IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 2, 2019, accepted December 3, 2019, date of publication December 10, 2019, date of current version December 31, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2958830

Privacy Protection for Telecare Medicine Information Systems with Multiple Servers Using a Biometric-based Authenticated Key Agreement Scheme

CHIN-LAUNG LEI AND YUN-HSIN CHUANG
Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan
Corresponding author: Yun-Hsin Chuang (ntueeyun@gmail.com)

This work was supported by the Ministry of Science and Technology, Taiwan, under Grant MOST 107-2221-E-002-033-MY3 and Grant MOST 108-2218-E-002-045.

ABSTRACT Telecare medical information systems (TMIS) allow patients remotely login medical service providers to acquire their medical information and track their health status through unsecured public networks. Hence, the privacy of patients is vulnerable to various types of security threats and attacks, such as the leakage of medical records or login footprints and the forgery attacks. Many anonymous three-factor authentication and key agreement (AKA) schemes have been proposed for TMIS with single server, but none of them is suited for TMIS with multiple servers. In this paper, we propose a biometric-based three-factor AKA scheme to protect user anonymity and untraceability in TMIS with multiple servers. We will construct a security model of a three-factor AKA scheme with user anonymity in TMIS with multiple servers, and give a formal security proof of the proposed scheme. The security of the proposed scheme is based on the elliptic curve decisional Diffie-Hellman problem assumption and hash function assumption. We will show that the proposed scheme is efficient enough for low-power mobile devices.

INDEX TERMS Biometric, three-factor, authentication, anonymity, untraceability, multi-server, TMIS.

I. INTRODUCTION

The demand for telemedicine services grows rapidly with the rise of health consciousness, the development of Internet of Things (IoT), and the dramatic growth of the world's older population. Telecare medical information systems (TMIS) allow patients to remotely login medical servers to enjoy healthcare or access medical records. How to transmit private information in public channels while keeping secrecy and patients' privacy becomes a new issue.

Numerous authentication and key agreement (AKA) schemes have been proposed from a simple password based scheme to two-factor and three-factor schemes. In 1981, Lamport [1] proposed the first password based authentication scheme. Password based authentication schemes cannot withstand the replay attacks and have to maintain the password files or verification tables; Hwang *et al.*'s [2] proposed the first two-factor authentication scheme in 1990 to overcome these problems. Two-factor authentication schemes verify the user by user's password and smart card. Recently, three-factor authentication schemes get more attention because that they can prevent stolen smart card attack. ***Three-factor*** authentication schemes verify the user by a combination of three different factors: the knowledge, the possession, and the inherent categories. Many present three-factor AKA schemes verify the user by password, smart card, and biometric.

For personal privacy, patients want to access medical servers anonymously. Many ***anonymous*** AKA schemes are proposed to prevent the leakage of user's identity. In ordinary anonymous authentication schemes, even though a user uses an anonymous identity to login, the relationship between each login is exposed since the user uses identical anonymous identity in each login. Recently, the concept of ***untraceability*** has been proposed to overcome this problem, there is no identical or related information would be transmitted in different sessions.

A patient usually communicates to the same medical service provider (server) through unreliable channels in TMIS with single server. In TMIS with multiple servers, a patient communicates to various servers through unreliable channels.

The associate editor coordinating the review of this manuscript and approving it for publication was Zijian Zhang.

The various servers can be doctors, case managers, health centers, clinics, hospitals, etc. These servers should be regards as independent entities with distinct private keys. Otherwise, the malicious server would masquerade as a patient or another medical server.

Many anonymous three-factor AKA schemes have been proposed for TMIS with ***single server***. In 2013, Das and Goswami [3] proposed an anonymity preserving AKA scheme for connected health care. Later on, Wen [4] pointed out the security defects of Das-Goswami scheme, such as user impersonation attack and without user anonymity, and proposed an improvement. In 2014, Xie *et al.* [5] showed that Wen's scheme [4] is vulnerable to the offline password guessing attack and without user anonymity. In 2015, Xu and Wu [6] showed that Xie *et al*.'s scheme [5] is vulnerable to the De-synchronization attack. In 2014, Tan [7] proposed a three-factor AKA scheme for single server TMIS. Later on, Arshad and Nikooghadam [8] pointed out that Tan's scheme [7] is vulnerable to replay attacks. In 2015, Das [9] and Lu *et al.* [10] showed that Arshad-Nikooghadam scheme [8] cannot withstand offline password guessing and user impersonation attacks, and proposed improvements. Later, Amin *et al.* [11] and Jiang *et al.* [12] demonstrated that Lu *et al.*'s scheme [10] is insecure against user anonymity, new smart card issue, patient impersonation, and medical server impersonation attacks; they both proposed an improvement. In 2014, Mishra *et al.* [13] improved an un-anonymous biometrics based AKA scheme [14] to achieve user anonymity. In 2015, Amin and Biswas [15] showed that the Mishra *et al*.'s protocol [13] cannot withstand server impersonation, session key computation, and smart card stolen attacks, and proposed an improvement. However, in 2016, Wazid *et al*. [16] showed that Amin *et al.*'s scheme [11] is vulnerable to privileged insider attack through both smart card stolen and offline password guessing attacks, and also showed that Amin-Biswas's scheme [15] is vulnerable to privileged-insider, stolen smart card, and offline password guessing, user impersonation as well as strong replay attacks. In 2016, Jiang *et al.* [17] proposed a privacy preserving three-factor AKA scheme for e-Health clouds. However, Irshad and Chaudhry [18] identified a flaw in the mutual authentication phase of Jiang *et al.*'s scheme [17] that an adversary may launch a denial-of-service attack (DoS) against the server. In 2017, Zhang *et al.* [19] proposed a privacy protection for TMIS using a chaotic map-based three-factor AKA scheme.

To the best of our knowledge, there is no anonymous three-factor AKA scheme proposed for TMIS with ***multiple servers***. Recently, some anonymous three-factor AKA schemes have been proposed for ***multi-server*** environment. Although they are not specifically designed for TMIS, they are suitable for TMIS with ***multiple servers***. Let us discuss these schemes in the following.

In 2015, Lu *et al.* [20] proposed a biometrics and smart cards-based authentication scheme for multi-server environments that provides strong user anonymity. However, Chaudhry *et al.*'s [21] pointed out that Lu *et al.*'s scheme [20] is defenseless against user impersonation attack, and proposed an improvement. In the same year, He and Wang [22] proposed a biometrics-based AKA scheme for multi-server environment with strong user anonymity. However, Odelu *et al*. [23] showed that He-Wang scheme fails to prevent known session temporary information attack, and their scheme cannot prevent the reply attack and impersonation attack; they further proposed an improvement.

Also in 2015, Amin and Biswas [24] found that Hsieh and Leu's two-factor authentication scheme [25] is vulnerable to user anonymity, password guessing, and server masquerading attacks, and the password change phase is inefficient; they modified it to be a three-factor authentication scheme. In 2017, Chandrakar and Om [26] showed that Amin-Biswas scheme [24] cannot prevent identity and password guessing, user untraceability, user-server impersonation, and privileged insider attacks. They further proposed an improvement. However, Chuang and Lei [27] found that Chandrakar-Om scheme [26] is vulnerable to malignant server attack; any user who has ever login a server, the server would get the user's secrets to impersonate the user. In the same year, Chandrakar and Om [28] proposed another anonymous three-factor remote authentication scheme for multi-server environment using ECC. Unfortunately, Chuang and Lei [27] showed that Chandrakar-Om scheme [28] is vulnerable to insider attack; any user can impersonate another user.

In 2016, Park and Park [29] pointed out that a two-factor authentication scheme proposed by Chang *et al.* [30] is vulnerable to off-line password guessing attacks, and further proposed a three-factor authentication using elliptic curve cryptosystem, and proposed an improvement. However, the Gateway node (registration center) needs to store and manage user's temporal identity table in Park-Park scheme [29]. Also in 2016, Irshad *et al.* [31] proposed an anonymous multi-server authenticated key agreement based on chaotic map without engaging registration center, which the servers have to store public keys of all users. In 2017, Amin *et al.* [32] proposed an anonymous multi-server authentication protocol using multiple registration servers. Their scheme uses the unique identity to achieve user anonymity, but the unique identity repeats in each login session that their scheme does not achieve user untraceability. Also in 2017, Reddy *et al.* [33] proposed an AKA for multi-server environment. In 2019, Xu *et al.* [34] indicated that Reddy *et al.*'s scheme [33] lacks untraceability for users and is susceptible to privileged insider attacks, and proposed an improvement. In 2018, Qi *et al.* [35] proposed a secure biometrics-based AKA protocol for multi-server TMIS using ECC; however, the management of server's public keys is an issue.

