Attribute-Based Traceable Anonymous Proxy Signature Strategy for Mobile Healthcare

Dacheng Meng[1], Wenbo Wang[1], Entao Luo[1], and Guojun Wang[2(✉)]

[1] School of Information Science and Engineering,
Central South University, Changsha 410083, China
[2] School of Computer Science and Educational Software,
Guangzhou University, Guangzhou 510006, China
csgjwang@gmail.com

Abstract. In mobile healthcare, with the gradual development of the validity of electronic information, the use of electronic signatures as electronic prescriptions for medical users has gradually been adopted by various medical institutions. This electronic prescription can simplify the complex medical treatment of patients, while reducing the burden of healthcare providers, so the medical treatment process can be more standardized, rational, humane. Because of the importance of medical signatures, in order to solve the problem that doctors cannot provide a signature also needs to consider the case of proxy signature. For proxy signature, the legality and privacy disclosure of the agents need to be considered. In the existing signature system, proxy negotiation is wildly used to grant attorney, however, this authorization process is complex, which cannot provide fine-grained access control for the identity of the agent. In this paper, based on attribute-based encryption, we propose a traceable proxy signature scheme, only when the user's attributes satisfy the access policy, the user can decrypt the corresponding ciphertext to obtain the proxy signature right. The program can solve the signature issue in case of the doctors absence, while solving the problem of attorney abuse. Meanwhile, the authorization is completed in central authority, thus, the computational overhead is greatly reduced, a simple, safe and efficient proxy signature scheme can be achieved.

Keywords: Mobile healthcare · Digital signature · Anonymous proxy · Attribute-based encryption · Traceability

1 Introduction

With the rapid development of the Internet, mobile healthcare has gradually become a hot research topic. In mobile healthcare, doctors can directly using electronic signature to issue electronic prescription, so the patients do not need to get signed by a doctor to take medicine, and the patient also can have physical examination without getting signed by a doctor according to the electronic prescriptions. By using electronic prescriptions, electronic medical records and

G. Wang et al. (Eds.): SpaCCS 2016, LNCS 10066, pp. 178–189, 2016.
DOI: 10.1007/978-3-319-49148-6_16

electronic inspection report, facilitate the hospital staff and the difficulties of medical treatment of patients have been greatly reduced. In the mobile healthcare system, the electronic medical records, electronic prescriptions, electronic inspection reports are called electronic medical documents, the medical documents record patients condition from medical treatment until the end of the diagnosis and the treatment of all the relevant condition changes, report check and the full course of treatment, and for a doctor, also record all the examination, judgment, treatment of the whole process, which contain personal privacy information. If the privacy is obtained by an attacker, patients' safety of life and property will be greatly impacted, so these privacy information can only be viewed or signed by professional doctors. However, in many cases, doctors can not personally carry out signatures, such as doctors are busy with surgery or business trip, then you need to find an agent to help doctors deal with these issues [1]. In the actual situation, many agents in order to protect their own privacy may be not willing to reveal his identity to the original signature. Therefore, agent identity anonymous is necessary. However, in the anonymous, the legitimacy of the agent should also be considered, if the agent uses the right of signature to make some illegal behavior, which needs timely tracking to the proxy to ensure efficiency of signature. Waters et al. proposes an identity-based encryption scheme [2], and on this basis a standard model of identity based signature scheme is given, but this method is relatively single, can not adapt to a variety of circumstances under the signature. To solve this problem, Kim et al. propose a proxy signature scheme [3], proxy signature can combine other signature technology to produce digital signature scheme. However, the signature of the agency privacy lack effective protection. Then Yu et al. propose a proved secure anonymous proxy signature scheme [4], the scheme combines proxy signature and ring signature, which realize the anonymity proxy signature and the protection of the proxy signature, but the program is not traceable and signature verification efficiency is low.

2 Preliminaries

In this section, some preliminaries related to bilinear maps, complexity assumptions and access structure are presented.

2.1 Bilinear Maps

Let G and G' be two multiplicative cyclic groups with big prime order p. Let g be a generator of G. Let be a bilinear map $e : G \times G \rightarrow G'$ with the following properties [5]:

(1) Bilinearity For all and the equation holds.
(2) Non-degeneracy $e(g, g) \neq 1$.
(3) Computability There exists an efficient algorithm to compute bilinear map $e : G \times G \rightarrow G'$.

