J Med Syst (2018) 42: 219
https://doi.org/10.1007/s10916-018-1047-x

SYSTEMS-LEVEL QUALITY IMPROVEMENT

Privacy-Preserving and Efficient Truly Three-Factor Authentication Scheme for Telecare Medical Information Systems

Dongqing Xu[1] · Jianhua Chen[1] · Shu Zhang[1] · Qin Liu[2]

Received: 1 November 2017 / Accepted: 27 August 2018 / Published online: 2 October 2018

Abstract
Significant development of information technologies has made Telecare Medical Information Systems (TMISs) increasingly popular. In a TMIS, patients upload their medical data through smart devices to obtain a doctor's diagnosis. However, these smart devices have limited computing and storage capacities, so it is difficult to store substantial patient information and to support time-consuming operations. Moreover, although many three-factor authentication protocols have been proposed for TMISs, the problems of privacy leaks and other security flaws are serious. In addition, authentication factors are verified at the user side in most protocols, giving users a high level of trust and resulting in a potential lack of security. In this paper, we propose a novel efficient truly three-factor authentication protocol for TMISs. In our proposed protocol, three factors (i.e., password, smart card and biometrics) are verified at the server side, which reduces the storage and computational burden of the user side. Additionally, our proposed protocol uses only lightweight operators and is thus efficient. A formal proof analysis demonstrates that our proposed protocol is provably secure in the random oracle model. The performance evaluation shows that the proposed protocol is very efficient and suitable for TMISs.

Keywords Authentication scheme · Three-factor · Telecare medical information system · Random oracle model · Biometrics protection

Introduction

The electronic-health (e-health) system, which greatly facilitates people's life, is developing rapidly. Many applications, such as personal health records (PHRs) and Telecare Medical Information Systems (TMISs), have been developed to store medical records on the cloud and to diagnose patients remotely. TMISs, among the most popular applications of e-health systems, have received much attention from patients and doctors. In a basic TMIS, as shown in Fig. 1, some sensors are installed on the patient to collect medical data in a timely manner. These data are transferred to the medical server for storage and are used to build the patient's records. Then, the doctor can make an accurate diagnosis on the basis of the records. By means of the TMIS, patients are also able to receive good medical care at home without having to frequently rush to a hospital. Moreover, TMISs save time for patients in addition to saving many lives because doctors are able to diagnose patients in time.

However, the users (including doctors, patients, and relatives) of TMISs usually have limited computing and storage capabilities. A TMIS where the users are responsible for expensive and time-consuming computations is unsuitable for practical applications. Therefore, it is critical to minimize the user's operations in a TMIS.

In addition to the efficiency of the user side, privacy protection and security issues have become a major obstacle to the adoption of TMISs. Because of the openness of the Internet, adversaries can intercept and tamper with exchanged messages. As a result, TMISs face major challenges with respect to privacy protection and other security problems.

This article is part of the Topical Collection on *Systems-Level Quality Improvement*

✉ Qin Liu
qinliu@whu.edu.cn

[1] School of Mathematics and Statistics, Wuhan University, Wuhan, China

[2] Key Laboratory of Aerospace Information Security and Trusted Computing Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China

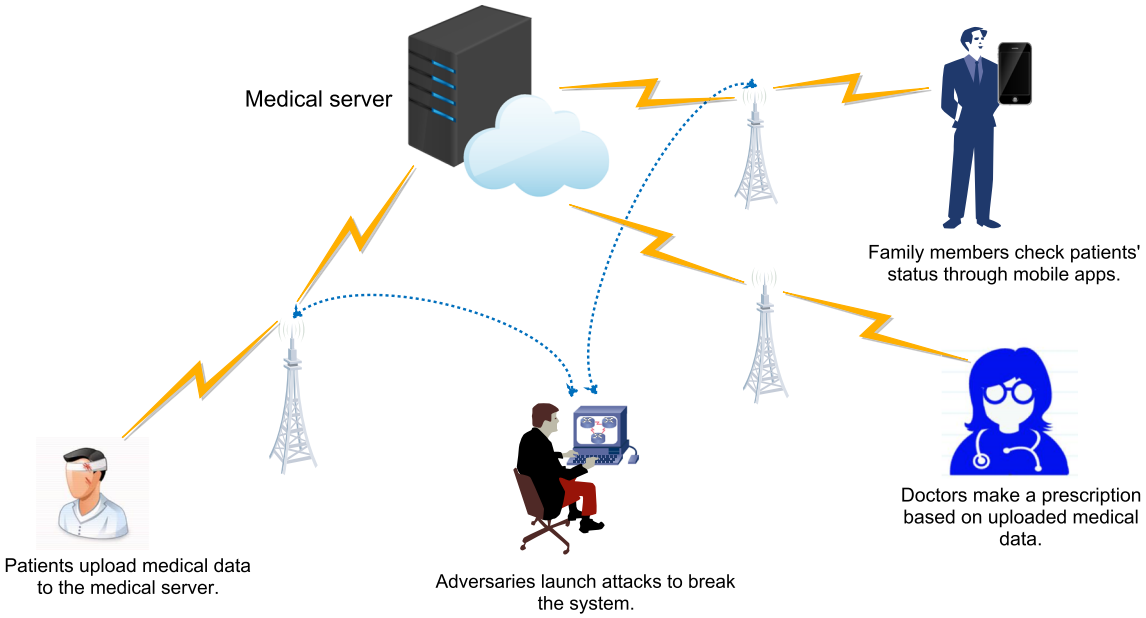

Fig. 1 A basic medical application scenario of a TMIS

The consequences of leaking medical data are serious. Imagine that a famous entrepreneur, whose health condition has a large impact on his/her company's market value, is accessing a TMIS to treat his/her disease at home. If an adversary (we will use $\mathcal{A}$ to represent an adversary) intercepts the entrepreneur's medical data, he/she is able to know the severity of the entrepreneur's illness. Then, the adversary has the ability to reveal the illness to the media for financial gains. This scenario shows that privacy protection is a critical consideration in a TMIS. Imagine another case where an adversary tampers with the medical data transmitted from a patient. Then, the doctor could make a false diagnosis that will lead to a serious medical accident.

Two-factor authentication protocols have been proposed to solve the privacy leakage and security weakness issues in TMISs [5, 12, 27, 29]. In this type of protocol, a user utilizes a password and a smart card as the two authentication factors. In addition, the user and the server authenticate each other, and an identical session key is shared between the user and the server at the end of the protocol. However, problems remain in two-factor authentication schemes. First, passwords and smart cards are easily lost or stolen. Second, authorized users can share passwords and smart cards with unauthorized users, and the system is hard to stop. Additionally, Wang et al. [24] demonstrated that two-factor authentication protocols cannot protect user privacy without using asymmetric cryptography. Wang et al. [24] also reported that if the verification information of the user's password is stored on the smart card, then the adversary can launch password guessing attacks and other attacks after extracting the data stored on the smart card via a side channel attack. However, authentication protocols using public key cryptography are not suitable for TMISs because of the expensive computing operations.

Biometrics have been introduced to solve the problems with two-factor authentication protocols. The combination of passwords, smart cards and biometrics can resist loss or theft of passwords and smart cards, so three-factor authentication schemes overcome some of the problems presented by two-factor authentication protocols. Therefore, many three-factor (i.e., password, smart card and biometrics) authentication protocols have been proposed [2, 20]. Some researchers built three-factor authentication protocols based on public key cryptography [11, 19, 26, 30, 31]. Although there is considerable improvement in security, these schemes use expensive computing operations are not suitable for users with limited computing power.

