Confidentiality-Preserving Personal Health Records in Tele-Healthcare System Using Authenticated Certificateless Encryption

Rui Guo[1,2], Huixian Shi[3]
(Corresponding author: Rui Guo)
National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications[1]
Xi'an 710121, P.R. China
(Email: guorui@xupt.edu.cn)
State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications[2]
School of Mathematics and Information Science, Shaanxi Normal University[3]
(Received Aug. 3, 2016; revised and accepted Nov. 15 & Dec. 25, 2016)

Abstract

Wireless Medical Sensor Networks (WMSN) facilitate the traditional healthcare systems, however, due to the public transmission, the healthcare system in WMSN also faces some serious security and privacy challenges. These are major concerns in the Health Insurance Portability and Accountability Act. Especially, integrity and confidentiality of patient physiological data are two key issues in privacy protection, which must be considered and addressed firstly. Therefore, the security and privacy in such systems should be enforced via authentication as well as encryption. This paper presents an authenticated certificateless public key encryption scheme for protecting the integrity and confidentiality of the patient sensitive information in tele-healthcare system simultaneously. The security of this protocol is based on the hardness of the bilinear Diffie-Hellman problem, and we prove that it is secure in the random oracle model. Our analysis and comparisons with related protocols show that this scheme is a viable encryption for tele-healthcare system.

Keywords: Authentication; Bilinear Pairing; Certificateless Public Key Encryption; Privacy; Tele-healthcare System

1 Introduction

Wireless Medical Sensor Networks (WMSN) are the networks of medical sensors with small, limited memory and low battery power, for enabling to offer professional, individualized and real-time medical services [9]. In WMSN, the wearable sensor in the patient's body transmits his/her physiological signals (e.g., blood pressure, pulse oxi-meter and temperature, etc.) to the doctor via a wireless channel. In the transmission, lacking of necessary security protection may divulge the patient's privacy, and then cause that the adversary eavesdrops and distorts actual data to misadvise the patients with these false diagnoses and treatments [22].

The Health Insurance Portability and Accountability Act (HIPAA) [3], as a guideline for privacy and security regulations, was presented in 1996. This Act stated that the integrity and confidentiality of the personal health records (PHR) between patient and doctor should be ensured. Therefore, in order to protect the patient's privacy, authentication mechanisms and encryption protocols among patient, the medical server (MS) and doctor are essential in tele-healthcare systems (THS).

1.1 Related Works

Wu et al. [26] proposed an authentication protocol with a new phase named the pre-computing phase for the telecare medicine information system (TMIS). In that phase, the entity computes costly and time consuming exponential operations and stores them into a smart card. When these values are needed, the entity enables to ex-tract them from the device rapidly to raise performance. In 2012, He et al. [7] pointed out that Wu et al.'s scheme suffered from the impersonation attack to the insider attack. In order to overcome this weakness, they also proposed a more secure authentication scheme for TMIS. Following these two works, the different authentication protocols [14, 23, 28] were presented to ensure that the data cannot be distorted by an illegal entity.

With regard to the confidentiality of data, in public key cryptography, a public key infrastructure (PKI) is responsible for providing an assurance through the certificates issued by a certification authority (CA). However, this PKI must manage the certificate in revocation, storage, distribution and verification, which places a huge cost on the entity [10]. To avoid these disadvantages, Shamir [19] put forward the notion of identity based public key cryptography (ID-PKC) by deriving the user's public key directly from its identity information, such as email address and IP address. Moreover, Boneh and Franklin [4] presented a practical identity based encryption (IBE) firstly. Nevertheless, the inherent key escrow problem in ID-PKC is a great drawback [15]. Al-Riyami and Paterson [1] introduced a new paradigm called certificateless public key cryptography (CL-PKC) to get rid of the above flaws. Then, they proved that their certificateless encryption (CLE) is secure in the random oracle model. Based on that scheme, Guo et al. [6] proposed a provably secure CLE scheme for TMIS, which protects the confidentiality of the PHR efficiently.

A common characteristic of above schemes is that each protocol satisfies only one requirement of HIPAA. In 2002, Lynn [12] proposed an authenticated IBE firstly, which integrated the authentication with encryption on the basis of Boneh-Franklin's IBE system [4] and ensured the integrity and confidentiality of the data simultaneously. Unfortunately, there is a security defect that the private key generator (PKG) has the ability of impersonating any user to recover confidential messages. After that, Cheng and Comley [5] constructed an authenticated CLE to prevent the malicious PKG from eavesdropping the privacy information. In addition, as a special authenticated encryption, the signcryption also achieves the same purpose [27]. Barbosa and Farshim [2] proposed the first certificateless signcryption scheme in 2008. However, their construction is vulnerable to the malicious-but-passive key generation center (KGC) at-tacks. In the same year, Wu and Chen [25] designed a more efficient certificateless signcryption scheme and introduced the public verifiability into it. Shamila et al. [17] claimed that the scheme in [25] could not provide the confidentiality of data. Liu et al. [11] introduced an efficient certificateless signcryption scheme with the security proof in the standard model. But Sharmila et al. and Weng et al. [18, 24] pointed out that their security proof is not sound and the scheme is in fact insecure. In 2015, Huang et al. [8] proposed a new efficient convertible multi-authenticated encryption scheme for mobile communication which the signature was cooperatively produced by a group of signers instead of a signal signer. Based on factoring and discrete logarithms,Tsai et al. [21] recently designed a publicly verifiable authenticated encryption scheme. They also claimed that even if either factoring or discrete logarithms is broken, their scheme still could keep the authentication, integration and confidentiality of the message.

1.2 Our Contributions

In this paper, we put forward an authenticated CLE (Auth-CLE) scheme in THS for protecting PHR. The authentication phase is added in decryption to protect the integrity and confidentiality of ciphertext at the same time. Furthermore, we prove that our scheme is secure in the random oracle model, provided that the bilinear Diffie-Hellman (BDH) problem is intractable. At last, we compare the cost of the computation and communication between our proposal and others by the evaluations and experiments and it concludes that our protocol offers better performances in efficiency.

The remainder of this paper is organized as follows. Section 2 addresses some preliminaries such as bilinear pairing, complexity assumption and the model of Auth-CLE. Section 3 proposes an Auth-CLE scheme and proves its security in the random oracle model. Section 4 compares the proposed scheme with some other related schemes from two points. Finally, we conclude the paper in Section 5.

2 Preliminaries

2.1 Bilinear Pairing

Let G_1 be a cyclic additive group generated by a point P, whose order is p, G_2 be a multiplicative group of the same order. Assuming that the bilinear pairing is a map $\hat{e}: G_1 \times G_1 \rightarrow G_2$ with the following properties:

Bilinearity: For any $X, Y \in G_1$ and $a, b \in Z_p$, we have $\hat{e}(aX, bY) = \hat{e}(X, Y)^{ab}$.

Non-degeneracy: For any $X, Y \in G_1$, $\hat{e}(X, Y) \neq 1_{G_2}$, where 1_{G_2} denotes the identity element of the group G_2.