2.2 Bilinear Diffie-Hellman Inversion Assumption

In order to prove the security of the ATAPS scheme, we introduce l-BDHI assumption used in [6]. The l-BDHI problem in G is as follows: Given g, h and g^{y^i} in G for $i = 1, 2, ..., l$ as input for some unknown random $y \in Z_p^*$, output $W \in G'$ to decide whether $W = e(g, g)^{y^{l+1}}$. We say that a polynomial-time adversary $\mathcal{A}$ has advantage ε in solving the decisional l-BDHI problem (G, G') if $|Pr[A(g, h, y, e(g, h)^{y^{l+1}}) = 0] - Pr[A(g, h, y, e(g, h)^{y^z}) = 0]| \geq \varepsilon$, Where the probability is taken over random y, z and the random bits consumed by $\mathcal{A}$.

Definition 1. *We say that the (t, ε)-l-BDHI assumption holds in (G, G') if no t-time algorithm has the probability at least ε in solving the l-BDHI problem for non-negligible ε [7].*

2.3 Access Structure and Access Tree

Definition 2 *(Access structure [8]). Let $\{P_1, P_2, ..., P_n\}$ be a set of parties. A collection $A \subseteq 2^{\{P_1, P_2, ..., P_n\}}$ is monotone if $\forall B, C$: if $B \in A$ and $B \in C$. An access structure (respectively, monotonic access structure) is a collection (respectively, monotone collection) A of non-empty subsets of $\{P_1, P_2, ..., P_n\}$, i.e. $A \subseteq 2^{\{P_1, P_2, ..., P_n\}} \backslash \{0\}$. The sets in A are called the authorized sets, and the sets not in A are called the unauthorized sets.*

2.4 Access Tree with Time-Specific Attributes

We denote γ as an access tree. Each non-leaf node of the tree represents a threshold gate, described by a threshold value and its children [9]. If num_x is the number of children of a node x and k_x is its threshold value, the $0 < k < num_x$ holds. The threshold gate is an OR gate when threshold value $k_x = 1$. If threshold value of node x of the tree is associated with a time instant t_x. If the t_x belongs to a time interval $[t_{L,x}, t_{R,x}]$, which is associated with the corresponding attribute x in the ciphertext, we let value $k_x = 1$.

Some functions are defined in order to facilitate dealing with γ. In γ, the function $parent(x)$ is represented as the parent of the node x. The component of attributes is associated with the leaf node x in γ, also defines an ordering between the children of a node which are numbered from 1 to num. The function $index(x)$ returns such a number associated with the node x, where the index values are uniquely allocated to nodes in γ for a given key [10].

In the following we will describe how to satisfy an access tree width attributes and time constraints. Let Γ be a with root r. Γ_x is represented as the subtree of Γ with the root node at x. For the root r of Γ, we denote Γ_r. If a set of attributes S satisfies Γ_x, we denote it as $\Gamma_x(S) = 1$. $\Gamma_x(S)$ is calculated recursively as follows: If x is a non-leaf node, evaluate $\Gamma_x(S)$ returns 1 if and only if at least k_x children return 1. If x is a node belongs to the last layer from bottom, then $\Gamma_x(S)$ returns 1 if and only if the current time instant t_x associated with leaf node (attribute) in the access tree belongs to time interval $[t_{L,x}, t_{R,x}]$ associated with the corresponding attribute x in the ciphertext, that is $t_x \in [t_{L,x}, t_{R,x}]$.

2.5 Security Model

In the model, an attacker can be preset to two categories:

(1) External attackers: the attacker $\mathcal{A}_1$ only knows public key of the original signer and the proxy signer;
(2) Internal attackers: the attacker $\mathcal{A}_2$ has access to the proxy signature key;

For the first attacker, now give a formal security game:

Setup: Challenger runs algorithm Setup and gives the public key PK to adversary.

Phase 1: Adversary repeatedly generates private keys of corresponding attributes set $S_1, S_2, ..., S_{q_1}$.

Challenge: Adversary offers two message M_0, M_1 with same length. Besides, attributes sets $S_1, S_2, ..., S_{q_1}$ provided by adversary cannot satisfy access policy A^*. Challenger randomly choose $b \in \{0, 1\}$, encrypt M_b under A^*, and send ciphertext CT^* to adversary.

