Personal Health Records Integrity Verification Using Attribute Based Proxy Signature in Cloud Computing

Ximeng Liu[1], Jianfeng Ma[2], Jinbo Xiong[2], Tao Zhang[2], and Qi Li[2]

[1] School of Telecommunications Engineering, Xidian University, Xi'an 710071, China
snbnix@gmail.com
[2] School of Computer Science and Technology, Xidian University, Xi'an, 710071, China

Abstract. Personal health records (PHRs) have been appeared as patient -centric model for health information exchange, which are often outsourced to be stored in cloud services. However, the integrity and privacy of the PHRs are cause for concern that personal health information could be compromised. The principal method to guarantee integrity of PHRs is using signature mechanism when a PHR owner use the PHR to generate signature and a user is able to verify the PHR by using the signature. In some scenario, PHR owner can not sign the PHR by himself/herself, he/her wants to delegate its sign ability to other people to sign the PHR. In order to solve delegation of original signer's capabilities to guarantee integrity of PHR and the anonymity of the signer, attribute based proxy signature scheme(ABPS) for personal health records was first proposed in this paper. We formalize and construct the ABPS. Our scheme is proved to be existentially unforgeable against chosen message attack in the standard model. Analysis shows that our ABPS is more appropriate for cloud computing environment to guarantee integrity of PHRs.

Keywords: Attribute based signature, Proxy signature, PHRs, Cloud computing.

1 Introduction

In recent years, personal health records(PHRs) have emerged as a patient-centric model of health information exchange. A PHRs service allows a patient to create, manage, and control one's personal health data in one place through the Web, which have made the storage, retrieval and sharing of the medical information more efficient. Because it cost highly to build and maintain the data center, PHRs are often outsourced to third party severs (such as cloud data center) to stored. Patients also want to control their sensitive personal information. However, the cloud data center can not always be fully trusted. Recently, some architecture of storing PHRs in cloud computing have be proposed in [13] [10] [5]. Li et al.[11] enforced access control for outsourcing PHRs and attribute based encryption.

M. Pathan, G. Wei, and G. Fortino (Eds.): IDCS 2013, LNCS 8223, pp. 238–251, 2013.

When a doctor takes PHRs from cloud data center to assessed for disease, it should make sure that the PHRs can not modified by the cloud center. Also, patients sometime do not want his/her identify exposures to the doctor and want to flexible control their privacy.

Attribute based signatures(ABS)[19] scheme offers fine-grained access control in anonymity authentication systems which extends identity-based signature where the signer is associate with a set of attributes instead of a single identity string. It provides a powerful way for users to control their privacy: the patient chooses the subset of their attributes relevanting for the specific scenario in signing PHRs. Any doctor who has the attributes set containing all attributes above could issue the signature. Considering the following scenario, a patient signs PHR with access structure {"paediatrician" AND "hospital A" AND "internal medicine"} and uploads it to cloud center. When a doctor has these three attributes could verify the PHR's integrity, that is to say, a internal medicine paediatrician in hospital A could verify the PHR. In some cases, a patient could not sign the PHR by himself/herself(such as go aborad which could not access internet). The patient wants delegate his/her sign ability to proxy signer(such as the patient's relatives). The PHRs could only be signed by the patient, or signed by a proxy signer authorized by the patient. We want a scheme that an original signers with attribute sets can authorize a designated person as proxy signer which could sign PHRs on behalf of him/her.

In this paper, we propose a scheme called attribute based proxy signature (ABPS) scheme in order to solve the problem mentioned above. The ABPS scheme allows a designated proxy signer with its attribute set to sign the message on behalf of the original signer. The proposed scheme allows users to control their privacy flexibly.

1.1 Related Work

This paper constructs cryptographic primitive to keep personal health records integrity for outsourcing data to the cloud severs. In this subsection, we primarily introduce some related work in cryptography.

Attribute Based Signature. In basic ABE, an important application of the fuzzy identity based encryption (FIBE)[18], a user encrypts a message with a set of n attributes such that users whose decryption key have at least t common attributes with the ciphertext attribute set can decrypt the message. We call this scheme threshold attribute-based encryption (t-ABE) for describe simplicity. Yang et al.[24] introduced a new cryptographic primitive called fuzzy identity based signature (FIBS) which the signature analogue of the FIBE. Shahandashti[20] proposed a threshold attribute-based signature construction for both small attribute universe and large attribute universe. Due to FIBS scheme can not control signer's privacy, Maji et al.[14] introduced an ABS scheme can provide strong privacy guarantee for the signer and strong unforgeability guarantee for the verifier. In order to sign messages with any subset

of their attributes issued from an attribute center, Li and Kim[8] gave a hidden attribute-based signatures without anonymity revocation scheme which can reach anonymity and unforgeability. Li et al.[7] proposed a new construction of ABS supporting flexible threshold predicate which could compact the signature size and improve the verification time. Liu et al.[12] proposed a new attribute based multi-signature scheme to reduce the bandwidth needed to transmit attribute based signatures which is more appropriate for the wireless nature where bandwidth is a bottleneck.