A. OUR CONTRIBUTION

In this paper, we proposed a secure three-factor AKA scheme for a TMIS with multiple servers, which achieves user anonymity and untraceability; meanwhile, no public keys and password tables need to be maintained. We add on-line update phase to avoid the involvement of the registration center in each mutual authentication phase.

We construct a security model of a three-factor AKA scheme with user anonymity in TMIS with multiple servers, and give a formal security proof of the proposed scheme. We also show that the proposed scheme is efficient enough for low-power mobile devices.

Generally speaking, there are two kinds of user anonymity: weak anonymity and strong anonymity. *Weak anonymity*: Protect the real identities of users from outsiders; only the participants in the session can get the real identity of the user. In some situations, the servers (medical service providers) need to obtain user's real identity in TMIS to provide medical service, such as tracking and retrieval of health records; an AKA scheme with weak anonymity is suitable for this kind of situation. *Strong anonymity*: It not only achieves the weak anonymity, but also protects the real identities of users from the logged-in servers. In our scheme, if a user wants to protect his/her real identity from the logged-in servers, then he/she can use a pseudonym as his/her identity in the registration phase to achieve strong anonymity.

B. ORGANIZATION

The rest of the paper is organized as follows: The preliminaries are elaborated in Section II. In Section III, we will introduce the framework and the threat model of TMIS with multiple servers and construct a security model of a three-factor remote AKA with user anonymity in TMIS with multiple servers. The proposed scheme and the formal proof are presented in Section IV and Section V, respectively. Section VI shows the performance analysis and comparison. We draw the conclusion and the future work in Section VII.

II. PRELIMINARIES

In this section, we briefly introduce the elliptic curve group [36]–[38], fuzzy extractor [39], and the underlying hard mathematical problems [38].

A. ELLIPTIC CURVE CRYPTOGRAPHY

Let p be a prime number, and let F_p denotes the field of integers modulo p. An elliptic curve E over F_p is defined by an equation of the form $y^2 = x^3 + ax + b$, where $a, b \in F_p$ satisfy $4a^3 + 27b^2 \neq 0 \pmod p$. A pair (x, y), where $x, y \in F_p$, is a point on the curve if (x, y) satisfies $y^2 = x^3 + ax + b$. The set of all the points on E is denoted by $E(F_p)$. Let P be a point in $E(F_p)$, and suppose that P has prime order n. Then the cyclic subgroup of $E(F_p)$ generated by P is $G = \{\infty, P, 2P, 3P, \ldots, (n-1)P\}$.

Given an elliptic curve E defined over a finite field F_p, there are three hard mathematical problems [38]:

1) ***Elliptic curve discrete logarithm (ECDL) problem:*** Given a point $Q=dP\in G$, determine the integer d.
2) ***Elliptic curve decisional Diffie-Hellman (ECDDH) problem:*** Given a point $P \in E(F_p)$ of order n, and points $A = aP$, $B = bP$, and $C = cP$ in $G =< P >$, determine whether $C = abP$ or, equivalently, whether $c \equiv ab \pmod n$.
3) ***Elliptic curve computational Diffie-Hellman (ECCDH) problem:*** Given a point $P \in E(F_p)$ of order n, and points $A = aP$, $B = bP \in G$, find the point $C = abP$.

B. FUZZY EXTRACTOR

Many biometric based authentication schemes refer to Dodis *et al.*'s article [39]; readers may refer to it for the details. We briefly describe the definition of generate function Gen and reproduce function Rep in the following.

Definition 1: An $(M, m, l, t, \varepsilon)$-fuzzy extractor is a pair of randomized procedures, "generate" (Gen) and "reproduce" (Rep), with the following properties:

1) *The generation procedure Gen on input $B\in M$ outputs an extracted string $R\in\{0, 1\}^l$ and a helper string $P\in\{0, 1\}^*$.*
2) *The reproduction procedure Rep takes an element $B'\in M$ and a bit string $PP\in\{0, 1\}^*$ as inputs. The correctness property of fuzzy extractors guarantees that if $dis(B, B')\leq t$ and SP, PP were generated by $(SP, PP)\leftarrow Gen(B)$, then $Rep(B', PP) = SP$. If $dis(B, B') > t$, then no guarantee is provided about the output of Rep.*
3) *The security property guarantees that for any distribution W on M of min-entropy m, the string SP is nearly uniform even for those who observe PP: if $(SP, PP)\leftarrow Gen(W)$, then $SD((SP, PP),(U_l, PP))\leq \varepsilon$.*

C. MATHMATICAL ASSUMPTIONS

The security of the proposed scheme is based on the following assumptions:

Assumption 1 (ECDDH Assumption): No polynomial-time algorithm can solve the Elliptic curve decisional Diffie-Hellman (ECDDH) problem with non-negligible advantage.

Assumption 2 (Hash Function Assumption): There exists a secure one-way hash function $H:X = \{0,1\}^* \rightarrow Y = Z_p^*$, which satisfies the following requirements:

1) *Preimage Resistance: Given any $y \in Y$, it is hard to find $x \in X$ such that $H(x) = y$.*
2) *Second Preimage Resistance: Given any $x\in X$, it is hard to find $x'\in X$ such that $x'\neq x$ and $H(x') = H(x)$.*
3) *Collision Resistance: It is hard to find $x, x'\in X$ such that $x'\neq x$ and $H(x') = H(x)$.*

D. NOTATIONS

The notations used in this paper are summarized in Table 1.

III. FRAMEWORK AND SECURITY

We introduce the TMIS and construct a security model of anonymous three-factor AKA for TMIS with multiple servers.

TABLE 1. **Notations.**

Notation	Meaning
RC	Trusted registration center in TMIS
S_j	The j-th telecare servers in TMIS
U_i	The i-th patients in TMIS
ID_α	The identity of the participant α
PW_i	The password of the patient U_i
B_i	The biometric of the patient U_i
x	The master private key of RC
X	The public key of RC
ΔT	The maximum transmission delay
$\|\|$	String concatenation operation
$\oplus$	Exclusive-or operation
TG_{mul}	Time of executing a multiplication of points
TG_{add}	Time of executing a addition of points
T_{inv}	Time of executing an inversion of scalars
T_{mul}	Time of executing an multiplication of scalars
T_h	Time of executing a one-way hash function
T_{sym}	Time of executing a symmetric encryption/ decryption
T_C	Time of performing a Chebyshev chaotic map operation
T_{kdf}	Time of executing a one-way key derivation function

A. FRAMWORK OF TMIS

In a TMIS with multiple servers, there are one trusted registration center (RC), various medical service providers (Servers), and numerous patients (Users). RC is in charge of system setup, the registration affairs, and keeping the secret key of the system. Servers may be doctors, case managers, clinics, hospitals, health centers, and so on. To protect the privacy of users, servers are regarded as independent entities with distinct private keys. Any server cannot compromise the secrecy of the session between a user and another server. Each user has a low-power mobile device to communicate to servers.

Initially, RC established the system. Each server and user must be registered on the RC through a secure channel when joining the system, and the RC will generate its private key and send it back through a secure channel. After registration, each user makes on-line update through a public channel to get the necessary information before he/she logs into an unfamiliar server. Then, users can use his/her private key and the necessary information to log into servers remotely, authenticate mutually and establish common session keys for secure communication in public channels. Figure 1 illustrates the framework of the TMIS.

B. THREAT MODEL

The following are the assumptions about the attacker's capabilities.

CA1. A legitimate user and a legitimate server can behave as an attacker.

CA2. An attacker can eavesdrop, replay, insert, delete, or modify any message over an unreliable channel.

CA3. An attacker can offline enumerate all the (ID, PW) pairs in the Cartesian product $D_{ID} \times D_{PW}$ within polynomial time [40].

CA4. An attacker can steal the user's smart card and extract the secret data from it using the power consumption analysis [41], [42].

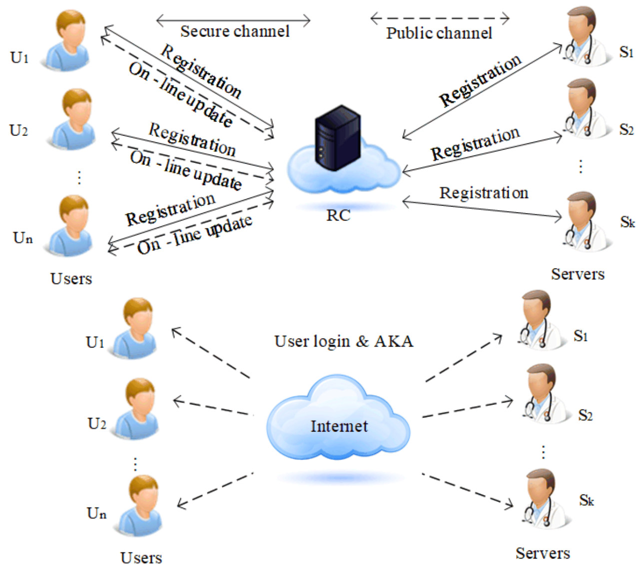


FIGURE 1. **The framework of TMIS.**

CA5. An attacker might fake the biometric [43].