Phase 2: Adversary provides attributes sets $S_{q_1+1}, S_{q_2+1}, ..., S_q$, and these sets cannot satisfy the access policy, repeat phase 1.

Guess: Adversary output the guess b' of b.

In the above game, the advantage of $\mathcal{A}_1$ is $Pr[b' = b] - \frac{1}{2}$. Note that this model can be used in phase 1 and phase 2 to allow the decryption of the adversary query to be extended to handle the case of chosen plaintext attack.

Definition 3. *In above security game, this scheme is secure if the adversary has the advantage that can be ignored in polynomial time.*

For the second attackers, game between the attacker $\mathcal{A}_2$ and the challenger $\mathcal{C}$ can be described as follows:

Setup: Challenger $\mathcal{C}$ runs algorithm Setup and sends public key PK to adversary. Next, challenger runs algorithm K to generate authorization key, and runs algorithm E to encrypt authorization certificate, then upload the ciphertext with access policy to the cloud.

Authorization asks: $\mathcal{A}_2$ ask the authorization of certificate for authority center. The authority runs algorithm V to verify the identity of attacker. When passing the verification, attacker obtains the certificate.

Proxy signature ask: $\mathcal{A}_2$ asks the proxy signature for any message $m \in \{0, 1\}^*$ from $\mathcal{C}$. If necessary, $\mathcal{C}$ firstly runs agent protocol (D, P) and generates the certificate of w. $\mathcal{C}$ runs proxy signature algorithm PS to generate the proxy signature $p\sigma$ about message $m \in \{0, 1\}^*$ with satisfying the certificate w, and then sends $p\sigma$ to $\mathcal{A}_2$.

Output: Game over, the adversary $\mathcal{A}_2$ outputs $m^*, w^*, p\sigma^*$. If the following conditions are established, then the attacker wins the game:

(1) Adversary does not query the authority of certificate w^*;
(2) Adversary does not query the proxy signature of $m^*, p\sigma^*$;
(3) $PV(m^*, p\sigma^*, y_A^*, y_B^*) = 1$.

The possibility to win the game is ε for adversary $\mathcal{A}_2$ within time t, after q_d times authority query and q_s times proxy signature query, $\mathcal{A}_2$ is called $(\varepsilon, t, q_d, q_s)$ attacker of proxy signature scheme. If the ε is negligible, the scheme is safe for the original signer.

3 Proposed Scheme

In this section, we propose an attribute-based anonymous proxy signature scheme and security analysis.

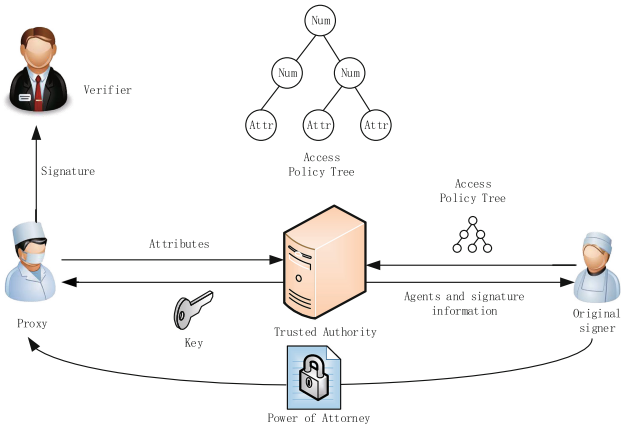

Fig. 1. System architecture model

3.1 Scheme Description

Suppose Alice is an origin signer, $\mu = \{\mu_1, \mu_2, ..., \mu_n\}$ is the collection of proxy signers, $\mu_i (1 \leq i \leq n)$ is a certain proxy signer. The scheme contains the following parts:

(1) Setup: G_0, G_1 are cyclic group with order p, bilinear mapping $e : G_0 \times G_0 \rightarrow G_1$ generator $g \in G_0$. randomly select security parameter κ this security parameter determines the scale of group. Meanwhile, this algorithm defines Lagrange coefficient $\Delta_{i,s} \in Z_p$, S is an element in Z_p: $\Delta_{i,s}(x) = \prod_{j \in S, j \neq i} \frac{x-j}{i-j}$. By hash function $H : \{0,1\}^* \rightarrow G_0$, any attribute of binary string description can be mapped to any random group. The encryption hash function is $H_0 : \{0,1\}^* \times G_0 \rightarrow Z_q^*$, $H_1 : \{0,1\}^* \rightarrow \{0,1\}^k$. Randomly selects $\alpha, \beta \in Z_p$ and generates system public key: $PK = \{G_0, g, h = g^\beta, e(g,g)^\alpha\}$, master key: $MK = \beta, g^\alpha$.