Another approach is to construct three-factor authentication protocols based on symmetric cryptography or hash functions [2, 6, 16–18, 21]. These protocols are efficient because they use only symmetric encryptions, hash functions, biohash functions and so forth. However, these three-factor authentication protocols face difficulties with protecting user privacy, including user's biometrics [2, 6, 16–18].

Fan et al. [8] proposed the notion of the truly three-factor authentication protocol. They considered that some malicious users may not comply with the protocol and not validate passwords and biometrics. Therefore, the three factors of the users should be verified at the server side rather than at the user side. In their proposed truly three-factor authentication protocol, the server is able to verify the three factors of the user to reduce the problem of user imitation and improve the security of the protocol.

Several three-factor authentication protocols authenticating user's factors in server side have been proposed [8, 32]. However, these protocols are either not secure [8] or not truly three-factor authentication protocols [32]. Thus, the challenge remains to propose a truly efficient and secure three-factor protocol for TMISs.

Our research contributions

In this paper, we propose a novel three-factor authentication scheme for TMISs. Our major research contributions are summarized as follows:

1. **Truly three-factor authentication scheme for TMISs.** The three factors (i.e., password, smart card and biometrics) are all authenticated at the server side in our scheme. In previous TMIS schemes, the server depends on the user to authenticate the three factors, which is a weak point of these schemes. By contrast, in our proposed scheme, the user's password and biometric template are stored on the server's table. The server checks whether the user's password and biometrics are valid each time the user logs in. Moreover, the user also sends a dynamic identity stored on the smart card to the server so that the server can verify the user's smart card. Therefore, our proposed scheme is the first truly three-factor authentication scheme for TMISs.
2. **Strong privacy protection.** In the proposed scheme, passwords and biometrics are stored on the server side and are protected by secure hash functions and random numbers. Thus, no one other than the user can obtain the user's passwords and biometrics. In terms of anonymity and untraceability, our proposed protocol adopts a dynamic identity mechanism. The adversary is unable to obtain the user's real identity and trace the user's actions because of this dynamic mechanism.
3. **Efficiency.** Our proposed scheme is efficient since the scheme is mainly based on hash functions, symmetric cryptosystems and some other efficient operators.
4. **Lightweight for users.** The user's operations are minimal in our proposed scheme. The user does not need to verify the three authentication factors (i.e., password, smart card and biometrics) and only needs to execute some lightweight operations, which increases the efficiency of the user side.
5. **Change passwords and biometrics freely.** In a truly three-factor authentication protocol, the password and biometric templates are stored at the server side instead of on the smart card, so it is difficult for users to change passwords and biometrics. By contrast, our proposed protocol is user-friendly, that is, users are able to change passwords and biometrics freely.

Organization of the paper

The remainder of this paper is organized as follows. In Related work, we review the related literature. Background and notation introduces background material, including the formal security model, TMIS security requirements, and notation used in the proposed protocol. The protocol and its security analysis are presented in The proposed protocol and Security analysis, respectively. In Security properties and performance analysis, we compare the performance of our scheme with that of related schemes. Conclusion concludes this paper.

Related work

Public key cryptography has been applied to the access control structure of e-health systems. Three-factor authentication protocols based on discrete logarithms [19] and chaotic mapping and bilinear pairings have been proposed [20, 23, 33]. Since elliptic curve cryptography (ECC) has advantages over other public key cryptosystems in terms of computation time and communication size, several ECC-based three-factor authentication protocols have also been proposed [25, 26, 31]. Although the security of three-factor authentication protocols based on public key cryptography is constantly improving [11], these protocols remain unsuitable for TMISs with limited computing power because power multiplication and point multiplication operations are time-consuming and unsuitable for resource-constrained environments, especially on the user side.

Authentication protocols mainly using lightweight computing operations, such as symmetric encryption and hash functions, were proposed to solve the problems of protocols based on public key cryptography. Li and Hwang [17] proposed an efficient three-factor authentication protocol using only hash functions. Li et al. [18], Das et al. [6] and An [2] also proposed such kind of key exchange scheme respectively. However, those protocols still have many security issues. For example, the protocol in Li and Hwang [17] was found to be unable to withstand man-in-the-middle attacks [18], and the protocol in [6] was found to be prone to insider attacks, password guessing attacks, and impersonation attacks [2].

Additionally, the protection of a user's biometrics in three-factor authentication protocols is also a problem. Passwords can be changed, but the user's biometrics are unique and cannot be altered. Once biometrics are leaked, the consequences are serious. Therefore, both efficiency and security should be considered in three-factor authentication schemes.

A variety of tools are available for managing biometrics in authentication protocols, such as fuzzy commitments [15], fuzzy vaults [13, 22], error-correcting [9], fuzzy

extractors [7, 14], and biohash functions [32]. Among these tools, the biohash function has one-way properties similar to hash functions and can conveniently handle biometric errors. Therefore, biohash functions, which can take into account both efficiency and security, are considered to be suitable for managing biometrics.

Several three-factor authentication protocols based on biohash functions have recently been proposed [1, 4, 10, 21, 32]. Amongst them, Chaudhry et al. [4] reported that Mir et al.'s protocol [21] cannot resist stolen smart card attacks and that user anonymity violation attacks are still possible [21]. Chaudhry et al. [4] then proposed an improved protocol. Additionally, the protocol in [10], which combines biohash functions with public key cryptography, is a useful exploration for designing authentication protocols in TMISs.

Fan et al. [8] proposed the concept of the truly three-factor authentication protocol. They believed that the three factors should be verified at the server side rather than at the user side, which is potentially insecure for protocols since the traditional verification at the user side gives a high level of trust to the user. However, dishonest users may not authenticate these factors and directly communicate with the server. In this case, the server cannot determine whether he/she is communicating with an honest user and has no idea whether the verification at the user side is executed correctly. However, a truly three-factor authentication protocol is not easy to achieve. Verifying the three factors at the server side means that the user's password and biometrics or related values must be stored at the server side. Therefore, it is difficult to ensure that the server does not know the user's password and biometrics or related values. The scheme proposed by the authors had several disadvantages. For example, Yeh et al. [31] reported that Fan et al.'s scheme is unable to resist insider attacks.

Zhang et al. [32] recently proposed a three-factor authentication scheme with key agreement based on biohash functions. In their scheme, the user's biometrics are verified at the server side, whereas the password and smart card are not. Thus, it is not a truly three-factor authentication scheme. Additionally, after receiving a login request from a user, the server has to determine whether there is a value W in the database that is equal to user's message, which reduces the efficiency of their scheme. Additionally, Zhang et al.'s protocol [32] is not user-friendly because it does not provide a password and biometrics change service.

Background and notation

Security model

In this subsection, we extend previously reported security models [3, 28, 32] for three-factor authentication protocols. Our improved model defines a $TestID(\cdot)$ oracle to capture the notion of user anonymity. In addition, we define a $CorruptSC(\cdot)$ oracle to simulate stolen smart card attacks, and we define a $CorruptDB(\cdot)$ oracle to simulate stolen verifier table attacks.

Participants Two types of participants exist in protocol $\mathcal{P}$, i.e., users $U_i \in User$ and servers $S_j \in Server$. U_i^a (resp. S_j^a) represents the a^{th} instance of U_i (resp. S_j). acc_U^i is a Boolean variable that indicates whether U_i accepts. $acc_U^i = 1$ indicates that U_i accepts, and $acc_U^i = 0$ indicates that U_i rejects.

Partnering In the protocol, the session identifier in each session is unique. If U_i and S_j share the same non-null session identifier, then U_i and S_j are partnered.

