Remote Data Integrity Checking and Sharing in Cloud-Based Health Internet of Things

Huaqun WANG††††a, Keqiu LI†††b, Nonmembers, Kaoru OTA††††c, Member, and Jian SHEN††d, Nonmember

SUMMARY In the health IoT (Internet of Things), the specialized sensor devices can be used to monitor remote health and notify the emergency information, e.g., blood pressure, heart rate, etc. These data can help the doctors to rescue the patients. In cloud-based health IoT, patients’ medical/health data is managed by the cloud service providers. Secure storage and privacy preservation are indispensable for the outsourced medical/health data in cloud computing. In this paper, we study the integrity checking and sharing of outsourced private medical/health records for critical patients in public clouds (ICS). The patient can check his own medical/health data integrity and retrieve them. When a patient is in coma, some authorized entities and hospital can cooperate to share the patient’s necessary medical/health data in order to rescue the patient. The paper studies the system model, security model and concrete scheme for ICS in public clouds. Based on the bilinear pairing technique, we design an efficient ICS protocol. Through security analysis and performance analysis, the proposed protocol is provably secure and efficient.

key words: remote data integrity checking, public cloud, data sharing

1. Introduction

IoT can be used in the healthcare applications. It plays a significant role from managing chronic diseases at one end of the spectrum to preventing disease at the other, such as clinical care and remote monitoring. In the life of every people, the generated medical data and the health data is very large. In the medical research, the researchers will process massive medical data. Massive data processing and information security risk call for the new computation model as an alternative to conventional computing. As a new computation model, cloud computing has become a reality along with the development of network and computer technology. Cloud computing provides a flexible, dynamic, resilient and cost effective infrastructure for the business environments. For the patients and hospital, the remote medical/health data integrity checking and sharing can be performed by the patients and hospital. Since the medical/health data is outside the control of patients and hospital, cloud service providers are more responsible for the security of application services, especially in public clouds. Based on public cloud server’s benefits and security risks, the paper focuses on privacy-preserving remote medical/health data integrity checking and sharing for critical patients in public clouds.

Throughout the paper, “privacy-preserving ICS” is simplified as “ICS”, “public cloud server” is simplified as “PCS”.

1.1 Motivation

Along with the development of IoT, it is widely used in the field of healthcare. By using the IoT, massive health data is generated. On the other hand, when the patient goes to the hospital, the medical data is also generated. These generated medical/health data can be used in order to rescue the patients. In the cloud-based health IoT, the medical/health data is stored in public clouds. In public clouds, the hospitals and patients access PCS via Internet. To protect the hospital’s benefits, it is important to prevent unauthorized entities to check these data integrity. If the malicious competitors can check these data integrity, they can evaluate these data size. Then, the competitors can evaluate the hospital’s daily business volume. Then, the competitors can take measures to prevail the hospital. In order to protect these data, the remote data integrity checking can only be performed by the hospital besides of the patient. Usually, in order to protect the patients’ privacy, only the patient can retrieve his own medical/health data. Unfortunately, the critical patients may be in a coma before reaching the hospital. In this case, the hospital and the patient’s relatives should be able to cooperate to share the patient’s medical/health data in order to rescue him. This real social requirement motivates us to study privacy-preserving remote medical/health data integrity checking and sharing for critical patients in public clouds.

1.2 Related Work

The rapidly developed IoT has been widely applied in the medical/health field [1], [2]. Based on the special properties of medical/health data, when IoT is used in the medical/health field, some security risks emerge [3], [4]. In 2015, Wu et al. proposed employment of the regenerating codes
and symmetric-key encryption with a Blom based key management [5]. Their scheme can repair the lost fragment and protect data secrecy. When the medical/health data is stored on the PCS, the remote data integrity checking and sharing become an urgent security problem which needs to be solved. In 2014, Wang et al. proposed the fair remote retrieval of outsourced private medical records in electronic health networks [6]. Lu et al. proposed privacy preserving opportunistic computing framework for mobile-health care emergency [7]. In 2016, He et al. introduced a novel network security architecture for cloud computing considering characteristics of cloud computing [8]. Their scheme can protect external and internal traffics in cloud computing. It can also attain flexible scalability with respect to virtual middle box load and achieve fault-tolerant among virtual middle box failure.

Since these privacy-aware medical/health data is stored on the untrusted third party PCS, it is important to keep these data uncorrupted and privacy-preserving. As an efficient remote data integrity checking model, the concept of provable data integrity (PDP) was proposed by G. Ateniese et al. in 2007 [9]. They also designed two concrete statically secure PDP schemes. Since PDP is a very efficient remote data integrity checking model, many researchers proposed a variety of PDP security models and concrete schemes [10]–[14].

Privacy-preservance is an important security issue in cloud computing. In 2013, Wang et al. proposed the privacy-preserving public auditing in cloud computing [15]. Nabeel et al. proposed privacy preserving policy based content sharing in public clouds [16]. Guo et al. proposed the variable threshold-value authentication architecture for wireless mesh networks [17]. In the medical/health data cloud storage, privacy preservation is especially important. Only the patient and the authorized entities can get the patient’s medical/health data. PDP protocols are classified into two categories: private PDP and public PDP. Some private information is necessary to perform private PDP. The private information is not needed for public PDP.