(2) Key Generation: Origin signer randomly selects $x_0 \in Z_q^*$ as the private key, and the public key is $Y_0 = x_0 \cdot g$. Similarly, each proxy signer selects $x_i \in Z_q^*$ as private key, the public key is $Y_0 = x_0 \cdot g$.

(3) Signature Stage: Before this phase, origin signer needs to encrypt authorization certificate and sends it with access policy to the authority. The authorization of signature is conducted by attributes, only when the agent satisfies the access policy designed by origin signer, he/she can be authorized.

(1) Construction of signature authorization certificate: Origin signer generates certificate m_w which contains the validate time of proxy signature authorization, identity of origin signer, the identity of all the proxy signer and the scope of the signing of the message. Selects a random number $\theta \in Z_q^*$ and computes $\Theta = \theta g, \lambda = \theta + x_0 H_0(m_w, \Theta) \bmod p$, then sends the $\{m_w, \Theta, \lambda\}$ to the authority center.

(2) Encryption algorithm: This algorithm encrypts m_w under the access structure τ. Algorithm firstly chooses a polynomial q_x for every node (including the leaf node) in τ. To begin with the root R. from top to the bottom, selects the polynomial. The degree d_x of the polynomial q_x of the node x is one less than the threshold k_x, which is $d_x = k_x - 1$.

The algorithm selects a random number $s \in Z_p$ from the root R and sets $q_R(0) = s$. Then, the algorithm selects d_R points from q_R to define q_R. As for other vertex x, sets $q_x(0) = q_{parent(x)}(index(x))$, randomly selects other d_x points to define q_x. Suppose Y is the collection of leaf nodes in τ, so we can get the ciphertext under the access tree τ:

$$CT = \{\tau, \widetilde{C} = m_w \cdot e(g,g)^{\alpha s}, C = h^s, \forall y \in Y : C_y = g^{q_y(0)}, C'_y = H(att(y))^{q_y(0)}\}$$

(3) Access key generation algorithm: This algorithm inputs the attributes set S and outputs the secrete key denoted by S. Algorithm firstly selects a random number $r \in Z_p$, and for each randomly selects $j \in S$, then computes the private key:

$$SK = \{D = g^{(\alpha+r)/\beta}, \forall j \in S : D_j = g^r \cdot H(j)^{r_j}, D_j^* = g^{r_j}\}$$

(4) Decryption algorithm: Decryption algorithm is a recursive algorithm. For the sake of simplicity, we present the simplest form of decryption algorithm in this paper. Firstly, we define the recursive algorithm $Decrypt(PK, CT, x)$, ciphertext CT, the private key SK associated with attributes set $S$the node x in τ are the input. when x is the leaf node, set $i = att(x)$, if $iinS$, then

$$\begin{aligned} DecryptNode(CT, SK, x) &= \frac{e(D_i, C_x)}{e(D'_i, C'_x)} \\ &= \frac{e(g^r \cdot H(i)^{r_i}, g^{q_x(0)})}{e(g^{r_i}, H(i)^{q_x(0)})} = e(g,g)^{rq_x(0)} \end{aligned} \tag{1}$$

if $i \notin S$, then $Decrypt(PK, CT, x) = \perp$.