Proxy Signature. Mambo et al.[15] first proposed a new signature scheme called proxy signature. In this scheme, the original signer authorized a designated proxy signer to sign the message on behalf. After that, proxy signatures have found numerous practical applications, such as mobile communications[17], distributed systems[16], grid computing[3] and mobile agent applications[6]. Boldyreva [1] was first presented the formal definition and security notion for proxy signature. Their work was proved to be security against adaptive chosen-message attack. Huang et al.[4] proposed a proxy signatureschemes which was proved to be existential unforgeable in the stand model. After Boneh and Franklin[2] used bilinear groups to construct identity-based encryption, a lot of identity-based proxy signature schemes were proposed. Xu et al.[23] formalized the notion of security for ID-based proxy signature schemes and proposed a scheme based on the bilinear pairings. But their schemes could not reach the notion of adaptively chosen message and chosen identity attacker in identity based system. Wu et al.[22] redefined the security models of identity based proxy signature to capture the most stringent attacks against adaptively chosen message and chosen identity attacker. Furthermore, many extensions of the basic proxy signature primitive had been considered include threshold proxy signatures [21] and blind proxy signatures [15].

1.2 Our Contributions

In this work, we make the following contributions. (1) We define a scheme called attribute based proxy signature(ABPS) for PHRs. We also formalize the model of ABPS and give security model for ABPS. (2) The concrete construction of the ABPS scheme is proposed in this paper. (3) We prove our ABPS scheme is existential unforgeability in the standard model by using the computational Diffie-Hellman assumption. Analysis shows that our ABPS scheme is more appropriate for cloud computing environment to keep PHRs integrity and keep PHR owners anonymity.

1.3 Organization

The rest of the paper organized as follows: In section 2, we review some concepts about bilinear pairing, complexity assumption and flexible threshold predicate. In section 3, we give the a formal model and its security model of the ABPS

scheme for PHRs. The specific construction about the ABPS scheme for PHRs is presented in section 4. In section 5, we give security and performance analysis for the ABPS scheme. And we conclude this paper in section 6.

2 Preliminaries

In this section we introduce bilinear maps, complexity assumptions and flexible threshold predicate which is associated with our construction.

2.1 Bilinear Maps

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic groups of prime order p with the multiplication. Let g be a generator of $\mathbb{G}$ and e be a bilinear map. Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map has the following properties:

1. Bilinearity: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
2. Non-degeneracy: $e(g, g) \neq 1$.
3. Computability: There is efficient algorithm to compute bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$.

Notice that the map e is symmetric since $e(u^a, v^b) = e(u, v)^{ab} = e(u^b, v^a)$.

2.2 Complexity Assumptions

Definition 1. *The challenger choose $a, b \in \mathbb{Z}_p$ at random and output (g, g^a, g^b). The Computational Diffie-Hellman(CDH) problem is to compute g^{ab}. An adversary $\mathcal{A}$ has at least an ϵ if*

$$\left|\Pr[\mathcal{A}(g, g^a, g^b) = g^{ab}]\right| \geq \epsilon$$

The computational (t, ϵ)-DH assumption holds if no t-time adversary has at least ϵ advantage in solving the above game.

2.3 Flexible Threshold Predicate

In this paper, we use predicates Υ consisting of thresholds gates. All predicates $\Upsilon_{k,\omega^*}(\cdot) \to 0/1$ for ω^* with threshold value k. If the number of attribute in $\omega' \cap \omega^*$ exceeds threshold k, it outputs 1. Otherwise, it outputs 0.

$$\Upsilon_{k,\omega^*}(\omega') = \begin{cases} 1, & |\omega' \cap \omega^*| \geq k \\ 0, & otherwise \end{cases}$$

2.4 Lagrange Interpolation

In this subsection, we describe Lagrange interpolation which is used in the ABMS schemes. Given d points $q(1), \cdots, q(d)$ on a $d-1$ degree polynomial, we can use Lagrange interpolation to compute $q(i)$ for any $i \in \mathbb{Z}_p$. Let S be a d-element set. We define the Lagrange coefficient $\Delta_{j,S}(i)$ of $q(j)$ in the computation of $q(i)$ as:

$$\Delta_{j,S}(i) = \prod_{\eta \in S, \eta \neq j} \frac{i-\eta}{j-\eta}$$

3 Formal Models and Security Model for ABPS

3.1 Formal Models

The attribute based proxy signature can be described as a collection of the following seven algorithms:

Setup: This algorithm runs by the authority which inputs the security parameter and generates the public parameters *params* of the scheme and the master secret key. The authority entity publishes *params* and keeps the master secret to itself.

Extract: This algorithm runs by authority to generate a private key for the entity involve in the PHR system. It inputs an attribute set ω, the master key and *params* and outputs the private key of ω. After generating private keys for all entities participating in the scheme is generated, the authority distributes the private keys to their respective owner through a secure channel.