CA6. An attacker can successfully guess the password, extract the secret data from smart card, and fake the biometric individually, but break them at the same time is not feasible in polynomial time.

C. ADVERSARIAL MODEL

According to the thereat model, we define the adversarial model of anonymous three-factor remote AKA in TMIS with multiple servers. In the adversarial model, the TMIS environment contains three kinds of participants: a trusted RC, n users $U = \{U_i | \text{ for } i = 1, \ldots, n\}$, and k servers $S = \{S_j | \text{ for } j = 1, \ldots, k\}$. Each user U_i and each server S_j have unique identities ID_{Ui} and ID_{Sj}, respectively. Let Π_α^s denote the s-th instance of the participant $\alpha \in U \cup S$. We assume that an adversary $\mathcal{A}$ is a probabilistic polynomial-time (PPT) algorithm and potentially control all communications by accessing to a set of oracles described below. An adversary can send eight kinds of queries: Hash, Extract, Send, Execute, Reveal, Rot, Corrupt, and Test queries. In the adversarial model, there is a Simulator $\mathcal{B}$ (oracles) who responds to queries of an adversary as below.

- ***Hash*** (m): $\mathcal{B}$ keeps an initially empty list for each hash function. When receiving the hash query along with a message m, the same response is returned if the query has been asked before. Otherwise, $\mathcal{B}$ selects a random value r, records the pair (m, r), and returns r to $\mathcal{A}$.
- ***Extract*** (ID): $\mathcal{B}$ computes the private key associated with ID and returns it to $\mathcal{A}$. This query models the chosen identity attack. (CA1)
- ***Send*** (Π_α^s, m): $\mathcal{B}$ executes the protocol according to m and responds the corresponding results to $\mathcal{A}$. This query models the active attack. (CA2)
- ***Execute*** (U_α, S_β): $\mathcal{B}$ gives $\mathcal{A}$ the complete transcripts of an honest execution between U_α and S_β. This query models the passive attack. (CA2)

- ***Reveal*** (Π_α^s): There are two kinds of reveal query. (CA1)
  - ***Reveal*$_{\mathbf{SK}}$** (Π_α^s): $\mathcal{B}$ gives $\mathcal{A}$ the corresponding session key *SK* if the instance Π_α^s has accepted the session; otherwise, it returns a null value. This query models the known-session-key attack, in the sense that an adversary cannot reveal other session keys when it compromises a session key.
  - ***Reveal*$_{\mathbf{ID}}$** (Π_α^s): $\mathcal{B}$ gives $\mathcal{A}$ the identity of α. This query models the anonymity attack, in the sense that an adversary cannot reveal the identity of the target user when it compromises the identities of other users.
- ***Rot*** (U_α, M): This query models the secrecy of three-factor authentication, where M withstands the type of the factor. Even if an adversary $\mathcal{A}$ gets any two factors, it still cannot impersonate the user U_α. (CA3) (CA4) (CA5)
- ***Corrupt*** (Π_α^s): $\mathcal{B}$ gives $\mathcal{A}$ the private keys of α. This query models full forward secrecy, in the sense that if an adversary knows the private key of the participant α, it cannot compute any previous session keys established by the participant. (CA3)
- ***Test*** (Π_α^s): $\mathcal{A}$ is allowed to make a single Test query at any time during the game. There are two kinds of test query as follows:
  - ***Test*$_{\mathbf{SK}}$** (Π_α^s): This query is used to define the advantage of $\mathcal{A}$, who breaks the session key secrecy. When $\mathcal{A}$ asks this query to an instance (Π_α^s) for $\alpha \in U \cup S$, Simulation $\mathcal{B}$ chooses a random bit $b \in \{0,1\}$. Simulation $\mathcal{B}$ returns the session key if $b=1$; or returns a random value if $b=0$.
  - ***Test*$_{\mathbf{ID}}$** (Π_α^s): This query is used to define the advantage of $\mathcal{A}$, who breaks the anonymity of α's identity. When $\mathcal{A}$ asks this query to an instance (Π_α^s) for $\alpha \in U$, $\mathcal{B}$ chooses a random bit $b \in \{0,1\}$. $\mathcal{B}$ returns the identity of α if $b = 1$; or returns a random value if $b = 0$.

D. DEFINITIONS OF SECUTIRY

To demonstrate the security of the ID-based MAKA scheme for multi-server environment, we give definitions of security in this subsection. Let Π_α^s denote the s-th instance of the participant αin the adversarial model.

Definition 2: Π_α^s and Π_β^t, where $\alpha \in U$ and $\beta \in S$, are said to be ***partners*** *if they can authenticate mutually and accept a common session key.*

Definition 3: An oracle Π_α^s with its partner Π_β^t is said ***fresh*** *(or holds a fresh key SK) if the following two conditions hold:*

1) *Π_α^s and Π_β^t accept the same session key SK$\neq$ NULL while both of them are not requested by Reveal query.*
2) *No Corrupt query has been asked before the query Send(Π_α^s, m) or Send (Π_β^t, m) has been asked.*
3) *At most two types of the query Rot(Π_α^s, m) have been asked.*

Definition 4: Let **Succ** *denote the event that $\mathcal{A}$ correctly guesses the bit b chosen in the Test query. If $\mathcal{A}$ asks a Test(Π_α^s) and guesses the bit b, the successful advantage (probability) of $\mathcal{A}$ in attacking the attacked scheme $\mathcal{P}$ is defined as $Adv_{\mathcal{P}}(\mathcal{A})= |2\cdot Pr[\mathbf{Succ}]-1|$.*

Definition 5: A three-factor AKA scheme for TMIS with multiple servers offers existential ***unforgeability*** *and maintains* ***session key secrecy***, ***full forward secrecy***, *and* ***user anonymity*** *against adaptive chosen ID attacks if no probabilistic polynomial time adversary $\mathcal{A}$ has a non-negligible advantage in the following game played between an adversary $\mathcal{A}$ and infinite set of oracles Π_α^s for $\alpha \in U \cup S$ and $s \in N$.*

1) *A long-term key is assigned to each user and server through the initialization phase related to the security parameter.*
2) *$\mathcal{A}$ may ask several queries and get back the results from the corresponding oracles.*
3) *$\mathcal{A}$ may asks at most two types of Rot queries for the same user.*
4) *There is no Reveal (Π_α^s) query or Corrupt (ID_α) query asked before the Test (Π_α^s) query.*
5) *$\mathcal{A}$ may ask other queries during asking the Test (Π_α^s) query where Π_α^s is fresh. $\mathcal{A}$ outputs its guess b' for the bit b which is chosen in the Test (Π_α^s) query eventually and the game is terminated.*

IV. PROPOSED SCHEME

Our scheme is composed of five phases: the setup phase, the registration phase, the on-line update phase, the login and AKA phase, and the password and biometric change phase.

A. SETUP PHASE

RC selects a large prime p, an elliptic curve $E_p(a,b)$ over a finite field F_p, a base point $P \in E_p(a,b)$, a one-way hash function $h()$:$\{0, 1\}^* \rightarrow Z_p^*$, and fuzzy extractor functions *Gen*() and *Rep*(). *RC* chooses $x \in Z_p^*$, keeps x as the master private key, and computes $X = x \cdot P$. *RC* decides the maximum transmission delay ΔT, and lets Pub = $\{X, h(), Gen(), Rep(), p, E_p, P, \Delta T\}$ be public parameters.

B. REGISTRATION PHASE

1) SERVER REGISTRATION PHASE

When a new server S_j is to be registered, the following steps are performed.

Step 1: S_j freely chooses an identity ID_{Sj}, and sends it to *RC* through a secure channel.

Step 2: After receiving ID_{Sj} from the server, *RC* computes $k_{Sj} = h(ID_{Sj} || x)$, and sends $\{k_{Sj}, \text{Pub}\}$ to S_j through a secure channel.

2) USER REGISTRATION PHASE

When a patient U_i wants to be a legal user in TMIS, he/she performs the following steps with *RC* through a secure channel, as shown in Figure 2.