Now consider the recursive case when x is not the leaf node. The working methods of the algorithm $Decrypt(PK, CT, x)$ are as follows: for all the leaf nodes z in x, calculates $F_z = Decrypt(PK, CT, z)$. Suppose S_x is the collection of leaf node z with size k_x and satisfying $F_z \neq \perp$. If there is no such collection, then the node is not satisfied, and the function returns $\perp$; otherwise calculates $F_x = \prod F_z^{\Delta_{i,s'_x}(0)}$, where $i = index(z), S'_x = \{index(z) : z \in S_x\}$.

$$\begin{aligned} F_x &= \prod_{z \in S(x)} F_z^{\Delta_{i,s'_x}(0)} = \prod_{z \in S(x)} (e(g,g)^{r \cdot q_z(0)})^{\Delta_{i,s'_x}(0)} \\ &= \prod_{z \in S(x)} (e(g,g)^{r \cdot q_{parent(z)}(index(z))})^{\Delta_{i,s'_x}(0)} \\ &= \prod_{z \in S(x)} (e(g,g)^{r \cdot q_x(0) \cdot \Delta_{i,s'_x}(0)} \\ &= e(g,g)^{r \cdot q_x(0)} \end{aligned} \tag{2}$$

After defining function $DecryptNode$, we define decryption algorithm. This algorithm first runs $Decrypt(CT, SK, R)$, R is the root of tree τ. If the tree satisfies S, the algorithm sets:

$$A = Decrypt(CT, SK, R) = e(g,g)^{rq_R(0)} \tag{3}$$

Decrypting by the following decryption algorithm:

$$\widetilde{C} / \frac{e(C, D)}{A} = \widetilde{C} / \frac{e(h^s, g^{(\alpha + r)/\beta})}{e(g,g)^{rs}} = m_w \tag{4}$$

The signature authorization is carried out in the authority center, if the attributes of proxy signer match the access policy, she/he can decrypt the ciphertext $\widetilde{C}$ to get m_w. The authority center randomly selects k_i and computes $PID_i = H_1(k_i, ID_i)$ as the identity of the proxy signer $\mu_i$where ID_i is the real identity of μ_i. Then, the authority sends Θ, λ, PID_i to proxy signer through secure channel. When proxy signer receives Θ, λ, PID_i, the proxy μ_i verify the equation $\lambda g = \Theta + H_0(m_w, \Theta)Y_0$. If the equation is correct, the authority will accept authorization, or will reject it.

(4) Signature Phase: After obtaining the access authorization, the agent will calculate the proxy private key, and replace the original signature on file according to the definition of proxy authorization to.

(1) Generation of signature private key: After obtaining m_w, the signer randomly selects $k \in Z_p^*$ and calculates signature private key $psk_s = k(\lambda + x_s H_0(m_w, \Theta))$.

(2) Signing: The process is relatively simple, only needs to calculate the four signature components.

$$\begin{aligned} V &= k \cdot H_0(m_w, \Theta) \\ \hat{Y} &= k \sum_{i=1, i \neq s}^{n} (Y_0 + Y_i) \\ \sigma_s &= psk_s^{-1} \cdot H(m_w || m) \\ \Theta' &= k\Theta \end{aligned} \tag{5}$$

Signature can be obtained after calculating the above signature component

$$\sigma = \{\sigma_s, m, m_w, \Theta', \Theta, V, \hat{Y}, PID_s\}$$

(5) Validation Phase: After signing the documents, the verifier needs to verify the signature when viewing the file. According to the public key of proxy signer $Y_0, Y_1, ..., Y_n$ and the given anonymous proxy signature s, verifier verifies the following equation:

$$e(n\Theta' + v\sum_{i=1}^{n}(Y_0 + Y_1), \sigma_s) = e(g, H(m_w||m))e((n-1)\Theta' + H_0(m_w, \Theta)\hat{Y}, \sigma_s) \tag{6}$$

If the equation is correct, verifier will accept signature, or will reject it.

3.2 Correctness Verification

In the description of the scheme in detail in section, an agent after the access to the agency that is able to file for the signature, this scheme for the validity of the signature can be directly by the following equation:

$$\begin{aligned} & t_i g = (x_i h_0(m_w, K_i) + k_i)g = h_0(m_w, K_i)x_i g + k_i g = Y_i h_0(m_w, K_i) + K_i \\ & \lambda g = (\theta + x_0 H_0(m_w, \Theta))g = \theta g + H_0(m_w, \Theta)x_0 g = \Theta + H_0(m_w, \Theta)Y_0 \\ & e(\sum_{i=1}^{n}(R' + V(Y_0 + Y_i)), \sigma_s) \\ & = e(\sum_{i=1, i \neq s}^{n} \Theta' + V(Y_0 + Y_s), \sigma_s)e(\Theta' + V(Y_0 + Y_s), \sigma_s) \\ & = e(\sum_{i=1, i \neq s}^{n} \Theta' + V(Y_0 + Y_s), \sigma_s)e(\Theta' + V(Y_0 + Y_s), psk^{-1}H(m_w||m)) \\ & = e(P, H(m_w||m))e((n-1)\Theta' + H_0(m_w, \Theta)\hat{Y}, \sigma_s) \end{aligned} \tag{7}$$