1.3 Our Contribution and Organization

In this paper, we propose the novel concept of ICS. Then, we give the formal system model and security model of ICS. By using the bilinear pairings, an efficient ICS protocol is designed. Through security analysis and performance analysis, our ICS protocol is provably secure and efficient.

The rest of the paper is organized as follows. Section 2 introduces the preliminaries and Sect. 3 describes our pairing-based ICS protocol and analyzes its security. Section 4 gives the performance analysis of our ICS protocol. Finally, Sect. 5 concludes this paper.

2. Preliminaries

ICS system model and security model are proposed in this section. After that, bilinear pairings and some corresponding difficult problems are also described below.

2.1 System Model and Security Model

ICS system consists of four different network entities: Patient, Hospital, AuthSet, PCS. They can be identified below.

1. Patient, whose medical/health data will be uploaded to PCS for maintenance and computation, is individual human being;
2. Hospital, which diagnoses the patients and generates the medical/health data for the patients, can be physicians or medical institutions;
3. AuthSet, which is the patient’s authorized entity set, can cooperate with the hospital to share the patient’s medical/health data;
4. PCS, which is managed by cloud service provider, has significant storage space and computation resource to maintain the patients’ data.

Definition 1 (ICS). A ICS protocol is a collection of seven polynomial time algorithms (SetUp, EncTagGen, CheckTagSign, GenProof, GenRetr, CheckProof, Retrieval) among PCS, Hospital, Patient, and AuthSet such that:

1. SetUp(1k) → (params, sk, pk) is a probabilistic system parameters and key generation algorithm to setup the protocol where k is a security parameter, sk is the secret key and pk is the public key. params is the public system parameters. IDj gets its private/public key pair (xj, Xj) and a symmetric encryption key skj. The hospital’s private/public key pair are (y, Y). PCS’s private/public key pair are (z, Z). The patient IDj prepares the warrant ωj and the corresponding certificate Sign(ωj). IDj creates skj’s secret shares and distributes the shares among AuthSetj.
2. EncTagGen(xj, skj, Y, Mj) → (Ti,j) is run by the patient IDj to generate the verification metadata, where Mj is IDj’s medical/health data and Ti,j is IDj’s i-th block tag.
3. CheckTagSign(Fij, Ti,j, ωj, Sign(ωj)) → (“success”, “failure”) is run by PCS to check whether the medical/health block-tag pair (Fij, Ti,j) and the warrant-signature pair (ωj, Sign(ωj)) are valid or not.
4. GenProof(F, χ, χ, Z) → Vp is run by PCS to generate the proof of integrity. F is the stored file and Z is the stored tags. χ is the challenge from the verifier.
5. GenRetr(F, χ, Z) → Vr is run by PCS to generate the retrieval response.
6. CheckProof(χ, χ, Z, Vp) → (“success”, “failure”) is run by the patient IDj or the hospital to check the data integrity.
7. Retrieval((xj, skj) or (y, skj’s valid share set), Xj, Y, Z, χ, Vr) → Mj is run by the patient IDj or the cooperation of hospital and AuthSetj to retrieve IDj’s remote medical/health data Mj.

The following definitions 2, 3, 4 define the security against the malicious PCS forgery, restrictive remote medical/health data integrity checking and restrictive retrievabil-
ity.

**Definition 2 (Integrity Against Malicious PCS).** ICS protocol satisfies the integrity if any PPT (probabilistic polynomial time) adversary \( A \) (i.e., malicious PCS) can win the ICS game only with negligible probability. ICS game between the challenger \( C \) and the adversary \( A \) is described below:

1. **Setup:** In this phase, the system parameters are created. Let the patient set be \( \mathcal{P} \). The patient \( ID_j \)'s private/public key pair \( (x_j, X_j) \), the hospital's private/public key pair \( (y, Y) \) and PCS's private/public key pair \( (z, Z) \) are created where \( ID_j \in \mathcal{P} \). Let \( ID_j \)'s symmetric encryption/decryption key be \( sk_j \). The private keys \( x_j, y \) are kept secret. The parameters \( (z, X_j, Y, Z, \text{params}, ID_j \in \mathcal{P}) \) are sent to \( A \).

2. **First-Phase Queries:** \( A \) adaptively queries \( C \) below:
   - **Hash query.** Input the hash queries adaptively, \( C \) responds the corresponding hash values to \( A \).
   - **Tag query.** Input the medical/health block \( F_{i,j} \) for the patient \( ID_p \), \( C \) calculates the tag \( T_{i,j} \) by \( \text{TagGen}(x_j, F_{i,j}) \) and sends it to \( A \). Without loss of generality, let \( (F_{i,j}, T_{i,j}) \) be the queried block-tag pair set and \( I_1 = \{(i, j)\} \) in First-Phase Queries.

3. **Challenge:** \( C \) generates a data integrity challenge \( \text{chal}_p \) which defines the challenged block-tag pair index collection \( I_v = \{(i_1, j_1), (i_2, j_2), \cdots, (i_c, j_c)\} \), where \( I_v \subseteq I_1 \) and \( c \) is a positive integer. \( C \) is queried to provide the proof of integrity checking for the medical/health blocks \( F_{i_1,j_1}, \cdots, F_{i_c,j_c} \).

4. **Second-Phase Queries:** Similar to First-Phase Queries. Let the queried medical/health block-tag pair set be \( \{F_{i,j}, T_{i,j}\} \) and \( I_2 = \{(i, j)\} \) in Second-Phase Queries. The restriction is that \( I_v \not\subseteq \{(i, j)\} \).

