J Supercomput (2018) 74:3685–3703
https://doi.org/10.1007/s11227-016-1886-5

An anonymous and provably secure biometric-based authentication scheme using chaotic maps for accessing medical drop box data

Imran Khan[1] · Shehzad Ashraf Chaudhry[1] ·
Muhammad Sher[1] · Javed I. Khan[2] ·
Muhammad Khurram Khan[3]

Published online: 6 October 2016

Abstract Telecare medicine information systems (TMISs) provides a platform to the participating medical entities to share medical data over an insecure public channel. Medical drop box (MDB) is used for the said purpose, where electronic health record (EHR) is maintained for national health information exchange (NHIX). EHR is a crucial part of MDB. Therefore, the main challenge in NHIX is to restrict MDB access to only the authenticated entities. Very Recently, Moon et al. introduced a biometrics-based authentication scheme using chaotic maps for TMISs. The authors claimed that their scheme is efficient and robust in terms of its usage and implementation. However, this paper unveils that due to storage of verifier table on server, their scheme is having scalability and efficiency issues. Furthermore, the use of the same parameters IM_1 and IM_2 during different login requests makes the scheme traceable. Therefore, an improved scheme using chaotic maps has been proposed in this paper, which pro-

✉ Shehzad Ashraf Chaudhry
shahzad@iiu.edu.pk

Imran Khan
imran.khan@iiu.edu.pk

Muhammad Sher
m.sher@iiu.edu.pk

Javed I. Khan
javed@cs.kent.edu

Muhammad Khurram Khan
mkhurram@ksu.edu.sa

[1] Department of Computer Science and Software Engineering, International Islamic University, Islamabad, Pakistan

[2] Department of Computer Science, Kent State University, Kent, OH, USA

[3] Center of Excellence in Information Assurance, King Saud University, Riyadh, Saudi Arabia

vides user anonymity and untraceability along with computational efficiency. The security of the proposed scheme is evaluated in detail through the random oracle model. The analysis reveals that the proposed scheme is robust and secure against the known attacks. Moreover, analysis is further verified through popular automated tool ProVerif.

Keywords Chaotic maps · Medical drop box · Privacy · Authentication · National health information exchange · Electronic health record · Authentication · Biometrics · Anonymity violation · ProVerif · TMIS

1 Introduction

A modern healthcare system is highly important for the well-being of any country, especially in developing countries. Unfortunately, the mainstream of the healthcare industry is working at a sub-optimal level, as the use of health information systems (HIS) even at individual hospital/institution level where it is nearly non-existent [32]. Therefore, that is why there is a great need to improve the current system with information and communication technology.

National health information exchange (NHIX) is a rapidly evolving cyber-infrastructure technology enabling sharing of healthcare-related data within a geographic region electronically among healthcare-related autonomous entities, such as physicians, hospitals, test laboratories, health insurance companies, emerging health information organizations (HIO), and even government departments. A number of NHIX are currently being in the deployment process in USA under the federal initiative as the key cyber infrastructure for healthcare [30]. The health information technology (HIT) infrastructure is also under development in the United States [23,38]. Its cornerstone is a national health information network (NHIN) initiative, which will create a national health information exchange (NHIX) system in the US [36,38]. Health information exchange (HIX) is the meeting point technology for the mobilization of healthcare information transmitted electronically across organizations within a region, community, or hospital system [16]. The US HITECH Act is funding the building of this national health IT infrastructure, in which patients' data are to fire across a national health information highway [31,34].

One of the main problems is the numerous variety and formats of health-related data. Health level 7 (HL7) is an international organization, which has taken the responsibility to develop standards for the exchange of variety of electronic health data [33]. Because, nowadays, people travels different parts of the world frequently ever before, the demand for medical data exchange between different countries is becoming much greater. Then, during a foreign travel, a person may need medical attention. In such cases, doctors in the hospitals may need to have access to the previous history of the patient for proper treatment plans. To facilitate this process, clinical data must be exchangeable electronically in some standard format [36]. In this way, the integrity and permanence of patient's health information can be maintained [27].

2 Existing work

With the rapid development in information and communication technologies, there is an ever growing demand for customized healthcare systems in medical industry. The demand for medical services and health promotions activities is also increasing with increase in population [27]. There is a need to cut down on monitory and time expanses of individuals and to form a communication bridge among patients and other entities of the system. Such system should provide seamless access to the information by any authorized entity, such as doctors provide treatment services to their patients after accessing the patient's information [27,36]. The main problem in these systems is that an unauthenticated entity must not be allowed access and also an authenticated entity must not be able to access information which is not authorized.

An authentication process is required for the medical entities to ensure that a particular user is authorized to access the data from MDB. Several authentication and key agreement schemes are proposed [2–5,13–15,17,24] for different systems. Some of the authentication schemes are based on chaos theory [11,29], because it has the better performance than the traditional cryptography approaches [11,29]. Xiao et al. [39] introduced the authentication key agreement protocol based on chaotic maps using the random numbers in 2007. After the Xiao scheme, another user anonymity preserving authentication and key agreement scheme based on chaotic maps was introduced by Tseng et al. [35]. Niu et al. [28] proves that the scheme introduced by Tseng et al. was not be able to provide the user anonymity and he introduced his improved scheme by removing the flaws in Tseng et al.'s scheme. However, Xue et al. [41] also mention in his article that the Niu et al.'s scheme can be exposed in man-in-middle attack. Guo et al. [12] introduced the password authentication key agreement scheme using smart cards also based on chaotic maps. Lin [21] both introduced their updated versions of Guo et al.'s scheme to improve the deficiencies of user anonymity. Jiang et al. [18] proved that the scheme introduced by Hao et al. did not establish the session key properly and possible launch of stolen smart card attack. They tried to improve their scheme and to remove the flaws in Hao et al.'s scheme. After that Li et al. [20] presented that the schemes introduced by Jiang et al. were not be able to stop the service misuse attack from non-registered user and their scheme provides the user information during his authentication process. Li et al. adjusted some minor modifications in Lee's scheme and introduce their enhanced scheme after addressing the limitations. In early 2015, Lu et al. [22] found some weaknesses, including: lack of local verification, vulnerability to impersonation attack, leakage of session key, etc., in Li et al.'s scheme. They further proposed an new scheme and stressed the merits of their scheme. However, Moon et al. [26] found that besides their claim, Lu et al.'s scheme cannot withstand server and user impersonation attacks. Furthermore, their scheme entails correctness issues and is not scalable. An improvement of Lu et al.'s scheme is also proposed by Moon et al. [26]. However, the scheme proposed by Moon et al. uses verification table on server side which effects the scalability and incurs the searching time during authentication. Furthermore, the use of the same parameters IM_1 and IM_2 during authentication makes the scheme traceable. This paper mainly focuses on an improved authentication scheme for the medical entities using medical drop box in NHIX.

In the remainder of this paper: Sect. 3 briefly presents some preliminaries about medical drop box, its security needs the notation guide, along with fundamentals of Chebyshev chaotic maps and hash functions. Section 4 presents a brief reviews of Moon et al.'s chaotic map-based authentication scheme, whereas cryptanalysis of Moon et al.'s scheme is solicited in Sect. 5. The proposed scheme is presented in Sect. 6. In Sect. 7, we present security analysis of proposed scheme, while its automated analysis is solicited in Sect. 8. The comparative performance analysis is illustrated in Sect. 9. Finally, a brief conclusion is made in Sect. 10