StandardSign: This algorithm runs by PHR owner on input a message m, an attribute set ω, a private key d and *params*. It generates the signature σ of ω on m. The entity with attribute set ω will use this algorithm for signing.

StandardVerify: This algorithm runs by verifier on input a signature σ, a message m, attribute set and *params*. It outputs *accept* if a valid signature on message for attribute set or outputs *reject* otherwise.

DelegationGen: This algorithm runs by the PHR owner on input system's public parameters *params*, PHR owner's secret key sk_A, the delegation warrant θ which include the restriction on the class of message delegated, the attribute set of the PHR owner, the attribute set of the proxy user and period of delegation, etc. It outputs delegation σ_θ to proxy user on behalf of the PHR owner.

ProxySign: This algorithm runs by proxy user on input public parameters *params*, proxy signature sk_p, a warrant θ and a message m which satisfies θ. The algorithm outputs a proxy signature $p\sigma_p$ on message m.

ProxyVerify: This algorithm runs by verifier on input public parameters *params*, proxy signature $p\sigma_p$, a warrant θ and a message m. If $p\sigma_p$ is a valid proxy signature for m, the algorithm outputs 1 or outputs 0 otherwise.

3.2 Security Models

In the model defined in [4], they divide potential attackers into the three kinds. We also use these three attackers to define security model for ABPS scheme:

1. **Type 1** ($\mathcal{A}_1$): This type of adversary $\mathcal{A}_1$ only has the public key of PHR owner and proxy user.

2. **Type 2** ($\mathcal{A}_2$): This type of adversary $\mathcal{A}_2$ not only has the public key of PHR owner and proxy user but also has the private key of the proxy user.

3. **Type 3** ($\mathcal{A}_3$): This type of adversary $\mathcal{A}_3$ only has the public key of PHR owner and proxy user, he also has the private key of the PHR owner.

It is easy to find that ABPS scheme is secure against Type 2(or Type 3) adversary, the scheme is also secure against Type 1 adversary. Here, we focus on define the existential unforgeability of the ABPS scheme.

Existential Unforgeability against Adaptive $\mathcal{A}_2$ Adversary
In order to define the secure model of ABPS against adaptive $\mathcal{A}_2$ adversary, we define following game between a challenger $\mathcal{B}$ and an adversary $\mathcal{A}_2$.

Setup: The challenger $\mathcal{B}$ runs the *Setup* algorithm and obtains both the public parameters *params* and the master secret key. $\mathcal{B}$ gives the *params* to adversary and keeps the master secret key by itself.

Queries. The adversary $\mathcal{A}_2$ adaptively makes a polynomial bounded number of queries to the challenger. Each query can be one of the following:

-**Extract query:** The adversary $\mathcal{A}_2$ can ask for the private key of any attribute set ω. The challenger responds by running the *Extract* algorithm and gives the private key to adversary.

-**Delegation queries:** $\mathcal{A}_2$ adaptively make request the delegation on the warrant θ. $\mathcal{B}$ runs the *DelegationGen* algorithm to obtain σ_θ and return σ_θ to the adversary $\mathcal{A}_2$.

-**ProxySign queries:** $\mathcal{A}_2$ can adaptively request the proxy signature on message m under the warrant θ. $\mathcal{B}$ first runs *DelegationGen* algorithm to generate the delegation on the warrant θ. Then $\mathcal{B}$ runs the *ProxySign* algorithm to obtain signature $p\sigma_p$ and return $p\sigma_p$ to the adversary $\mathcal{A}_2$.

Output: Eventually, $\mathcal{A}_2$ halts and outputting a forgery such that :

1). θ^* has not been requested as one of the *Delegation* queries.

2). (m^*, θ^*) has not been requested as one of the *ProxySign* queries.

3). σ^* is a valid proxy signaature of the message m^* under the warrant θ^*.

The type 2 adversary $\mathcal{A}_2$ can adaptively submit the *ProxySign* queries under warrant whose delegation is unknow to $\mathcal{A}_2$. The only restrictions are when $\mathcal{A}_2$ outputs the forgery $(m^*, \theta^*, p\sigma_p^*)$ which θ^* can not be submitted as one of the *Delegation* queries or (m^*, θ^*) can not be submitted as one of the *ProxySign* queries.

Definition 2. *The attribute based proxy signature scheme is* $(t, q_e, q_D, q_{PS}, \epsilon)$*-secure against type 2 adversary* $\mathcal{A}_2$ *if no* t*-time adversary* $\mathcal{A}_2$ *making* q_e *Extract queries,* q_D *Delegation queries,* q_{PS} *ProxySign queries can win the above game with advantage more than* ϵ*.*

Existential Unforgeability against Adaptive $\mathcal{A}_3$ Adversary
In order to define the secure model of ABPS against adaptive $\mathcal{A}_3$ adversary, we define following game between a challenger $\mathcal{B}$ and an adversary $\mathcal{A}_3$.