5. **Forge:** Finally, \( A \) forges a data integrity proof \( V_p \) for the medical/health blocks indicated by \( \text{chal}_p \) and returns \( V_p \) to \( C \).

In the above ICS game, we say that \( A \) wins if

\[
\Pr\left[ \text{CheckProof}(y, X_j, Y, Z, \text{chal}_p, V_p, ID_j \in \mathcal{P}) \rightarrow \text{"success"} \right] \geq \frac{1}{p(k)}
\]

where \( p(k) \) is a polynomial of the security parameter \( k \).

**Definition 3 (Restrictive Integrity Checking).** In the remote medical/health data integrity checking, only the following restrictive entities have the ability to perform the data integrity checking protocol:

1. The hospital can perform the remote medical/health data integrity checking for all the patients.
2. The individual patient can perform the integrity checking only for his own remote medical/health data.
3. Except the hospital, patient and PCS, the other entity cannot perform the remote medical/health data integrity checking.

**Definition 4 (Restrictive Retrievability).** In the remote medical/health data retrievability, the patient \( ID_j \) creates the retrievability control set \( \mathcal{R} = \{R_{1,j}, R_{2,j}, \cdots, R_{n,j}\} \). It satisfies the following requirements:

1. The patient \( ID_j \) can retrieve his own remote medical/health data by himself, i.e., \( \{ID_j\} \in \mathcal{R} \).
2. For the other retrievability control set \( R_{i,j} \), the entities in \( R_{i,j} \) can cooperate to retrieve \( ID_j \)'s remote medical/health data under the help of the hospital.
3. For any entity set \( S_A \) if \( R_{i,j} \not\subseteq S_A \) for every \( 1 \leq i \leq n \), the entities in \( S_A \) cannot retrieve \( ID_j \)'s remote medical/health data even if they collude.

A secure ICS protocol also needs to guarantee that after validating the PCS-generated proof, the verifier can also be convinced that all of his outsourced data has been kept intact with a high probability. The following security definition gives the security property.

**Definition 5 ((\( \rho, \delta \)) Security).** An ICS protocol is \((\rho, \delta)\)-secure if PCS corrupted \( \rho \) fraction of the whole medical/health blocks, the probability that the corrupted blocks are detected is at least \( \delta \).

### 2.2 Bilinear Pairings and Difficult Problems

Let \( \mathcal{G}_1 \) and \( \mathcal{G}_2 \) be two cyclic multiplicative groups with the same prime order \( q \). Let \( \hat{e} : \mathcal{G}_1 \times \mathcal{G}_1 \rightarrow \mathcal{G}_2 \) be a bilinear map. \( \hat{e} \) can be constructed by the modified Weil or Tate pairings on elliptic curves [18], [19]. The group with such a map \( \hat{e} \) is called a bilinear group, on which the Computational Diffie-Hellman (CDH) problem is assumed hard while the Decisional Diffie-Hellman (DDH) problem is easy [20].

**Definition 6 (Gap Diffie-Hellman (GDH) Group).** Let \( g \) is the generator of \( \mathcal{G}_1 \). Given \( g, g^a, g^b, g^c \in \mathcal{G}_1 \) while \( a, b, c \in \mathbb{Z}_q^* \) are unknown, it is recognized that there exists an efficient algorithm to determine whether \( ab \equiv c \mod q \) holds by verifying \( \hat{e}(g^a, g^b) = \hat{e}(g, g^c) \) in polynomial time (DDH problem), while there exist no efficient algorithms to compute \( g^{ab} \in \mathcal{G}_1 \) with non-negligible probability within polynomial time (CDH problem). An algorithm \( A \) is said to \((t, \epsilon)\)-break the CDH problem on \( \mathcal{G}_1 \) if \( A \) runs in time at most \( t \), and the following CDH advantage is at least \( \epsilon \).

\[
\text{Adv}_{A}^{\text{CDH}} = \Pr[ A(g, g^a, g^b) \rightarrow g^{ab} : \forall a, b \in \mathbb{Z}_q^* ] \geq \epsilon
\]

The probability is taken over the choice of \( a, b \) and \( A \)'s coin tosses. A group \( \mathcal{G}_1 \) is a \((t, \epsilon)\)-GDH group if the DDH problem on \( \mathcal{G}_1 \) is efficiently computable and no algorithm \((t, \epsilon)\)-breaks the CDH problem on \( \mathcal{G}_1 \).

**Definition 7 (Bilinear Diffie-Hellman Problem (BDHP) assumption).** Given \( g, g^a, g^b, g^c \) for unknown \( a, b, c \in \mathbb{Z}_q^* \), it is difficult to compute \( W = \hat{e}(g, g)^{abc} \in \mathcal{G}_2 \).
3. Our Proposed Pairing-Based ICS Protocol

3.1 ICS Protocol Construction

Our ICS protocol consists of seven phases: (SetUp, EncTagGen, CheckTagSign, GenProof, GenRetr, CheckProof, Retrieval) among PCS, Hospital, Patient, and AuthSet. Suppose there are \( n_p \) patients whose set is denoted as \( \mathcal{P} \), i.e., \( \mathcal{P} = \{ ID_1, ID_2, \ldots, ID_{n_p} \} \). Suppose the patient \( ID_j \) will upload \( n_j \) block-tag pairs. Denote the corresponding block-patient index set as \( \mathcal{B} \times \mathcal{P} = \{(i, j), 1 \leq j \leq n_p, 1 \leq i \leq n_j \} \). Let \( n \) denote the whole block number, i.e., \( n = \sum_{j=1}^{n_p} n_j \). Let \( f \) and \( \Omega \) be two pseudo-random functions, and let \( \pi \) be a pseudo-random permutation. Let \( H, h \) be two cryptographic hash functions. They are described below.