3 Preliminaries

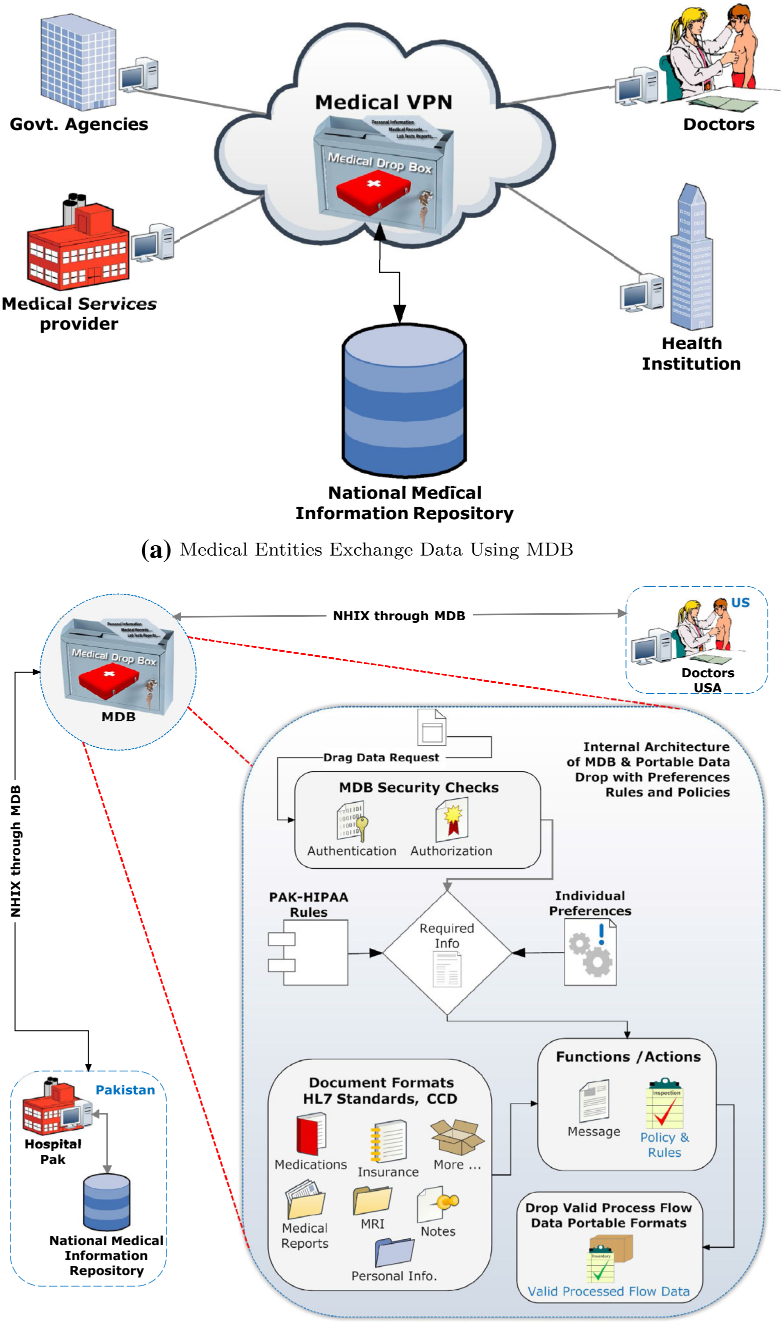
This section explains the basic architecture for medical drop box in NHIX environments, the need of security for such systems, notations used throughout the paper and fundamental hard problems concerning hash functions, Chebyshev chaotic maps, and the conjoint model for adversarial capabilities.

3.1 Medical drop box

NHIX has the potential to improve the lives of millions of people. To meet the standards for the NHIX, MDB must support reliable and secure transfer of data among the various systems of the world and also facilitate to access and retrieval of data with all its privacy. MDB is a standard application implemented according to HL7 and HIT standards, so the information can be exchanged between the health service providers and health-related industry. The MDB is a small-scale exploration focused on one individual's information. The MDB can be envisioned as a portable folder which will enable any person to carry all of his/her electronic medical information and share it with his/her medical service provider. However, this covering the necessities of the healthcare industry and allowing them to communicate clinical data internally as well as exchange of medical information with other parts of the world. Figure 1 shows the information access by medical entities through the MDB.

3.2 Medical drop box security

Medical information security is the most important aspect in MDB for data exchange. Security of medical data in all level must be ensured during, i.e., access, transmission, and storage. Before an entity or a user executes a function for access or to exchange the information, the access control list and authentication process will execute to verify whether this entity has the permission to perform that operation. Therefore, the MDB makes sure that the entity is authenticated and eligible to perform the particular operation. The privacy of individual/patient and authenticity of medical entity is very important. However, as we have stated, the challenges are multifaceted and are extremely complex. In an effort to advance healthcare privacy and exchange of protected health information, we proposed a road map of how we could authenticate



(a) Medical Entities Exchange Data Using MDB

(b) MDB Internal processing Architecture

Fig. 1 Access architecture of medical drop box

Table 1 Notation blueprint

Notations	Description	Notations	Description
$\mathcal{S}$	The server	$\mathcal{U}_i$	The legitimate user/client
ID_i	Identity of user/client	PW_i	Password of user/client
u	Distinct arbitrary number of $\mathcal{U}_i$	v	Distinct arbitrary number of $\mathcal{S}$
k_{u}	Secret key of $\mathcal{U}_i$	k_{s}	Secret key of $\mathcal{S}$
$\|$	String concatenation operator	$\oplus$	Bitwise XOR operation
$h(.)$	A one-way hash function	$\mathcal{A}$	The attacker/adversary

the medical entities of MDB. These entities are responsible for exchanging protected health information between different stakeholders/entities using MDB.

3.3 Notations

All notations used in this paper are itemized in Table 1.

3.4 Hash functions

A hash function $H : \{0, 1\}^* \rightarrow Z_q^*$ produces fixed-size output code $C = H(S)$ by taking random size input string S. The produced output is often designated as hash value or hash code. Trivial modification in the input string S can bring nontrivial change in the output C. Following are the characteristics to nominate a function as a secure hash function:

- Computationally, it is effortless to compute $C_t = H(P_t)$ if P_t and $H(.)$ are specified.
- It is tedious task to know two inputs P_1 and P_2, such that $H(P_1) = H(P_2)$. This characteristic is recognized as collision resistance property (CRP).

Definition 1 (*CRP for hash functions*) Consider $H(.)$ as a prearranged collision resistant (CR) secure hash function. The likelihood that $\mathcal{A}$ (an adversary) can extract a pair $(P_1 \neq P_2)$, where $H(P_1) = H(P_2)$ is established as $\mathrm{Advnt}_{\mathcal{A}}^{\mathrm{HASH}}(t) = \mathrm{Pr0}[(P_1, P_2) \Leftarrow_r \mathcal{A} : (P_1 \neq P_2)$ and $H(P_1) = H(P_2)]$, where $\mathcal{A}$ can select any arbitrary pair (P_1, P_2) through polynomial time t. Referring to the CR property, the advantage carried by $\mathcal{A}$ is as follows: $\mathrm{Advnt}_{\mathcal{A}}^{\mathrm{HASH}}(t) \leq \epsilon$ for any sufficiently large $\epsilon > 0$.

3.5 Chebyshev chaotic maps

Let k be an integer and r be a variable form a set $[-1, 1]$, the Chebyshev polynomial $\mathrm{Tn}(x) : [-1, 1] \rightarrow [-1, 1]$ is defined as $T_k(r) = \cos(k\arccos(r))$. Given $n >= 2$, $T_0(r) = 1$ and $T_1(r) = r$, the recurrent relationship of Chebyshev polynomial map $T_k : R \rightarrow R$ is defined as follows: $T_k(r) = 2rT_{k-1}(r) - T_{k-2}(r)$. There are two main features pertaining to Chebyshev polynomial. (1) Chaotic feature the polynomial $T_k(r) : [-1, 1] \rightarrow [-1, 1]$ of degree k where $k >= 1$ represents a chaotic map with $f * (r) = 1/(\pi\sqrt{1 - r^2})$ (invariant density) for all positive Lyapunov exponent k. (2) Semigroup feature the polynomial over an interval $[-\infty, \infty]$ is defined as follows: $T_k(r) = (2rT_{k-1}(r) - T_{k-2}(r)) \bmod p$, where $k >= 2$, $r \in [-\infty, \infty]$, and p a large range prime number. Besides, $T_x(T_y(r)) = T_{xy}(r) = T_y(T_x(r)) \bmod p$.

Definition 2 (*Chebyshev chaotic map-based discrete logarithm problem (CMDLP)*) Given the polynomial K and r, compute k, such that $K = T_k(r)$. The likelihood that a polynomial time (t) bound $\mathcal{A}$ (an adversary) can compute k is as given: $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{CMDLP}}(t) = \mathrm{Prb}[(\mathcal{A}(= K, r) = k : k \in Z_p, K = T_k(r)]$. The CMDLP supposition implies that $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{CMDLP}}(t) \leq \epsilon$, for sufficiently large $\epsilon > 0$.