Computability: There exists an efficient algorithm to compute $\hat{e}(X, Y)$ for any $X, Y \in G_1$.

2.2 Complexity Assumption

Considering the following computational hardness assumption in $< G_1, G_2, \hat{e} >$ as above, which is the basis of our scheme's security.

Definition 1. *Bilinear Diffie-Hellman (BDH) problem: Given* $< P, xP, yP, zP > \in G_1$ *with uniformly random choices of* $x, y, z \in Z_p^*$*, compute* $\hat{e}(P, P)^{xyz} \in G_2$.

The BDH assumption is that there is no polynomial time algorithm that can solve the BDH problem with non-negligible probability.

Let algorithm $\mathcal{A}$ be a BDH adversary who has an advantage ε in solving the BDH problem if $Pr[\mathcal{A}(< P, xP, yP, zP >) = \hat{e}(P, P)^{xyz}] = \varepsilon$. This probability is measured over random choices of $x, y, z \in Z_p^*$ and the point P. Adversary $\mathcal{A}$ solves the BDH problem with ε if and only if the advantage of $\mathcal{A}$ is greater than ε. The BDH problem is said to be ε-intractable if there is no algorithm that $\mathcal{A}$ solves this problem with ε.

2.3 Syntax

Different from the traditional CL-PKE scheme in [1], an Auth-CLE scheme consists of seven probabilistic, polynomial time (PPT) algorithms: *Setup*, *Partial-Private-Key-Extract*, *Set-Secret-Value*, *Set-Private-Key*, *Set-Public-Key*, *Authenticated-Encrypt* and *Authenticated-Decrypt*. These algorithms are defined as follows:

Setup: On input a security parameter 1^k, this algorithm returns the system parameters *params*, master public key *mpk* and the master secret key *msk*. The system parameters *params* include the plaintext space $\mathcal{M}$ and the ciphertext space $\mathcal{C}$. After this algorithm is over, the KGC publishes *params* and *mpk*, then keeps the *msk* secretly.

Partial-Private-Key-Extract: On input *params*, *msk* and an identity ID for the entity, KGC executes this algorithm and returns the partial private key D_{ID} to entity via a confidential and authentic channel.

Set-Secret-Value: On input *params* and an identity ID, entity executes this algorithm and returns entity's secret value x_{ID}.

Set-Private-Key: On input *params*, entity's partial private key D_{ID} and secret value x_{ID}, this algorithm returns the entity's full private key SK_{ID}.

Set-Public-Key: On input *params*, *mpk* and entity's secret value x_{ID}, this algorithm re-turns the public key PK_{ID} to the entity.

Authenticated-Encrypt: Running by a sender. On input *params*, message $M \in \mathcal{M}$, the receiver's identity ID_R, the public keys of receiver PK_{ID_R} and the secret value of sender x_{ID_S}, this algorithm returns a ciphertext $C \in \mathcal{C}$.

Authenticated-Decrypt: Running this deterministic algorithm by a receiver. On input *params*, ciphertext $C \in \mathcal{C}$, the sender's public key PK_{ID_S} and a private key of receiver's SK_{ID_R}, this algorithm returns and verifies a message $M \in \mathcal{M}$, which is either a plaintext message or a "*Reject*" message.

2.4 Security model for Auth-CLE

In the Auth-CLE, there are two types of adversaries with different capabilities, Type I and Type II adversaries. A difference between these two attackers is that $\mathcal{A}_I$ does not have access to the master secret key of KGC while $\mathcal{A}_{II}$ does have. Specifically, the adversary $\mathcal{A}_I$ in Type I represents a normal third party attacker against the Auth-CLE scheme. That is, $\mathcal{A}_I$ is not allowed to access to the master secret key but it may request the public keys and replace them with the values of its choice. By contrast, adversary $\mathcal{A}_{II}$ in Type II represents a malicious KGC who can generate the partial private keys of users, and it is allowed to have access to the master secret key but not replace a public key.

Definition 2. *An Auth-CLE scheme is IND-CCA secure if neither polynomial bounded adversary $\mathcal{A}$ of Type I nor Type II has a non-negligible advantage against the challenger in the following game:*

Setup: The challenger $\mathcal{CH}$ takes a security parameter 1^k as inputs and runs the *Setup* algorithm, then it sends the resulting system parameters *params* and *mpk* to $\mathcal{A}$. If $\mathcal{A}$ is of Type I, $\mathcal{CH}$ keeps the master secret key *msk* to itself. Otherwise, returns *msk* to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ is given access to the following oracles:

1) Partial-Key-Extract-Oracle: Upon receiving a partial key query for a user's identity ID, $\mathcal{CH}$ computes D_{ID} and returns it to $\mathcal{A}$. (Note that it is only useful to Type I adversary.)
2) Private-Key-Request-Oracle: Upon receiving a private key query for a user's identity ID, $\mathcal{CH}$ computes SK_{ID} and returns it to $\mathcal{A}$. It outputs $\perp$ (denotes failure) if the user's public key has been replaced (in the case of Type I adversary).
3) Public-Key-Request-Oracle: Upon receiving a public key query for a user's identity ID, $\mathcal{CH}$ computes PK_{ID} and returns it to $\mathcal{A}$.
4) Public-Key-Replace-Oracle: For identity ID and a valid public key, $\mathcal{A}$ replaces the associated user's public key with the new one of its choice (this is only for Type I adversary). The new value will be recorded and used by $\mathcal{CH}$ in the coming computations or responses to the adversary's queries.
5) Authenticated-Decryption-Oracle: On input a ciphertext and an identity, $\mathcal{CH}$ returns the correct decryption of ciphertext using the private key corresponding to the current value of the public key associated with the identity of the user, even if the corresponding public key for the user ID has been replaced.

Challenge Phase: Once $\mathcal{A}$ decides that *Phase 1* is over, it outputs and submits two messages (M_0, M_1), together with a challenge identity ID^* of uncorrupted secret key. Note that $\mathcal{A}$ is not allowed to know the private key of ID^* in anyway. The challenger $\mathcal{CH}$ picks a random bit $\beta \in 0, 1$ and computes C^*, which is the encryption of M_β under the current public key PK_{ID^*} for ID^*. If the output of the encryption is $\perp$, $\mathcal{A}$ immediately losses the game. Otherwise, C^* is delivered to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ issues a second sequence of queries as in *Phase* 1. A decryption query on the challenge ciphertext for C^* the combination of ID^* and PK_{ID^*} is not allowed.