Step 1: U_i freely chooses an identity ID_{Ui} and a password PW_i. Note that ID_{Ui} can be either the real identity of U_i or just a pseudonym to achieve strong

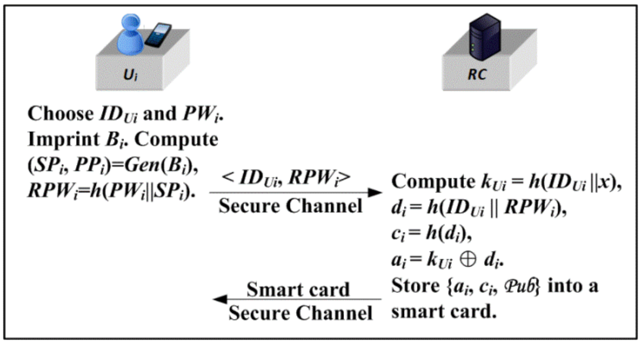


FIGURE 2. User registration phase.

anonymity. U_i then imprints the biometric B_i via a sensor and uses *Gen* function on B_i to produce the private key SP_i and the public key PP_i, *i.e.*, $(SP_i, PP_i) = Gen(B_i)$. U_i computes $RPW_i = h(PW_i||SP_i)$ and sends $< ID_{Ui}, RPW_i >$ to RC through a secure channel.

Step 2: After receiving a request message from the user, RC computes $k_{Ui} = h(ID_{Ui}||x)$, $d_i = h(ID_{Ui}||RPW_i)$, $c_i = h(d_i)$, and $a_i = k_{Ui} \oplus d_i$. RC stores $\{a_i, c_i, \text{Pub}\}$ into a smart card, and sent it to U_i through a secure channel.

Step 3: After receiving the smart card from RC, the user stores PP_i into the smart card. Finally, the smart card contains $\{PP_i, a_i, c_i, \text{Pub}\}$.

C. ON-LINE UPDATE PHASE

Before the user U_i logs into an unfamiliar server S_j, he/she has to run the on-line update phase once to get the public key PK_j of S_j, and the common secret key C_{ij} between U_i and S_j. U_i can delete $<ID_{Sj}, PK_j, C_{ij}>$, which are stored in the smart card, at any time after the on-line update phase. But after then, U_i has to execute the on-line update phase again to get $<ID_{Sj}, PK_j, C_{ij}>$ before U_i logs into server S_j. U_i can ask a batch of on-line update phase for different servers, and can ask for the same server more than once. The on-line update phase is illustrated in Figure 3 and performed as following steps:

Step 1: U_i inputs identity ID_{Ui} and password PW_i to the smart card and imprints the biometric impression B_i at the sensor. U_i's smart card produces the private key SP_i^* by executing *Rep* function on B_i and PP_i, *i.e.*, $SP_i^* = Rep(B_i,PP_i)$. U_i's smart card computes $RPW_i^* = h(PW_i||SP_i^*)$, $d_i^* = h(ID_{Ui}||RPW_i^*)$, and $c_i^* = h(d_i^*)$, and checks if $c_i^* = c_i$. If so, the validity of U_i is confirmed, and then continues the procedure. Otherwise, U_i's smart card terminates it.

Step 2: U_i's smart card generates a random nonce $n \in Z_p^*$ and computes $N = n{\cdot}P$, $K = n{\cdot}X$, and $DID = ID_{Ui} \oplus K$. U_i then sends $<DID, ID_{Sj}, N>$ to RC through an untrustworthy channel.

Step 3: After receiving a request message from the user, RC computes $K = x{\cdot}N$, $ID_{Ui}^* = DID \oplus K$, $k_{Ui} = h(ID_{Ui}^*||x)$, $k_{Sj} = h(ID_{Sj}||x)$, $PK_j = k_{Sj} \cdot P$, and $C_{ij} = (k_{Ui}^{-1} \cdot h(k_{Sj}||ID_{Ui}^*)){\cdot}P$. Finally, RC computes $v = h(ID_{Ui}^*||k_{Ui}|| PK_j||C_{ij}||K)$, and sends $<PK_j, C_{ij}, v>$ to U_i through an untrustworthy channel.

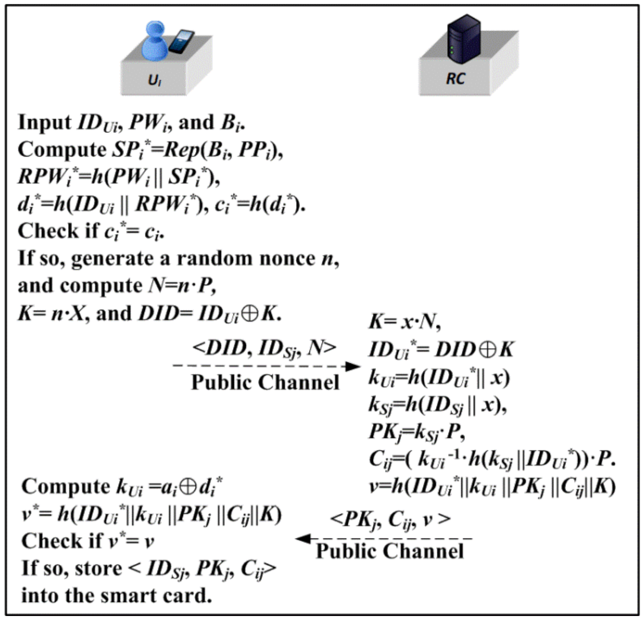


FIGURE 3. On-line update phase.

Step 4: U_i's smart card computes $k_{Ui} = a_i \oplus d_i^*$ and $v^* = h(ID_{Ui}^*||k_{Ui}||PK_j||C_{ij}||K)$, and checks if $v^* = v$. If so, stores $<ID_{Sj}, PK_j, C_{ij}>$ into the smart card.

D. LOGIN AND AKA PHASE

When a user U_i wants to log into a server S_j, the following steps are performed. Figure 4 illustrates the login and AKA phase.

Step 1: Same as Step 1 in the on-line update phase.

Step 2: U_i generates a random nonce n_U and timestamp T_U, and then computes $N_U = n_U{\cdot}P$, and $k_{Ui} = a_i \oplus d_i^*$. U_i finds PK_j and C_{ij} corresponding to S_j's identity ID_{Sj} in the smart card, and computes $Q_{U\_1} = n_U{\cdot}PK_j$, $Q_{U\_2} = (n_U \cdot k_{Ui}) \cdot C_{ij}$, $DID = ID_{Ui} \oplus Q_{U\_1}$, and $v_U = h(ID_{Ui}|| Q_{U\_1}||Q_{U\_2}||T_U)$. U_i then transmits $<ID_{Sj}, DID, N_U, T_U, v_U>$ to S_j.

Step 3: When the server S_j receives the login request message from U_i, S_j generates a timestamp T_S, and verifies if $T_S - T_U \leq \Delta T$. If not, rejects the login request; otherwise, continues the process. S_j computes $Q_{U\_1} = k_{Sj} \cdot N_U$, $ID_{Ui}^* = DID \oplus Q_{U\_1}$, $Q_{U\_2} = h(k_{Sj}||ID_{Ui}^*){\cdot}N_U$, and $v_U^* = h(ID_{Ui}^*|| Q_{U\_1}||Q_{U\_2}||T_U)$. Checks if $v_U^* = v_U$. If not, rejects the login request; otherwise, continues the process. S_j generates a random nonce n_S in Z_p^* and computes $N_S = n_S \cdot P$, the common session key $SK_{ij} = h(Q_{U\_1}||Q_{U\_2}||N_S)$, and $v_S = h(ID_{Ui}^*||ID_{Sj}|| SK_{ij} ||T_U||T_S)$. S_j then sends $< N_S, T_S, v_S >$ to U_i.

Step 4: After receiving $< N_S, T_S, v_S >$, U_i generates a timestamp T_U', and verifies if $T_U' - T_S \leq \Delta T$. If not, rejects the login request; otherwise, continues the process.

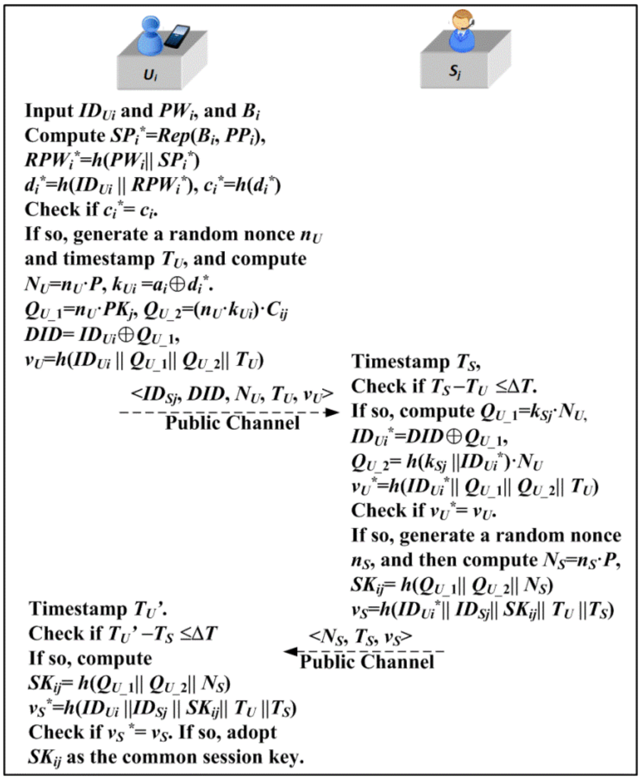


FIGURE 4. Login and AKA phase.