3.6 Adversarial model

In this paper, we have taken into consideration the conjoint model for adversarial capabilities as presented by [6,9,10]. Following are the considerations:

1. $\mathcal{A}$ have control over entire public communication link. $\mathcal{A}$ (an adversary) is capable to interrupt, rerun, amend, eliminate, or can transmit a new forged message.
2. The stored parameters in a smart card are liable to expose by $\mathcal{A}$ (an adversary).
3. A user/server of the system can act as $\mathcal{A}$ (an adversary).
4. The identities pertaining to users and server/s are known to insiders.
5. The private key of the server cannot be compromised.

4 Review of Moon et al.'s scheme

Registration, login, and authentication phases of Moon et al.'s scheme are presented in this section as follows:

4.1 Registration phase

The registration process is performed as follows:

Step R1: $\mathcal{U}_i$ provides scan of his/her BIO_i. Then, $\mathcal{U}_i$ chooses the ID_i and PW_i. $\mathcal{U}_i$ proceeds and determines $\mathrm{PWD} = h_1(\mathrm{PW}_i \| H(\mathrm{BIO}_i))$, then sends registration request in the form of $\{\mathrm{ID}_i, \mathrm{PWD}\}$ to server $\mathcal{S}$ over secure channel.

Step R2: After receiving request from $\mathcal{U}_i$, $\mathcal{S}$ verifies the existence the of ID_i in the repository. If ID_i does not exist, $\mathcal{S}$ produces unique random number k_u. Then, $\mathcal{S}$ determines $K = h_1(\mathrm{ID}_i \| \mathrm{PWD})$, $\mathrm{IM}_1 = K \oplus h_1(k_\mathrm{u})$, $\mathrm{IM}_2 =$

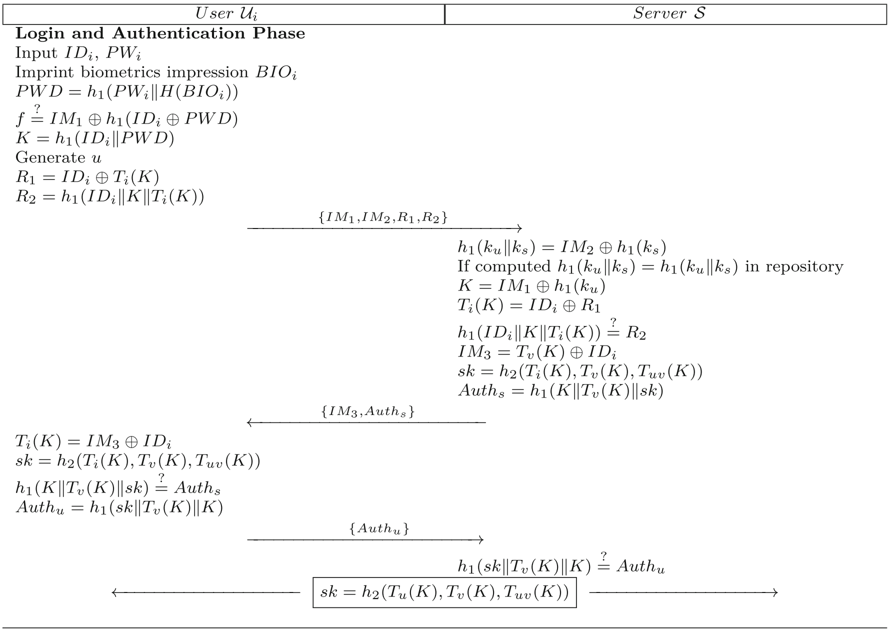

User $\mathcal{U}_i$
Server $\mathcal{S}$
Login and Authentication Phase
Input ID_i, PW_i
Imprint biometrics impression BIO_i
$PWD = h_1(PW_i \| H(BIO_i))$
$f \overset{?}{=} IM_1 \oplus h_1(ID_i \oplus PWD)$
$K = h_1(ID_i \| PWD)$
Generate u
$R_1 = ID_i \oplus T_i(K)$
$R_2 = h_1(ID_i \| K \| T_i(K))$
$\xrightarrow{\{IM_1, IM_2, R_1, R_2\}}$
$h_1(k_u \| k_s) = IM_2 \oplus h_1(k_s)$
If computed $h_1(k_u \| k_s) = h_1(k_u \| k_s)$ in repository
$K = IM_1 \oplus h_1(k_u)$
$T_i(K) = ID_i \oplus R_1$
$h_1(ID_i \| K \| T_i(K)) \overset{?}{=} R_2$
$IM_3 = T_v(K) \oplus ID_i$
$sk = h_2(T_i(K), T_v(K), T_{uv}(K))$
$Auth_s = h_1(K \| T_v(K) \| sk)$
$\xleftarrow{\{IM_3, Auth_s\}}$
$T_i(K) = IM_3 \oplus ID_i$
$sk = h_2(T_i(K), T_v(K), T_{uv}(K))$
$h_1(K \| T_v(K) \| sk) \overset{?}{=} Auth_s$
$Auth_u = h_1(sk \| T_v(K) \| K)$
$\xrightarrow{\{Auth_u\}}$
$h_1(sk \| T_v(K) \| K) \overset{?}{=} Auth_u$
$\longleftarrow \boxed{sk = h_2(T_u(K), T_v(K), T_{uv}(K))} \longrightarrow$

Fig. 2 Moon et al.'s scheme

$h_1(k_u \| k_s) \oplus h_1(k_s)$, and $f = \mathrm{IM}_1 \oplus h_1(\mathrm{ID}_i \oplus \mathrm{PWD})$. At the end, $\mathcal{S}$ store $\{\mathrm{ID}_i \oplus h_1(k_u \| k_s), k_u \oplus k_s, h_1(k_u \| k_s)\}$ in its repository.

Step R3: $\mathcal{S}$ issue a smart card containing $\{\mathrm{IM}_1, \mathrm{IM}_2, f, h_1(.), h_2(.), H(.)\}$ and then send this smart card to $\mathcal{U}_i$.

4.2 Login and authentication phase

The login and authentication as illustrated in Fig. 2 is performed as follows:

Step LP1 User $\mathcal{U}_i$ provides his/her ID_i and PW_i and imprints his/her biometrics impression as BIO_i, then computes $\mathrm{PWD} = h_1(\mathrm{PW}_i \| H(\mathrm{BIO}_i))$, and verifies $f \overset{?}{=} \mathrm{IM}_1 \oplus h_1(\mathrm{ID}_i \oplus \mathrm{PW}_i)$, after successful verification, the $\mathcal{U}_i$ steps forward and computes $K = h_1(\mathrm{ID}_i \| \mathrm{PW}_i)$. $\mathcal{U}_i$ generates u and calculates $R_1 = \mathrm{ID}_i \oplus T_i(K)$ and $R_2 = h_1(\mathrm{ID}_i \| K \| T_i(K))$. At the end, $\mathcal{U}_i$ sends login request towards server in the form of $\{\mathrm{IM}_1, \mathrm{IM}_2, R_1, R_2\}$.

Step LP2 The server $\mathcal{S}$ receives the login request from $\mathcal{U}_i$ and calculates $h_1(k_u \| k_s) = \mathrm{IM}_2 \oplus h_1(k_s)$. If computed $h_1(k_u \| k_s)$ is present in the repository, then $\mathcal{S}$ proceeds and calculates $K = \mathrm{IM}_1 \oplus h_1(k_u)$ and $T_i(K) = \mathrm{ID}_i \oplus R_1$. $\mathcal{S}$ verifies $h_1(\mathrm{ID}_i \| K \| T_i(K)) \overset{?}{=} R_2$, successful verification results in computation of $\mathrm{IM}_3 = T_v(K) \oplus \mathrm{ID}_i$, $\mathrm{sk} = h_2(T_i(K), T_v(K), T_{uv}(K))$ and $\mathrm{Auth}_s = h_1(K \| T_v(K) \| \mathrm{sk})$. At the end, server generates challenge message in the form of $\{\mathrm{IM}_3, \mathrm{Auth}_s\}$.