3.3 Safety Analysis

Definition 4. *If in the polynomial time, the adversary can win the above game with negligible advantage, so the proposed scheme can achieve CPA security.*

Theorem 1. *If the adversary can break the security model, there is at least one polynomial time algorithm which can solve the DBDH problem without the negligible advantage.*

Proof. Suppose the adversary $\mathcal{A}$ can break the MHM-ABE algorithm with the nonnegligible advantage according to the security model, then we will prove the DBDH problem can be solved with the nonnegligible advantage $\frac{\varepsilon}{2}$.

Define the bilinear mapping $e : G_0 \times G_0 \to G_1$, G_0 is a multiply cyclic group with order p and generator g. First, the challenger of DBDH flips a coin b, and sets: $(g, A, B, C, Z) := \{(g, g^a, g^b, g^c, e(g,g)^{abc}), b = 0(g, g^a, g^b, g^c, e(g,g)^z), b = 1$, where $a, b, c, z \in Z_p$ are ransom numbers. Challenger then sends $(g, A, B, C, Z) = (g, g^a, g^b, g^c, Z)$ to simulator, in the following DBDH game, the simulator acts as the challenger.

(**1**) **System Initialization:** The adversary $\mathcal{A}$ chooses an access policy T^*.

(**2**) **System Setup:** Simulator $\mathcal{C}$ runs the parameter initialization algorithm in the proposed scheme, and generates the system public key

$$PK_0 = (G_0, g, h = g^\beta, f = g^{\frac{1}{\beta}}, e(g,g)^\alpha) \tag{8}$$

System master key $MK = (\beta, g^\alpha)$, $\mathcal{C}$ keeps MK_0, and sends PK_0 to adversary.

(**3**) **Query Phase 1:** The adversary requests the secrete key for the attribute sets $\{A_1, A_2, ..., A_q\}$, but any $A_i, 1 \le i \le q$ cannot satisfy the access tree T^*, simulator will call the secrete key construction method to calculate:

$$\begin{aligned} Du^{(k)} &= g^{\frac{\alpha^{(k)} + ru^{(k)}}{\beta_{k,1}}}, \\ Du_{i,j}^{(k)} &= g^{ru_i^{(k)}} \cdot H(au_{i,j}^{(k)})^{ru_{i,j}^{(k)}}, \\ Du'^{(k)}_{i,j} &= g^{ru_{i,j}^{(k)}} \end{aligned} \tag{9}$$

then sends $SK_i, 1 \le i \le q$ to adversary.

(**4**) **Challenge Phase:** Adversary chooses the plaintext M_0, M_1 with the same length and sends them to $\mathcal{C}$, $\mathcal{C}$ flips a coin μ and $\mu \in \{0, 1\}$, then encrypts M_μ by T^*, finally sends the ciphertext CT^* to adversary.

$$\begin{aligned} &CT^* = \{T^*, \tilde{C} = M_\mu \cdot Z, \{C^{(w)} = h_{w,1}^\theta, \bar{C}^{(w)} = h_{w,2}^\theta \\ &\forall y^{(w)} \in Y^{(w)} : C_y^{(w)} = g^{q_y(0)}, C'^{(w)}_y = H(attr)^{q_y(0)}, \\ &\forall x^{(w)} \in X^{(w)} : \hat{C}_x^{(w)} = h_{w,2}^{q_x(0)}\}_{w=1}^W\} \end{aligned} \tag{10}$$

When $b = 0$, we defineand $Z = e(g,g)^{abc}$ set $c = \theta$, so the ciphertext CT^* is a ciphertext, because $\tilde{C} = M_\mu \cdot Z = M_\mu \cdot e(g,g)^{abc} = M_\mu \cdot e(g,g)^{\alpha\theta}$. Otherwise, when $b = 1$, $Z = (g,g)^z, \tilde{C} = M_\mu \cdot Z = M_\mu \cdot e(g,g)^z$, Z is randomly chosen and dosen't relate to system, so $\tilde{C}$ is a random generator in G_0 and contains nothing about M_μ.