\[
\begin{align*}
    f & : \mathbb{Z}_q^* \times \{1, 2, \ldots, n\} \rightarrow \mathbb{Z}_q^*, \\
    \Omega & : \mathbb{Z}_q^* \times (\mathcal{B} \times \mathcal{P}) \rightarrow \mathbb{Z}_q^*, \\
    \pi & : \mathbb{Z}_q^* \times \{1, 2, \ldots, n\} \rightarrow \mathcal{B} \times \mathcal{P}, \\
    H & : \mathbb{G}_2 \times \{0, 1\}^* \rightarrow \mathbb{Z}_q, \quad h : \mathbb{Z}_q^* \rightarrow \mathcal{G}_1^*
\end{align*}
\]

Let \( g \) be a generator of \( \mathcal{G}_1 \). Without loss of generality, we only consider the patient \( ID_j \) in the concrete scheme construction. Denote \( ID_j \)'s medical/health data as \( M_j \). The patient \( ID_j \) picks a random \( x_j \in \mathbb{Z}_q^* \) as his private key and computes \( X_j = g^{x_j} \) as his public key. On the other hand, \( ID_j \) picks a random \( s_{kj} \in \mathbb{Z}_q^* \) as his symmetric encryption/decryption key. The hospital picks a random number \( y \in \mathbb{Z}_q^* \) as its private key and computes \( Y = g^y \) as its public key. PCS picks a random number \( z \in \mathbb{Z}_q^* \) as its private key and computes \( Z = g^z \) as its public key. The phases of our proposed ICS protocol are described below.

SetUp: The patient \( ID_j \) delegates his remote medical/health data retrieval capability to the authorized entity set \( \text{AuthSet}_j \). Suppose that \( n_j \) entities share \( ID_j \)'s symmetric key and \( th_j \) entities can retrieve \( ID_j \)'s remote medical/health data by cooperating with the hospital. Let the authorized entity set be \( \text{AuthSet}_j = \{ D_{j1}, D_{j2}, \ldots, D_{jth_j} \} \) and \( D_{jj} \in \mathbb{G}_1^* \) (it can be realized by using the hash function \( H : \{0, 1\}^* \rightarrow \mathbb{Z}_q^* \)). Let \( (\text{Sign}, \text{Verify}) \) be secure signature/verification algorithm pairs. In order to restrict the authorized entities’ behaviors, for every entity \( D_{ji}, ID_j \) creates the warrant \( \omega_i \) and \( \text{Sign} (\omega_i) \) by using his own private key \( x_j \). The warrant \( \omega_i \) describes the rules which must be obeyed by \( D_{ji} \). For the symmetric encryption key \( s_{kj}, ID_j \) generates the corresponding shares below.

1. Pick the random \( a_{j1}, a_{j2}, \ldots, a_{jn_j-1} \in \mathbb{Z}_q^* \) and get the following polynomial with the order \( th_j - 1 \):

\[
\mathcal{F}_j(x) = s_{kj} + \sum_{i=1}^{th_j-1} a_{ji} x^i
\]

2. Compute the share \( ss_{j,i} = \mathcal{F}_j(D_{ji}) \) for every entity \( D_{ji} \). Then, \( ID_j \) sends \( (ss_{j,i}, \omega_i, \text{Sign} (\omega_i)) \) to \( D_{ji} \).

3. Every entity \( D_{ji} \) verifies whether the warrant-signature pair \( (\omega_j, \text{Sign} (\omega_j)) \) is valid. If it is valid, \( D_{ji} \) keeps the warrant-signature pair \( (\omega_j, \text{Sign} (\omega_j)) \) and the corresponding secret share \( ss_{j,i} \) of the symmetric encryption/decryption key \( s_k \) of the patient \( ID_j \).

Finally, \( ID_j \) picks a public random element \( u_j \in \mathcal{G}_1 \). The final system parameters are

\[
\text{params} = \{ \mathcal{G}_1, \mathcal{G}_2, \hat{\epsilon}, f, \Omega, \pi, h, X_j, Y, Z, u_j, q, ID_j \in \mathcal{P} \}
\]

EncTagGen \( (x_j, s_{kj}, M_j, \Omega, \pi, Y, Z) \): After finishing the medical/health advices, the patient \( ID_j \) gets his own medical/health data \( M_j \). Taking use of the symmetric encryption algorithm, \( ID_j \) gets the ciphertext \( F_j = E_{sk_j}(M_j) \). Then, \( F_j \) is divided into \( n_j \) blocks, i.e., \( F_j = (F_{j1}, F_{j2}, \ldots, F_{jn_j}) \). For the block \( F_{ji} \), the patient \( ID_j \) computes \( t_j = H(\hat{\epsilon}(X_j, Y^y), \omega_j) \). When the above procedures are performed \( n_j \) times, all the block-tag pairs are created. At last, the patient \( ID_j \) uploads his block-tag pairs collection \( \{(F_{ji}, T_{ji}), 1 \leq i \leq n_j \} \) and the warrant-signature pairs \( (\omega_j, \text{Sign} (\omega_j)) \) to PCS. PCS stores the block-tag pairs and the warrant-signature pairs \( (\omega_j, \text{Sign} (\omega_j)) \). The patient deletes these block-tag pairs \( \{(F_{ji}, T_{ji}), 1 \leq i \leq n_j \} \) from its local storage.