Setup: The challenger $\mathcal{B}$ runs the *Setup* algorithm and obtains both the public parameters *params* and the master secret key. $\mathcal{B}$ gives the *params* to adversary and keeps the master secret key by itself.

Queries. The adversary $\mathcal{A}_3$ adaptively makes a polynomial bounded number of queries to $\mathcal{B}$. Each query can be one of the following:

-**Extract query:** The adversary $\mathcal{A}_3$ can ask for the private key of any attribute set ω. $\mathcal{B}$ responds by running the *Extract* algorithm and gives the private key to $\mathcal{A}_3$.

-**Delegation queries:** $\mathcal{A}_3$ adaptively make request the delegation on the warrant θ. $\mathcal{B}$ runs the *DelegationGen* algorithm to obtain σ_θ and return σ_θ to the adversary $\mathcal{A}_3$.

- **ProxySign queries:** $\mathcal{A}_3$ can adaptively request the proxy signature on message m under the warrant θ. $\mathcal{B}$ first runs *DelegationGen* algorithm to generate the delegation on the warrant θ. Then $\mathcal{B}$ runs the *ProxySign* algorithm to obtain signature $p\sigma_p$ and return $p\sigma_p$ to the adversary $\mathcal{A}_2$.

Output: Eventually, $\mathcal{A}_3$ halts and outputting a forgery such that :

1). (m^*, θ^*) has not been requested as one of the *ProxySign* queries.

2). σ^* is a valid proxy signaature of the message m^* under the warrant θ^*.

Definition 3. *The attribute based proxy multi-signature scheme is* $(t, q_e, q_D, q_{PS}, \epsilon)$*-secure against type 2 adversary* $\mathcal{A}_2$ *if no t-time adversary* $\mathcal{A}_2$ *making* q_e *Extract queries,* q_D *Delegation queries,* q_{PS} *ProxySign queries can win the above game with advantage more than* ϵ*.*

4 Our Constructions

In this section, we give the concrete construction of attribute based proxy signature scheme.

4.1 Overview of the ABPS Scheme for PHRs

The main goal of our attribute based proxy signature scheme guarantees integrity of PHRs and allows patients to flexible control their privacy. Meanwhile, it solves delegation problem when patients can not sign the PHRs by himself/herself which need to delegate his/her signing ability to proxy user on behalf of him/her. As fig. 1 shows, there are PHR owner, proxy user, verifier(Doctor, emergency staff) and authority involved in the system. The authority first generates a master key and defines a common universe of attributes, such as "paediatrician", "hospital A", "internal medicine", "physician". Then, authority uses the master key and attribute sets to generate user's private keys and send them to the corresponding users involve in the system respectively. PHR owner could sign the PHR by himself/herself, or generates a warrant which includes the restrictions on the proxy signer. After that, the PHR owner uses the warrant to generate delegation and sends it to the proxy signer together with the warrant. When the proxy user receives warrant and delegation, he/she can use his/her own private key and attribute set to proxy sign the delegation on behalf of original signer.

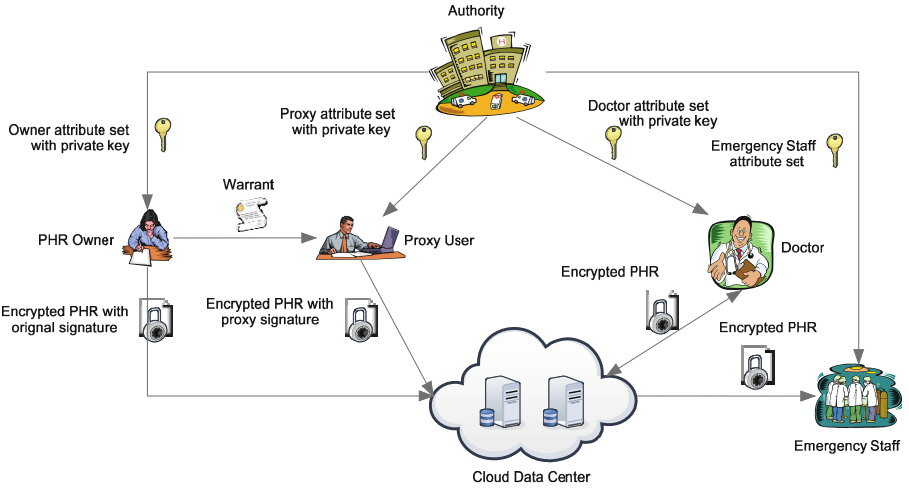

Fig. 1. ABPS for PHRs

The proxy signature combine with the encrypted PHR are sended to cloud data center(third party servers) to stored. When a doctor is requested to diagnose the PHR owner whether has be infected with some diseases or not, he should first retrieval the PHR from cloud data center and then decrypt the the PHR. In order to guarantee the the PHR is not modified by the cloud data center, he must first use the PHR owner's attribute set, proxy user's set and warrant declared by the PHR owner to check the integrity of the PHR. When the PHR passed the verification, it shows that the PHR is not be modified. The doctor can use the information present in the PHR to diagnosed the PHR owner's health condition condition. After that, the doctor uses his own private key to sign the PHR, encrypts the PHR with PHR owner's attribute set and sends back to the cloud data center. When the PHR owner is sent to the hospital in emergency, emergency staff could decrypt the PHR and verify the signature to indicate that the PHR is not falsified by other and believe the authenticity of the PHR. The concrete construction of ABPS will be presented in the next subsection.