Guess: Finally, $\mathcal{A}$ outputs its guess β' for β. The adversary wins the game if $\beta' = \beta$ and the advantage of $\mathcal{A}$ in this game is defined to be $Adv(\mathcal{A}) = | \ Pr(\beta' = \beta) - \frac{1}{2} \ |$. The adversary $\mathcal{A}$ breaks an IND-CCA secure Auth-CLE scheme with $(q_H, q_{par}, q_{pub}, q_{prv}, q_D, \varepsilon)$ if and only if the guessing advantage of $\mathcal{A}$ that makes q_H times to the random oracle $H(\cdot)$, q_{par} times *Partial-Key-Extract-Oracle*, q_{pub} times *Public-Key-Request-Oracle*, q_{prv} times *Private-Key-Request-Oracle* and q_D times *Authenticated-Decryption-Oracle* queries is greater than ε. The scheme is said to be $(q_H, q_{par}, q_{pub}, q_{prv}, q_D, \varepsilon)$-IND-CCA secure if there is no attacker $\mathcal{A}$ that breaks IND-CCA secure scheme with $(q_H, q_{par}, q_{pub}, q_{prv}, q_D, \varepsilon)$.

3 Our Protocol

In this section, we propose an Auth-CLE scheme to protect the integrity and confiden-tiality of data between the patient and the doctor.

3.1 Construction

The proposed Auth-CLE scheme consists of the following seven PPT algorithms.

Setup: Let G_1, G_2 be cyclic groups of prime order p with an arbitrary generator $P \in G_1$, $\hat{e} : G_1 \times G_1 \to G_2$ be a bilinear pairing. The MS selects $s \in Z_p^*$ at random and computes $P_{pub} = sP$ as master public key. Then, it chooses three collision resistant hash functions $H_1 : \{0,1\}^l \to G_1^*$, $H_2 : G_2 \to \{0,1\}^m$ and $H_3 : G_1 \times \{0,1\}^m \to G_1^*$, where l, m denotes the bit-length of identity and plaintext respectively. The system parameters are $params = \{G_1, G_2, \hat{e}, P, P_{pub}, H_1, H_2, H_3\}$ and the master secret key is $msk = s$.

Partial-Private-Key-Extract: On input patient's identity $\text{ID}_P \in \{0,1\}^l$, MS computes $Q_{ID_P} = H_1(\text{ID}_p)$ and sends the partial private key $D_{ID_P} = s \cdot Q_{ID_P} \in G_1^*$ to patient via a secure channel.

Set-Secret-Value: On input *params*, doctor's identity ID_D and patient's identity ID_P, doctor picks a secret value $\omega \in Z_p^*$ and returns $x_{ID_D} = \omega$ as his/her secret value. Correspondingly, the patient chooses $x_{ID_P} = \upsilon \in Z_p^*$ as his/her secret value.

Set-Private-Key: On input *params*, D_{ID_D} and x_{ID_D}, D_{ID_P} and x_{ID_P}, the doctor obtains the private key SK_{ID_D} by computing $SK_{ID_D} = \omega \cdot D_{ID_D}$. The patient gets his/her private key $SK_{ID_P} = \upsilon \cdot D_{ID_P}$.

Set-Public-Key: On input *params*, *mpk*, x_{ID_D} and x_{ID_P}, this algorithm returns $PK_{ID_D} = \omega P_{pub} = \omega sP$, $PK_{ID_P} = \upsilon P_{pub} = \upsilon sP$ as the public keys of doctor and patient respectively.

Authenticated-Encrypt: To encrypt $M \in \{0,1\}^m$, the doctor selects a random value $r \in Z_p^*$ and computes

$$Q_{ID_P} = H_1(ID_P),$$

$$\begin{aligned} c_1 &= r \cdot P, \\ c_2 &= M \oplus H_2(\hat{e}(H_1(ID_P), PK_{ID_P})^r), \\ c_3 &= H_3(\omega \cdot PK_{ID_P}, M). \end{aligned}$$

Then, set the ciphertext to $C = (c_1, c_2, c_3)$ and transmit it to the patient via the WMSN.

Authenticated-Decrypt: To decrypt ciphertext $C = (c_1, c_2, c_3)$ for the patient with private key SK_{ID_P}, he/she computes

$$M' = c_2 \oplus H_2(\hat{e}(SK_{ID_P}, c_1)).$$

After that, check $c_3 = H_3(\upsilon \cdot PK_{ID_D}, M')$. If not, reject the ciphertext. Otherwise, output M' as plaintext. Consistency of the scheme is clear since

$$\begin{aligned} \hat{e}(H_1(ID_P), PK_{ID_P})^r &= \hat{e}(H_1(ID_P), \upsilon sP)^r \\ &= \hat{e}(H_1(ID_P), P)^{\upsilon sr} \\ &= \hat{e}(\upsilon s H_1(ID_P), rP) \\ &= \hat{e}(SK_{ID_P}, c_1) \end{aligned}$$

by bilinearity.

3.2 Confidentiality Analysis

Theorem 1. *Given H_1, H_2 and H_3 are three collision resistant hash functions. The Auth-CLE scheme is IND-CCA secure in the random oracle model assuming that the BDH problem is intractable.*

This theorem following from two lemmas will show that our Auth-CLE scheme is secure against the Type I and Type II attacker whose behaviors are as described in **Definition 2**.

Lemma 1. *The Auth-CLE scheme is $(q_{H_1}, q_{H_2}, q_{par}, q_{pub}, q_{prv}, q_D, \varepsilon_I)$-IND-CCA secure against the Type I attacker $\mathcal{A}$ in the random oracle assuming the BDH problem is ε'_I-intractable, where*

$$\varepsilon'_I > \frac{1}{q_{H_2}}\left(\frac{2\varepsilon_I}{e(q_{prv} + q_{par} + 1)} - \frac{q_D q_{H_1}}{2^l} - \frac{q_D}{p}\right).$$

Proof. In this lemma, a Type I $\mathcal{A}$ models an "outside" adversary $\mathcal{A}_I$, who replaces the public key of arbitrary identities but cannot corrupt the master secret key.

Let $\mathcal{A}_I$ be a Type I IND-CCA adversary against our scheme. Suppose $\mathcal{A}_I$ has the advantage ε'_I, makes q_{H_i} queries to random oracle $H_i (i = 1, 2)$ and q_D decryption queries. We show how to construct a PPT algorithm $\mathcal{B}$ to solve the BDH problem with instance of (P, aP, bP, cP) by interacting with $\mathcal{A}_I$.

At the beginning, $\mathcal{B}$ simulates the algorithm *Setup* for $\mathcal{A}_I$ by supplying $\mathcal{A}_I$ with $params = \{G_1, G_2, \hat{e}, P, P_{pub}, H_1, H_2, H_3\}$, where H_1, H_2 and H_3 are random oracles that will be controlled by $\mathcal{B}$. $\mathcal{B}$ chooses an index I uniformly at random with $1 \le I \le q_{H_1}$.

The $\mathcal{A}_I$ adversary may make queries of the random oracles $H_i(i = 1, 2)$ at any time during its attack. $\mathcal{B}$ responds as follows:

H_1 queries: $\mathcal{B}$ maintains a list of tuples $< \mathrm{ID}_i, Q_i, t_i >$ in H_1-**List** L_1. On receiving a query ID_i to H_1, $\mathcal{B}$ responds as follows:

1) If ID_i already appears on the list L_1 in a tuple $< \mathrm{ID}_i, Q_i, t_i >$, $\mathcal{B}$ responds Q_i as an answer.
2) Otherwise, if $i \neq I$, choose $t_i \in Z_p^*$ at random and compute $Q_i = t_i P$, add $< \mathrm{ID}_i, Q_i, t_i >$ to L_1, then return Q_i as an answer.
3) If $i = I$, add $< \mathrm{ID}_i, Q_i = aP, * >$ to L_1 and return $Q_i = aP$ as an answer (where "$*$" denotes the arbitrary value).