Adversary ability To simulate a realistic adversary, the adversary can query the following oracles:

- $Execute(U_i^a, S_j^b)$: This query simulates eavesdropping attacks, where an adversary obtains session messages between U_i^a and S_j^b.
- $Hash(m, h(m))$: In this query, the oracle searches for the existence of $(m, h(m))$ in the hash list. If it exists, the oracle returns $(m, h(m))$ for the adversary; otherwise, the oracle selects a random string k to return to the adversary and then stores (m, k) in the list.
- $Biohash(m, h_{Bio}(m))$: In this query, the oracle compares m and m^*, which is in the biohash list, to determine whether the difference between m and m^* is within a tolerable range. If yes, the oracle returns $h_{Bio}(m)$ to the adversary; otherwise, the oracle selects a random string r for the adversary and stores (m, r) in the list.
- $Send(U_i^a / S_j^b, m)$: This query simulates active attacks, where an adversary can obtain the corresponding messages according to the execution of the protocol.
- $Reveal(U_i^a)$: This query simulates known session key attack, where an adversary obtains the session key held by U_i^a.
- $CorruptSC(U_i^a)$: This query simulates stolen smart card attacks, where the adversary obtains the information stored on the smart card.
- $CorruptDB(S_j^b)$: This query simulates stolen verifier table attacks, where the adversary is able to obtain information stored on the verifier table at the server side.
- $TestID(U_i^a)$: This query tests user anonymity and can only be asked for once. In this query, the oracle randomly tosses a coin b. If $b = 1$, then the adversary obtains the user's real identity. If $b = 0$, then the adversary obtains a random element in the identity space.

- $Test(U_i^a/S_j^b)$: By throwing a coin b, this query captures the notion of the semantic security of the session key. When $b = 1$, the adversary obtains the session key; when $b = 0$, the adversary obtains a random string with the same bit length as the session key.

Freshness U_i^a is fresh if:

1. acc_U^i=1.
2. No $Reveal(U_i^a/S_j^b)$ is queried by adversary $\mathcal{A}$.

AKE Security (Semantic Security) In the $Test$ query, the adversary $\mathcal{A}$ outputs his/her guess b'. If $b' = b$, then $\mathcal{A}$ wins the game. The advantage for $\mathcal{A}$ to break the protocol is defined as:

$$Adv_P^{ake} = 2Pr[b' = b] - 1.$$

We say that a protocol $\mathcal{P}$ is AKE secure if Adv_P^{ake} is negligible.

Security requirements

A truly three-factor authentication scheme for TMISs should satisfy the following security requirements.

User anonymity: To protect user privacy, the protocol should ensure the anonymity of all the users, i.e., an adversary should be unable to obtain the users' real identities from the exchanged messages.

Untraceability: To provide greater security for user privacy, the protocol should support untraceability, i.e., an adversary should be unable to track users' behaviors from the exchanged messages.

Mutual authentication: The protocol should support mutual authentication, i.e., the user U_i and the medical server S in a TMIS should be able to authenticate each other.

Session key agreement: In the protocol, the user U_i and the medical server S should share an identical session key for further communication.

Known key security: Even if the adversary obtains the current session key, he/she should be unable to know the previous session keys.

Perfect forward secrecy: Even if the adversary obtains the long secret keys of the participants, he/she should not be able to acquire the current and previous session keys.

Three-factor secrecy: To guarantee the security of a user's private keys, the scheme should permit three-factor (i.e., password, smart card and biometrics) secrecy, i.e., an adversary should be unable to mimic a legal user even if he/she obtains any two of the three factors.

Biometrics protection: The scheme should ensure that the user's biometrics cannot be compromised.

Resistance to various attacks: To maintain security in open networks, the scheme should resist stolen verifier attacks, privileged insider attacks, user impersonation attacks, server spoofing attacks, replay attacks, and de-synchronization attacks.

Notation

Table 1 presents the notation used in the proposed scheme.

The proposed protocol

In this section, we present a concrete construction of the proposed protocol. Our proposed scheme consists of five phases, i.e., the registration, login, mutual authentication, password and biometrics update, and smart card revocation.

Registration phase

To become a legitimate user in the system, a new user registers with the medical server by performing the following three steps, as shown in Table 2.

1. User U_i selects ID_i and PW_i as his/her identity and password, respectively. Then, U_i collects his/her biometrics T_i at the sensor and generates a random

Table 1 Notation

Notation	Description
U_i	The i^{th} user (could be a patient, relative, or doctor) who participates in a phase
S	The medical server
SC	The smart card
ID_i	The identity of U_i
ID_{sc}	The identity of the smart card SC
DID_1, DID_2, DID_3	The dynamic identity of U_i
PW_i	Passwords of U_i
x	The private key of S
T_i	The biometric template of U_i
B_i, B_{i1}, B_{i2}	The biometric data of U_i
N_1, N_2, N_3	Random numbers
r_1, r_2, r_3, r_4	Random numbers
$E_x(\cdot)$	Symmetric encryption with x
$D_x(\cdot)$	Symmetric decryption with x
$h(\cdot)$	A one-way hash function
$h_{Bio}(\cdot)$	A biohash function
$\stackrel{?}{=}$	Relational comparison operator
$\|$	Concatenation of two strings
$\oplus$	Bitwise exclusive-OR operation

Table 2 Registration phase of our scheme

U_i	S
Chooses ID_i, PW_i	
Inputs T_i	
Generates r_1	
$C_1 = h(ID_i \| \| PW_i \| \| h_{Bio}(T_i))$	
$C_2 = T_i \oplus h(PW_i) \oplus r_1$	
$\xrightarrow{< ID_i, C_1, C_2 >}$	
	$M = h(h_{Bio}(C_2) \| \| x)$
	$Y = M \oplus C_1$
	$ID = ID_i \| \| ID_{sc}$
	Chooses N_1
	$DID_1 = E_x(ID \| \| N_1)$
	Stores $\{ID, C_2\}$ in database
	Embeds $\{ID_{sc}, h(\cdot), h_{Bio}(\cdot), Y, DID_1\}$ into SC
	$\xleftarrow{< SC >}$
$Z = r_1 \oplus h_{Bio}(T_i) \oplus h(PW_i)$	
Stores Z into SC	

Table 3 Login phase of our scheme

U_i/SC	S
Inputs ID_i, PW_i, B_i	
$C_1^* = h(ID_i \| \| PW_i \| \| h_{Bio}(B_i))$	
$M^* = Y \oplus h(C_1^*)$	
$r_1^* = Z \oplus h_{Bio}(B_i) \oplus h(PW_i)$	
Generates r_2	
$C_3 = r_2 \oplus h_{Bio}(B_i \oplus h(PW_i) \oplus r_1^*)$	
$C_4 = B_i \oplus h(PW_i) \oplus r_1^* \oplus h(M^* \| \| r_2)$	
	$\xrightarrow{m_1 =< DID_1, C_3, C_4 >}$

nonce r_1. Subsequently, U_i computes $C_1 = h(ID_i \| \| PW_i \| \| h_{Bio}(T_i))$ and $C_2 = T_i \oplus h(PW_i) \oplus r_1$. Then, U_i sends his/her registration request $< ID_i, C_1, C_2 >$ to the medical server S via a secure channel.

2. Upon obtaining the registration request from U_i, S calculates $M = h(h_{Bio}(C_2) \| \| x)$, $Y = M \oplus C_1$, and $ID = ID_i \| \| ID_{sc}$. Then, S generates a random number N_1 and computes $DID_1 = E_x(ID \| \| N_1)$. Subsequently, S stores $\{ID, C_2\}$ in a database for further verification processes. S also embeds $\{ID_{sc}, h(\cdot), h_{Bio}(\cdot), Y, DID_1\}$ into a smart card SC. Then, S delivers the smart card SC to U_i through a secure channel.
3. After receiving the smart card SC, U_i calculates $Z = r_1 \oplus h_{Bio}(T_i) \oplus h(PW_i)$ and writes Z into the smart card SC. Finally, the smart card SC contains parameters $\{ID_{sc}, h(\cdot), h_{Bio}(\cdot), Y, DID_1, Z\}$.