4.2 Attribute Based Proxy Signature Scheme

Setup: This algorithm first defines the attributes in the universe U as the element in $\mathbb{Z}_p$. A $d-1$ default attribute set from $\mathbb{Z}_p$ is given as $\Omega = \{\Omega_1, \Omega_2, \cdots, \Omega_{d-1}\}$. It selects a random generator $g \in \mathbb{G}$, a random $\alpha \in \mathbb{Z}_p^*$ and computes $g_1 = g^\alpha \in \mathbb{G}$. Next, it picks a random element g_2 and computes $A = e(g_1, g_2)$. After that it chooses $t_1, \cdots, t_{n+1}$ uniformly at random from $\mathbb{G}$. Let N be the set $\{1, \cdots, n+1\}$ and we define a function T , as:

$$T(x) = g_2^{x^n} \prod_{j=1}^{n+1} t_j^{\Delta_{j,N}(x)}$$

Finally, the algorithm selects random values y' from $\mathbb{Z}_p$, a random vector $\mathbf{y} = (y_1, y_2, \cdots, y_k)$ from $\mathbb{Z}_p^k$ and computes $\mathbf{U} = (u_1, u_2, \cdots, u_k) = (g^{y_1}, g^{y_2}, \cdots, g^{y_k})$. The public parameters are

$$params = (\mathbb{G}, \mathbb{G}_T, e, g, g_1, g_2, t_1, \cdots, t_{n+1}, \mathbf{U}, A)$$

The master key are

$$\text{MSK} = \alpha$$

Extract: This algorithm generates a private key for an attribute set ω related with users involved in the system. It takes the following steps:

1) Firstly, it chooses a $d-1$ degree polynomial at random with $q(0) = \alpha$.

2) It then generates a new attribute set $\hat{\omega} = \omega \cup \Omega$. For each $i \in \hat{\omega}$, the algorithm chooses and computes $d_{i0} = g_2^{q(i)} \cdot T(i)^{r_i}$, $d_{i1} = g^{r_i}$.

3) Finally, it outputs

$$D_i = (d_{i0}, d_{i1})_{i \in \hat{\omega}}$$

as the private key.

StandardSign: This algorithm inputs a private key for the attribute set ω, message m and predicate $\Upsilon_{k,\omega^*}(\cdot)$. In order to sign message m with predicate $\Upsilon_{k,\omega^*}(\cdot)$, i.e., to prove owning at least k attribute among an n-elements ω^*. It selects a k-element form the subset $\omega' \subseteq \omega \cap \omega^*$ and works as follows:

(1) First, it selects a default attribute subset $\Omega' \subseteq \Omega$ with $|\Omega'| = d - k$ and chooses $n + d - k$ random values $r_i' \in Z_p$ for $i \in \omega^* \cup \Omega'$.

(2) It then computes

$$\sigma_0 = [\prod\nolimits_{i \in \omega' \cup \Omega'} d_{i0}^{\Delta_{i,S}(0)}][\prod\nolimits_{i \in \omega^* \cup \Omega'} T(i)^{r'_i}](u' \prod_{j \in \mathcal{M}} u_j^{m_j})^{r_s}$$

$$\{\sigma_i = d_{i1}^{\Delta_{i,S}(0)} g^{r'_i}\}_{i \in \omega' \cup \Omega'}, \{\sigma_i = g^{r'_i}\}_{i \in \omega^*/\omega'}, \sigma_0' = g^{r_s}$$

(3) Finally, the algorithm outputs the signature:

$$\sigma = (\sigma_0, \{\sigma_i\}_{i \in \omega^* \cup \Omega'}, \sigma_0')$$

StandardVerify: In order to verify the correctness of the signature $\sigma = (\sigma_0, \{\sigma_i\}_{i \in \omega^* \cup \Omega'}, \sigma_0')$ on m with threshold k for attributes set $\omega^* \cup \Omega'$, it checkes the following equation holds:

$$\frac{e(g, \sigma_0)}{[\prod_{i \in \omega^* \cup \Omega'} e(T(i), \sigma_i)]\, e(u' \prod\limits_{j \in \mathcal{M}} u_j^{m_j}, \sigma_0')} = A$$

If the equation holds, it indicates that the signature is indeed from some user with k attributes among ω^*. Otherwise, it denotes the signature is not valid.