H_2 queries: $\mathcal{B}$ maintains a list of tuples $< \mathrm{ID}_i, e_i, R_i >$ in H_2-List L_2. On receiving a query $< \mathrm{ID}_i, e_i >$ to H_2, $\mathcal{B}$ responds as follows:

1) If ID_i already appears on the list L_2 in a tuple $< \mathrm{ID}_i, e_i, R_i >$, $\mathcal{B}$ responds R_i as an answer.
2) Otherwise, pick $R_i \in \{0, 1\}^m$at random, add $< \mathrm{ID}_i, e_i, R_i >$ to L_2 and return R_i as an answer.

Phase 1: $\mathcal{A}_I$ issues a sequence of polynomially bounded number of the following oracle queries.

Partial-Key-Extract-Oracle: $\mathcal{B}$ maintains a **PartialKeyList** of tuples $< \mathrm{ID}_i, D_i >$. On receiving a query ID_i, $\mathcal{B}$ responds as follows:

1) If $< \mathrm{ID}_i, D_i >$ exist in **PartialKeyList**, return D_i as an answer.
2) Otherwise, pick i at random, so that $\Pr[i \neq I]=\delta$ (δ will be determined later.). If $i \neq I$, search L_1 for a tuple $< \mathrm{ID}_i, Q_i, t_i >$, compute $D_i = t_i P_{pub}$, add $< \mathrm{ID}_i, D_i >$ to the **PartialKeyList** and return D_i as an answer.
3) If $i = I$, return "Abort" and terminate.

Private-Key-Request-Oracle: $\mathcal{B}$ maintains a **PrivateKeyList** of tuples $< \mathrm{ID}_i, x_i, D_i >$. On receiving a query ID_i, $\mathcal{B}$ responds as follows:

1) If $< \mathrm{ID}_i, x_i, D_i >$ exist in **PrivateKeyList**, return $< x_i, D_i >$ as an answer.
2) Otherwise, if $i \neq I$, run the simulation algorithm *Public-Key-Request-Oracle* to get a tuple $< \mathrm{ID}_i, x_i, PK_i >$ and *Partial-Key-Extract-Oracle* to get a tuple $< \mathrm{ID}_i, D_i >$, add $< \mathrm{ID}_i, x_i, D_i >$ to the **PrivateKeyList** and return $< x_i, D_i >$ as an answer. (Note that if the corresponding public key has been replaced, such a private key query is not allowed.)
3) If $i = I$, return "Abort" and terminate.

Public-Key-Request-Oracle: $\mathcal{B}$ maintains a **PublicKeyList** of tuples $< \mathrm{ID}_i, x_i, PK_i >$. On receiving a query ID_i, $\mathcal{B}$ responds as follows:

1) If $< \mathrm{ID}_i, x_i, PK_i >$ exist in **PublicKeyList**, return PK_i as an answer.
2) Otherwise, if $i \neq I$ choose $x_i \in Z_p^*$ and compute $PK_i = x_i P_{pub} = bP$, add $< \mathrm{ID}_i, x_i, PK_i >$ to the **PublicKeyList** and return PK_i as an answer.
3) If $i = I$, add $< \mathrm{ID}_i, *, PK_i = bQ_i >$ to **PublicKeyList** and return PK_i as an answer.

Public-Key-Replace-Oracle: $\mathcal{A}_I$ may replace any public key with a new value of its choice and $\mathcal{B}$ records all the changes.

Auth-Decryption-Oracle: On receiving a query $< \mathrm{ID}_i, PK_i, C >$, where $C = (c_1, c_2, c_3)$. $\mathcal{B}$ responds as follows:

1) If $i \neq I$ and PK_i is the correct public key (not a replaced one), $\mathcal{B}$ decrypts C by using the corresponding private key.
2) Otherwise, search L_2 for a tuple $< \mathrm{ID}_i, e_i, R_i >$. If such a tuple exists, $\mathcal{B}$ retrieves the related R_i to compute $M = c_2 \oplus R_i$ and returns M as an answer.
3) Otherwise, $\mathcal{B}$ picks $R_i \in \{0, 1\}^m$ at random, computes $M = c_2 \oplus R_i$ and returns M as an answer. Add $< \mathrm{ID}_i, e_i, R_i >$ to L_2.

Challenge Phase: $\mathcal{A}_I$ then outputs two messages (M_0, M_1) and a challenge identity ID^*. On receiving a challenge query $< \mathrm{ID}^*, (M_0, M_1) >$:

1) If $\mathrm{ID}^* \neq \mathrm{ID}_i$, $\mathcal{B}$ aborts the game.
2) Otherwise, $\mathcal{B}$ sets $c_1^* = cP$ and defines $c_2^* = H_2(\hat{e}(SK_{ID^*}, c_1^*)) \oplus M_\beta$, $c_3^* = H_3(\omega^* \cdot PK_{ID'}, M_\beta)$ (note that $\mathcal{B}$ does not know c and ω^*, $PK_{ID'}$ is the sender's public key), returns $C^* = (c_1^*, c_2^*, c_3^*)$ as a target ciphertext.

Phase 2: $\mathcal{A}_I$ requests in the same way as in *Phase* 1. Moreover, no *Private-Key-Request-Oracle* on ID^* is allowed and no *Auth-Decryption-Oracle* can be made on the ciphertext C^* for the combination of identity ID^* and public key PK_{ID^*} that encrypted plaintext M_β.

Guess: $\mathcal{A}_I$ should make a guess β' for β. The adversary wins the game if $\beta' = \beta$. Then, $\mathcal{B}$ will be able to solve the BDH problem by computing

$$\hat{e}(PK_i, c_1^*) = \hat{e}(bQ_i, cP) = \hat{e}(baP, cP) = \hat{e}(P, P)^{abc}.$$

Analysis: By **Ask**H_2^*, we denote the event that (ID_i^*, e_i^*) has been queried to H_2. Also, by **Ask**H_1^*, we denote the event that ID_i^* has been queried to H_1. If **Ask**H_2^* happens, $\mathcal{B}$ will be able to solve the BDH problem by choosing a tuple $< \mathrm{ID}_i, e_i, R_i >$ from L_2 and computing $H_2(e_i)$ with the probability at least $\frac{1}{q_{H_2}}$. Hence, we have $\varepsilon'_I \geq \frac{1}{q_{H_2}}\Pr[\mathbf{Ask}H_2^*]$.

It is easy to notice that if $\mathcal{B}$ does not abort these oracles, the simulations of *Partial-Key-Extract-Oracle*, *Private-Key-Request-Oracle*, *Public-Key-Request-Oracle* and the simulated target ciphertext is identically distributed as the real one from the construction.