Login phase

If U_i wants to access services provided by the medical server S, he/she needs to perform the following steps to login to S. Table 3 shows details of the user login process.

1. The user U_i inputs ID_i, PW_i, and his biometric information B_i into the terminal device. U_i then inserts the smart card into the terminal card reader.
2. The smart card computes $C_1^* = h(ID_i \| \| PW_i \| \| h_{Bio}(B_i))$, $M^* = Y \oplus h\left(C_1^*\right)$, and $r_1^* = Z \oplus h_{Bio}(B_i) \oplus h(PW_i)$.
3. For the login request message, the smart card generates a random nonce r_2 and calculates $C_3 = r_2 \oplus h_{Bio}(B_i \oplus h(PW_i) \oplus r_1^*)$ and $C_4 = B_i \oplus h(PW_i) \oplus r_1^* \oplus h(M^* \| \| r_2)$. Then, U_i sends $m_1 =< DID_1, C_3, C_4 >$ to the medical server S.

Mutual authentication phase

In this phase, U_i and S authenticate each other and then negotiate a session key for the next communication. Table 4 shows details of the mutual authentication process between a user and a medical server.

1. After receiving the login request from U_i, S decrypts DID_1 to obtain $ID' \| \| N_1'$. Then, S checks whether ID' is in the database to verify the validation of U_i. If ID' is valid, S retrieves C_2 to compute $M' = h(h_{Bio}(C_2) \| \| x)$, $r_2' = C_3 \oplus h_{Bio}(C_2)$, and $(B_i \oplus h(PW_i) \oplus r_1)' = C_4 \oplus h(M' \| \| r_2')$. Subsequently, S checks whether the difference between $(B_i \oplus h(PW_i) \oplus r_1)'$ and C_2 is less than a bearable threshold [32][33]. If it is not, S terminates the session; otherwise, S authenticates U_i and continues to the next step.
2. S generates a random nonce N_2 and calculates $DID_2 = E_x(ID \| \| N_2)$. Then, S chooses another random nonce r_3 and computes $C_5 = (r_3 \| \| DID_2) \oplus h((B_i \oplus h(PW_i) \oplus r_1)')$, $SK_{su} = h(M' \| \| r_2' \| \| r_3)$, and $Auth_1 = h((B_i \oplus h(PW_i) \oplus r_1)' \| \| SK_{su} \| \| r_2' \| \| r_3)$ (we assume that the output of $h((B_i \oplus h(PW_i) \oplus r_1)')$ has the same bit length with $(r_3 \| \| DID_2)$. And $h(\cdot)$ in the protocol represents a family of hash functions, so its output length could be different). Subsequently, S sends $m_2 =< C_5, Auth_1 >$ to U_i.
3. After obtaining message m_2 from S, U_i computes $r_3^* \| \| DID_2^* = C_5 \oplus h(B_i \oplus h(PW_i) \oplus r_1^*)$ and $SK_{us} = h(M^* \| \| r_2 \| \| r_3^*)$. Then, U_i verifies whether $Auth_1$ is equal to $h(B_i \oplus h(PW_i) \oplus r_1^* \| \| SK_{us} \| \| r_2 \| \| r_3^*)$. If this verification does not hold, U_i aborts the session; otherwise, U_i authenticates the medical server S. U_i

Table 4 Mutual authentication phase of our scheme

U_i/SC	S
$\xrightarrow{m_1 =< DID_1, C_3, C_4 >}$	
	$D_x(DID_1) = ID'\|\|N_1'$,
	$M' = h(h_{Bio}(C_2)\|\|x)$,
	$r_2' = C_3 \oplus h_{Bio}(C_2)$,
	$(B_i \oplus h(PW_i) \oplus r_1)' = C_4 \oplus h(M'\|\|r_2')$,
	Compare $(B_i \oplus h(PW_i) \oplus r_1)'$ with C_2,
	Chooses N_2,
	$DID_2 = E_x(ID\|\|N_2)$,
	Chooses r_3,
	$C_5 = (r_3\|\|DID_2) \oplus h((B_i \oplus h(PW_i) \oplus r_1)')$,
	$SK_{su} = h(M'\|\|r_2'\|\|r_3)$,
	$Auth_1 = h((B_i \oplus h(PW_i) \oplus r_1)'\|\|SK_{su}\|\|r_2'\|\|r_3)$,
	$\xleftarrow{m_2 =< C_5, Auth_1 >}$
$r_3^*\|\|DID_2^* = C_5 \oplus h(B_i \oplus h(PW_i) \oplus r_1^*)$,	
$SK_{us} = h(M^*\|\|r_2\|\|r_3^*)$,	
$Auth_1 \stackrel{?}{=} h(B_i \oplus h(PW_i) \oplus r_1^*\|\|SK_{us}\|\|r_2\|\|r_3^*)$,	
$Auth_2 = h(ID\|\|(B_i \oplus h(PW_i) \oplus r_1^*)\|\|SK_{us}\|\|r_2\|\|r_3^*)$,	
$\xrightarrow{m_3 =< Auth_2 >}$	
	$Auth_2 \stackrel{?}{=} h(ID\|\|(B_i \oplus h(PW_i) \oplus r_1)'\|\|SK_{su}\|\|r_2'\|\|r_3)$

also accepts the session key SK_{us} and the dynamic identity DID_2^* for the next login. Then, U_i computes $Auth_2 = h(ID||(B_i \oplus h(PW_i) \oplus r_1^*)||SK_{us}||r_2||r_3^*)$ and launches $m_3 =< Auth_2 >$ to S.

4. Upon receiving message m_3 from U_i, S verifies whether $Auth_2$ is equal to $h(ID||(B_i \oplus h(PW_i) \oplus r_1)'||SK_{su}||r_2'||r_3)$. If this verification condition does not hold, S aborts the session; otherwise, S ensures that message m_3 comes from U_i. Finally, S accepts the session key SK_{su}.

Remark DID_1 and ID_{sc} are two variables embedded into the user's smart card by the server. When the user logs in, the server's verification of DID_1 represents the server's authentication of the user's smart card. Without DID_1, the server would not consider the login request to be a valid login request message from U_i. Moreover, the server's verification of $Auth_2$, which contains ID_{sc}, also indicates the server's authentication of the user's smart card.

In our proposed protocol, the server simultaneously, rather than separately, authenticates both the user's passwords and biometrics. If the difference between $(B_i \oplus h(PW_i) \oplus r_1)'$ and C_2 is less than an acceptable threshold, the server accepts the passwords and biometrics contained in the login message sent by the user.

Passwords and biometrics update phase

A user U_i is able to update his/her passwords and biometrics during this phase, which is conducted with the help of the medical server since the password and biometrics template is stored at the server side.