DelegationGen: In order to delegate the PHR owner's signing capability to the proxy user, the PHR owner first makes a warrant θ which includes the restrictions on the class of messages delegated, the PHR owner's attributess set, proxy user's attribute sets, public parameters and the period of validity, etc. Let

θ be an m-bit message to be signed by the original signer a. θ_j denotes the j-th bit of θ and $W \subseteq \{1, \cdots, m\}$ be the set of all j for which $\theta_j = 1$. The PHR owner's delegation is generated as follows. First, it chooses r'_i, r_a randomly in $\mathbb{Z}_p^*$, then the delegation is constructed as

$$\sigma_\theta = (\sigma_{\theta 0}, \{\sigma_{\theta i}\}_{i \in \omega_a^* \cup \Omega'_a}, \sigma_{\theta 2})$$

where

$$\sigma_{\theta 0} = [\prod\nolimits_{i \in \omega_{a'} \cup \Omega_{a'}} d_{i0}^{\Delta_{i,S}(0)}][\prod\nolimits_{i \in \omega_a^* \cup \Omega'_a} T(i)^{r'_i}](u' \prod_{j \in W} u_j^{\tau_j})^{r_a}$$

$$\{\sigma_{ai} = d_{i1}^{\Delta_{i,S_k}(0)} g^{r'_i}\}_{i \in \omega'_a \cup \Omega'_a}, \{\sigma_{ai} = g^{r'_i}\}_{i \in \omega_a^* / \omega'_a}, \sigma_{\theta 2} = g^{r_a}.$$

Finally, the PHR owner sends the σ_θ with the warrant θ to the proxy user b.

ProxySign: Let m be an m'-bit message to be signed by PHR owner and m_d denote the d-th bit of m, and $\mathcal{M} \subseteq \{1, 2, \cdots, m'\}$ be the set of all d for which $m_d = 1$. The proxy signature is generated as follows. The proxy user chooses random value $r'_i, r'_a, r_m \in \mathbb{Z}_p$, then the signature is constructed as:

$$p\sigma_p = (p\sigma_{p0}, \{p\sigma_{ai}\}_{i \in \omega_a^* \cup \Omega'_a}, \{p\sigma_{bi}\}_{i \in \omega_b^* \cup \Omega'_b}, p\sigma_{p2}, p\sigma_{p3}).$$

where

$$p\sigma_p = \sigma_{\theta 0}(u' \prod_{j \in W} u_j^{\tau_j})^{r'_a}[\prod\nolimits_{i \in \omega'_b \cup \Omega'_b} d_{i0}^{\Delta_{i,S}(0)}] \cdot [\prod\nolimits_{i \in \omega_b^* \cup \Omega'_b} T(i)^{r'_i}](u' \prod_{j \in \mathcal{M}} u_j^{m_j})^{r_m}$$

$$\{p\sigma_{ai}\}_{i \in \omega'_a \cup \Omega'_a} = \{\sigma_{ai}\}_{i \in \omega'_a \cup \Omega'_a}, \{p\sigma_{ai}\}_{i \in \omega_a^* / \omega'_a} = \{\sigma_{ai}\}_{i \in \omega_a^* / \omega'_a}$$

$$\{p\sigma_{bi}\}_{i \in \omega'_b \cup \Omega'_b} = \{\sigma_{bi}\}_{i \in \omega'_b \cup \Omega'_b}, \{p\sigma_{bi}\}_{i \in \omega_b^* / \omega'_b} = \{\sigma_{bi}\}_{i \in \omega_b^* / \omega'_b}$$

$$p\sigma_{p2} = \sigma_{\theta 2} \cdot g^{r'_a}, p\sigma_{p3} = g^{r_m}.$$

ProxyVerify: Given the public parameters, a warrant $\theta \in \{0,1\}^m$, a message $m \in \{0,1\}^{m'}$ and a signature $p\sigma_p$. A verifier accepts $p\sigma_p$ if the following equality holds:

$$e(p\sigma_{p0}, g) = A_a \cdot A_b \left[\prod_{i \in \omega_a^* \cup \Omega'_a} e(T(i), p\sigma_{ai})\right] e(u' \prod_{j \in \mathcal{M}} u_j^{m_j}, p\sigma_{p3}) \cdot \left[\prod_{i \in \omega_b^* \cup \Omega'_b} e(T(i), p\sigma_{bi})\right] e(u' \prod_{j \in W} u_j^{\tau_j}, p\sigma_{p2})$$

5 Security and Performance Analysis

In this section, we first show our ABPS scheme is existentially unforgeable against Type 2 and Type 3 adversary. Then, we give an analysis to show our ABPS is more appropriate for cloud computing environment to keep PHRs integrity.