U_i computes $SK_{ij} = h(Q_{U\_1}||Q_{U\_2}||N_S)$ and $v_S^* = h(ID_{Ui}||ID_{Sj}|| SK_{ij}||T_U||T_S)$, and checks if $v_S^* = v_S$. If so, U_i adopts SK_{ij} as the common session key.

E. PASSWORD AND BIOMETRIC CHANGE PHASE

When a user U_i wants to change the password or biometric impression, U_i can change them on his/her own by performing the following steps.

Step 1: Same as Step 1 in the on-line update phase.

Step 2: U_i inputs the new password PW_i^{new}, and imprints new biometric impression B_i^{new}. U_i's smart card computes $(SP_i^{new}, PP_i^{new}) = Gen(B_i^{new})$, $RPW_i^{new} = h(PW_i^{new}||SP_i^{new})$, $d_i^{new} = h(ID_i||RPW_i^{new})$, $a_i^{new} = a_i \oplus d_i^* \oplus d_i^{new}$, and $c_i^{new} = h(d_i^{new})$. U_i's smart card then replaces a_i, c_i, and PP_i with a_i^{new}, c_i^{new}, and PP_i^{new}, respectively.

V. SECURITY ANALYSIS

In this section, we analyze the proposed scheme in the random oracle model [44]. The random oracle model assumes that the hash function is actually a true random function and it produces a random value for each new query. In the random oracle model, the security of the proposed scheme is based on the ECDDH problem. We formally prove that the proposed scheme offers unforgeability, session key secrecy, and full forward secrecy, and provides user anonymity.

Theorem 1: The proposed scheme offers existential unforgeability, session key secrecy and full forward secrecy against adaptive chosen ID attacks under ECDDH assumption and hash function assumption.

Proof: Suppose that there exists a PPT adversary A who can break the unforgeability or session key secrecy or full forward secrecy of the proposed scheme with non- negligible advantage ε, running time T, and given ID_U and ID_S. Then we can construct an algorithm $\mathcal{B}$ to solve ECDDH problem with non-negligible advantage. Let q_U and q_S denotes the numbers of users and servers, respectively. $\mathcal{B}$ is given an instance (p, E_p, P, $A = aP$, $B = bP$, and $C = cP$) of the ECDDH problem. Then B's goal is to determine whether $C = abP$. $\mathcal{B}$ runs $\mathcal{A}$ as a subroutine and simulates its attack environment. First, $\mathcal{B}$ chooses x and sets the public system parameters Pub $= \{X, h(), Gen(), Rep(), p, E_p, P, \Delta T\}$ by letting $X = xP$. $\mathcal{B}$ permeates the *ECDDH* problem into the queries on user U (ID_U) and server S (ID_S), which are asked by $\mathcal{A}$. $\mathcal{B}$ lets $k_U^{-1} \cdot P = A$ and $h(k_V||ID_U) \cdot P = B$. Without loss of generality, assume that $\mathcal{A}$ does not ask queries on the same message more than once, and the user instance Π_α^s and the server instance Π_β^t are partners. $\mathcal{B}$ maintains the list L_H to ensure identical responding and avoid collision of the hash queries, and is L_H empty in the beginning. $\mathcal{B}$ adds $(ID_V||x, k_V)$, $(k_V||ID_U$, NULL), and $(ID_U||x$, NULL) to L_H. $\mathcal{B}$ simulates the oracle queries of $\mathcal{A}$ as follows:

- ***Hash*** (m): When $\mathcal{A}$ makes an H-query for m, $\mathcal{B}$ returns r if $(m,r) \in L_H$. Otherwise, $\mathcal{B}$ returns a random value r and adds (m,r) to L_H.
- ***Extract*** (ID): There are two types of extract query.
  - ***Extract*****user** (ID_α, RPW_α): When $\mathcal{A}$ asks a user Extract-query on $ID_\alpha \neq ID_U$, $\mathcal{B}$ makes $Hash(ID_\alpha||x)$ query to get k_α, makes $Hash$ $(ID_\alpha||RPW_\alpha)$ query to get d_α, makes $Hash(d_\alpha)$ query to get c_α, computes $a_\alpha = k_\alpha \oplus d_\alpha$, and returns $\{a_\alpha, c_\alpha, \text{Pub}\}$ to $\mathcal{A}$.
  - ***Extract*****server** (ID_β): When $\mathcal{A}$ asks a server Extract-query on $ID_\beta \neq ID_S$, $\mathcal{B}$ makes $Hash(ID_\beta||x)$ query to get k_β, and then returns $\{k_\beta, \text{Pub}\}$ to $\mathcal{A}$.
- ***Send*** (Π_α^s, m): There are four types of send query.
  - ***Send*****update** $(\Pi_\alpha^s$, Start): When $\mathcal{A}$ asks this query on ID_α, $\mathcal{B}$ generates a random nonce n and computes $N = n \cdot P$, $K = n \cdot X$, and $DID = ID_\alpha \oplus K$. $\mathcal{B}$ returns $< DID, ID_\beta, N >$ to $\mathcal{A}$.
  - ***Send*****update** $(\Pi_\alpha^s, <DID, ID_\beta, N>)$: $\mathcal{B}$ computes $K = x \cdot N$ and $ID_\alpha = DID \oplus K$, and makes $Hash(ID_\alpha||x)$ query to get k_α and $Hash(ID_\beta||x)$ query to get k_β. If $ID_\alpha = ID_U$ and $ID_\beta = ID_S$, then $\mathcal{B}$ lets $C_{US} = C$. If $ID_\alpha = ID_U$ and $ID_\beta \neq ID_S$, then $\mathcal{B}$ lets $C_{U\beta} = h(k_\beta||ID_U) \cdot A$. If $ID_\alpha \neq ID_U$, then $\mathcal{B}$ computes $PK_\beta = k_\beta \cdot P$, and $C_{\alpha\beta} = (k_\alpha^{-1} \cdot h(k_\beta||ID_\alpha)) \cdot P$. $\mathcal{B}$ then makes $Hash(ID_\alpha||k_\alpha||PK_\beta||C_{\alpha\beta}||K)$ query to get v and returns $<PK_\beta, C_{\alpha\beta}, v>$ to $\mathcal{A}$.
  - ***Send*****MAKA** $(\Pi_\alpha^s$, Start): $\mathcal{B}$ generates a random nonce n_α and timestamp T_α, and then makes $Hash(ID_\beta||x)$ query to get k_β, and $Hash(k_\beta||ID_\alpha)$