Then, we evaluate the simulation of *Auth-Decryption-Oracle*. If a public key PK_i has not been replaced nor PK_i has not been produced by reselecting $x_i \in Z_p^*$, the simulation is perfect as $\mathcal{B}$ knows the private key SK_i corresponding to PK_i. Otherwise, a simulation error may occur while $\mathcal{B}$ running the decryption oracle simulation specified above. Let **DecErr** be this event. Suppose $< \mathrm{ID}_i, PK_i, C >$, where $C = (c_1, c_2, c_3)$ and $PK_i = x_i P_{pub}$, has been issued as a valid decryption query. Even if C is valid, there is a possibility that C can be produced without querying (ID_i, e_i) to H_2.

Let **Valid** be an event that C is valid, $\mathbf{Ask}H_2$ and $\mathbf{Ask}H_1$ respectively be events that (ID_i, e_i) has been queried to H_2 and ID_i has been queried to H_1. Since **DecErr** is an event that $\mathbf{Valid}|\neg\mathbf{Ask}H_2$ happens during the entire simulation and q_D *Auth-Decryption-Oracle* queries are made, we have $\Pr[\mathbf{DecErr}]=q_D\Pr[\mathbf{Valid}|\neg\mathbf{Ask}H_2]$. However,

$$\begin{aligned}\Pr[\mathbf{Valid}|\neg\mathbf{Ask}H_2] &\leq \Pr[\mathbf{Valid} \wedge \mathbf{Ask}H_1|\neg\mathbf{Ask}H_2] \\ &+ \Pr[\mathbf{Valid} \wedge \neg\mathbf{Ask}H_1|\neg\mathbf{Ask}H_2] \\ &\leq \Pr[\mathbf{Ask}H_1|\neg\mathbf{Ask}H_2] \\ &+ \Pr[\mathbf{Valid}|\neg\mathbf{Ask}H_1 \wedge \neg\mathbf{Ask}H_2] \\ &\leq \frac{q_{H_1}}{2^l} + \frac{1}{p}\end{aligned}$$

Let the event $(\mathbf{Ask}H_2^* \vee \mathbf{DecErr})|\neg\mathbf{Abort}$ be denoted by **E**, where **Abort** denotes an event that $\mathcal{B}$ aborts during the simulation. The probability $\neg\mathbf{Abort}$ that happens is given by $\delta^{q_{prv}+q_{par}}(1-\delta)$ which is maximized at $\delta = 1 - \frac{1}{q_{prv}+q_{par}+1}$. Hence, we have $\Pr[\neg\mathbf{Abort}] \leq \frac{1}{e(q_{prv}+q_{par}+1)}$, where e denotes the base of the natural logarithm.

If **E** does not happen, it is clear that $\mathcal{A}_I$ does not gain any advantage greater than $\frac{1}{2}$ to guess β due to the randomness of the output of the random oracle H_2. Namely, we have $\Pr[\beta' = \beta|\neg\mathbf{E}] \leq \frac{1}{2}$.

By Definition 2, we have

$$\begin{aligned}\varepsilon_I &< |\Pr[\beta' = \beta] - \frac{1}{2}| \\ &= |\Pr[\beta' = \beta|\neg\mathbf{E}]\Pr[\neg\mathbf{E}] + \Pr[\beta' = \beta|\mathbf{E}]\Pr[\mathbf{E}] - \frac{1}{2}| \\ &\leq |\frac{1}{2}\Pr[\neg\mathbf{E}] + \Pr[\mathbf{E}] - \frac{1}{2}| \\ &= |\frac{1}{2}(1 - \Pr[\mathbf{E}]) + \Pr[\mathbf{E}] - \frac{1}{2}| \\ &= \frac{1}{2}\Pr[\mathbf{E}] \\ &\leq \frac{\Pr[\mathbf{Ask}H_2^*] + \Pr[\mathbf{Ask}H_1^*|\neg\mathbf{Ask}H_2^*] + \Pr[\mathbf{DecErr}]}{2\Pr[\neg\mathbf{Abort}]} \\ &\leq \frac{e(q_{prv} + q_{par} + 1)}{2}(q_{H_2}\varepsilon'_I + \frac{q_D q_{H_1}}{2^l} + \frac{q_D}{p})\end{aligned}$$

Consequently, we obtain

$$\varepsilon'_I > \frac{1}{q_{H_2}}(\frac{2\varepsilon_I}{e(q_{prv} + q_{par} + 1)} - \frac{q_D q_{H_1}}{2^l} - \frac{q_D}{p}).$$

□

Lemma 2. *The Auth-CLE scheme is* $(q_{H_1}, q_{H_2}, q_{pub}, q_{prv}, q_D, \varepsilon_{II})$*-IND-CCA secure against the Type II attacker* $\mathcal{A}$ *in the random oracle assuming the BDH problem is* ε'_{II}*-intractable, where*

$$\varepsilon'_{II} > \frac{1}{q_{H_2}}(\frac{2\varepsilon_{II}}{e(q_{prv} + 1)} - \frac{q_D q_{H_1}}{2^l} - \frac{q_D}{p}).$$

Proof. In this lemma, a Type II $\mathcal{A}$ models an "inside" adversary $\mathcal{A}_{II}$, who has access to *msk* but cannot replace public key of entity.

Let $\mathcal{A}_{II}$ be a Type II IND-CCA adversary against our scheme. Suppose $\mathcal{A}_{II}$ has the advantage ε'_{II}, makes q_{H_i} queries to random oracle $H_i (i = 1, 2)$ and q_D decryption queries. We show how to construct a PPT algorithm $\mathcal{B}$ to solve the BDH problem with instance of (P, aP, bP, cP) by interacting with $\mathcal{A}_{II}$.

At the beginning, $\mathcal{B}$ simulates the algorithm *Setup* for $\mathcal{A}_{II}$ by supplying $\mathcal{A}_{II}$ with *params* $= \{G_1, G_2, \hat{e}, P, P_{pub}, H_1, H_2, H_3\}$, where H_1, H_2 and H_3 are random oracles that will be controlled by $\mathcal{B}$. $\mathcal{B}$ chooses an index I uniformly at random with $1 \leq I \leq q_{H_1}$.

The adversary $\mathcal{A}_{II}$ may make queries of the random oracles $H_i (i = 1, 2)$ at any time during its attack. $\mathcal{B}$ responds as follows:

H_1 **queries:** $\mathcal{B}$ maintains a list of tuples $< \mathrm{ID}_i, Q_i >$ in H_1-**List** L_1. On receiving a query ID_i to H_1, $\mathcal{B}$ responds as follows:

1) If ID_i already appears on the list L_1 in a tuple $< \mathrm{ID}_i, Q_i >$, $\mathcal{B}$ responds Q_i as an answer.
2) Otherwise, if $i \neq I$, choose $Q_i \in G_1^*$ at random and add $< \mathrm{ID}_i, Q_i >$ to L_1, return Q_i as an answer.