Step LP3 After getting challenge message from $\mathcal{S}$, $\mathcal{U}_i$ determines $T_i(K) = \mathrm{IM}_3 \oplus \mathrm{ID}_i$ and $\mathrm{sk} = h_2(T_i(K), T_v(K), T_{uv}(K))$. Then, $\mathcal{U}_i$ verifies

$h_1(K\|T_v(K)\|\mathrm{sk}) \overset{?}{=} \mathrm{Auth_s}$, and after successful verification, it computes $\mathrm{Auth_u} = h_1(\mathrm{sk}\|T_v(K)\|K)$. $\mathcal{U}_i$ sends computed $\mathrm{Auth_u}$ towards server.

Step LP4 $\mathcal{S}$ receives $\mathrm{Auth_u}$ and verifies $h_1(\mathrm{sk}\|T_v(K)\|K) \overset{?}{=} \mathrm{Auth_u}$. Successful verification results in exchange of session key $\mathrm{sk} = h_2(T_i(K), T_v(K), T_{uv}(K))$ between the user and server.

5 Cryptanalysis of Moon et al.'s scheme

This section highlights that Moon et al.'s scheme does not provide scalability and incurs the searching time during authentication. Moreover, the scheme is liable to traceability attack.

5.1 Stolen verifier attack

In Moon et al.'s scheme, server $\mathcal{S}$ stores IM_1 and IM_2 in smart card and also maintains verifier table/database; therefore, stolen verifier attack can be attempted on Moon et al's scheme. Because $h_1(k_\mathrm{u})$ can be easily extracted from IM_1, similarly, by knowing public identity, $h_1(k_\mathrm{u}\|k_\mathrm{s})$ can also be extracted from stolen verifier. Leakage of $h_1(k_\mathrm{u}\|k_\mathrm{s})$ can be used to extract $h_1(k_\mathrm{s})$ from IM_2. An adversary after getting $h_1(k_\mathrm{s})$ can guess k_s and verify his guessed value using $h_1(k_\mathrm{s})$.

5.2 Scalability issue

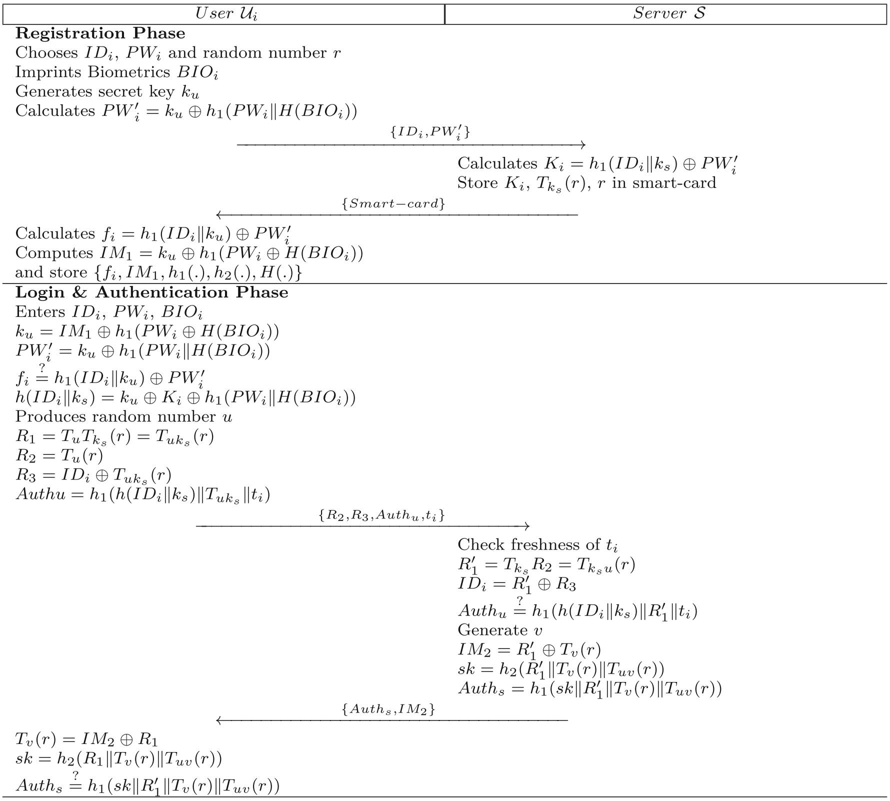
Since in Moon et al.'s scheme, server maintains a verifier table for each unique user in table. Therefore, this verifier table impose limitations on number of users along with computation time to scan and match verifier table entities.

5.3 Traceability

Anonymity is subject to two aims: (1) user's original identity cannot be exposed to $\mathcal{A}$ (an adversary) and (ii) it cannot be known by $\mathcal{A}$ whether or not two separate sessions are commenced by identical user. Moon et al. claimed their scheme to fulfill anonymity, but in login and authentication phase of their scheme, two parameters IM_1 and IM_2 are sent in request message from user. These two parameters remain the same in several login requests. Hence, their scheme does not fulfill a requirement to term it as anonymous. Hence, in their scheme, the user is traceable, which violates the user anonymity.

6 Proposed scheme

This section presents the detail of improved scheme which is illustrated in Fig. 3 as follows:



User $\mathcal{U}_i$ | *Server $\mathcal{S}$*

Registration Phase

Chooses ID_i, PW_i and random number r

Imprints Biometrics BIO_i

Generates secret key k_u

Calculates $PW_i' = k_u \oplus h_1(PW_i \| H(BIO_i))$

$\xrightarrow{\{ID_i, PW_i'\}}$

Calculates $K_i = h_1(ID_i \| k_s) \oplus PW_i'$

Store K_i, $T_{k_s}(r)$, r in smart-card

$\xleftarrow{\{Smart-card\}}$

Calculates $f_i = h_1(ID_i \| k_u) \oplus PW_i'$

Computes $IM_1 = k_u \oplus h_1(PW_i \oplus H(BIO_i))$

and store $\{f_i, IM_1, h_1(.), h_2(.), H(.)\}$

Login & Authentication Phase

Enters ID_i, PW_i, BIO_i

$k_u = IM_1 \oplus h_1(PW_i \oplus H(BIO_i))$

$PW_i' = k_u \oplus h_1(PW_i \| H(BIO_i))$

$f_i \overset{?}{=} h_1(ID_i \| k_u) \oplus PW_i'$

$h(ID_i \| k_s) = k_u \oplus K_i \oplus h_1(PW_i \| H(BIO_i))$

Produces random number u

$R_1 = T_u T_{k_s}(r) = T_{uk_s}(r)$

$R_2 = T_u(r)$

$R_3 = ID_i \oplus T_{uk_s}(r)$

$Authu = h_1(h(ID_i \| k_s) \| T_{uk_s} \| t_i)$

$\xrightarrow{\{R_2, R_3, Auth_u, t_i\}}$

Check freshness of t_i

$R_1' = T_{k_s} R_2 = T_{k_s u}(r)$

$ID_i = R_1' \oplus R_3$

$Auth_u \overset{?}{=} h_1(h(ID_i \| k_s) \| R_1' \| t_i)$

Generate v

$IM_2 = R_1' \oplus T_v(r)$

$sk = h_2(R_1' \| T_v(r) \| T_{uv}(r))$

$Auth_s = h_1(sk \| R_1' \| T_v(r) \| T_{uv}(r))$

$\xleftarrow{\{Auth_s, IM_2\}}$

$T_v(r) = IM_2 \oplus R_1$

$sk = h_2(R_1 \| T_v(r) \| T_{uv}(r))$

$Auth_s \overset{?}{=} h_1(sk \| R_1' \| T_v(r) \| T_{uv}(r))$

Fig. 3 Proposed scheme

6.1 Registration

Step RS1 The user $\mathcal{U}_i$ chooses ID_i, PW_i, random r, and a secret key k_u. Then, $\mathcal{U}_i$ provides scan of his/her biometrics as BIO_i and determines $\mathrm{PW}_i' = h_1(\mathrm{PW}_i \| H(\mathrm{BIO}_i))$. After that $\mathcal{U}_i$ sends registration request $\{\mathrm{ID}_i, \mathrm{PW}_i'\}$ to $\mathcal{S}$ over protected channel.