query to get $h(k_\beta||ID_\alpha)$.If $ID_\alpha = ID_U$ and $ID_\beta = ID_S$, $\mathcal{B}$ then lets $Q_{\alpha\_2} = n_\alpha \cdot B$; otherwise, $\mathcal{B}$ computes $N_\alpha = n_\alpha \cdot P$ and $Q_{\alpha\_2} = h(k_\beta||ID_\alpha) \cdot N_\alpha$. $\mathcal{B}$ then computes $Q_{\alpha\_1} = k_\beta \cdot N_\alpha$ and $DID = ID_\alpha \oplus Q_{\alpha\_1}$. $\mathcal{B}$ makes $Hash(ID_\alpha||Q_{\alpha\_1}||Q_{\alpha\_2}||T_\alpha)$ query to get v_α. $\mathcal{B}$ returns $<ID_\beta, DID, N_\alpha, T_\alpha, v_\alpha>$.
  - ***Send*$_{\mathbf{MAKA}}$** ($_\beta\Pi^t$, $<ID_\beta$, DID, N_α, T_α, $v_\alpha>$): $\mathcal{B}$ generates a random nonce n_β and a timestamp T_β, and verifies if $T_\beta - T_\alpha \leq \Delta T$. If not, $\mathcal{B}$ returns "Reject". $\mathcal{B}$ makes $Hash(ID_\beta||x)$ query to get k_β, and computes $N_\beta = n_\beta \cdot P$, $Q_{\alpha\_1} = k_\beta \cdot N_\alpha$, $ID_\alpha = DID \oplus Q_{\alpha\_1}$. If $ID_\alpha = ID_U$ and $ID_\beta = ID_S$, $\mathcal{B}$ then uses v_α to find $((ID_\alpha||Q_{\alpha\_1}||Q_{\alpha\_2}||T_\alpha), v_\alpha)$ in L_H to get Q_{α_2}; otherwise, $\mathcal{B}$ makes $Hash(k_\beta||ID_\alpha)$ query to get $h(k_\beta||ID_\alpha)$, and computes $Q_{\alpha\_2} = h(k_\beta||ID_\alpha) \cdot N_\alpha$. $\mathcal{B}$ makes $Hash(ID_\alpha||Q_{\alpha\_1}||Q_{\alpha\_2}|| T_\alpha)$ query to get v_α^*. Checks if $v_\alpha^* = v_\alpha$. If not, $\mathcal{B}$ returns "Reject". $\mathcal{B}$ makes $Hash(Q_{\alpha\_1}|| Q_{\alpha\_2}||N_\beta)$ query to get $SK_{\alpha\beta}$, and $Hash(ID_\alpha||ID_\beta||SK_{\alpha\beta}|| T_\alpha||T_\beta)$ query to get v_β. $\mathcal{B}$ returns $< N_\beta, T_\beta, v_\beta >$.
- ***Execute*** (U_α, S_β): When $\mathcal{A}$ asks an Execute(ID_α, ID_β) query, $\mathcal{B}$ returns the transcript $<(DID, ID_\beta, N)$, $(PK_\beta, C_{\alpha\beta}, v)$, $(ID_\beta, DID, N_\alpha, T_\alpha, v_\alpha)$, $(N_\beta, T_\beta, v_\beta)>$ by using the above simulation of Send queries.
- ***Reveal*** (Π_α^s): There are two types of reveal query as follows:
  - ***Reveal*$_{\mathbf{SK}}$** (Π_α^s): $\mathcal{B}$ returns $SK_{\alpha\beta}$ by using the above simulation of Send queries if the instance Π_α^s has accepted the session; otherwise, $\mathcal{B}$ returns a null value.
  - ***Reveal*$_{\mathbf{ID}}$** (Π_α^s): $\mathcal{B}$ returns ID_α.
- ***Rot*** ($\boldsymbol{U}_\alpha$, M): At most two types of Rot query can be asked for a user $\boldsymbol{U}_\alpha$. $\mathcal{B}$ reacts by the following three types of Rot query.
  - ***Rot*** ($\boldsymbol{U}_\alpha$, PW): $\mathcal{B}$ returns PW_α.
  - ***Rot*** ($\boldsymbol{U}_\alpha$, BI): $\mathcal{B}$ returns B_α.
  - ***Rot*** ($\boldsymbol{U}_\alpha$, SC): $\mathcal{B}$ makes ***Extract***$_{\text{user}}$(ID_α, RPW_α) query to get $\{a_\alpha, c_\alpha, \text{Pub}\}$, and then returns $\{a_\alpha, c_\alpha, \text{Pub}\}$.
- ***Corrupt*** (Π_α^s): When $\mathcal{A}$ asks a Corrupt (ID_α) query, then $\mathcal{B}$ makes Extract (ID_α, RPW_α) query to get $\{a_\alpha, c_\alpha, \text{Pub}\}$, and then returns PW_α, B_α, and $\{a_\alpha, c_\alpha, \text{Pub}\}$ to $\mathcal{A}$.
- ***Test*$_{\mathbf{SK}}$** (Π_α^s): $\mathcal{B}$ randomly chooses a bit $b \in\{0,1\}$. $\mathcal{B}$ returns $SK_{\alpha\beta}$ if $b = 1$, and else returns a random value.

If $\mathcal{A}$ answers $b = 1$ to the $Test_{SK}$ query, then $\mathcal{B}$ answers $C = abP$ to the ECDDH problem. If $\mathcal{A}$ answers $b \neq 1$ to the $Test_{SK}$ query, then $\mathcal{B}$ answers $C \neq abP$ to the ECDDH problem. The success probability of $\mathcal{B}$ depends on the event that $\mathcal{A}$ asks the $Test_{SK}$ query on user U (ID_U) and server S (ID_S) and correctly guesses b in the $Test_{SK}$ query. In the above simulation, the probability that $\mathcal{A}$ asks the $Test_{SK}$ query in the l-th session is $1/q_U \cdot q_S$. If $\mathcal{A}$ correctly guesses b in the $Test_{SK}$ query with a non-negligible advantage ε, then $\mathcal{B}$ solves the ECDDH problem with a non-negligible advantage $\varepsilon/q_U \cdot q_S$. By Assumption 1, no polynomial-time algorithm can solve ECDDH problem with non-negligible advantage, it is a contradiction. Hence, there is no PPT time adversary $\mathcal{A}$ has a non-negligible advantage in the above game played between $\mathcal{A}$ and $\mathcal{B}$. Then by Definition 5, the proposed scheme offers existential unforgeability, session key secrecy and full forward secrecy against adaptive chosen ID attacks. ■

Theorem 2: The proposed scheme maintains user anonymity under ECDDH and hash function assumptions.

Proof: Suppose that there exists a PPT adversary $\mathcal{A}$ who can break the anonymity of the proposed scheme with running time T, advantage ε. Then we can construct an algorithm $\mathcal{B}$ to solve ECDDH problem with non-negligible advantage. Let q_U, q_S, and q_{ns}, respectively, denote the numbers of users, servers, and sessions. $\mathcal{B}$ is given an instance (p, E_p, P, $A = aP$, $B = bP$, and $C = cP$) of the elliptic curve decision Diffie-Hellman problem. Then $\mathcal{B}$'s goal is to determine whether $C = abP$. $\mathcal{B}$ runs $\mathcal{A}$ as a subroutine and simulates its attack environment. First, $\mathcal{B}$ chooses x and sets the public system parameters Pub = $\{X, h(), Gen(), Rep(), p, E_p, P, \Delta T\}$ by letting $X = x{\cdot}P$. $\mathcal{B}$ gives the public parameters to $\mathcal{A}$. $\mathcal{B}$ permeates *ECDDH* problem into the queries, which are asked by $\mathcal{A}$ in the l-session, on user $U(ID_U)$ and server S (ID_S). Without loss of generality, assume that $\mathcal{A}$ does not ask queries on the same message more than once, and the user instance Π_α^s and the server instance Π_β^t are partners. $\mathcal{B}$ maintains the list L_H to ensure identical responding and avoid collision of the hash queries. $\mathcal{B}$ simulates the oracle queries of $\mathcal{A}$ as follows:

- ***Hash***(m), ***Extract***(ID), ***Send*$_{\mathbf{update}}$** (Π_α^s, Start), ***Send*$_{\mathbf{MAKA}}$** (Π_α^s, Start), ***Send*$_{\mathbf{MAKA}}$** (Π_β^t, $<ID_\beta$, DID, N_α, T_α, $v_\alpha>$), ***Execute***(U_α, S_β), ***Reveal*** (Π_α^s), ***Rot*** ($\boldsymbol{U}_\alpha$, M), and ***Corrupt*** (Π_α^s) are identical to those queries in the proof of Theorem 1.
  - ***Send*$_{\mathbf{update}}$** (Π_α^s, $<DID$, ID_β, $N>$): When $\mathcal{A}$ asks this query, $\mathcal{B}$ computes $K{=}x{\cdot}N$ and $ID_\alpha = DID \oplus K$, and makes $Hash(ID_\alpha||x)$ query to get k_α. $\mathcal{B}$ makes $Hash(ID_\beta||x)$ query to get k_β, and computes $PK_\beta = k_\beta \cdot P$, and $C_{\alpha\beta} = (k_\alpha^{-1} \cdot h(k_\beta||ID_\alpha)){\cdot}P$. If $ID_\alpha = ID_U$ and $ID_\beta = ID_S$, then $\mathcal{B}$ lets $PK_\beta = PK_S = B$. $\mathcal{B}$ then makes $Hash(ID_\alpha||k_\alpha||PK_\beta||C_{\alpha\beta}||K)$ query to get v and returns $<PK_\beta, C_{\alpha\beta}, v>$ to $\mathcal{A}$.
- ***Test*$_{\mathbf{ID}}$** (Π_α^s): When $\mathcal{A}$ makes a Test query, $\mathcal{B}$ randomly chooses a bit $b \in\{0,1\}$. $\mathcal{B}$ then returns ID_α if $b = 1$, and else returns a random number.

If $\mathcal{A}$ answers $b = 1$ to the $Test_{ID}$ query, then $\mathcal{B}$ answers $C = abP$ to the ECDDH problem. If $\mathcal{A}$ answers $b \neq 1$ to the $Test_{ID}$ query, then $\mathcal{B}$ answers $C \neq abP$ to the ECDDH problem. The success probability of $\mathcal{B}$ depends on the event that $\mathcal{A}$ asks the $Test_{ID}$ query for the user U (ID_U) and the server S (ID_S) in the l-session. In the above simulation, the probability that $\mathcal{A}$ asks the $Test_{ID}$ query for ID_U is $1/q_U$, and asks the Send query for ID_S in the l-session is $1/q_{S \cdot q_{ns}}$. If $\mathcal{A}$ correctly guesses b in the $Test_{ID}$ query with non-negligible advantage ε, then $\mathcal{B}$ solves *mECCDH* problem with non-negligible advantage at least $\varepsilon/q_U \cdot q_S \cdot q_{ns}$. By Assumption 1, no polynomial-time algorithm can solve

TABLE 2. Execution times of operations.

