Conclusion

We have revisited Wu et al.'s scheme and showed lack of pre-smart card authentication and it's disadvantage. We identify the flows in Lee et al.'s scheme to present efficient login phase. We discuss "why Wu et al.'s and Lee et al.'s schemes do not resist insider attack". Further, we present an improved authentication scheme using smart card for integrated EPR information system. Our scheme could resists active and passive attacks including found in Wu et al.'s and Lee et al.'s schemes. It also reduces the computation overhead and supports smart card pre-authentication.

Acknowledgments. The authors extend their sincere appreciations to the Deanship of Scientific Research at King Saud University for its funding this Prolific Research Group (PRG-1436-16)

References

1. Chaturvedi, A., Mishra, D., Mukhopadhyay, S.: Improved biometric-based three-factor remote user authentication scheme with key agreement using smart card. In: Information Systems Security, pp. 63–77. Springer (2013)
2. Mishra, D.: Understanding security failures of two authentication and key agreement schemes for telecare medicine information systems. Journal of medical systems 39(3), 1–8 (2015)
3. Chen, C.L., Tsaur, W.J., Chen, Y.Y., Chang, Y.C.: A secure mobile drm system based on cloud architecture. Computer Science and Information Systems 11(3), 925–941 (2014)
4. Kumari, S. and Khan, M. K.: Cryptanalysis and improvement of a robust smart-card-based remote user password authentication scheme, International Journal of Communication Systems 27(12) 3939–3955 (2014).

5. Mishra, D. and Mukhopadhyay, S.: Privacy rights management in multiparty multilevel DRM system, Proceedings of the International Conference on Advances in Computing, Communications and Informatics 625–631, 2012.
6. Mishra, D. and Mukhopadhyay, S.: Cryptanalysis of Yang et al.s digital rights management authentication scheme based on smart card, Recent Trends in Computer Networks and Distributed Systems Security, 288–297 (2014).
7. Gao, T., Guo, N., Yim, K., Wang, Q.: Pps: A privacy-preserving security scheme for multi-operator wireless mesh networks with enhanced user experience. Computer Science and Information Systems 11(3), 975–999 (2014)
8. Mishra, D.: On the security flaws in id-based password authentication schemes for telecare medical information systems. Journal of medical systems 39(1), 1–16 (2015)
9. Goorman, E., Berg, M.: Modelling nursing activities: electronic patient records and their discontents. Nursing inquiry 7(1), 3–9 (2000)
10. Mishra, D., Chaturvedi, A. and Mukhopadhyay, S.: Cryptanalysis and Improvement of Jiang et al.'s Smart Card Based Remote User Authentication Scheme, arXiv preprint arXiv:1312.4793 (2013).
11. Khan, M.K., Kim, S.K., Alghathbar, K.: Cryptanalysis and security enhancement of a more efficient & secure dynamic id-based remote user authentication scheme. Computer Communications 34(3), 305–309 (2011)
12. Kumari, S., Khan, M.K.: Cryptanalysis and improvement of a robust smart-card-based remote user password authentication scheme. International Journal of Communication Systems 27(12), 3939–3955 (2014)
13. Lee, T.F., Chang, I.P., Lin, T.H., Wang, C.C.: A secure and efficient password-based user authentication scheme using smart cards for the integrated epr information system. Journal of medical systems 37(3), 1–7 (2013)
14. Mishra, D., Chaturvedi, A., Mukhopadhyay, S.: An improved biometric–based remote user authentication scheme for connected healthcare. International Journal of Ad Hoc and Ubiquitous Computing 18(1-2), 75–84 (2015)
15. Mishra, D., Kumar, V., Mukhopadhyay, S.: A pairing-free identity based authentication framework for cloud computing. In: Network and System Security, pp. 721–727. Springer (2013)
16. Mishra, D., Mukhopadhyay, S., Chaturvedi, A., Kumari, S., Khan, M.K.: Cryptanalysis and improvement of yan et al.s biometric-based authentication scheme for telecare medicine information systems. Journal of medical systems 38(6), 1–12 (2014)
17. Mishra, D., Mukhopadhyay, S., Kumari, S., Khan, M.K., Chaturvedi, A.: Security enhancement of a biometric based authentication scheme for telecare medicine information systems with nonce. Journal of medical systems 38(5), 1–11 (2014)
18. Kumari, S., Gupta, M. K., Khan, M. K. and Li, X.: An improved timestamp-based password authentication scheme: comments, cryptanalysis and improvement, Security and Communication Networks, 7(11) 1921–1932, (2014).
19. Neuman, B.C., Ts' O, T.: Kerberos: An authentication service for computer networks. Communications Magazine, IEEE 32(9), 33–38 (1994)
20. Mishra, D., Srinivas, J., Mukhopadhyay, S.: A secure and efficient chaotic map-based authenticated key agreement scheme for telecare medicine information systems. Journal of medical systems 38(10), 1–10 (2014)
21. Rigby, M., Roberts, R., Williams, J.: Objectives and prerequisites to success for integrated patient records. Computer methods and programs in biomedicine 48(1), 121–125 (1995)
22. Mishra, D.: A Study On ID-based Authentication Schemes for Telecare Medical Information System. arXiv preprint, arXiv:1311.0151 (2013).
23. Khan, M. K., and Kumari, S.: Cryptanalysis and Improvement of An Efficient and Secure Dynamic ID-based Authentication Scheme for Telecare Medical Information Systems, Security and Communication Networks, 7(2) 399–408 (2014).