CheckTagSign \( ((z, X_j, Y, \omega_j, \text{Sign} (\omega_j)), (F_{ji}, T_{ji}), 1 \leq i \leq n_j) \): Upon receiving \( (\omega_j, \text{Sign} (\omega_j)) \), PCS verifies its validity by using the corresponding verification algorithm \( \text{Verify} \). If it is invalid, output “failure”. Otherwise, for every \( 1 \leq i \leq n_j \), PCS computes \( \hat{t}_j = H(\hat{\epsilon}(X_j, Y^y), \omega_j) \). Then, it verifies whether the following formula holds: \( \hat{t}_j = \hat{\epsilon}(h(W_{ji} u_j^{F_{ji}}), X_j) \). If it holds, then PCS accepts it. Otherwise, PCS rejects it and responds “failure”.

GenProof \( (F, \text{chal}_p, \Sigma) \): Let the challenge be \( \text{chal}_p = (c, k_1, k_2) \) where \( 1 \leq c \leq n, k_1 \in \mathbb{Z}_q^*, k_2 \in \mathbb{Z}_q^* \). Let \( F \) be the set of the blocks. Let \( \Sigma \) be the set of the tags. The hospital queries PCS for medical/health data integrity proof of \( c \) file blocks. \( k_1 \) is used as the random key of the pseudo-random permutation \( \pi \). \( k_2 \) is used as the random key of the pseudo-random function \( f \). PCS performs the procedures below.

1. For \( 1 \leq i \leq c \), PCS computes the indices and coefficients below: \( (l_i, j_i) = \pi_k (i), a_i = f_{k_2} (i) \)

2. The set \( \{(l_i, j_i), 1 \leq i \leq c\} \) is divided into many subsets \( S_j \) based on the different patients. For the same patient \( ID_j \), let \( S_j \) be the subset \( \{(l_i, j_i), j_i \text{ is constant}, 1 \leq i \leq c\} \). Thus, \( S_j \) describes the challenged medical/health data blocks of the patient \( ID_j \). Denote \( S = \{S_j, 1 \leq i \leq c\} \).

Note: For the different \( i \), maybe, the mapped \( j \) are the same, i.e., there exist more challenged blocks for the patient \( ID_j \). Of course, maybe, there doesn’t exist challenged blocks for some patients.

3. For \( 1 \leq i \leq c \), compute
\[ T = \prod_{i=1}^{c} T_{l_i,j_i}, \quad \hat{F}_j = \sum_{(l_i,j_i) \in S_j} a_i F_{l_i,j_i} \]

4. Denote \( \hat{F} = \{ \hat{F}_j, S_j \in S \} \). Output \( V = (\hat{F}, T) \) to the hospital.

**GenRetr\( \text{(} F, \text{cha}_t, \Sigma \)\): Suppose the patient \( ID_j \) wants to retrieve his own medical/health data blocks \( (l_1, j_1), (l_2, j_2), \ldots, (l_i, j_i) \). \( ID_j \) sends the challenge \( cha_l = (l_1, j_1), (l_2, j_2), \ldots, (l_i, j_i) \) where \( k_2 \in \mathbb{Z}^* \). Upon receiving the retrieval challenge \( cha_l, \) from the patient \( ID_j \), PCS performs the procedures below:

1. For \( 1 \leq i \leq c \), compute the coefficients: \( a_i = f_{i,j}(i) \).
2. For \( 1 \leq i \leq c \), compute \( T = \prod_{j=1}^{c} T_{l_i,j_i} \).
3. Output \( V_r = (F_{l_1,j_1}, F_{l_2,j_2}, \ldots, F_{l_c,j_c}) \) to \( ID_j \).

When the hospital and \( AuthSet_j \) cooperate to query PCS to retrieve the patient \( ID_j \)'s medical/health data, they sends the challenge \( cha_l = (\omega_j, Sign(\omega_j), (l_1, j_1), (l_2, j_2), \ldots, (l_i, j_i), k_2) \) to PCS. PCS verifies the warrant-signature pair \((\omega_j, Sign(\omega_j))\). If it is valid and the query complies with the warrant \( \omega_j \), PCS performs the same procedures as \( ID_j \)'s retrieval query. Otherwise, rejects it.

**CheckProof\( \text{(} y, X_j, Y, Z, cha_l, V_r, ID_j \in \mathcal{P} \)\): Upon receiving the response \( V_r \) from PCS, the hospital performs the procedures below:

1. For \( 1 \leq i \leq c \), compute the indexes and coefficients below: \( (l_i, j_i) = n_{l_i,j_i}(i) \), \( a_i = f_{i,j}(i) \).
2. For \( 1 \leq i \leq c \), compute \( \hat{t}_j = H(\hat{e}(X_j, Z)^{t_j}, \omega_j) \).
3. Check whether the following formula holds.

\[
\hat{e}(T, g) \overset?= \prod_{S_j \in S} \hat{e} \left( \prod_{(l_i,j_i) \in S_j} h(\Omega_j, (l_i, j_i))^{a_i} u_{j_i}^{\hat{c}_i}, X_j \right) \tag{1}
\]

If (1) holds, then the hospital outputs “success”. Otherwise, the hospital outputs “failure”.