3) If $i = I$, add $< \mathrm{ID}_i, Q_i = aP, * >$ to L_1 and return $Q_i = aP$ as an answer (where "$*$" denotes the arbitrary value).

H_2 queries: $\mathcal{B}$ maintains a list of tuples $< \mathrm{ID}_i, e_i, R_i >$ in H_2-List L_2. On receiving a query $< \mathrm{ID}_i, e_i >$ to H_2, $\mathcal{B}$ responds as follows:

1) If ID_i already appears on the list L_2 in a tuple $< \mathrm{ID}_i, e_i, R_i >$, $\mathcal{B}$ responds R_i as an answer.

2) Otherwise, pick $R_i \in \{0,1\}^m$ at random, add $< \mathrm{ID}_i, e_i, R_i >$ to L_2 and return R_i as an answer.

Phase 1: $\mathcal{A}_{II}$ issues a sequence of polynomially bounded number of the following oracle queries.

Private-Key-Request-Oracle: $\mathcal{B}$ maintains a **PrivateKeyList** of tuples $< \mathrm{ID}_i, x_i, D_i >$. On receiving a query ID_i, $\mathcal{B}$ responds as follows:

1) If $< \mathrm{ID}_i, x_i, D_i >$ exist in **PrivateKeyList**, return $< x_i, D_i >$ as an answer.

2) Otherwise, pick i at random, so that $\Pr[i \neq I] = \delta$ (δ is the same as it in the proof of **Lemma 1**). If $i \neq I$, run the simulation algorithm *Public-Key-Request-Oracle* to get a tuple $< \mathrm{ID}_i, x_i, PK_i >$, pick $s \in Z_p^*$ and compute $D_i = sH_1(\mathrm{ID}_i)$, add $< \mathrm{ID}_i, x_i, D_i >$ to the **PrivateKeyList** and return $< x_i, D_i >$ as an answer.

3) If $i = I$, return "Abort" and terminate.

Public-Key-Request-Oracle: $\mathcal{B}$ maintains a **PublicKeyList** of tuples $< \mathrm{ID}_i, x_i, PK_i >$. On receiving a query ID_i, $\mathcal{B}$ responds as follows:

1) If $< \mathrm{ID}_i, x_i, PK_i >$ exist in **PublicKeyList**, return PK_i as an answer.

2) Otherwise, if $i \neq I$ choose $x_i \in Z_p^*$, compute $PK_i = x_iP$, add $< \mathrm{ID}_i, x_i, PK_i >$ to the **PublicKeyList**, return PK_i as an answer.

3) If $i = I$, set $PK_i = bP_{pub} = sbP$, add $< \mathrm{ID}_i, *, PK_i >$ to **PublicKeyList** and return PK_i as an answer.

Auth-Decryption-Oracle: On receiving a query $< \mathrm{ID}_i, PK_i, C >$, where $C = (c_1, c_2, c_3)$. $\mathcal{B}$ responds as follows:

1) If $i \neq I$, $\mathcal{B}$ decrypts C by using the private key $< x_i, D_i >$.

2) Otherwise, search L_2 for a tuple $< \mathrm{ID}_i, e_i, R_i >$. If such a tuple exists, $\mathcal{B}$ retrieves the related R_i to compute $M = c_2 \oplus R_i$ and returns M as an answer.

3) Otherwise, $\mathcal{B}$ picks $R_i \in \{0,1\}^m$ at random, computes $M = c_2 \oplus R_i$ and returns M as an answer. Add $< \mathrm{ID}_i, e_i, R_i >$ to L_2.

Challenge Phase: $\mathcal{A}_{II}$ then outputs two messages (M_0, M_1) and a challenge identity ID^*. On receiving a challenge query $< \mathrm{ID}^*, (M_0, M_1) >$:

1) If $\mathrm{ID}^* \neq \mathrm{ID}_i$, $\mathcal{B}$ aborts the game.

2) Otherwise, $\mathcal{B}$ sets $c_1^* = s^{-1}cP$ and defines $c_2^* = H_2(\hat{e}(SK_{ID^*}, c_1^*)) \oplus M_\beta$, $c_3^* = H_3(\omega^* \cdot PK_{ID'}, M_\beta)$ (note that $\mathcal{B}$ does not know c and ω^*, $PK_{ID'}$ is the sender's public key), returns $C^* = (c_1^*, c_2^*, c_3^*)$ as a target ciphertext.

Phase 2: $\mathcal{A}_{II}$ requests the same methods that it used in *Phase* 1. Moreover, no *Private-Key-Request-Oracle* on ID^* is allowed and no *Auth-Decryption-Oracle* can be made on the ciphertext C^* for the combination of identity ID^* and public key PK_{ID^*} that encrypted plaintext M_β.

Guess: $\mathcal{A}_{II}$ should make a guess β' for β. The adversary wins the game if $\beta' = \beta$. Then, $\mathcal{B}$ will be able to solve the BDH problem by computing

$\hat{e}(aPK_{ID^*}, c_1^*) = \hat{e}(absP, s^{-1}cP) = \hat{e}(P,P)^{abss^{-1}c} = \hat{e}(P,P)^{abc}$.

Analysis: Similar to *Analysis* in the proof of **Lemma 1**.

□

Consequently, we obtain

$$\varepsilon'_{II} > \frac{1}{q_{H_2}}\left(\frac{2\varepsilon_{II}}{e(q_{prv}+1)} - \frac{q_D q_{H_1}}{2^l} - \frac{q_D}{p}\right).$$

These two lemmas complete the proof of **Theorem 1**.

3.3 Unforgeability Analysis

Theorem 2. *Suppose H_1, H_2 and H_3 are three collision resistant hash functions, and $\mathcal{A}$ is an adversary that can forge a ciphertext with advantage ε by making q_{H_3} queries to random oracle H_3 and q_D queries to* Auth-Decryption-Oracle. *Then, there exists a PPT algorithm $\mathcal{B}$ that can solve the BDH problem with advantage at least*

$$Adv(\mathcal{B}) = \frac{\varepsilon}{\left(\frac{2}{q_{H_3}(q_{H_3}-1)}\right)^2 q_D}$$

Proof. We show how to construct a PPT algorithm $\mathcal{B}$ to solve the BDH problem with instance of (P, aP, bP, cP) by interacting with $\mathcal{A}$.

At the beginning, $\mathcal{B}$ simulates the algorithm *Setup* for $\mathcal{A}$ by supplying $\mathcal{A}$ with *params* $= \{G_1, G_2, \hat{e}, P, P_{pub}, H_1, H_2, H_3\}$, where H_1, H_2 and H_3 are random oracles that will be controlled by $\mathcal{B}$. There are two lists L_i that store the answers on H_i queries ($i = 2, 3$) and a list of possible bilinear pairing answers L_e.