24. Takeda, H., Matsumura, Y., Kuwata, S., Nakano, H., Sakamoto, N., Yamamoto, R.: Architecture for networked electronic patient record systems. International Journal of Medical Informatics 60(2), 161–167 (2000)
25. Tang, C., Wu, D.O.: Mobile privacy in wireless networks-revisited. Wireless Communications, IEEE Transactions on 7(3), 1035–1042 (2008)
26. Wang, D., Liu, D., Chen, Y.: A mechanism to verify the integrity of computer-based patient records. J China Assoc Med Inform 10, 71–84 (1999)
27. Winkelman, W.J., Leonard, K.J., Rossos, P.G.: Patient-perceived usefulness of online electronic medical records: employing grounded theory in the development of information and communication technologies for use by patients living with chronic illness. Journal of the American Medical Informatics Association 12(3), 306–314 (2005)
28. Wu, Z.Y., Chung, Y., Lai, F., Chen, T.S.: A password-based user authentication scheme for the integrated epr information system. Journal of medical systems 36(2), 631–638 (2012)

Dr. Muhammad Khurram Khan is currently working at the Center Excellence in Information Assurance, King Saud University, Saudi Arabia. He has edited seven books and proceedings published by Springer-Verlag and IEEE. He has published more than 200 papers in international journals and conferences and he is an inventor of 10 U.S. PCT patents. Dr. Khan is the Editor-in-Chief of a well-reputed journal Telecommunication Systems (Springer). He is also on the editorial boards of several International SCI journals. Dr. Khurram is one of the organizing chairs of several top-class international conferences and he is also on the program committee of dozens of conferences. He is a recipient of several national and international awards for his research contributions. His current research interests include Cybersecurity, biometrics, multimedia security, and digital authentication.

Ankita Chaturvedi has received his Ph.D. from Indian Institute of Technology Roorkee, India in 2012. Currently, he is working as the post-doctoral fellow in the Department of Mathematics, Indian Institute of Technology Kharagpur, India. His research interests include Boolean functions and cryptographic protocols.

Dheerendra Mishra has received his Ph.D. from Indian Institute of Technology Kharagpur, India in 2014. Currently, he is working as the Assistant Professor in the Department of Mathematics, the LNM Institute of Information Technology, Jaipur, India. His research interests include digital rights management systems, authentication and key agreement protocols, security and privacy.

Dr. Saru Kumari is currently an Assistant Professor with the Department of Mathematics, Ch. Charan Singh University, Meerut, Uttar Pradesh, India. She received her Ph.D. degree in Mathematics in 2012 from CCS University, Meerut, UP, India. She has published more than 42 research papers in reputed International journals and conferences. She is a reviewer of more than a dozen of reputed journals including SCI-Indexed. Her current research interests include information security, digital authentication, security of wireless sensor networks, and applied mathematics.

Received: October 29, 2014; Accepted: May 3, 2015.