(**5**) **Query Phase 2:** Repeat the operation in query phase 1.

(**6**) **Guess:** Adversary outputs the guess of μ. If it is correct, which means $\mu = \mu'$, the simulator outputs $b' = 0$, which means the received tuple is DBDH tuple $(g, g^a, g^b, g^c, e(g,g)^{abc})$. Otherwise, the simulator outputs $b' = 1$, which means the received tuple is the random tuple $(g, g^a, g^b, g^c, e(g,g)^z)$.

In the above DBDH game, if $b = 1$, the adversary dosen't receive any information about M_μ, so $Pr[\mu' \neq \mu | b = 1] = \frac{1}{2}$. When $\mu' \neq \mu$, simulator guesses $b' = 1$, so $Pr[b' = b | b = 1] = \frac{1}{2}$.

If $b = 0$, the adversary can get the ciphertext M_μ, according to the definition, the adversary can break our scheme with the nonnegligible advantage, so $Pr[\mu' \neq \mu | b = 0] = \frac{1}{2} + \varepsilon$. When $\mu' = \mu$, simulator guesses $b' = 0$, so $Pr[\mu' \neq \mu | b = 1] = \frac{1}{2} + \varepsilon$.

Generally, the advantage that simulator in the above DBDH game can rightly guess $b' = b$ is:

$$\begin{aligned} Adv_c &= Pr[b' = b] - \frac{1}{2} \\ &= \frac{1}{2} Pr[b' = b | b = 1] + \frac{1}{2} Pr[b' = b | b = 0] - \frac{1}{2} \\ &= \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot (\frac{1}{2} + \varepsilon) - \frac{1}{2} \\ &= \frac{\varepsilon}{2} \end{aligned} \tag{11}$$

From the above analysis, if adversary $\mathcal{A}$ can break the security model with the nonnegligible advantage ε, then there exists an algorithm which can solve the DBDH problem with the advantage $\frac{\varepsilon}{2}$ in polynomial time.

Verifiability. In the signature $\sigma = \{\sigma_s, m, m_w, R', R, V, \hat{Y}, PID_s\}$, there is a proxy authorization m_w, and the participation of the original signer's public key is needed in the verification. Therefore, the verifier is convinced that the anonymous proxy signature by the original signer's authorization, which can meet the verifiability.

Traceability. In the case of disputes, the verifier will send the signature to the authorization server, the authorization server can reveal the identity of anonymous proxy signature. When receiving the proxy signature, the authorization server extracts PID_i from signature and searches corresponding ID_i from the stored information, so as to determine the identity of the proxy signature, so the traceability can be met.

3.4 Performance Analysis

In this section, the security and computing performance of the proposed scheme is compared with Yu. The comparison results are shown in Tables 1 and 2, e is bilinear mapping, P_a and P_b are the group multiplication and addition operations, respectively, n is the number of proxy signer, k represent a number of the property. Table 1 is the proposed scheme compared with Yu et al. on security,

Table 1. Safety comparison

Scheme	Anonymity	Unforgeability	Traceability
Yu	Yes	No	No
Our	Yes	Yes	Yes

Table 2. Performance comparison

Scheme	Keygen	Authorization	Signatures	Verification
Yu	Same	P_a	$(3n-2)P_a + (n+1)P_b$	$(n+1)e + nP_a + 2nP_a$
Our	Same	$ke + P_a$	$3P_a + (n-1)P_b$	$3e + 2P_a + 2nP_a$

Table 2 is compared with the proposed scheme and Yu et al. on computational cost.

From the table we can see that in the key generation phase, the two schemes have the same efficiency. During the delegation stage, the scheme in this paper has lower efficiency compared to Yu's scheme, but the authorization phase is completed in the trusted authority, which dose not occupy the signer and the proxys computing resource. In the signature generation and verification phase, when $n > 2$, the computational efficiency of the proposed scheme is higher than Yu's scheme, and with the increase of N, the efficiency advantages are more apparent. To implement anonymity, the number of the proxy signer is far greater than 2. Thus, the computational efficiency of the scheme in this paper is better than Yu's the anonymous signature scheme.