1. First, U_i inserts the smart card SC into the card reader of the terminal device. Then, U_i enters ID_i, PW_i, and B_{i1} on the terminal device. U_i also inputs new password PW_i^{new} and new biometrics B_{i2} to replace the original password and biometrics stored at the server side.
2. The smart card SC calculates $C_1^{**} = h(ID_i||PW_i||h_{Bio}(B_{i1}))$, $M^{**} = Y \oplus h\left(C_1^{**}\right)$, and $r_1^{**} = Z \oplus h_{Bio}(B_{i1}) \oplus h(PW_i)$.
3. For the update message, the smart card SC chooses a random number r_4 and computes $C_6 = r_4 \oplus h_{Bio}(B_{i1} \oplus h(PW_i) \oplus r_1^{**})$, $C_7 = B_{i1} \oplus h(PW_i) \oplus r_1^{**} \oplus h(M^{**}||r_4)$, and $C_8 = B_{i2} \oplus h(PW_i^{new}) \oplus r_1^{**} \oplus h(M^{**}||r_4)$. Then, U_i sends $< DID_2, C_6, C_7, , C_8 >$ to S.
4. After obtaining the update message from U_i, S computes $D_x(DID_2) = ID''||N_2'$. Then, S checks whether ID'' is in the database to verify the validity of U_i. If ID'' is valid, S retrieves C_2 to calculate $M'' = h(h_{Bio}(C_2)||x)$, $r_4' = C_6 \oplus h_{Bio}(C_2)$, $(B_{i1} \oplus h(PW_i) \oplus r_1)' = C_7 \oplus h(M''||r_4')$, and $(B_{i2} \oplus h(PW_i^{new}) \oplus r_1)' = C_8 \oplus h(M''||r_4')$. Subsequently, S checks whether the difference between $(B_{i1} \oplus h(PW_i) \oplus r_1)'$ and C_2 is less than an acceptable threshold [32][33]. If it is not, S aborts the update request; otherwise, it proceeds to the next step.
5. S selects a random number N_3 and computes $DID_3 = E_x(ID||N_3)$. Then, S generates another random number r_5 and calculates $C_9 = (r_5||DID_3) \oplus h((B_{i2} \oplus h(PW_i^{new}) \oplus r_1)')$ and $Auth_3 = h((B_{i2} \oplus h(PW_i^{new}) \oplus r_1)'||r_4'||r_5)$ (we assume that the output of $h((B_{i2} \oplus h(PW_i^{new}) \oplus r_1)')$ has the same bit length with $(r_5||DID_3)$ as above). Subsequently, S sends $< C_9, Auth_3 >$ to U_i.
6. Upon obtaining the message $< C_9, Auth_3 >$ from S, U_i calculates $r_5^*||DID_3^* = C_9 \oplus h(B_{i2} \oplus h(PW_i^{new}) \oplus r_1^{**})$. Then, U_i verifies whether $Auth_3$ is equal to $h(B_{i2} \oplus h(PW_i^{new}) \oplus r_1^{**}||r_4||r_5^*)$. If this verification condition does not hold, U_i terminates the update request; otherwise, U_i calculates $Auth_4 = h(ID||(B_{i2} \oplus h(PW_i^{new}) \oplus r_1^{**})||r_4||r_5^*)$ and sends $< Auth_4 >$ to S.

7. After receiving $< \ Auth_4 \ >$ from U_i, S verifies whether $Auth_4$ is equal to $h(ID||(B_{i2} \oplus h(PW_i^{new}) \oplus r_1)'||r_4'||r_5)$. If this verification condition does not hold, S terminates the update request; otherwise, S assures that the message $< \ Auth_4 \ >$ comes from the user U_i and then replaces C_2 with $(B_{i2} \oplus h(PW_i^{new}) \oplus r_1)'$ for further verification of U_i, which completes the password and biometrics update phase.

Smart card revocation

If U_i loses his/her smart card, he/she can use their ID_i to apply to the server for a new smart card. This phase is the same as the user registration phase. The user utilizes the same ID_i as before, and selects new passwords PW_i, new biometrics T_i, and new random number r_1, to calculate $C_1 = h(ID_i||PW_i||h_{Bio}(T_i))$ and $C_2 = T_i \oplus h(PW_i) \oplus r_1$. Then, the user sends registration request $< ID_i, C_1, C_2 >$ to the medical server S via a secure channel. The subsequent steps are the same as those in the registration phase. When issuing a new smart card to the user, the server also changes the value of C_2 corresponding to the ID_i in the database. Therefore, even if an adversary uses the lost smart card to login, he/she cannot be authenticated by the server.

Security analysis

Formal security proof

Theorem 1 *Assume that $\mathcal{A}$ executes $Execute$ queries q_{exe} times, $Send$ queries q_{send} times, $Hash$ queries q_h times, and $Biohash$ queries q_b times to breach the AKE security of the protocol. Let $|H_1|$, $|H_b|$, and $|T|$ represent the distribution space of the hash function, the distribution space of the biohash function, and the number of items in the server's table, respectively. q_t denotes the number of $\mathcal{A}$'s guessing attempt towards the server. Let D_P and D_B represent the distribution spaces of user's password and biometrics, respectively. $|D_P|$ and $|D_B|$ represent the size of D_P and D_B, respectively, and $|D_B|$ is much larger than $|D_P|$. n represents the total number of times that a user might log in, and the server's secret key x has length of l bits. In addition, $Adv_{E_x}^{SE}(t)$ represents the advantage for the adversary to break the symmetric encryption in time t. Then, we have*

$$Adv_P^{ake}(\mathcal{A}) \leq 2Adv_{E_x}^{SE}(t) + \frac{q_h^2}{|H_1|} + \frac{q_b^2}{|H_b|} + \frac{(q_{send}+q_{exe})^2}{n} + \frac{2q_{send}}{n} + max\{\frac{2q_{send}}{|D_P||D_B|}, \frac{q_t}{2^{l-1} \cdot |T|}\}.$$

Proof We create a series of games defined as G_i $(0 \leq i \leq 4)$ to show that the proposed protocol is provably secure. In each game G_i, Suc_i denotes the event that $\mathcal{A}$ guesses b correctly in the $Test$ query.

$GameG_0$: G_0 is real attacks in the random oracle model. Thus, we have

$$Adv_P^{ake}(\mathcal{A}) = 2Pr[Suc_0] - 1. \quad (1)$$

$GameG_1$: In this game, we simulate $Hash$ oracles and $Biohash$ oracles by maintaining the hash lists. The other oracles, including $Execute$, $Send$, $Reveal$, $CorruptSC$, $CorruptDB$, $TestID$, and $Test$, are the same as those in the original attacks. If the adversary makes the correct judgement in the $TestID$ query, then the adversary can break the symmetric encryption with the same probability. Thus, we have

$$Pr[Suc_1] - Pr[Suc_0] \leq Adv_{E_x}^{SE}(t). \quad (2)$$

$GameG_2$: G_2 is the same as the previous game except that the game is terminated if the message transcripts and hash queries have collisions. The probability of collisions on message transcripts is $\frac{(q_{send}+q_{exe})^2}{2n}$, and the probability of collisions on hash queries is $\frac{q_h^2}{2|H_1|} + \frac{q_b^2}{2|H_b|}$, according to the birthday paradox. Therefore, we have

$$Pr[Suc_2] - Pr[Suc_1] \leq \frac{q_h^2}{2|H_1|} + \frac{q_b^2}{2|H_b|} + \frac{(q_{send}+q_{exe})^2}{2n}. \quad (3)$$

$GameG_3$: G_3 is the same as the previous game, but it is rejected if $\mathcal{A}$ guesses $Auth_i$, $Auth_j$, and $Auth$ without querying the corresponding $Hash$ oracle $h(\cdot)$. Hence, we obtain

$$Pr[Suc_3] - Pr[Suc_2] \leq \frac{q_{send}}{n}. \quad (4)$$

$GameG_4$: In this game, $\mathcal{A}$ queries the $CorruptSC$ oracle or $CorruptDB$ oracle. Thus, the following two cases exist:

$Case1$: In this case, $\mathcal{A}$ queries $CorruptSC$ and obtains parameters stored on the smart card. Then, $\mathcal{A}$ launches dictionary attacks with possible passwords and biometrics. Thus, we obtain:

$$Pr[Suc_4] - Pr[Suc_3] \leq \frac{q_{send}}{|D_P||D_B|}. \quad (5)$$

$Case2$: In this case, $\mathcal{A}$ queries $CorruptDB$ and then obtains parameters stored in the verifier table. After receiving the table $\{ID, C_2\}$, $\mathcal{A}$ tries all C_2 to launch an online dictionary attack through computing $M' = h(h_{Bio}(C_2)||x)$ and $C_4 = C_2 \oplus h(M'||r_2')$, where r_2' is random nonce. $\mathcal{A}$ obtains C_3 through the Biohash oracle, and $\mathcal{A}$ also queries $Send(S, \{DID_1, C_3, C_4\})$. $\mathcal{A}$ then

uses an l-bit random value to replace the server's secret key x. Therefore, we have:

$$Pr[Suc_4] - Pr[Suc_3] \leq \frac{q_t}{2^l \cdot |T|}. \quad (6)$$

□

Since $\mathcal{A}$ is unable to query $CorruptSC$ and $CorruptDB$ simultaneously, $\mathcal{A}$ can choose case 1 or case 2 as the final game G_3. In G_3, $\mathcal{A}$ queries the $Test$ oracle and guesses b. Thus, we have:

$$Pr[Suc_3] = \frac{1}{2}. \quad (7)$$

If $\mathcal{A}$ chooses case 1, from Eqs. 1, 2, 3, 4, 5 and 7, we obtain:

$$Adv_P^{ake}(\mathcal{A}) \leq 2Adv_{E_x}^{SE}(t) + \frac{q_h^2}{|H_1|} + \frac{q_b^2}{|H_b|} + \frac{(q_{send} + q_{exe})^2}{n} + \frac{2q_{send}}{n} + \frac{2q_{send}}{|D_P||D_B|}.$$

If $\mathcal{A}$ chooses case 2, from Eqs. 1, 2, 3, 4, 6 and 7, we obtain:

$$Adv_P^{ake}(\mathcal{A}) \leq 2Adv_{E_x}^{SE}(t) + \frac{q_h^2}{|H_1|} + \frac{q_b^2}{|H_b|} + \frac{(q_{send} + q_{exe})^2}{n} + \frac{2q_{send}}{n} + \frac{q_t}{2^{l-1} \cdot |T|}.$$

Therefore, we have:

$$Adv_P^{ake}(\mathcal{A}) \leq 2Adv_{E_x}^{SE}(t) + \frac{q_h^2}{|H_1|} + \frac{q_b^2}{|H_b|} + \frac{(q_{send} + q_{exe})^2}{n} + \frac{2q_{send}}{n} + max\{\frac{2q_{send}}{|D_P||D_B|}, \frac{q_t}{2^{l-1} \cdot |T|}\}. \quad (8)$$

Theorem 1 is proven by Eq. 8.

Analysis of security requirements

In this subsection, we will demonstrate that the proposed protocol satisfies all the security requirements listed in Section 2.

User anonymity : In the protocol, U_i's identity ID_i is included in DID_1, where $DID_1 = E_x(ID||N_1)$. The adversary is unable to obtain the real identity ID_i from the intercepted information. Therefore, our scheme provides user anonymity.

Untraceability : In the protocol, the login request message $< DID_1, C_3, C_4 >$ is different each time. U_i receives a different DID_x for each login, and U_i generates a random nonce to compute C_3 and C_4 each time. Moreover, $Auth_2$ is a random hash string in each execution of the protocol. Thus, $\mathcal{A}$ is unable to trace a user's actions from the public messages. Therefore, our protocol provides untraceability for users.

Mutual authentication: In the proposed protocol, only the user with the correct DID_x can be authenticated by S. Moreover, only the medical server with the right secret value x can handle the login request message and respond with right challenge message $< C_5, Auth_1 >$. Therefore, our proposed scheme provides mutual authentication.

Session key agreement: In the proposed protocol, U_i computes the session key $SK_{us} = h(M^*||r_2||r_3^*)$, and S calculates the session key $SK_{su} = h(M'||r_2^*||r_3)$. Hence, U_i and S share an identical session key $SK_{us} = SK_{su}$. Therefore, our proposed protocol provides session key agreement.

Known key security: In the proposed scheme, U_i and S share the session key $SK_{us} = SK_{su} = h(M^*||r_2||r_3^*)$ at the end of the mutual authentication scheme. To compute the session key, U_i chooses a random number r_2, and S chooses a random number r_3. These numbers are random and different each time. Thus, even if $\mathcal{A}$ obtains a session key, he/she still does not know the other session keys. Therefore, our proposed scheme provides known key security.

Perfect forward secrecy: In the proposed scheme, even if $\mathcal{A}$ obtains the medical server's secret value x, he/she still does not have M to compute the session key $SK_{us} = SK_{su} = h(M^*||r_2||r_3^*)$. Additionally, $\mathcal{A}$ also cannot obtain r_2 and r_3 to compute the session key. Therefore, our proposed scheme provides perfect forward secrecy.

Three-factor secrecy: Three-factor secrecy means that even if $\mathcal{A}$ obtains any two of the three factors, he/she still cannot mimic a legal user. To generate a correct login request message as a valid user, $\mathcal{A}$ has to compute $C_4 = B_i \oplus h(PW_i) \oplus r_1^* \oplus h(M^*||r_2)$. Here, we demonstrate that even if $\mathcal{A}$ acquires any two of the three factors, he/she still cannot obtain M to compute C_4, which is an essential part of the login request message.

$Case1$: In this case, we assume that $\mathcal{A}$ obtains U_i's password and smart card. To compute $M^* = Y \oplus h(C_1^*)$, $\mathcal{A}$ needs to obtain C_1^*. However, without B_i, $\mathcal{A}$ is unable to calculate $C_1^* = h(ID_i||PW_i||h_{Bio}(B_i))$. Thus, in this case, $\mathcal{A}$ cannot successfully forge a login request message.

$Case2$: In this case, we assume that $\mathcal{A}$ obtains U_i's password and biometrics. To compute $M^* = Y \oplus h(C_1^*)$, $\mathcal{A}$ needs to obtain Y. However, without the smart card, $\mathcal{A}$ is unable to acquire Y, which is stored on the smart card. Thus, in this case, $\mathcal{A}$ cannot successfully mimic a legal user to forge a login request message.

$Case3$: In this case, we assume that $\mathcal{A}$ obtains U_i's biometrics and smart card. To calculate $M^* = Y \oplus h(C_1^*)$, $\mathcal{A}$ needs to acquire C_1^*. However, without PW_i, $\mathcal{A}$ is unable to calculate $C_1^* = h(ID_i||PW_i||h_{Bio}(B_i))$. Thus, in this case, $\mathcal{A}$ cannot successfully forge a login request message.

These three cases demonstrate that our scheme provides three-factor secrecy.

Biometrics protection: In the proposed scheme, we guarantee that biometrics are possessed only by the user himself and that $\mathcal{A}$ cannot obtain the user's biometrics. On the user side, $Z = r_1 \oplus h_{Bio}(T_i) \oplus h(PW_i)$ is stored on the smart card. The biometric template T_i is protected by secure hash functions and random number r_1. Thus, $\mathcal{A}$ is unable to obtain the user's biometric template T_i. On the server side, $C_2 = T_i \oplus h(PW_i) \oplus r_1$ is stored in the server's database. The biometric template T_i is also protected by secure hash functions and random number r_1. Hence, $\mathcal{A}$ cannot obtain the user's biometric template T_i at the server side. Therefore, our protocol provides biometrics protection.

Resistance to stolen verifier attacks: In the proposed scheme, ID and C_2 are stored on the verifier. If the verifier is stolen by $\mathcal{A}$, $\mathcal{A}$ still cannot successfully launch attacks. $\mathcal{A}$ is unable to trace a user's actions with only ID. Moreover, $\mathcal{A}$ is unable to mimic a medical server with only C_2. Additionally, $\mathcal{A}$ cannot obtain the user's biometric template T_i from $C_2 = T_i \oplus h(PW_i) \oplus r_1$, as demonstrated above. Therefore, our proposed scheme is resistant to stolen verifier attacks.