Operations	Execution Time	Platform
TG_{mul}	17.71 ms	Xilinx VirtexII-Pro XC2VP30 FPGA device with maximal clock frequency 25.51 MHz [45]
T_{inv}	1.24 ms	
TG_{add}	0.06276 ms	
T_{mul}	0.00286 ms	
T_h	0.065 ms	8 MHz MSP430 family [46]

TABLE 3. The estimated times on the user side.

	Computational cost	Execution Time
User registration phase	$1T_h$	0.065 ms
On-line update phase	$2TG_{mul} + 4T_h$	35.68 ms
Login and AKA phase	$3TG_{mul} + T_{mul} + 6T_h$	53.52286 ms

ECDDH problem with non-negligible advantage, it is a contradiction. Hence, there is no PPT time adversary $\mathcal{A}$ has a non-negligible advantage in the above game played between $\mathcal{A}$ and $\mathcal{B}$. Then by Definition 5, the proposed scheme offers existential user anonymity against adaptive chosen ID attacks. ■

VI. PERFORMANCE ANALYSIS AND COMPARISONS

Vliegen *et al.* [45] described the implementation of elliptic curve cryptography over prime fields on the Xilinx VirtexII-Pro XC2VP30 FPGA device with maximal clock frequency 25.51 MHz, the execution times of TG_{mul}, T_{inv}, TG_{add}, and T_{mul} are 17.71 milliseconds (ms), 1.24 ms, 0.06276 ms, 0.00286 ms, respectively. In [46], the execution time of a hash function is 0.065 ms, in which the implementation is performed on the MSP430 family with a frequency of 8 MHz. The execution times of operations are summarized in Table 2.

Table 3 shows the estimated executing times on the user side. In our scheme, the estimated execution time of a user during registration phase, on-line update phase, and login and AKA phase are only 0.065 ms, 35.68 ms, and 53.52286 ms, respectively. Obviously, our scheme is well suited for the low-power mobile devices.

The comparisons of our scheme and the relevant three-factor AKA schemes, which are suitable for TMIS with multiple servers, are summarized in Table 4. These schemes all achieve both user anonymity and untraceability except Amin *et al.*'s [32] scheme.

In Chaudhry *et al.*'s [21], Odelu *et al.*'s [23], Park-Park scheme [29], Amin *et al.*'s [32], and Xu *et al.*'s [34] scheme, the registration center or gateway node has to store and maintain password tables.

In Odelu *et al.*'s [23], Park and Park [29], Amin *et al.*'s [32], and Qi *et al.*'s [35] scheme, the registration center or gateway node has to be involved in each user login and MAKA phase; it may cause the traffic bottleneck.

In Chaudhry *et al*'s. [21], Irshad *et al.* [31], Xu *et al.*'s [34] scheme, and Qi *et al.*'s [35] scheme, there are public keys need to be managed and public. Verifying the authenticity of public keys is an issue.

Only our scheme, the registration center does not need to maintain any table, and is not involved in the user login and MAKA phases; meanwhile, no public key needs to be managed. Moreover, our scheme keeps the efficiency and is suitable for low power devices.

VII. CONCLUSION AND FUTURE WORK

In this paper, we proposed a biometric based three-factor AKA scheme that is suited for TMIS with multiple servers,

TABLE 4. Comparisons of our scheme and relevant schemes.

	Amin *et al.* [32]	Odelu *et al.* [23]	Park-Park [29]	Chaudhry *et al.* [21]	Xu *et al.* [34]	Qi *et al.* [35]	Irshad *et al.* [31]	Our Scheme
User anonymity	Y	Y	Y	Y	Y	Y	Y	Y
User untraceability	N	Y	Y	Y	Y	Y	Y	Y
Without storing password tables	N (RC)	N (RC)	N (RC)	N (Users)	N (RC& Users)	Y	Y	Y (Optional)
Without RC involvement in MAKA phase	N	N	N	Y	Y	N	Y	Y
Without public-key tables	Y	Y	Y	N (Servers)	N (Servers)	N (Servers)	N (Servers & Users)	Y
Computation cost of RC in the login and MAKA phase	$7\,T_h$	$3T_{sym}+1TG_{mul}+10T_h$	$11\,T_h$	0	0	$1T_{sym}+TG_{mul}+T_{kdf}+5T_h$	0	0
Computation cost of user in the login and MAKA phase	$9\,T_h$	$1T_{sym}+3TG_{mul}+7T_h$	$2TG_{mul}+10T_h$	$1T_{sym}+15T_h$	$3TG_{mul}+10T_h$	$3TG_{mul}+6T_h$	$4T_C+7T_h$	$3TG_{mul}+T_{mul}+6T_h$
Computation cost of server in the login and MAKA phase	$6\,T_h$	$2T_{sym}+2TG_{mul}+6T_h$	$3TG_{mul}+4T_h$	$1T_{sym}+12T_h$	$3TG_{mul}+6T_h$	$1T_{sym}+4TG_{mul}+T_{kdf}+4T_h$	$4T_C+4T_h$	$3TG_{mul}+4T_h$

and achieves strong user anonymity and user untraceability. We constructed a security model of a three-factor AKA scheme with user anonymity for TMIS with multiple servers. We gave the formal proof of the proposed scheme in the random oracle model, and the security of the proposed scheme is based on the ECDDH and hash function assumptions. We estimated the executing times on low-power mobile devices to show that our scheme is efficient enough. Moreover, we compared our scheme with relevant three-factor AKA schemes to show the contributions of our scheme.

In the proposed scheme, a user needs to run the on-line update phase once before he/she logs into an unfamiliar server. Our future work is to modify the proposed scheme to be free from on-line update; meanwhile, retain all advantages.

ACKNOWLEDGMENT

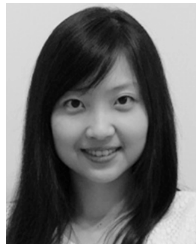
The authors thank the anonymous referees for their valuable comments and constructive suggestions. This research was supported by the Ministry of Science and Technology, Taiwan, under Grants MOST 107-2221-E-002-033-MY3 and MOST 108-2218-E-002-045.