4 Conclusion

This paper mainly focuses on attribute-based access control, and how to apply this method to authorize to signature proxy in mobile healthcare. The privacy of proxy can be protected through anonymity and the malicious users can be traced when controversy occurred. Although some research results have been achieved, there are still some problems that need to be modified and concerned:

In the existing attribute-based schemes, the bilinear mapping is wildly used, because of its complexity, when attributes are large, the computational efficiency is not good enough. It needs further study that how to reduce time and computational overhead or design a new access structure to reduce the number of matching. Research on how to apply attribute-based encryption into medical signature is still in the primary stage, in real application, there are more actual demand, such as how to revoke without updating all the access policy when a malicious user is traced, meanwhile, the security and confidentiality can be guaranteed, which is the follow-up research work. In the proposed attribute-based proxy signature scheme, we suppose each user only has one key, which means he/she only has one attributes set. But in real application scenarios, users may

have multiple identities, the agent may be also a doctor. How to cope with the multiple identities and prevent unauthorized illegal access also need to be addressed.

Acknowledgments. This work is supported in part by the National Natural Science Foundation of China under Grant Numbers 61632009, 61472451 and 61272151, and the High Level Talents Program of Higher Education in Guangdong Province under Funding Support Number 2016ZJ01. The Fundamental Research Funds for the Central Universities of Central South University 2016zzts339.

References

1. Santos-Pereira, C, Augusto, A.B, Cruz-Correia, R., et al.: A secure RBAC mobile agent access control model for healthcare institutions. In: Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems, pp. 349–354. IEEE (2013)
2. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005). doi:10.1007/11426639_7
3. Kim, Y.S., Chang, J.H.: Self proxy signature scheme. IJCSNS Int. J. Comput. Sci. Netw. Secur. **7**(2), 335–338 (2007)
4. Yu, Y., Xu, C., Huang, X., et al.: An efficient anonymous proxy signature scheme with provable security. Comput. Stan. Interfaces **31**(2), 348–353 (2009)
5. Zhou, J., Cao, Z., Dong, X., et al.: 4S: a secure and privacy-preserving key management scheme for cloud-assisted wireless body area network in m-healthcare social networks. Inf. Sci. **314**, 255–276 (2015)
6. Wang, G., Lu, R., Huang, C.: PSLP: Privacy-preserving single-layer perceptron learning for e-Healthcare. In 2015 10th International Conference on Information, Communications and Signal Processing (ICICS), pp. 1–5. IEEE (2015)
7. Son, J., Kim, J.D., Na, H.S., et al.: Dynamic access control model for privacy preserving personalized healthcare in cloud environment. Technol. Health Care **24**(S1), S123–S129 (2015)
8. Guo, F., Mu, Y., Susilo, W., et al.: CP-ABE with constant-size keys for lightweight devices. IEEE Trans. Inf. Forensics Secur. **9**(5), 763–771 (2014)
9. Bodong, C., et al.: A scheme supporting efficient attribute revocation for cloud storage based on CPABE. In: International Conference on Computer Science and Service System, pp. 736–740 (2014)
10. Schobel, J., Schickler, M., Pryss, R., Nienhaus, H., Reichert, M: Using vital sensors in mobile healthcare business applications: challenges, examples, lessons learned. In: International Conference on Web Information Systems and Technologies, pp. 509–518 (2013)
11. Yu, Y.C., Hou, T.W.: An efficient forward-secure group certificate digital signature scheme to enhance EMR authentication process. Med. BIol. Eng. Comput. **52**(5), 449–457 (2014)
12. Fadini, G.P., Albiero, M., Millioni, R., et al.: The molecular signature of impaired diabetic wound healing identifies serpinB3 as a healing biomarker. Diabetologia **57**(9), 1947–1956 (2014)
13. Rahman, F., Bhuiyan, M.Z.A., Ahamed, S.I.: A privacy preserving framework for RFID based healthcare systems. Future Gener. Comput. Syst. (2016). doi:10.1016/j.future.2016.06.001