Resistance to privileged insider attacks: In authentication schemes, the privileged insider may somehow obtain ID_i, C_1, ID, and C_2. With ID_i and C_1, $\mathcal{A}$ can do nothing. We have demonstrate that even if $\mathcal{A}$ acquires ID and C_2, the scheme is still secure. Therefore, our proposed scheme is able to resist privileged insider attacks.

Resistance to user impersonation attacks: To mimic a legal user, $\mathcal{A}$ needs to calculate $C_4 = B_i \oplus h(PW_i) \oplus r_1^* \oplus h(M^*||r_2)$. We have demonstrated that even if $\mathcal{A}$ obtains any two of the three factors, he/she still cannot compute C_4 to forge a correct login request message to mimic a user. Therefore, our proposed scheme is free from user impersonation attacks.

Resistance to server spoofing attacks: In the proposed scheme, S has its own secret key x to handle the login request message and generate a correct challenge message. $\mathcal{A}$ is unable to mimic a legal medical server without the secret key x. Hence, our proposed scheme is free from server spoofing attacks.

Resistance to replay attacks: In the proposed scheme, the messages exchanged between U_i and S are different each time because of the random numbers. If an adversary sends a previously used message to a participant in the protocol, the participant can check its validity. Therefore, our proposed scheme is free from replay attacks.

Resistance to de-synchronization attacks: In the proposed scheme, after computing the session key, S sends DID_2, which is included in the challenge message, to U_i. If the challenge message is blocked and U_i does not receive DID_2 within a given time, U_i will still use DID_1 for login, which is permitted by S (in this case, C_3 and C_4 should be different each time to resist replay attacks). Therefore, our proposed scheme is free from de-synchronization attacks.

Security properties and performance analysis

In this section, we compare our scheme with related three-factor authentication schemes [18, 26, 32] in terms of security performance, computational complexity, and communication complexity.

Comparison of security properties

Table 5 shows the security performance comparison of our scheme and related schemes, i.e., Li et al.'s scheme [18], Wu et al.'s scheme [26], and Zhang et al.'s scheme [32]. In Table 5, Y stands for yes, indicating that the protocol provides the security property or can resist the attack; N stands for no, indicating that the protocol does not provide the security property or cannot resist the attack; and – indicates that the protocol does not provide the corresponding security property.

As shown in the table, Li et al.'s protocol [18] does not provide user anonymity or traceability and thus does not protect user privacy. Wu et al.'s scheme [26] does not provide three-factor secrecy and cannot resist user impersonation attacks. In Zhang et al.'s protocol [32], users cannot freely update passwords and biometrics. Only our proposed protocol meets all the security requirements and is a truly three-factor authentication scheme.

Comparison of computational complexity

We evaluate the computational complexity of the proposed scheme and compare it with that of the schemes in [18, 26, 32] based on the experimental data in [30].

The program in [30] is executed on a computer with a 2.1 GHz Intel(R) Core(TM) 2T6570 CPU, 4 GB RAM, and a 32-bit system. The software used includes Visual C++ and the MIRACL library. The evaluation involves hash function, biohash function, symmetric encryption, and elliptic scalar multiplication operations. The actual runtime for these operations is presented in Table 6. We denote T_h, T_{bh}, T_s, and T_{smul} as the time cost for a hash function, the time cost for a biohash function, the time cost for a symmetric encryption/decryption, and the time cost for a scalar multiplication, respectively.

Table 5 Comparison of security properties

	Li et al. [18]	Wu et al. [26]	Zhang et al. [32]	Ours
C1	*N*	*Y*	*Y*	*Y*
C2	*N*	*Y*	*Y*	*Y*
C3	*Y*	*Y*	*Y*	*Y*
C4	*Y*	*Y*	*Y*	*Y*
C5	*Y*	*Y*	*Y*	*Y*
C6	*Y*	*Y*	*Y*	*Y*
C7	*N*	*N*	*Y*	*Y*
C8	*Y*	*Y*	*Y*	*Y*
C9	*Y*	*Y*	*Y*	*Y*
C10	*Y*	*Y*	*Y*	*Y*
C11	*N*	*N*	*Y*	*Y*
C12	*Y*	*Y*	*Y*	*Y*
C13	*Y*	*Y*	*Y*	*Y*
C14	–	*Y*	*Y*	*Y*
C15	*N*	*N*	*N*	*Y*

C1: Provides user anonymity
C2: Provides untraceability
C3: Provides mutual authentication
C4: Provides session key agreement
C5: Provides known key security
C6: Provides perfect forward secrecy
C7: Provides three-factor secrecy
C8: Provides biometrics protection
C9: Resists stolen verifier attacks
C10: Resists privileged insider attacks
C11: Resists user impersonation attacks
C12: Resists server spoofing attacks
C13: Resists replay attacks
C14: Resists de-synchronization attacks
C15: Changes passwords and biometrics freely

The total computational time, including both the user side and the server side, is $11T_h \approx 0.0044ms$ in the login and mutual authentication phases of Li et al.'s protocol [18]. The total computational time, including both the user side and the server side, is $12T_h + 4T_{smul} + 4T_s \approx 29.9376ms$ in the login and mutual authentication phases of Wu et al.'s protocol [26]. The total computational time, including both the user side and the server side, is $19T_h + 4T_{bh} \approx 0.0476ms$ in the login and mutual authentication phases of Zhang et al.'s protocol [32]. Finally, the total computational time, including both the user side and the server side, is $2T_s + 22T_h + 5T_{bh} \approx 0.3194ms$ in the login and mutual authentication phases of our proposed protocol.

Table 6 Runtime of related operations

Operations	Time (ms)
Time cost for a hash function (T_h)	0.0004
Time cost for a biohash function (T_{bh})	0.01
Time cost for a symmetric encryption/decryption (T_s)	0.1303
Time cost for a scalar multiplication (T_{smul})	7.3529

Table 7 Comparison of computational cost (milliseconds)

Scheme	Total computational time
Li et al. [18]	$11T_h \approx 0.0044$
Wu et al. [26]	$12T_h + 4T_{smul} + 4T_s \approx 29.9376$
Zhang et al. [32]	$19T_h + 4T_{bh} \approx 0.0476$
Ours	$2T_s + 22T_h + 5T_{bh} \approx 0.3194$

The computational cost comparison is shown in Table 7. Note that Wu et al.'s protocol [26] is based on ECC and consumes the most time. The remaining three protocols are based on hash functions, biohash functions, and symmetric cryptography. Although Li et al.'s protocol [18] and Zhang et al.'s protocol [32] are less time-consuming, only our proposed scheme is a truly secure three-factor authentication scheme in which users can freely change passwords and biometrics.

Comparison of communication overhead

Based on the experimental testbed in [30], the size of an elliptic curve point is 160 bits, the size of a hash function output is 160 bits, the size of a biohash function output is 160 bits, and the size of a symmetric encryption is 128 bits.

The size of an exchanged message in Li et al.'s protocol [18] is 832 bits (we assume that the user's identity has a length of 32 bits in Li et al.'s protocol [18]). The size of an exchanged message in Wu et al.'s protocol [26] is 896 bits (we assume that the random number has a length of 160 bits in Wu et al.'s protocol [26]). The size of an exchanged message in Zhang et al.'s protocol [32] is 1056 bits. Finally, the size of an exchanged message $< DID_1, C_3, C_4, C_5, Auth_1, Auth_2 >$ in our proposed scheme is $128+160+160+128+160+160+160 = 1056$ bits.