Step RS2 Upon reception, $\mathcal{S}$ computes $K_i = h_1(\mathrm{ID}_i \| k_\mathrm{s}) \oplus \mathrm{PW}_i'$. $\mathcal{S}$ then stores K_i, $T_{k_\mathrm{s}}(r)$, r, $h_1(.)$, and $H(.)$ into smart card and handover it to $\mathcal{U}_i$.

Step RS3 User $\mathcal{U}_i$ determines $f_i = h_1(\mathrm{ID}_i \| k_\mathrm{u}) \oplus \mathrm{PW}_i'$ and, $\mathrm{IM}_1 = k_\mathrm{u} \oplus h_1(\mathrm{PW}_i \oplus H(\mathrm{BIO}_i))$ using his/her private key k_u. At the end, $\mathcal{U}_i$ stores computed f_i and IM_1 into smart card. Finally, smart card held by user $\mathcal{U}_i$ contains $\{f_i, \mathrm{IM}_1, h_1(.), h_2(.), H(.)\}$.

6.2 Login

The login phase proceeds as follows: $\mathcal{U}_i$ enters the card into specific reader and provides $\mathrm{ID}_i, \mathrm{PW}_i$. Then, $\mathcal{U}_i$ scans his/her biometrics BIO_i.$\mathcal{U}_i$ computes $k_\mathrm{u} =$

$\mathrm{IM}_1 \oplus h_1(\mathrm{PW}_i \oplus H(\mathrm{BIO}_i))$ and $\mathrm{PW}'_i = k_\mathrm{u} \oplus h_1(\mathrm{PW}_i \| H(\mathrm{BIO}_i))$. $\mathcal{U}_i$ then verifies whether $f_i \overset{?}{=} h_1(\mathrm{ID}_i \| k_\mathrm{u}) \oplus \mathrm{PW}'_i$ is true or not. If it is true, then $\mathcal{U}_i$ calculates $h(\mathrm{ID}_i \| k_\mathrm{s}) = k_\mathrm{u} \oplus K_i \oplus h_1(\mathrm{PW}_i \| H(\mathrm{BIO}_i))$. After that $\mathcal{U}_i$ produces u and computes $R_1 = T_\mathrm{u} T_{k_\mathrm{s}}(r) = T_{uk_\mathrm{s}}(r)$, $R_2 = T_\mathrm{u}(r)$, $R_3 = \mathrm{ID}_i \oplus T_{uk_\mathrm{s}}(r)$, $and Auth_\mathrm{u} = h_1(h(\mathrm{ID}_i \| k_\mathrm{s}) \| T_{uk_\mathrm{s}} \| t_i)$. At the end, $\mathcal{U}_i$ sends $\{R_2, R_3, \mathrm{Auth}_\mathrm{u}, t_i\}$ to $\mathcal{S}$, where t_i is freshly generated timestamp.

6.3 Authentication

Initially, the server $\mathcal{S}$ checks the freshness of t_i if t_i is fresh $\mathcal{S}$ responds to login request from $\mathcal{U}_i$ as follows:

Step AS 1 $\mathcal{S}$ computes $R'_1 = T_{k_\mathrm{s}} R_2 = T_{k_\mathrm{s} u}(r)$, $\mathrm{ID}_i = R'_1 \oplus R_3$ using his/her private key k_s. $\mathcal{S}$ authenticates $\mathcal{U}_i$ after successful verification of $\mathrm{Auth}_\mathrm{u} \overset{?}{=} h_1(h(\mathrm{ID}_i \| k_\mathrm{s}) \| R'_1 \| t_i)$. Server $\mathcal{S}$ then produces a random number v and calculates $\mathrm{IM}_2 = R'_1 \oplus T_v(r)$, $\mathrm{sk} = h_2(R'_1 \| T_v(r) \| T_{uv}(r))$, and $\mathrm{Auth}_\mathrm{s} = h_1(\mathrm{sk} \| R'_1 \| T_v(r) \| T_{uv}(r))$. After that server $\mathcal{S}$ sends a challenge message $\{\mathrm{Auth}_\mathrm{s}, \mathrm{IM}_2\}$ to $\mathcal{U}_i$.

Step AS 2 User $\mathcal{U}_i$ after receiving challenge message computes $T_v(r) = \mathrm{IM}_2 \oplus R'_1$, $\mathrm{sk} = h_2(R_1 \| T_v(r) \| T_{uv}(r))$ and authenticates server after successful verification of $\mathrm{Auth}_\mathrm{s} \overset{?}{=} h_1(\mathrm{sk} \| R'_1 \| T_v(r) \| T_{uv}(r))$. Then, common session key is exchanged between $\mathcal{U}_i$ and $\mathcal{S}$.

7 Security analysis

Following subsections performs formal and informal security analysis of proposed scheme. Furthermore, the analysis is also verified using the widespread automated tool ProVerif in the following section.

7.1 Formal security analysis

The incumbent analysis to illustrate the security of proposed scheme is adopted from [25,37]. The oracles for the said purposes are defined as follows:

- Reveal: This oracle returns x from given $T_x(r)$ and r unconditionally.
- Extract: This oracle outputs a string S from the one-way hash function $R = h(S)$ unconditionally.

Theorem 1 *The proposed scheme is verifiable as secure besides an attacker $\mathcal{A}$ for extraction of $\mathcal{U}_i$'s identity (ID_i), the private key (k_s) of $\mathcal{S}$, and the computed session key SK between $\mathcal{U}_i$ and $\mathcal{S}$ under the hardness assumption of CMDLP and ruminating the secure hash function as random oracle.*

Proof 1 Let an adversary $\mathcal{A}$ is having the abilities to extract $\mathcal{U}_i$'s ID_i, $\mathcal{S}$'s secret key k_s, and the session key sk shared among $\mathcal{U}_i$ and $\mathcal{S}$. $\mathcal{A}$ can execute the algorithmic experiment $\mathrm{EXP1}_{\mathcal{A},\mathrm{PCMAS}}^{\mathrm{CMDLP,OWHASH}}$ against our proposed chaotic map-based authentication scheme. The success probability of said experiment is defined as follows: $\mathrm{Suc}_1 = |\mathrm{Prob}[\mathrm{EXP1}_{\mathcal{A},\mathrm{PCMAS}}^{\mathrm{CMDLP,OWHASH}} = 1] - 1|$. For the mentioned experiment, $\mathcal{A}$ can make, q_{ex} and q_{rv}, extract and reveal queries, respectively. Stating the experiment $\mathrm{EXP1}_{\mathcal{A},\mathrm{PCMAS}}^{\mathrm{CMDLP,OWHASH}}$, $\mathcal{A}$ can excerpt user identity ID_i, the shared key sk, the secret authentication parameter $h(\mathrm{ID}_i\|k_\mathrm{s})$, and server's private key k_s. If $\mathcal{A}$ can: (1) invert the hash function and (2) break CMDLP. However, mentioning Definition 1 inverting one-way hash function is computationally infeasible. Likewise, stating Definition 2, breaking CMDLP is also computationally infeasible. Hence, proposed scheme is invincible against $\mathcal{A}$ to derive $\mathcal{U}_i$'s ID_i, $\mathcal{S}$'s secret key k_s, and the session key sk shared among $\mathcal{U}_i$ and $\mathcal{S}$.