REFERENCES

[1] L. Lamport, "Password authentication with insecure communication," *Commun. ACM*, vol. 24, no. 11, pp. 770–772, Nov. 1981.
[2] T. Hwang, Y. Chen, and C. J. Laih, "Non-interactive password authentications without password tables," in *Proc. IEEE TENCON*, Hong Kong, vol. 1, Sep. 1990, pp. 429–431.
[3] A. K. Das and A. Goswami, "A secure and efficient uniqueness-and-anonymity-preserving remote user authentication scheme for connected health care," *J. Med. Syst.*, vol. 37, no. 9948, pp. 1–16, Jun. 2013.
[4] F. Wen, "A robust uniqueness-and-anonymity-preserving remote user authentication scheme for connected health care," *J. Med. Syst.*, vol. 37, no. 9980, pp. 1–9, Dec. 2013.
[5] Q. Xie, W. Liu, S. Wang, L. Han, B. Hu, and T. Wu, "Improvement of a uniqueness-and-anonymity-preserving user authentication scheme for connected health care," *J. Med. Syst.*, vol. 38, no. 91, pp. 1–10, Sep. 2014.
[6] L. Xu and F. Wu, "Cryptanalysis and improvement of a user authentication scheme preserving uniqueness and anonymity for connected health care," *J. Med. Syst.*, vol. 39, no. 10, pp. 1–9, Feb. 2015.
[7] Z. Tan, "A user anonymity preserving three-factor authentication scheme for telecare medicine information systems," *J. Med. Syst.*, vol. 38, no. 16, pp. 1–9, Mar. 2014.
[8] H. Arshad and M. Nikooghadam, "Three-factor anonymous authentication and key agreement scheme for telecare medicine systems information," *J. Med. Syst.*, vol. 38, no. 6, pp. 1–12, Dec. 2014.
[9] A. K. Das, "A secure user anonymity-preserving three-factor remote user authentication scheme for the telecare medicine information systems," *J. Med. Syst.*, vol. 39, no. 30, pp. 1–20, Mar. 2015.
[10] Y. Lu, L. Li, H. Peng, and Y. Yang, "An enhanced biometric-based authentication scheme for telecare medicine information systems using elliptic curve cryptosystem," *J. Med. Syst.*, vol. 39, no. 32, pp. 1–8, Mar. 2015.
[11] R. Amin, S. H. Islam, G. P. Biswas, M. K. Khan, and M. S. Obaidat, "Design and analysis of an enhanced patient-server mutual authentication protocol for telecare medical information system," *J. Med. Syst.*, vol. 39, no. 137, pp. 1–20, Nov. 2015.
[12] Q. Jiang, Z. Chen, B. Li, J. Shen, L. Yang, and J. Ma, "Security analysis and improvement of bio-hashing based three-factor authentication scheme for telecare medical information systems," *J. Ambient Intell. Hum. Comput.*, vol. 9, no. 4, pp. 1061–1073, Aug. 2018.
[13] D. Mishra, S. Mukhopadhyay, A. Chaturvedi, S. Kumari, and M. Khan, "Cryptanalysis and improvement of Yan's biometric-based authentication scheme for telecare medicine information systems," *J. Med. Syst.*, vol. 38, no. 24, pp. 1–12, Jun. 2014.
[14] X. Yan, W. Li, P. Li, J. Wang, X. Hao, and P. Gong, "A secure biometrics based authentication scheme for telecare medicine information systems," *J. Med. Syst.*, vol. 37, no. 9972, pp. 1–6, Oct. 2013.
[15] R. Amin and G. P. Biswas, "A secure three-factor user authentication and key agreement protocol for TMIS with user anonymity," *J. Med. Syst.*, vol. 39, no. 78, pp. 1–19, Aug. 2015.
[16] M. Wazid, A. K. Das, S. Kumari, X. Li, and F. Wu, "Design of an efficient and provably secure anonymity preserving three-factor user authentication and key agreement scheme for TMIS," *Secur. Commun. Netw.*, vol. 9, no. 13, pp. 1983–2001, Sep. 2016.
[17] Q. Jiang, M. K. Khan, X. Lu, J. Ma, and D. He, "A privacy preserving three-factor authentication protocol for e-Health clouds," *J. Supercomput.*, vol. 72, no. 10, pp. 3826–3849, Oct. 2016.
[18] A. Irshad and S. A. Chaudhry, "Comments on 'A privacy preserving three-factor authentication protocol for e-Health clouds,'" *J. Supercomput.*, vol. 73, no. 4, pp. 1504–1508, Apr. 2017.
[19] L. Zhang, S. Zhu, and S. Tang, "Privacy protection for telecare medicine information systems using a chaotic map-based three-factor authenticated key agreement scheme," *IEEE J. Biomed. Health Informat.*, vol. 21, no. 2, pp. 465–475, Mar. 2017.
[20] Y. Lu, L. Li, H. Peng, and Y. Yang, "A biometrics and smart cards-based authentication scheme for multi-server environments," *Secur. Commun. Netw.* vol. 8, no. 17, pp. 3219–3228, 2015.
[21] S. A. Chaudhry, H. Naqvi, M. S. Farash, T. Shon, and M. Sher, "An improved and robust biometrics-based three factor authentication scheme for multiserver environments," *J. Supercomput.*, vol. 74, no. 8, pp. 3504–3520, Aug. 2018.
[22] D. He and D. Wang, "Robust biometrics-based authentication scheme for multiserver environment," *IEEE Syst. J.*, vol. 9, no. 3, pp. 816–823, Sep. 2015.
[23] V. Odelu, A. K. Das, and A. Goswami, "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 9, pp. 1953–1966, Sep. 2015.
[24] R. Amin and G. P. Biswas, "Design and analysis of bilinear pairing based mutual authentication and key agreement protocol usable in multi-server environment," *Wireless Pers. Commun.*, vol. 84, no. 1, pp. 439–462, 2015.
[25] W.-B. Hsieh and J.-S. Leu, "An anonymous mobile user authentication protocol using self-certified public keys based on multi-server architectures," *J. Supercomput.*, vol. 70, no. 1, pp. 133–148, 2014.
[26] P. Chandrakar and H. Om, "Cryptanalysis and improvement of a biometric-based remote user authentication protocol usable in a multiserver environment," *Trans. Emerg. Telecommun. Technol.*, vol. 28, no. 12, Dec. 2017, Art. no. e3200.
[27] Y. H. Chuang and C. L. Lei, "Cryptanalysis of four biometric based authentication schemes with privacy-preserving for multi-server environment," Nat. Taiwan Univ., Taipei, Taiwan, Tech. Rep., 2019.
[28] P. Chandrakar and H. Om, "A secure and robust anonymous three-factor remote user authentication scheme for multi-server environment using ECC," *Comput. Commun.*, vol. 110, pp. 26–34, Sep. 2017.
[29] Y. Park and Y. Park, "Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks," *Sensors*, vol. 16, no. 12, Dec. 2016, Art. no. 2123.
[30] I.-P. Chang, T.-F. Lee, T.-H. Lin, and C.-M. Liu, "Enhanced two-factor authentication and key agreement using dynamic identities in wireless sensor networks," *Sensors*, vol. 15, no. 12, pp. 29841–29854, 2015.
[31] A. Irshad, M. Sher, S. A. Chaudhary, H. Naqvi, and M. S. Farash, "An efficient and anonymous multi-server authenticated Key agreement based on chaotic map without engaging registration centre," *J. Supercomput.*, vol. 72, no. 4, pp. 1623–1644, 2016.
[32] R. Amin, H. S. Islam, M. S. Obaidat, G. P. Biswas, and K. F. Hsiao, "An anonymous and robust multi-server authentication protocol using multiple registration servers," *Int. J. Commun. Syst.*, vol. 30, pp. 1–14, Aug. 2017.
[33] A. G. Reddy, E.-J. Yoon, A. K. Das, V. Odelu, and K.-Y. Yoo, "Design of mutually authenticated key agreement protocol resistant to impersonation attacks for multi-server environment," *IEEE Access*, vol. 5, pp. 3622–3639, 2017.
[34] D. Xu, J. Chen, and Q. Liu, "Provably secure anonymous three-factor authentication scheme for multi-server environments," *J. Ambient Intell. Hum. Comput.*, vol. 10, no. 2, pp. 611–627, 2019.
[35] M. Qi, J. Chen, and Y. Chen, "A secure biometrics-based authentication key exchange protocol for multi-server TMIS using ECC," *Comput. Methods Programs Biomed.*, vol. 164, pp. 101–109, Oct. 2018.
[36] V. S. Miller, "Use of elliptic curves in cryptography," in *Proc. CRYPTO*, Santa Barbara, CA, USA, vol. 218, 1985, pp. 417–426.

[37] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comput.*, vol. 48, no. 177, pp. 203–209, 1987.
[38] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*. New York, NY, USA: Springer-Verlag, 2004.
[39] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Proc. EUROCRYPT*, Interlaken, Switzerland, vol. 3027, 2004, pp. 523–540.
[40] D. Wang and P. Wang, "Two birds with one stone: Two-factor authentication with security beyond conventional bound," *IEEE Trans. Depend. Sec. Comput.*, vol. 15, no. 4, pp. 708–722, Jul./Aug. 2018.
[41] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances Cryptology-CRYPTO*, vol. 1666, M. Wiener, Ed. Berlin, Germany: Springer, 1999, pp. 388–397.
[42] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 541–552, May 2002.
[43] S. Rane, Y. Wang, S. Drape, and P. Ishwar, "Secure biometrics: Concepts, authentication architectures, and challenges," *IEEE Signal Process. Mag.*, vol. 30, no. 5, pp. 51–64, Sep. 2013.
[44] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *Proc. CCS*, Fairfax, VA, USA, 1993, pp. 62–73.
[45] J. Vliegen, N. Mentens, J. Genoe, A. Braeken, S. Kubera, A. Touhafi, and I. Verbauwhede, "A compact FPGA-based architecture for elliptic curve cryptography over prime fields," in *Proc. ASAP*, Rennes, France, 2010, pp. 313–316.
[46] S. Cavalieri and G. Cutuli, "Implementing encryption and authentication in KNX using Diffie–Hellman and AES algorithms," in *Proc. IECON*, Porto, Portugal, 2009, pp. 2459–2464.

CHIN-LAUNG LEI received the B.S. degree in electrical engineering from National Taiwan University, Taipei, in 1980, and the Ph.D. degree in computer science from The University of Texas at Austin, in 1986. From 1986 to 1988, he was an Assistant Professor with the Computer and Information Science Department, Ohio State University, Columbus. In 1988, he joined the Faculty of the Department of Electrical Engineering, National Taiwan University, where he is currently a Professor. He has published more than 250 technical articles in scientific journals and conference proceedings. His current research interests include network security, cloud computing, the Internet of Things, and big data analytics. He is a co-winner of the first IEEE LICS Test of-Time Award.



YUN-HSIN CHUANG received the B.S. and M.S. degrees from the Department of Mathematics, National Changhua University of Education, Taiwan, in 2006 and 2010, respectively. She is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, National Taiwan University. Her research interests include network security, applied cryptography, mobile communication, and the Internet of Things.