Table 8 compares the communication bits of various schemes. Although the communication cost of the proposed

Table 8 Comparison of communication overhead

Scheme	Communication bits
Li et al. [18]	832
Wu et al. [26].	896
Zhang et al. [32]	1056
Ours	1664

protocol is slightly higher than that of the protocols in [18, 26, 32], our proposed protocol is more secure.

Conclusion

In this study, we constructed a truly three-factor authentication protocol for TMISs. In the proposed protocol, passwords and biometrics are stored at the server side for verification by the server. Each time a user logs in, he/she sends the value associated with the smart card to the server for verification. Thus, all three factors are verified at the server side. Moreover, our protocol does not use public key cryptosystems, which require time-consuming exponentiation and point multiplication operations. Instead, we use only symmetric cryptography, hash functions, and biohash functions, which makes our protocol efficient. Additionally, our protocol protects user privacy and fulfills all the security requirements of TMISs; therefore, our protocol is suitable for real-world applications. Formal security analysis demonstrates that our proposed protocol is provably secure in the extended security model.

With the advent of the post-quantum era, future research should focus on authentication protocols for TMISs that can withstand attacks from quantum computers.

Acknowledgements The work was supported by the National Natural Science Foundation of China under Grant 61572370, 61501333 and 61572379.

References

1. Amin R, Islam SK, Biswas GP, Khan MK, and Li X, Cryptanalysis and en- hancement of anonymity preserving remote user mutual authentication and session key agreement scheme for e-health care systems. Journal of Medical Systems 2015;39(11):140.
2. An Y, Security analysis and enhancements of an effective biometric-based remote user authentication scheme using smart cards. Journal of Biomedicine & Biotechnology 2012;2012(4):519723.
3. Bresson E, Chevassut O, and Pointcheval D, Security proofs for an efficient password-based key exchange. In: ACM Conference on Computer and Communications Security. 2003. p. 241–50.
4. Chaudhry SA, Naqvi H, and Khan MK, An enhanced lightweight anonymous biometric based authentication scheme for tmis. Multimedia Tools & Applications 2017;(3):1–22.
5. Chaudhry SA, Naqvi H, Shon T, Sher M, and Farash MS, Cryptanalysis and improvement of an improved two factor authentication protocol for telecare medical information systems. Journal of medical systems 2015;39(6):66.
6. Das AK, Analysis and improvement on an efficient biometric-based remote user authentication scheme using smart cards. Information Security Iet 2011;5(3):145–51.
7. Dodis Y, Ostrovsky R, Reyzin L, and Smith A, Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. Siam Journal on Computing 2008;38(1):97–139.
8. Fan CI, and Lin YH, Provably secure remote truly three-factor authentica- tion scheme with privacy protection on biometrics. IEEE Transactions on Information Forensics and Security 2009;4(4):933–45.
9. Hao F, Anderson R, and Daugman J, Combining crypto with biometrics effec- tively. IEEE Transactions on Computers 2006;55(9):1081–8.
10. Jiang Q, Chen Z, Li B, Shen J, Yang L, and Ma J, Security analysis and improvement of bio-hashing based three-factor authentication scheme for telecare medical information systems. Journal of Ambient Intelligence & Humanized Computing 2017;(5):1–13.
11. Jiang Q, Khan MK, Lu X, Ma J, and He D, A privacy preserving three-factor authentication protocol for e-health clouds. Journal of Supercomputing 2016;72(10):3826–49.
12. Jiang Q, Ma J, Yang C, Ma X, Shen J, and Chaudhry SA, Efficient end-to-end authentication protocol for wearable health monitoring systems . Computers & Electrical Engineering 2017;.
13. Jin Z, Teoh ABJ, Goi BM, and Tay YH, Biometric cryptosystems: A new bio- metric key binding and its implementation for fingerprint minutiae-based representation. Pattern Recognition 2016;56:50–62.
14. Kang H, Hori Y, Katashita T, Hagiwara M, and Iwamura K, Cryptographie key generation from puf data using efficient fuzzy extractors. In: International Conference on Advanced Communication Technology. 2014. p. 23–6.
15. Kelkboom EJC, Breebaart J, Buhan I, and Veldhuis RNJ, Maximum key size and classification performance of fuzzy commitment for gaussian modeled biometric sources. IEEE Transactions on Information Forensics & Security 2012;7(4):1225–41.
16. Khan MK, and Kumari S, An improved biometrics-based remote user authentication scheme with user anonymity. BioMed research international 2013;2013(5):491289.
17. Li CT, and Hwang MS, An efficient biometrics-based remote user authentica- tion scheme using smart cards. Academic Press Ltd., 2010.
18. Li X, Niu JW, Ma J, Wang WD, and Liu CL, Cryptanalysis and improvement of a biometrics-based remote user authentication scheme using smart cards. Journal of Network & Computer Applications 2011;34(1):73–9.
19. Li X, Wen Q, Li W, Zhang H, and Jin Z, Secure privacy-preserving biometric authentication scheme for telecare medicine information systems. Journal of Medical Systems 2014;38(11):139.
20. Li X, Wu F, Khan MK, Xu L, Shen J, and Jo M, A secure chaotic map-based remote authentication scheme for telecare medicine information systems. Future Generation Computer Systems 2017.
21. Mir O, and Nikooghadam M, A secure biometrics based authentication with key agreement scheme in telemedicine networks for e-health services. Wireless Personal Communications 2015;83(4):2439–61.
22. Nandakumar K, Jain AK, and Pankanti S, Fingerprint-based fuzzy vault: Implementation and performance. IEEE Transactions on Information Forensics & Security 2007;2(4):744–57.
23. Vallent TF, and Kim H, Three Factor Authentication Protocol Based on Bilin- ear Pairing. Springer Netherlands, 2013.
24. Wang D, and Wang P, Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks. Ad Hoc Networks 2014;20(2):1–15.
25. Wazid M, Das AK, Kumari S, Li X, and Wu F, Design of an efficient and provably secure anonymity preserving threefactor user authentication and key agreement scheme for tmis. Security & Communication Networks 2016;9(13):1983–2001.

26. Wu F, Xu L, Kumari S, and Li X, A novel and provably secure biometrics-based three-factor remote authentication scheme for mobile clientserver networks . Computers & Electrical Engineering 2015;45(C):274–85.
27. Wu ZY, Lee YC, Lai F, Lee HC, and Chung Y, A secure authentication scheme for telecare medicine information systems. Journal of medical systems 2012;36(3):1529–35.
28. Xie Q, Wong D, Wang G, Tan X, Chen K, and Fang L, Provably secure dynamic id-based anonymous two-factor authenticated key exchange protocol with extended security model. IEEE Transactions on Information Forensics & Security 2017;12(6):1382–92.
29. Xiong H, Tao J, and Yuan C, Enabling telecare medical information systems with strong authentication and anonymity. IEEE Access 2017;5:5648–61.
30. Xu L, and Wu F, Cryptanalysis and improvement of a user authentication scheme preserving uniqueness and anonymity for connected health care. J Med Syst 2015;39(2):10.
31. Yeh HL, Chen TH, Hu KJ, and Shih WK, Robust elliptic curve cryptography- based three factor user authentication providing privacy of biometric data. Iet Information Security 2013;7(3):247–52.
32. Zhang L, Zhang Y, Tang S, and Luo H, Privacy protection for e-health systems by means of dynamic authentication and three-factor key agreement. IEEE Transactions on Industrial Electronics ;PP(99):1–.
33. Zhang L, Zhu S, and Tang S, Privacy protection for telecare medicine in- formation systems using a chaotic map-based three-factor authenticated key agreement scheme. IEEE Journal of Biomedical & Health Informatics 2017;PP(99):1–.