H_2 queries: $\mathcal{B}$ maintains a list of tuples $< \mathrm{ID}, e, R >$ in **H_2-List** L_2. On receiving a query $< \mathrm{ID}, e >$ to H_2, $\mathcal{B}$ responds as follows:

1) If ID already appears on the list L_2 in a tuple $< \text{ID}, e, R >$, $\mathcal{B}$ responds R as an answer.
2) Otherwise, pick $R \in \{0,1\}^m$ randomly, add $< \text{ID}, e, R >$ to L_2 and return R as an answer.

H_3 queries: $\mathcal{B}$ supplies $\mathcal{A}$ with (P, aP) and sets $1 \leq i \neq j \leq q_{H_3}$. $\mathcal{B}$ responds as follows:

1) If it is the ith query, respond with bP and call ID a guessed identity.
2) If it is the jth query, respond with cP and call ID a guessed identity.
3) Otherwise, choose a random $W \in G_1^*$ and add $< T, M, W >$ to L_3, return $W = dP$ as an answer.

Private-Key-Request-Oracle: On input identity ID, $\mathcal{B}$ responds as follows:

1) If ID is a guessed identity, $\mathcal{B}$ fails.
2) Otherwise, the list L_3 must contain the tuple $< T, M, W >$ for some $d \in Z_p^*$ and $\mathcal{B}$ outputs adP as its private key.

Authenticated-Encryption-Oracle: Suppose $\mathcal{A}$ issues an encryption query for a plaintext M between doctor ID_D and patient ID_P.

1) If ID_D and patient ID_P are the guessed identity,$\mathcal{B}$ picks three random values $\{R, T\} \in G_1^*$ and $S \in \{0,1\}^m$, return $C = (R, S, H_3(T, M))$ as a ciphertext.
2) Otherwise, assume ID_P is not a guessed identity, the list L_3 must contain the tuple $< T, M, W >$ for some $d \in Z_p^*$. Then, the patient's private key is adP and the ciphertext is computed as described in the *Authenticated-Encrypt*. Return the ciphertext to $\mathcal{A}$.

Authenticated-Decryption-Oracle: Suppose $\mathcal{A}$ issues a decryption query for a ciphertext $C = (c_1, c_2, c_3)$ between doctor and patient.

1) If ID_P is the guessed identity, L_2 is examined for an entry of the form $< \text{ID}_P, e, R >$ for some e. If such an entry exists, e is added to the list L_e. $\mathcal{A}$ is notified that C is invalid even if it is valid.
2) Otherwise, assume ID_P is not a guessed identity, the list L_3 must contain the tuple $< T, M, W >$ for some $d \in Z_p^*$ and adP is its private key. Then, the ciphertext is decrypted as described in the *Authenticated-Decryption* algorithm. If this ciphertext is valid, the correspondingly plaintext is given to $\mathcal{A}$ and it wins.

Eventually, $\mathcal{A}$ decides that the game is over. If the list L_e is empty, $\mathcal{B}$ fails. Otherwise, $\mathcal{B}$ outputs a random element of L_e.

Analysis: The probability that $\mathcal{A}$ has never issued *Private-Key-Request-Oracle* query on one of the guessed identity is at least $C^2_{q_{H_3}}$. If $\mathcal{A}$ has submitted a valid ciphertext, it forges a ciphertext successfully between the guessed identity with at least probability $C^2_{q_{H_3}}$ (but this ciphertext is actually invalid).

If $e = \hat{e}(P, P)^{abc}$ is not on the list L_e, $\mathcal{A}$ cannot generate a correct forgery for H_2 is a random oracle. Therefore, the probability that $\mathcal{A}$ queries $H_2(e)$ is at least ε. If this happens, $\mathcal{B}$ cannot fail and output the correct value with probability at $\frac{1}{q_D}$. Then, we have

$$Adv(\mathcal{B}) = \frac{\varepsilon}{(\frac{2}{q_{H_3}(q_{H_3}-1)})^2 q_D}.$$

□

4 Comparisons

4.1 Computation Costs

First, we evaluate the computational cost of our scheme and others [2, 11, 17, 25] through combined implementation and simulation. We test the cryptographic operations in bilinear pairing, exponentiation and scalar multiplication (without considering the addition of two points, the hash function and exclusive-OR operations), and detailed time results on a PC with the Intel Core i5-2400 at a frequency of 3.1 GHz with 3 GB memory and Windows XP operating system, using the MIRACL (Version 5.6.1, [16]). For bilinear pairing, in order to implement it in practice efficiently, we employ the Fast-Tate-Pairing in MIRACL, which is defined over the MNT curve E/F_q [13] with characteristic a 160-bit prime and embedding degree 4. For ECC-based protocols, we choose the recommended parameters "secp192k1" [20]. Furthermore, we denote the length of an element in a multiplicative group to be 1024-bit. Based on the above parameter settings, the average running time of each operation in 100 times is obtained and demonstrated in Table 1. Then, the total running time to finish one round of "*Authenticated-Encrypt and Decrypt*" is illustrated in Table 2. For example, in *Authenticated-Encrypt* and *Decrypt* of our scheme, there are two bilinear pairing operations, one exponentiation and three scalar multication in the additive cyclic group in all; thus the total operation time is $2 \times 2.65 + 1 \times 3.75 + 3 \times 0.78 = 11.39$ ms. These indicate our scheme is more scalable and efficient than existing works.

4.2 Communication Costs

Next, we analyze the communication cost in terms of the bandwidth of the transmitted ciphertext (or signcrypted text). Suppose that the output of one-way hash function is 160-bit. In our protocol, the ciphertext contains two hash values and one point, thus the bandwidth of it is $(160 \times 2 + 192)/8 = 64$ bytes. In Barbosa and Farshim's

Table 1: Cryptography operation time

Fast-Tate-Pairing	**Exponentiation**	**Scalar Multiplication**
2.65 ms	3.75 ms	0.78 ms

Table 2: Comparison of the related schemes

Scheme	**Auth-Enc**	**Auth-Dec**	**Bandwidth**	**Total Time**
[2]	1P+1E+4S	5P+1S	68 bytes	23.55 ms
[11]	4E	5P	532 bytes	28.25 ms
[17]	5E	7E	276 bytes	45.00 ms
[25]	1P+4E+3S	3P+4S	108 bytes	31.06 ms
Ours	1P+1E+2S	1P+1S	64 bytes	11.39 ms

scheme [2], the signcrypted text contains two points and one hash, the bandwidth of it is $(192 \times 2 + 160)/8 = 68$ bytes. In Liu et al.'s scheme [11], the signcrypted text contains four elements of multiplicative group and one bilinear pairing, the bandwidth of it is $(1024 \times 4 + 160)/8 = 532$ bytes. In the scheme of [17], the signcrypted text contains two elements of multiplicative group and one hash value, the bandwidth of it is $(1024 \times 2 + 160)/8 = 276$ bytes. At last, in Wu and Chen's scheme [25], the signcrypted text contains two points, two hash values and one element in additive group, and therefore the bandwidth of it is $(192 \times 2 + 160 \times 2 + 160)/8 = 108$ bytes. The detailed comparison results are also listed in Table 2, which shows that the bandwidth of our scheme is the smallest.