Algorithm 1 $EXP1_{\mathcal{A},PCMAS}^{CMDLP,OWHASH}$

```
Eavesdrop login request message $\{R_2, R_3, Auth_\mathrm{u}, t_i\}$, Where $R_2 = T_\mathrm{u}(r)$, $R_3 = ID_i \oplus T_{uk_\mathrm{s}}(r)$, $Auth_\mathrm{u} = h_1(h(ID_i\|k_\mathrm{s})\|T_{uk_\mathrm{s}}\|t_i)$
Call Reveal on $R_2$ and get $u^* \leftarrow Reveal(R_2)$
Compute $R_1^* = T_\mathrm{u}^* T_{k_\mathrm{s}}(r)$ and $ID_i^* = R_3 \oplus R_1^*$
Call Extract oracle on $Auth_\mathrm{u}$ and get $(h(ID_i\|k_\mathrm{s})^*\|T_{uk_\mathrm{s}}^{**}\|t_i^*) \leftarrow Extract(Auth_\mathrm{u})$
if $(T_{uk_\mathrm{s}}^{**} = R_1^*)$ and $(t_i = t_i^*)$ then
    Compute $ID_i^* = R_1^* \oplus R_3$
    Call Extract on $h(ID_i\|k_\mathrm{s})^*$ and get $(ID_i^{**}\|k_\mathrm{s}^*) \leftarrow h(ID_i\|k_\mathrm{s})^*$
    if $(ID_i^* = ID_i^{**})$ then
        Accept $R_1^*$, $h(ID_i\|k_\mathrm{s})^*$ and $k_\mathrm{s}$
        Eavesdrop response message $Auth_\mathrm{s}$ and $IM_\mathrm{s}$, where $Auth_\mathrm{s} = h_1(\mathrm{sk}\|R_1'\|T_v(r)\|T_{uv}(r))$ and $IM_2 = R_1' \oplus T_v(r)$
        Call Extract on $Auth_\mathrm{s}$ and get $(sk^*\|R_1^{'*}\|T_v^*(r)\|T_{uv}^*(r)) \leftarrow h_1(sk\|R_1'\|T_v(r)\|T_{uv}(r))$
        if $(R_1^{'*} = R_1^*)$ then
            Accept $T_v^*(r)$, $T_{uv}^*(r)$ and $sk^*$
        else
            return Fail
        end if
    else
        return Fail
    end if
else
    return Fail
end if
```

7.2 Informal security analysis

This subsection presents the security and correctness of improved scheme under the same considerations as discussed earlier in Sect. 5. The in-depth investigation reveals that the proposed scheme is capable to resist against recognized attacks. Table 2 provides a comparative overview of proposed scheme's security with related schemes. The detailed description is as follows:

Table 2 Security Comparisons

Scheme	Proposed	[26]	[22]	[18]	[21]	[12]
Anonymity and untraceability	✔	✘	✘	✔	✔	✘
Mutual authentication	✔	✔	✔	✔	✔	✔
Resists forgery attack	✔	✔	✘	✘	✘	✘
Resists smart card theft attack	✔	✔	✘	✘	✘	✔
Resists replay attack	✔	✔	✔	✔	✘	✘
Provides forward secrecy	✔	✔	✔	✘	✘	✘
No verifier table	✔	✘	✘	✔	✔	✔
Resists password guessing attack	✔	✔	✔	✘	✘	✘

✘ no, ✔ yes

7.2.1 Anonymity and privacy

Since identity of user $\mathcal{U}_i$ is not transmitted through public conduit, rather R_2 and R_3 are transmitted towards server $\mathcal{S}$. Both of these parameters are generated uniquely during each session. Adversary $\mathcal{A}$ can only violate the anonymity of $\mathcal{U}_i$ once R_1 is computed, but R_1 calculation requires private key k_s. Therefore, proposed scheme proved to offer untraceability and anonymity.

7.2.2 Mutual authentication

User $\mathcal{U}_i$ is authenticated by server $\mathcal{S}$ after verifying $\text{Auth}_u \overset{?}{=} h_1(h(\text{ID}_i\|k_s)\|R_1'\|t_i)$. Calculation of $\text{Auth}_u = h_1(h(\text{ID}_i\|k_s) - \|T_{uk_s}\|t_i)$ demands valid smart card, password PW_i, and biometrics BIO_i of legal user. Therefore, adversary $\mathcal{A}$ can only betray server $\mathcal{S}$ if he/she has smart card, password PW_i, and biometrics BIO_i of user $\mathcal{U}_i$. Considering the other side, $\mathcal{U}_i$ authenticates $\mathcal{S}$ by verifying $\text{Auth}_s \overset{?}{=} h_1(\text{sk}\|R_1'\|T_v(r)\|T_{uv}(r))$. This computation demands secret key k_s of server $\mathcal{S}$, as discussed in Sect. 7.2.1. Therefore, adversary $\mathcal{A}$ must have private key k_s of server $\mathcal{S}$ to betray user $\mathcal{U}_i$. Therefore, it can be calculated that only legitimate $\mathcal{U}_i$ can clear authentication check from $\mathcal{S}$ and vice versa. Moreover, $\mathcal{U}_i$ is authenticated by server $\mathcal{S}$ after verifying $\text{Auth}_u \overset{?}{=} h_1(h(\text{ID}_i\|k_s)\|R_1'\|t_i)$. Calculation of $\text{Auth}_u = h_1(h(\text{ID}_i\|k_s)\|T_{uk_s}\|t_i)$ demands valid smart card, password PW_i, and biometrics BIO_i of legal user. Therefore, adversary $\mathcal{A}$ can only betray server $\mathcal{S}$, if he/she has smart card, password PW_i, and biometrics BIO_i of user $\mathcal{U}_i$. On the other hand, user $\mathcal{U}_i$ authenticates server $\mathcal{S}$ by verifying $\text{Auth}_s \overset{?}{=} h_1(\text{sk}\|R_1'\|T_v(r)\|T_{uv}(r))$. This computation demands secret key k_s of server $\mathcal{S}$, as discussed in Sect. 7.2.1. Therefore, adversary $\mathcal{A}$ must have private key k_s of server $\mathcal{S}$ to betray user $\mathcal{U}_i$. Therefore, it can be calculated that only legitimate user can clear authentication check from $\mathcal{S}$ and vice versa. Hence, proposed scheme offers mutual authentication.

7.2.3 *Server and user impersonation attacks*

Only authentic user $\mathcal{U}_i$ can generate login request $\{R_2, R_3, \mathrm{Auth}_{\mathrm{u}}, t_i\}$ and, in turn, only authentic server $\mathcal{S}$ can reply login request with challenge message $\{\mathrm{Auth}_{\mathrm{s}}, \mathrm{IM}_2\}$, as discussed in Sect. 7.2.2. Therefore, it is infeasible to launch impersonation attacks on the proposed scheme.

7.2.4 *Privileged insider attack*

User $\mathcal{U}_i$ calculates $\mathrm{PW}_i' = h_1(\mathrm{PW}_i \| H(\mathrm{BIO}_i)) \oplus k_{\mathrm{u}}$ and sends PW_i' along with ID_i towards server $\mathcal{S}$, since PW_i' calculation is done by concatenating PW_i with $H(\mathrm{BIO}_i)$, and moreover, these concatenated values are further protected by hash function. It becomes very difficult for an adversary $\mathcal{A}$ to calculate hash protected ingredients/values in polynomial time. Furthermore, server $\mathcal{S}$ does not have direct access to these hash protected values. Hence, it can be concluded that proposed scheme overcomes the privileged insider attack.

7.2.5 *Replay attack*

If an adversary $\mathcal{A}$ attempts to replay a former login request $\{R_2, R_3, \mathrm{Auth}_{\mathrm{u}}, t_i\}$. $\mathcal{S}$ will be able to identify replay message quickly after verifying freshness of the time stamp t_i. Therefore, $\mathcal{S}$ will be able to avoid replay messages. Hence, proposed scheme successfully turns down replay attack.

7.2.6 *Stolen verifier attack*

$\mathcal{S}$ uses his secret key k_{s} for manipulating login and authentication message and does not maintain verifier table. Therefore, it is confirmed that proposed scheme withstands stolen verifier attack.

7.2.7 *Denial of service attack*

The smart card verifies three entries identity ID_i, password PW_i, and biometrics BIO_i. The smart card will instantly reject the request if any one of these three entries is invalid. Hence, $\mathcal{U}_i$ will be safe from denial of service attack due to wrong entry.

7.2.8 *Password guessing attack*

Suppose an adversary $\mathcal{A}$ excerpts f_i, IM_1, $T_{k_{\mathrm{s}}}(r)$, r stored in smart card. Then, $\mathcal{A}$ is required to calculate $\mathrm{PW}_i' = h_1(\mathrm{PW}_i \| H(\mathrm{BIO}_i))$ and it is difficult to do so, because $\mathcal{A}$ has to get password PW_i and $H(\mathrm{BIO}_i)$ that are hash protected. Hence, offline password guessing attack is infeasible. Likewise, permission for invalid request makes it hard to initiate online password guessing attack. Hence, proposed scheme withstands both offline and online guessing attacks.