1. **Type 1** ($\mathcal{A}_1$): This type of adversary $\mathcal{A}_1$ only has the public parameters of the PHR owner (signer) and proxy user.
2. **Type 2** ($\mathcal{A}_2$): This type of adversary $\mathcal{A}_2$ not only has the public parameters of the PHR owner(signer) and proxy user but also has the private key of the proxy user.
3. **Type 3** ($\mathcal{A}_3$): This type of adversary $\mathcal{A}_3$ not only has the public parameters of the PHR owner and proxy user, but also has the private key of the PHR owner(signer).

It is easy to find that ABPS scheme is secure against Type 2(or Type 3) adversary, the scheme is also secure against Type 1 adversary. Here, we focus on define the existential unforgeability of the ABPS scheme.

5.1 Existential Unforgeability against Type 2 Adversary

Theorem 1. *The attribute based proxy signature scheme is* $(t, q_e, q_D, q_{PS}, \epsilon)$*-unforgeable against type 2 adversary* $\mathcal{A}_2$ *if the* (t', ϵ')*-CDH assumption holds in where*

$$\epsilon' \geq \frac{\epsilon}{16\binom{d-1}{d-k} q_{PS}(q_D + q_{PS})(m+1)^2 p^{2d}}$$

$$t' = t + \mathcal{O}((d(q_e + q_D + q_{PS}) + m(q_D + q_{PS}))\rho + (d(q_e + q_D) + q_{PS})\tau)$$

and ρ *and* τ *are the time for a multiplication and an exponentiation in* $\mathbb{G}$ *respectively. Where* $\mathcal{A}_2$ *making* q_e *Extract queries,* q_D *Delegation queries,* q_{PS} *ProxySign queries.*

Proof. Due to space limitations, the detailed proof will be shown in the full version of our work.

5.2 Existential Unforgeability against Type 3 Adversary

Theorem 2. *The attribute based proxy signature scheme is* $(t, q_e, q_{PS}, \epsilon)$*- unforgeable against type 3 adversary* $\mathcal{A}_3$ *if the* (t', ϵ')*-CDH assumption holds in where*

$$\epsilon' \geq \frac{\epsilon}{16\binom{d-1}{d-k} q_{PS}^2 (m+1)^2 p^{2d}}$$

$$t' = t + \mathcal{O}((d(q_e + q_{PS}) + m \cdot q_{PS})\rho + (d \cdot q_e + q_{PS})\tau)$$

and ρ *and* τ *are the time for a multiplication and an exponentiation in* $\mathbb{G}$ *respectively.*

Proof. It is similar to the proof of Theorem 1.

5.3 Performance Analysis

In this subsection, we compare our scheme with existing schemes to indicate our scheme is more suitable for verifying PHRs in cloud computing environment. Huang et al.[4] proposed a proxy signature scheme which has the delegation property. In their scheme, it allows original signer to delegate his/her signing ability to proxy signer on behalf. But this scheme can not reach fine-grained access control and allow user flexible control their privacy. Wu et al.[22] gave a stronger security notion of the proxy signature by allowing the adversaries to behave more adaptively in oracle accessing. But the security of their scheme is proven to be secure under random oracle model. It can neither reach fine-grained access control nor provide user anonymity. Li et al.[9] proposed an attribute based signature scheme which can keep signer anonymity and provide fine-grained access control for user to control their privacy. But this scheme is only proved in the random oracle model and do not have the delegation property which is not appropriate for PHR in the cloud environment to keep the PHR integrity. Our ABPS scheme can ensure anonymity for user to flexible to control the privacy. It can also delegate its signing ability to other person on behalf. Also, the security of ABPS scheme is proved to be existential unforgeability under the standard model. More important, the proposed ABPS is more appropriate for cloud computing environment to keep PHRs integrity which is not modified by the distrust severs.

Table 1. The comparison between ABPS and the existing schemes

Functionality/ Scheme	Huang et al.[4]	Wu et al.[22]	Li et al.[9]	Ours
User's anonymity	No	No	Yes	Yes
Fine-grained access control	No	No	Yes	Yes
Delegation property	Yes	Yes	No	Yes
Standard model	Yes	No	No	Yes
Provable secure	Yes	Yes	Yes	Yes
Data integrity	Yes	Yes	Yes	Yes
Pairing based	Yes	Yes	Yes	Yes
Existential unforgeability	Yes	Yes	Yes	Yes

6 Conclusion

In this paper, we first proposed a scheme called attribute based proxy signature. The ABPS scheme allowed a proxy signer to sign the message on behalf of a original PHR owner. We proved our ABPS scheme secure against existential forgery against Type 2 and Type 3 adversary. More important, we showed our ABPS scheme is appropriate for cloud computing environment to guarantee the integrity of PHR and anonymity of the PHR owners.

Acknowledgment. This research is supported by Changjiang Scholars and Innovative Research Team in University under grant No. IRT1078; The Key Program of NSFC-Guangdong Union Foundation under grant No. U1135002; The National Natural Science Foundation of China under grant No. 61370078; Major national S&T program under grant No. 2011ZX03005-002; The Fundamental Research Funds for the Central Universities under grant No. JY10000903001. We thank the sponsors for their support and the reviewers for helpful comments.