**Retrieval\( \text{(} x_j, sk_j \) or \( y, sk_j \)’s valid share set, \( X_j, Y, Z, cha_l, V_r \)\): The two cases can be considered. (1) The patient \( ID_j \) retrieves his own medical/health data. (2) The hospital and \( AuthSet_j \) cooperate to retrieve \( ID_j \)'s medical/health data.

The first case, \( ID_j \) retrieves his own medical/health data below:

1. For \( 1 \leq i \leq c \), compute the coefficients: \( a_i = f_{i, j}(i) \).
2. Compute \( \hat{F}_j = \sum_{i=1}^{c} a_i F_{l_i,j_i}, t_j = H(\hat{e}(X, Y) \times \omega_j) \).
3. Check whether the following formula holds.

\[
\hat{e}(T, g) \overset?= \hat{e} \left( \prod_{i=1}^{c} h(\Omega_j, (l_i, j_i))^{a_i} u_{j_i}^{\hat{c}_i}, X_j \right) \tag{2}
\]

If the formula (2) holds, the patient \( ID_j \) accepts the blocks \( F_j' = \{ F_{l_1,j_1}, F_{l_2,j_2}, \ldots, F_{l_c,j_c} \} \). Then, the corresponding plaintext \( M_j' = D_{sk_j}(F_j') \) can be retrieved by using the symmetric encryption key \( sk_j \). Otherwise, the patient \( ID_j \) rejects the response.

The second case, in the authorized set \( AuthSet_j \), suppose that \( th_j \) entities agree to retrieve the patient \( ID_j \)'s medical/health data. Let the \( th_j \) entities be \( D_{th_1}, D_{th_2}, \ldots, D_{th_{th_j}} \). Under the help of the hospital, they cooperate to perform the steps below:

1. For \( 1 \leq i \leq c \), compute the coefficients: \( a_i = f_{i,j}(i) \). After that, it computes \( \hat{F}_j = \sum_{i=1}^{c} a_i F_{l_i,j_i}, t_j = H(\hat{e}(X_j, Z)^{t_j}, \omega_j) \).
2. Check whether the following formula holds.

\[
\hat{e}(T, g) \overset?= \hat{e} \left( \prod_{i=1}^{c} h(\Omega_j, (l_i, j_i))^{a_i} u_{j_i}^{\hat{c}_i}, X_j \right) \tag{3}
\]

If the formula (3) holds, they accept the blocks \( F_j' = \{ F_{l_1,j_1}, F_{l_2,j_2}, \ldots, F_{l_c,j_c} \} \). By using their own shares, these \( th_j \) entities can compute \( sk_j \).

\[
sk_j = \sum_{i=1}^{th_j} \left( \prod_{j \in \mathbb{Z}^*} -\frac{D_{th_i}}{D_{th_j}} \right) u_{th_j}^{\hat{c}_{th_j}}
\]

The corresponding plaintext \( M_j' = D_{sk_j}(F_j') \) can be retrieved by using the symmetric encryption key \( sk_j \). Otherwise, reject the response.

3.2 Security Analysis

The correctness analysis and security analysis of our proposed ICS protocol are given by the lemmas and theorems below:

**Theorem 1.** If the patient \( ID_j \), hospital and PCS are honest and follow the proposed procedures, then any block-tag pair can pass PCS’s tag checking, i.e., \( \text{CheckTagSign} \) satisfies the correctness.

**Proof.** Since \( \text{Sign, Verify} \) is secure signature-verification algorithm pair, and \( (\omega_j, \text{Sign}(\omega_j)) \) is valid warrant-signature pair, thus, \( (\omega_j, \text{Sign}(\omega_j)) \) can pass the verification. According to the phases of \( \text{EncTagGen} \) and \( \text{CheckTagSign} \), the following formulas hold:

\[
\hat{t}_j = H(\hat{e}(X_j, Y)^{t_j}, \omega_j) = t
\]

\[
\hat{W}_{l_1,j} = \Omega_l(i, j) = \Omega_l(i, j) = W_{l_1,j}
\]

\[
\hat{e}(T, g) = \hat{e}(h(W_{l_1,j})F_{l_1,j}^{\omega_j}, g)
\]

\[
= \hat{e}(h(W_{l_1,j})F_{l_1,j}^{\omega_j}, g^{\omega_j}) = \hat{e}(h(W_{l_1,j})F_{l_1,j}^{\omega_j}, X_j)
\]

**Theorem 2.** If hospital and PCS are honest and follow the proposed procedures, the response \( V_r \) can pass the hospital’s data integrity checking, i.e., \( \text{CheckProof} \) satisfies the correctness.