8 Security validation through ProVerif

In this section, automated security analysis of proposed scheme is done through ProVerif [1,19], the tool widely used for automated security analysis. It can be applied to a variety of authentication schemes to verify the security and privacy. ProVerif also provides the support for different cryptographic primitives like: hash functions, symmetric and asymmetric cryptography, bit-commitment, and digital signature. The tool has the capability to evaluate the reachability stuff, communication declarations, and observational correspondence [40]. The reasoning capabilities are very much useful in security domain, because they permit to consider the emerging properties of confidentiality, privacy, traceability, verifiability, and authentication in analysis process. ProVerif is also used as a verifier for cryptographic protocols with fully automatic with respect to an boundless sessions and message size [7,8]. The simulation model of ProVerif is composed of three sections: (i) the declaration section; (ii) the process section; and (iii) main section. The declaration section as shown in Fig. 4 is used to declare all the variables, names, and constants along with cryptographic functions modeled as constructors, destructors, and equations. Likewise, process section as shown in Fig. 5 is used to define the working of each involved process and main section defines the working of protocol to be investigated. We have imprinted all the variables, names along with two communication channels in declaration section. In process section, we defined the server and user processes (*ServerS* and *UserUx*), respectively. Finally, to analyze the proposed scheme, three queries are simulated in main part as shown in Fig. 6; the results are solicited as follows:

1. RESULT inj-event(endServer(id)) ==> inj-event(beginServer(id)) is true.
2. RESULT inj-event(endUser(id_1265)) ==> inj-event(beginUser(id_1265)) is true.
3. RESULT not attacker(SK[]) is true.

(1) and (2) shows that the processes relating to $\mathcal{S}$ and $\mathcal{U}$ are executed and sacked normally, while (3) illustrates that attack query on session key SK is not successful. Hence, proposed scheme is correct and robust against session key discloser attack.

```
(******************* Channels *****************)
free SecCh:channel [private].   (*secure channel*)
free PubCh:channel.      (*public channel*)
(*************** Names & Variables *************)
free IDi:bitstring.
free BIOi:bitstring [private].
free Kui:bitstring [private].
free PWi:bitstring [private].
free r:bitstring.
free Ks:bitstring.
free Ksi:bitstring [private].
fun h(bitstring):bitstring.
fun h1(bitstring):bitstring.
fun h2(bitstring):bitstring.
fun XOR(bitstring,bitstring):bitstring.
fun Concat(bitstring,bitstring):bitstring.
fun CCM(bitstring,bitstring):bitstring.
fun CCM1(bitstring,bitstring,bitstring):bitstring.
```

Fig. 4 Declarations

```
(******************* processes ******************)
(*****************User Process*****************)
let UserUi =
(************ Registration*************)
new r1i:bitstring;
let PWi' = h1(Concat(PWi,h(BIOi))) in
out(SecCh,(IDi,PWi'));
in(SecCh,(IMi:bitstring));
let fi = XOR( h1(Concat(IDi,Kui)) , PWi') in
(********Login & Authentication********)
event begin_User (IDi);
let fi' = XOR(h1(Concat(IDi,Kui)),h1(Concat(PWi,h(BIOi))))in
if(fi = fi') then new u:bitstring;
let R1 = CCM1(u,Ks,r) in
let R2 = CCM(u,r) in
let R3 = XOR(IDi, CCM1(u,Ks,r) ) in new ti:bitstring;
let Authu = h1(Concat(h(Concat(IDi,Ks)),(CCM(u,Ks),ti))) in
out(PubCh,(R2,R3,Authu,ti));
in(PubCh,(Auths:bitstring,IMs:bitstring));
let Tvr = XOR (IMs,CCM1(Ks,u,r)) in
let sk = h2(Concat(R1,(Tvr,CCM(u,Tvr)))) in
let Auths' = h1(Concat(sk,(R1,Tvr,CCM(u,Tvr)))) in
if(Auths = Auths') then event end_User(IDi)
else 0.
(**************** Server Process ****************)
let ServerS =
(************ Registration*************)
in(SecCh,(xIDi:bitstring,PWi':bitstring));
let Ki = XOR( h1(Concat(IDi,Ks)) , PWi' ) in
let IMi = XOR(Ki,h1(Ksi)) in
out(SecCh,(IMi));
(********Login & Authentication********)
event begin_Server(Ks);
in(PubCh,(R2:bitstring,R3:bitstring,Authu:bitstring,ti:bitstring));
let R1' = CCM(Ks,R2) in
let IDi = XOR (R1',R3) in
let Authu' = h1(Concat(h(Concat(IDi,Ks)),(R1',ti))) in
if ( Authu = Authu' ) then new v:bitstring;
let IMs = XOR (R1',CCM(v,r)) in
let sk = h2(Concat(R1',(CCM(v,r),CCM(v,R2)))) in
let Auths =  h1(Concat(sk,(R1',CCM(v,r),CCM(v,R2)))) in
out(PubCh,(Auths,IMs));
event end_Server(Ks)
else 0.
```

Fig. 5 Processes

```
(******************* Events*******************)
event begin_User(bitstring).
event end_User(bitstring).
event begin_Server(bitstring).
event end_Server(bitstring).
(***********Process Replication*****************)
process ( (!ServerS) | (!UserUi) )
(******************queries*******************)
free SK:bitstring [private].
query attacker(SK).
query id:bitstring; inj event(end_User(id)) ==> inj event(begin_User(id)) .
query id:bitstring; inj event(end_Server(id)) ==> inj event(begin_Server(id)) .
```

Fig. 6 Main

9 Performance analysis

Performance comparison in terms of computation is presented in this section. For quick reference, formal computational terms are listed below:

Table 3 Computation comparisons

Scheme	User side	Server side	Total
Guo et al. [12]	$3T_{\mathrm{H}} + 2T_{\mathrm{C}} + 1T_{\mathrm{ED}}$	$2T_{\mathrm{H}} + 2T_{\mathrm{C}} + 1T_{\mathrm{ED}}$	$5T_{\mathrm{H}} + 4T_{\mathrm{C}} + 2T_{\mathrm{ED}}$
Lin and Han-Yu [21]	$4T_{\mathrm{H}} + 2T_{\mathrm{C}} + 1T_{\mathrm{ED}}$	$2T_{\mathrm{H}} + 3T_{\mathrm{C}} + 1T_{\mathrm{ED}}$	$6T_{\mathrm{H}} + 5T_{\mathrm{C}} + 2T_{\mathrm{ED}}$
Jiang et al. [18]	$2T_{\mathrm{H}} + 3T_{\mathrm{C}} + 1T_{\mathrm{ED}}$	$1T_{\mathrm{H}} + 3T_{\mathrm{C}} + 2T_{\mathrm{ED}}$	$3T_{\mathrm{H}} + 6T_{\mathrm{C}} + 3T_{\mathrm{ED}}$
Lu et al. [22]	$7T_{\mathrm{H}} + 2T_{\mathrm{C}}$	$5T_{\mathrm{H}} + 2T_{\mathrm{C}}$	$12T_{\mathrm{H}} + 4T_{\mathrm{C}}$
Moon et al. [26]	$6T_{\mathrm{H}} + 2T_{\mathrm{C}}$	$5T_{\mathrm{H}} + 2T_{\mathrm{C}}$	$10T_{\mathrm{H}} + 4T_{\mathrm{C}}$
Proposed scheme	$7T_{\mathrm{H}} + 3T_{\mathrm{C}}$	$3T_{\mathrm{H}} + 3T_{\mathrm{C}}$	$10T_{\mathrm{H}} + 6T_{\mathrm{C}}$

- T_{C} represents time for determining Chebyshev.
- T_{ED} represents time required to perform encryption and decryption.
- T_{H} refers to time requirement for one-way hash functions.

The performance comparison illustrated in Table 3 depicts that proposed scheme is lightweight in terms of computation cost as compared to listed-related schemes. Therefore, proposed scheme can be considered to perform better than the rest of the related schemes, although computation cost of proposed scheme is slightly greater than Moon et al. [26], but it is proved earlier in this paper that Moon et al.'s scheme does not provide many security features, whereas proposed scheme provides security against recognized security threats and, in turn, offers robustness and efficiency.

10 Conclusion

In this paper, we have analyzed a recent authentication scheme for TMIS based on Chebyshev chaotic maps. It has been proved that Moon et al.'s scheme does not provide proper user anonymity and is having some issues relating to scalability. Furthermore, their scheme is vulnerable to stolen verifier attack. Then, we proposed an improved scheme for accessing medical drop box data, which has been proved invincible against familiar attacks by rigorous informal and formal analysis backed by automated analysis performed in ProVerif.