5 Conclusions

In this paper, we propose an authenticated certificateless encryption scheme to ensure the confidentiality and integrity of the transmitted information between patient and doctor in THS, which satisfies the privacy requirements of HIPAA. Moreover, it is proved that our protocol is IND-CCA secure and the information cannot be forged in the random oracle model, relative to the hardness of the BDH problem. By the evaluation and simulation, a comparison in Table 2 concludes that the proposed scheme is advantageous over the related schemes in computation and communication cost.

Acknowledgments

This study was supported by the National Nature Science Foundation of China under grant NSFC 11501343, and the Open Foundation of State key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications) under grant SKLNST-2016-2-11. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Advances in Cryptology (ASIACRYPT'03)*, LNCS, vol. 2894, pp. 452–473, 2003.

[2] M. Barbosa and P. Farshim, "Certificateless signcryption," in *Proceedings of the 2008 ACM symposium on Information, Computer and Communications Security*, pp. 369–372, 2008.

[3] L. J. Blumberg, L. M. Nichols, "The health insurance portability and accountability act of 1996: Summary of provisions and anticipated effects", *Journal of Medical Practical Management*, vol. 14, no. 1, pp. pp. 13-8, 1998.

[4] D. Boneh and M. Franklin, "Identity based encryption from the Weil pairing," in *Advances in Cryptology (CRYPTO'01)*, LNCS, vol. 2139, pp. 213–229, 2001.

[5] Z. H. Cheng and R. Comley, "Efficient certificateless public key encryption," *IACR Cryptology ePrint Archive*, 2005. (`http://eprint.iacr.org/2005/012.pdf`)

[6] R. Guo, Q. Y. Wen, H. X. Shi, Z. P. Jin and H. Zhang, "An efficient and provably-secure certificateless public key encryption scheme for telecare medicine in-formation systems," *Journal of Medical Systems*, vol. 37, pp. 9965, 2013.

[7] D. B. He, J. H. Chen and R. Zhang, "A more secure authentication scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 36, pp. 1989–1995, 2012.

[8] H. F. Huang, P. H. Lin, and M. H. Tsai, "Convertible Multi-authenticated Encryption Scheme for Data Communication," *International Journal of Network Security*, vol. 17, no. 1, pp. 40–48, 2015.

[9] R. S. Istepanian, E. Jovanov and Y. T. Zhang, "Guest editorial introduction to the special section on m-health: Beyond seamless mobility and global wireless health-care connectivity," *IEEE Transac-*

tions on Information Technology in Biomedicine, vol. 8, no. 4, pp. 405–414, 2004.

[10] A. V. N. Krishna, A. H. Narayana, K. M. Vani, "Window method based cubic spline curve public key cryptography," *International Journal of Electronics and Information Engineering*, vol. 4, no. 2, pp. 94–102, 2016.

[11] Z. H. Liu, Y. P. Hu, X. S. Zhang and H. Ma, "Certificateless signcryption scheme in the standard model," *Information Sciences*, vol. 180, no. 3, pp. 452–464, 2010.

[12] B. Lynn, "Authenticated identity-based encryption," *IACR Cryptology ePrint Archive*, 2002. (`http://eprint.iacr.org/2002/072.pdf`)

[13] A. Miyaji, M. Nakavayashi and S. Takano, "New explicit conditions of elliptic curve traces for FR-reduction," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E84-A, no. 5, pp. 1234–1243, 2001.

[14] J. Moon, Y. Choi, J. Kim and D. Won, "An improvement of robust and efficient biometrics based password authentication scheme for telecare medicine information systems using extended chaotic maps," *Journal of Medical Systems*, vol. 40, no. 3, 2016.

[15] J. H. Oh, K. K. Lee and S. J. Moon, "How to solve key escrow and identity revocation in identity based encryption schemes," in *The First International Conference of Information Systems Security*, vol. 3803, pp. 290–303, 2005.

[16] M. Scott, *Miracl Library*, Apr. 15, 2017. (`http://certivox.com/`)

[17] S. S. D. Selvi, S. S. Vivek and C. P. Rangan, "Cryptanalysis of certificateless signcryption schemes and an efficient construction without pairing," *IACR Cryptology ePrint Archive*, 2009. (`http://eprint.iacr.org/2009/298.pdf`)

[18] S. S. D. Selvi, S. S. Vivek and C. P. Rangan, "Security weaknesses in two certificateless signcryption schemes," *IACR Cryptology ePrint Archive*, 2010. (`http://eprint.iacr.org/2010/092.pdf`)

[19] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology (CRYPTO'84)*, LNCS, vol. 196, pp. 47–53, 1985.

[20] The Certicom Research, *SEC2: Recommended Elliptic Curve Domain Parameters*, Version 1.5, 2005.

[21] C. Y. Tsai, C. Y. Liu, S. C. Tsaur and M. S. Hwang, "A Publicly Verifiable Authenticated Encryption Scheme Based on Factoring and Discrete Logarithms," *International Journal of Network Security*, vol. 19, no. 3, pp. 443–448, 2017.

[22] K. K. Venkatasubramanian, A. Banerjee and S. K. S. Gupta, "Plethysmogram-based secure inter-sensor communication in body area networks," in *IEEE Military Communications Conference*, pp. 1–7, 2008.

[23] J. H. Wei, X. X. Hu and W. F. Liu, "An improved authentication scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 36, pp. 3597–3604, 2012.

[24] J. Weng, G. X. Yao, R. H. Deng, M. R. Chen and X. X. Li, "Cryptanalysis of a certificateless signcryption scheme in the standard model," *Information Sciences*, vol. 181, no. 3, pp. 661–667, 2011.

[25] C. H. Wu and Z. X. Chen, "A new efficient certificateless signcryption scheme," in *International Symposium on Information Science and Engineering*, vol. 1, pp. 661–664, 2008.

[26] Z. Y. Wu, Y. C. Lee, F. Lai, H. C. Lee and Y. Chung, "A secure authentication scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 36, pp. 1529–1535, 2012.

[27] Y. Zheng, "Digital signcryption or how to achieve cost (signature & encryption) $\leq$ cost(signature) + cost (encryption)," in *Advances in Cryptology (CRYPTO'97)*, LNCS, vol. 1294, pp. 165–179, 1997.

[28] Z. A. Zhu, "An efficient authentication scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 36, no. 6, pp. 3833–3838, 2012.

Biography

Rui Guo received the Ph.D degrees in the Department of State Key Laboratory of Networking and Switch Technology, Beijing University of Posts and Telecommunications in 2014. Now, he is a lecturer in National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications. His present research interests include cryptography, information security and WSN.

Huixian Shi received the B.S. and Ph.D degrees in Department of Mathematics and Information Science from Shaanxi Normal University, Xi'an, China, in 2007 and 2013, respectively. Now she is a associate professor in Department of Department of Mathematics and Information Science in Shaanxi normal University. Her present research interests include model checking, fuzzy logic and uncertainty reasoning.