**Proof.** Let the challenge be \( cha_l = (c, k_1, k_2) \). According to the phases of \( \text{EncTagGen} \) and \( \text{GenProof} \), we know that \( t_j = \)
\[ H(\hat{e}(X_j, Z)^{y_j}, \omega_j) = H(\hat{e}(Y, Z)^{y_i}, \omega_j) = t_j, \hat{W}_{i,j} = \Omega_j(i, j) = \Omega_j(i, j) = W_{i,j}. \] Thus,

\[
\hat{e}(T, g) = \hat{e}\left( \prod_{i=1}^{c} T_{i,j,i}^{a_i}, g \right) = \hat{e}\left( \prod_{S_h \subset S} \prod_{(i,j) \in S_h} T_{i,j,i}^{a_i}, g \right) = \hat{e}\left( \prod_{S_h \subset S} \prod_{(i,j) \in S_h} (h(W_{i,j})u_{i,j}^{F_{i,j}})^{a_i}, g \right) = \hat{e}\left( \prod_{S_h \subset S} \prod_{(i,j) \in S_h} h(\Omega_j(i, j))^{a_i}u_{i,j}^{F_{i,j}}, X_j \right)
\]

Thus, the proposed ICS protocol satisfies restrictive proof of integrity. □

**Theorem 3.** If the patient ID\(_j\) (or hospital and AuthSet\(_j\)) and PCS are honest and follow the proposed procedures, ID\(_j\) (or hospital and AuthSet\(_j\)) can retrieve the queried medical/health data, i.e., Retrieval satisfies the correctness.

**Proof.** When the patient ID\(_j\) queries to retrieve his own medical/health data, the verification formula (2) holds based on the theorem 2. Then, it is straightforward to get the medical/health data by decrypting the ciphertext using his own symmetric encryption key sk\(_j\).

When the hospital and AuthSet\(_j\) cooperate to retrieve ID\(_j\)’s medical/health data, by Lagrange interpolation formula, the sk\(_j\) can be obtained below:

\[ sk_j = \sum_{i=1}^{n} \left( \frac{n}{n_j} \prod_{j \neq j_i} \frac{-D_{j,i}}{D_{j,i} - D_{j,i}} \right) x_{S,j,i} \]

After that, since \( t_j = H(\hat{e}(X_j, Z)^{y_i}, \omega_j) = H(\hat{e}(Y, Z)^{y_i}, \omega_j) \), they can also get ID\(_j\)’s medical/health data by performs the similar procedures as the patient ID\(_j\). □

**Theorem 4 (Possession Against Malicious PCS).** On the GDH group \( G_1 \), based on the difficulty of CDH problem, the proposed ICS protocol is existentially unforgeable in the random oracle model. That is, the proposed ICS protocol satisfies the security property of provable data integrity against malicious PCS.

**Proof.** It is similar with the Ref. [10]. We omit it due to the page limits. □

**Theorem 5 (Restrictive Proof of Possession).** For the remote medical/health data, the proposed ICS protocol satisfies restrictive proof of integrity.

**Proof.** From theorem 2, the hospital can perform all the patients’ medical/health data integrity checking. For the patient ID\(_j\), he can compute the parameter \( t_j \) and \( W_{i,j} \). Based on the two parameters, ID\(_j\) can perform the proof of his own medical/health data integrity below: \( \hat{e}(T, g) \overset{}{\rightarrow} \hat{e}(\prod_{(i,j) \in S_h} h(\Omega_j(i, j))^{a_i}u_{i,j}^{F_{i,j}}, X_j) \).

Except for the hospital, the patients and PCS, the third party has no ability to get \( t_j \) based on the difficulty of BDHP. Thus, the third party can not also compute \( \Omega_j(i, j) \). Finally, the third party can not perform the verification equation:

\[
\hat{e}(T, g) \overset{}{\rightarrow} \prod_{S_h \subset S} \hat{e}\left( \prod_{(i,j) \in S_h} h(\Omega_j(i, j))^{a_i}u_{i,j}^{F_{i,j}}, X_j \right)
\]

Thus, the proposed ICS protocol satisfies restrictive proof of integrity. □

**Theorem 6 (Restrictive Retrieval).** For the remote medical/health data, the proposed ICS protocol satisfies the property of restrictive retrieval.

**Proof.** From the theorem 3, the patients have the ability to retrieve their own medical/health data. In the patient ID\(_j\)’s authorized entities, if at least \( n_1 \) entities agree to retrieve ID\(_j\)’s remote medical/health data, these authorized entities and the hospital have the ability to cooperate to retrieve ID\(_j\)’s remote medical/health data.

On the contrary, if less than \( n_1 \) – 1 authorized entities agree to retrieve ID\(_j\)’s data, they only succeed with negligible probability. According to the process of symmetric encryption key distribution, the function \( F_j \) has the order \( n_1 - 1 \). It can be determined by at least \( n_1 \) points. Less than \( n_1 \) – 1 authorized entities can provide less than \( n_1 \) – 1 points. \( F_j \) can not be determined and the symmetric encryption key sk\(_j\) can not also be determined. Thus, they have no ability to retrieve ID\(_j\)’s remote medical/health data. □

**Theorem 7.** The proposed ICS protocol is \( (\frac{d}{n}, 1 - \left( \frac{n-d}{n} \right)^d) \)-secure since the probability \( P_R \) of detecting the corruption satisfies:

\[ 1 - \left( \frac{n - d}{n} \right)^d \leq P_R \leq 1 - \left( \frac{n - c + 1 - d}{n - c} \right)^c \]

where PCS has stored \( n = n_1 + n_2 + \cdots + n_p \) block-tag pairs for \( n_p \) patients, PCS has corrupted \( d \) block-tag pairs, and the challenge is \( \text{chal}_p = (c, k_1, k_2) \).