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

References

1. Abadi M, Blanchet B, Comon-Lundh H (2009) Models and proofs of protocol security: a progress report. In: Computer aided verification. Springer, New York, pp 35–49
2. Alizadeh M, Abolfazli S, Zamani M, Baharun S, Sakurai K (2016) Authentication in mobile cloud computing: a survey. J Netw Comput Appl 61:59–80
3. Alizadeh M, Baharun S, Zamani M, Khodadadi T, Darvishi M, Gholizadeh S, Ahmadi H (2015) Anonymity and untraceability assessment of authentication protocols in proxy mobile ipv6. J Teknol 72(5)
4. Alizadeh M, Zamani M, Baharun S, Hassan WH, Khodadadi T (2015) Security and privacy criteria to evaluate authentication mechanisms in proxy mobile ipv6. J Teknol 72(5)

5. Alizadeh M, Zamani M, Baharun S, Manaf AA, Sakurai K, Anada H, Keshavarz H, Chaudhry SA, Khan MK (2015) Cryptanalysis and improvement of a secure password authentication mechanism for seamless handover in proxy mobile ipv6 networks. PloS One 10(11):e0142,716
6. Cao X, Zhong S (2006) Breaking a remote user authentication scheme for multi-server architecture. IEEE Commun Lett 10(8):580–581. doi:10.1109/LCOMM.2006.1665116
7. Chaudhry SA, Farash M, Naqvi H, Sher M (2015) A secure and efficient authenticated encryption for electronic payment systems using elliptic curve cryptography. Electron Commer Res 1–27. doi:10.1007/s10660-015-9192-5
8. Chaudhry SA, Naqvi H, Sher M, Farash MS, Hassan M (2015) An improved and provably secure privacy preserving authentication protocol for sip. Peer-to-Peer Netw Appl. doi:10.1007/s12083-015-0400-9
9. Dolev D, Yao AC (1983) On the security of public key protocols. IEEE Trans Inf Theory 29(2):198–208. doi:10.1109/TIT.1983.1056650
10. Eisenbarth T, Kasper T, Moradi A, Paar C, Salmasizadeh M, Shalmani M (2008) On the power of power analysis in the real world: a complete break of the keeloq code hopping scheme. In: Wagner D (ed) Advances in cryptology, CRYPTO 2008. Lecture notes in computer science, vol 5157, pp 203–220. Springer, Berlin. doi:10.1007/978-3-540-85174-5_12
11. Gao B, Shi Y, Yang C, Li L, Wang L, Yang Y (2014) Stp-lwe: a variant of learning with error for a flexible encryption. In: Mathematical problems in engineering
12. Guo C, Chang CC (2013) Chaotic maps-based password-authenticated key agreement using smart cards. Commun Nonlinear Sci Numer Simul 18(6):1433–1440
13. He D, Kumar N, Shen H, Lee JH (2015) One-to-many authentication for access control in mobile pay-tv systems. Sci China Inf Sci 1–14. doi:10.1007/s11432-015-5469-5
14. He D, Zeadally S, Kumar N, Lee JH (2016) Anonymous authentication for wireless body area networks with provable security. IEEE Syst J 99:1–12. doi:10.1109/JSYST.2016.2544805
15. He D, Zeadally S, Wu L (2015) Certificateless public auditing scheme for cloud-assisted wireless body area networks. IEEE Syst J 99:1–10. doi:10.1109/JSYST.2015.2428620
16. Huang HC, Fang WC, Lai WH (2012) Secure medical information exchange with reversible data hiding. In: 2012 IEEE International Symposium on Circuits and Systems (ISCAS), pp 1424–1427
17. Irshad A, Sher M, Faisal MS, Ghani A, Ul Hassan M, Ch SA (2013) A secure authentication scheme for session initiation protocol by using ECC on the basis of the Tang and Liu scheme. Security Comm Networks 7:1210–1218. doi:10.1002/sec.834
18. Jiang Q, Ma J, Lu X, Tian Y (2014) Robust chaotic map-based authentication and key agreement scheme with strong anonymity for telecare medicine information systems. J Med Syst 38(2):1–8
19. Kumari S, Chaudhry SA, Wu F, Li X, Farash MS, Khan MK (2015) An improved smart card based authentication scheme for session initiation protocol. Peer-to-Peer Netw Appl 1–15. doi:10.1007/s12083-015-0409-0
20. Li CT, Lee CC, Weng CY (2014) A secure chaotic maps and smart cards based password authentication and key agreement scheme with user anonymity for telecare medicine information systems. J Med Syst 38(9):1–11
21. Lin HY (2015) Improved chaotic maps-based password-authenticated key agreement using smart cards. Commun Nonlinear Sci Numer Simul 20(2):482–488
22. Lu Y, Li L, Peng H, Xie D, Yang Y (2015) Robust and efficient biometrics based password authentication scheme for telecare medicine information systems using extended chaotic maps. J Med Syst 39(6):1–10
23. Maro JC, Platt R, Holmes JH, Strom BL, Hennessy S, Lazarus R, Brown JS (2009) Design of a national distributed health data network. Ann Intern Med 151(5):341–344
24. Mir O, Nikooghadam M (2015) A secure biometrics based authentication with key agreement scheme in telemedicine networks for e-health services. Wirel Pers Commun 83(4):2439–2461
25. Mishra D, Das AK, Mukhopadhyay S (2014) A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards. Expert Syst Appl 41(18):8129–8143
26. Moon J, Choi Y, Kim J, Won D (2016) An improvement of robust and efficient biometrics based password authentication scheme for telecare medicine information systems using extended chaotic maps. J Med Syst 40(3):1–11. doi:10.1007/s10916-015-0422-0
27. Mostashari F, Tripathi M, Kendall M (2009) A tale of two large community electronic health record extension projects. Health Affairs 28(2):345–356
28. Niu Y, Wang X (2011) An anonymous key agreement protocol based on chaotic maps. Commun Nonlinear Sci Numer Simul 16(4):1986–1992

29. Özkaynak F, Yavuz S (2013) Designing chaotic s-boxes based on time-delay chaotic system. Nonlinear Dyn 74(3):551–557
30. People H (2013) Conclusion and future directions: CDC health disparities and inequalities report—United States, 2013. In: CDC Health Disparities and Inequalities Report—United States, 2013, vol 62(3), p 184
31. Privacy N (2008) Security framework for electronic exchange of individually identifiable health information. Office of the National Coordinator for Health Information Technology, US Department of Health and Human Services, p 15
32. Qazi MS, Ali M (2009) Pakistan's health management information system: health managers' perspectives. J Pak Med Assoc (JPMA) 59(1):10
33. Sinha PK, Sunder G, Bendale P, Mantri M, Dande A (2012) Electronic health record: standards, coding systems, frameworks, and infrastructures. Wiley, New York
34. Ts Z, Chu J, Araki K, Yoshihara H (2014) Design and development of an international clinical data exchange system: the international layer function of the dolphin project. pubmed commons. J Am Med Inf Assoc 18(5):683–689
35. Tseng HR, Jan RH, Yang W (2009) A chaotic maps-based key agreement protocol that preserves user anonymity. In IEEE International Conference on Communications, 2009, ICC'09, pp 1–6
36. Vest JR (2012) Health information exchange: national and international approaches. Adv Health Care Manag 12:3–24
37. Wei J, Hu X, Liu W (2012) An improved authentication scheme for telecare medicine information systems. J Med Syst 36(6):3597–3604
38. West DM, Friedman A (2012) Health information exchanges and megachange. In: Governance studies at Brookings
39. Xiao D, Liao X, Wong K (2005) An efficient entire chaos-based scheme for deniable authentication. Chaos Solitons Fractals 23(4):1327–1331
40. Xie Q, Hu B, Dong N, Wong DS (2014) Anonymous three-party password-authenticated key exchange scheme for telecare medical information systems. PloS One 9(7):e102,747
41. Xue K, Hong P (2012) Security improvement on an anonymous key agreement protocol based on chaotic maps. Commun Nonlinear Sci Numer Simul 17(7):2969–2977