References

1. Boldyreva, A., Palacio, A., Warinschi, B.: Secure proxy signature schemes for delegation of signing rights. J. Cryptology 25(1), 57–115 (2012)
2. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
3. Foster, I., Kesselman, C., Tsudik, G., Tuecke, S.: A security architecture for computational grids. In: Proceedings of the 5th ACM Conference on Computer and Communications Security, pp. 83–92. ACM (1998)
4. Huang, X., Susilo, W., Mu, Y., Wu, W.: Proxy signature without random oracles. In: Cao, J., Stojmenovic, I., Jia, X., Das, S.K. (eds.) MSN 2006. LNCS, vol. 4325, pp. 473–484. Springer, Heidelberg (2006)
5. Huba, N., Zhang, Y.: Designing patient-centered personal health records (phrs): Health care professionals perspective on patient-generated data. Journal of Medical Systems 36(6), 3893–3905 (2012)
6. Lee, B., Kim, H., Kim, K.: Strong proxy signature and its applications. In: Proc. of SCIS, vol. 1, pp. 603–608 (2001)
7. Li, J., Au, M.H., Susilo, W., Xie, D., Ren, K.: Attribute-based signature and its applications. In: Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, pp. 60–69. ACM (2010)
8. Li, J., Kim, K.: Hidden attribute-based signatures without anonymity revocation. Information Sciences 180(9), 1681–1689 (2010)
9. Li, J., Wang, Q., Wang, C., Ren, K.: Enhancing attribute-based encryption with attribute hierarchy. Mobile Networks and Applications 16(5), 553–561 (2011)
10. Li, M., Yu, S., Ren, K., Lou, W.: Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings. In: Jajodia, S., Zhou, J. (eds.) SecureComm 2010. LNICST, vol. 50, pp. 89–106. Springer, Heidelberg (2010)
11. Li, M., Yu, S., Zheng, Y., Ren, K., Lou, W.: Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. IEEE Trans. Parallel Distrib. Syst. 24(1), 131–143 (2013)
12. Liu, X., Zhang, T., Ma, J., Zhu, H., Cai, F.: Efficient data integrity verification using attribute based multi-signature scheme in wireless network. In: 2013 5th International Conference on Intelligent Networking and Collaborative Systems (INCoS). IEEE (2013)
13. Löhr, H., Sadeghi, A.-R., Winandy, M.: Securing the e-health cloud. In: Proceedings of the 1st ACM International Health Informatics Symposium, pp. 220–229. ACM (2010)
14. Maji, H., Prabhakaran, M., Rosulek, M.: Attribute-based signatures: Achieving attribute-privacy and collusion-resistance. IACR Cryptology ePrint Archive 2008, 328 (2008)

15. Mambo, M., Usuda, K., Okamoto, E.: Proxy signatures: Delegation of the power to sign messages. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences 79(9), 1338–1354 (1996)
16. Neuman, B.: Proxy-based authorization and accounting for distributed systems. In: Proceedings the 13th International Conference on Distributed Computing Systems, pp. 283–291. IEEE (1993)
17. Park, H.-U., Lee, I.-Y.: A digital nominative proxy signature scheme for mobile communication. In: Qing, S., Okamoto, T., Zhou, J. (eds.) ICICS 2001. LNCS, vol. 2229, pp. 451–455. Springer, Heidelberg (2001)
18. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
19. Shahandashti, S.F., Safavi-Naini, R.: Threshold attribute-based signatures and their application to anonymous credential systems. In: Preneel, B. (ed.) AFRICACRYPT 2009. LNCS, vol. 5580, pp. 198–216. Springer, Heidelberg (2009)
20. Shahandashti, S.F., Safavi-Naini, R.: Threshold attribute-based signatures and their application to anonymous credential systems. IACR Cryptology ePrint Archive 2009, 126 (2009)
21. Sun, H.-M., Lee, N.-Y., Hwang, T.: Threshold proxy signatures. In: IEE Proceedings-Computers and Digital Techniques, vol. 146, pp. 259–263. IET (1999)
22. Wu, W., Mu, Y., Susilo, W., Seberry, J., Huang, X.: Identity-based proxy signature from pairings. In: Xiao, B., Yang, L.T., Ma, J., Muller-Schloer, C., Hua, Y. (eds.) ATC 2007. LNCS, vol. 4610, pp. 22–31. Springer, Heidelberg (2007)
23. Xu, J., Zhang, Z., Feng, D.: ID-based proxy signature using bilinear pairings. In: Chen, G., Pan, Y., Guo, M., Lu, J. (eds.) ISPA-WS 2005. LNCS, vol. 3759, pp. 359–367. Springer, Heidelberg (2005)
24. Yang, P., Cao, Z., Dong, X.: Fuzzy identity based signature. IACR Cryptology ePrint Archive 2008, 2 (2008)