**Proof.** It is similar with Ref. [9]. We omit it due to the page limits. □

### 4. Performance

We implemented our ICS scheme in order to demonstrate the effectiveness of our scheme. We used the C programming language with the GMP (GMP-5.0.5), Miracl and PBC (pbc-0.5.13) libraries. In the implementation, PCS ran on the laptop with the following features:

- **CPU:** Intel Core i7-3517U @ 1.90GHz
- **Physical Memory:** 4GB DDR3 1600MHz
Hospital, Patient and AuthSet ran on a laptop with the following features:

- CPU: CPU I PDC E6700 3.2GHz
- Physical Memory: DDR3 2G
- OS: Ubuntu 11.10 over VMWare-workstation-full-8.0.0

Our cryptographic choices were: i) an elliptic curve with 160-bit group order; and iii) AES (Advanced Encryption Standard) as the symmetric encryption algorithm. Figure 1 depicts PCS’s computation time cost in the phase of GenProof and GenRetr. In the X-axis we represent the number \( c \) of challenged blocks. The Y-axis represents PCS’s computation time in ms (i.e., milliseconds) in order to generate the proof or retrieve the block. Figure 2 depicts hospital computation time cost in CheckProof. The X-axis represents the number \( c \) of challenged blocks. The Y-axis represents hospital’s computation time (s) to check the proof, where \( p_1, p_2, p_3 \) denotes the challenged patients’ number. Thus, the challenged block number must be bigger than the challenged patients, i.e., \( p_1 \geq c, p_2 \geq c, p_3 \geq c \). Figure 3 depicts the time cost of hospital and AuthSet in the phase Retrieval. The X-axis represents the number \( c \) of challenged blocks. The Y-axis represents hospital and AuthSet’s computation time (ms) in order to retrieve \( c \) blocks in the phase Retrieval.

5. Conclusions

In this paper, we propose the concept of ICS protocol for critical patients in cloud-based health internet of things. This paper formalizes the system model and security model of ICS protocol. Based on the pairing, a concrete ICS protocol is designed. The proposed ICS protocol is provably secure and efficient by security analysis and performance analysis.

Acknowledgments

The work of H. Wang was supported in part by the National Natural Science Foundation of China under Grant (61272522), in part by the National Science Foundation of Liaoning Province under Grant (2014020147), and in part by the Program for Liaoning Excellent Talents in University under Grant (LR2014021), and in part by the CICAEET fund and the PAPD fund. This work of K. Li was partly supported by the National Science Foundation for Distinguished Young Scholars of China (61225010), and the State Key Program of National Natural Science of China (61432002).

References


Huaqun Wang received the BS degree in mathematics education from the Shandong Normal University, Jinan, China, in 1997, the MS degree in applied mathematics from the East China Normal University, Shanghai, China, in 2000, and the PhD degree in cryptography from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2006. Since then, he has been with Dalian Ocean University, Dalian, China, as a Full Professor. His research interests include applied cryptography, network security, and cloud computing security. Dr. Wang has published more than 50 papers. He has served in the program committee of several international conferences and the editor board of international journals.

Keqiu Li received the bachelor’s and master’s degrees from the Department of Applied Mathematics, Dalian University of Technology in 1994 and 1997, respectively. He received the PhD degree from the Graduate School of Information Science, Japan Advanced Institute of Science and Technology in 2005. He also has a two-year postdoctoral experience in the University of Tokyo, Japan. He is currently a professor in the School of Computer Science and Technology, Dalian University of Technology, China. He has published more than 100 technical papers, such as IEEE TPDS, ACM TOIT, and ACM TOMCCAP. He is an associate editor of IEEE TPDS and IEEE TC. His research interests include internet technology, data center networks, cloud computing and wireless networks. He is a senior member of the IEEE.

Kaoru Ota received M.S. degree in Computer Science from Oklahoma State University, USA in 2008 and Ph.D. degree in Computer Science and Engineering from The University of Aizu, Japan in 2012. She is currently an Assistant Professor with Department of Information and Electronic Engineering, Muroran Institute of Technology, Japan. From March 2010 to March 2011, she was a visiting scholar with BBCR group at University of Waterloo, Canada. Also she was a Japan Society of the Promotion of Science (JSPS) research fellow with Kato-Nishiyama Lab at Graduate School of Information Sciences at Tohoku University, Japan from April 2012 to April 2013. She has joined JSPS A3 foresight program as one of primary researchers since 2011 which is supported by Japanese, Chinese and Korean government. Dr. Ota’s research results have been published in 90 research papers in international journals, conferences and books. She is the Best Paper Award Winner of ICA3PP 2014, GPC 2015 and IEEE DASC 2015. She serves a Guest Editor of IEEE Wireless Communications, IEICE Transactions on Information and Systems and serves Editor of Peer-to-Peer Networking and Applications (Springer), Ad Hoc & Sensor Wireless Networks, International Journal of Embedded Systems (InderScience). Her research interests include wireless sensor networks, vehicular ad hoc networks, and ubiquitous computing.

Jian Shen received the B.E. degree from Nanjing University of Information Science and Technology, Nanjing, China, in 2007 and the M.E. and Ph.D. degrees in Computer Science from Chosun University, Gwangju, Korea, in 2009 and 2012, respectively. Since late 2012, he has been a faculty member in the School of Computer and Software at Nanjing University of Information Science and Technology, Nanjing, China. His research interests include computer networking, security systems, mobile computing and networking, ad-hoc networks and systems, and ubiquitous sensor networks.