# Analysis and Improvement of Patient Self-controllable Multi-level Privacy-preserving Cooperative Authentication Scheme

Yang Zhao[1], Feng Yue[1], Songyang Wu[2], Hu Xiong[1,3], and Zhiguang Qin[1]
*(Corresponding author: Songyang Wu)*

School of Computer Science and Engineering & University of Electronic Science and Technology of China[1]
No. 4, North Jianshe Road, Chenghua District, chengdu, Sichuan, 610054, China
The Third Research Institute of Ministry of Public Security[2]
No. 76, Yueyang Road, Shanghai, 201204, China
(Email: wusongyang@stars.org.cn)
State Key Laboratory of Information Security, Institute of Software & Chinese Academy of Sciences[3]
No. 19, Yuquan Road, Shijingshan District, Beijing, 100190, China
*(Received Mar. 29, 2015; revised and accepted May 15 & June 4, 2015)*

## Abstract

In 2014, a patient self-controllable multi-level privacy-preserving cooperative authentication scheme (PSMPA) was proposed for attempting to address the issue of data confidentiality and patients' identity privacy simultaneously when the personal healthcare record (PHR) is shared in the distributed m-healthcare cloud computing system. In this paper, we show the PSMPA scheme fails to achieve the two goals under the collusion attack. Furthermore, the scheme also suffers from forgery attack because of a flawed design in the transcript simulation phase. In order to avoid the attacks, we propose an improved PHR sharing scheme by incorporating ciphertext policy attribute-based encryption (CP-ABE) and attribute-based signature (ABS) as a possible solution.

*Keywords: Access control, attribute-based encryption, data confidentiality, identity privacy*

## 1 Introduction

Motivated by the remarkable development of cloud computing, more and more significant data is stored into the cloud for sharing, including personal health record (PHR) absolutely. The e-healthcare service attracts much more attention than the traditional approaches due to its fascinating features such as high efficiency, universal accessibility and low cost. The patients can share their PHRs in the cloud to obtain treatment recommendations from physicians or to provide medical research institutions with precious medical information. However, on account of storing PHRs in the cloud far away from the patients, the PHRs are out of their physical control. The data confidentiality and patient's identity privacy will face enormous threatens which are bound to the obstacles of its wide adoption. To minimize users' concerns as far as possible, a lot of data sharing schemes in distributed cloud computing system have been proposed so far where cryptography is utilized popularly.

It's natural to think of leveraging the access control in the e-healthcare scheme. Access control enables the patients to delegate different privilege for accessing the PHRs to whoever they desire with freedom. ABE is considered as the most optimal solution to realizing fine-grained access control for sensitive data in the cloud environment. A number of literatures on ABE have been published in the past. Especially, in 2006 Goyal *et al.* [3] proposed key-policy attribute-based encryption fine-grained access control of encrypted data which makes key management more efficient during data sharing. Similarly, Bethencourt *et al.* [1] put forward the concept of ciphertext-policy attribute-based encryption which is parallel with KP-ABE. CP-ABE and KP-ABE are applied to different scenes dependent on their respective specialties. Nevertheless, both of them are short of efficient and dynamic attribute revocation mechanism which is indispensable. Based on ABE mentioned in [1, 3], [4, 11, 12] are proposed one after another. However, the single attribute authority that is responsible for distributing attributes becomes the bottleneck of these schemes inevitably. In 2009, Chase *et al.* [2] figured out a solution called multiple-authority ABE where multiple attribute authorities are requested to involve in distributing attributes. On the basis of [2], Li *et al.* [6, 7] divided the members in the cloud into various security domains for the purpose of decreasing the key management complexity

further. In 2013, Lee *et al.* [5] carried on a comprehensive survey on the existing ABE schemes and ran an extended analysis on KP-ABE and CP-ABE. In the same year, Li *et al.* [8] proposed the first multi-authority attribute based encryption scheme realizing such expressive access policy and constant ciphertext size. As in the real world circumstance, the attributes are always in the different levels, Liu *et al.* [9] proposed a scheme called ciphertext-policy hierarchical attribute based encryption in 2014. The above schemes mainly concentrate on achieving data confidentiality, while the user's identity privacy is neglected.

Recently, Zhou *et al.* [13] proposed a novel PSMPA aiming at guaranteeing the patient's identity privacy. The PHRs are divided into patient's identity information and healthcare data creatively and each of them is encrypted respectively. No one can decrypt the patient's authentic identity except the directly authorized physicians the patient has appointed personally. By this means, they claimed their scheme can satisfy the requirement of identity privacy. However, we find that the parent's identity privacy and healthcare data are vulnerable because the PSMPA is unable to resist the collusion attack from the dishonest physicians. In addition, the scheme also suffers from forgery attack because of a flawed design in the transcript simulation phase. Incorporating CP-ABE and ABS, we propose an improved PHRs sharing scheme as a possible solution.

The rest paper is organized as follows: In the Section 2, we review Zhou *et al.* [13]'s PSMPA scheme in detail. Our attacks against the PSMPA scheme are demonstrated in Section 3. In Section 4, we show a possible solution and Section 5 is the final conclusion.

# 2 Review of the PSMPA Scheme

In this section, we carry out a detailed statement on the PSMPA scheme to prepare for the analysis and the attacks in Section 3.

## 2.1 Network Model

As is illustrated in Figure 1, in the m-healthcare cloud computing system, all the members are classified into three levels of security: the directly authorized physicians such as Bob in the local healthcare provider, the indirectly authorized physicians such as Jack, Tom and Jim in the remote healthcare providers and the unauthorized persons such as Black. The directly authorized physicians are authorized by the patients and can not only access the patient's personal health record but also recognize the patient's identity. The indirectly authorized physicians are authorized by the directly authorized physicians for medical consultant or some research purposes (since they are not authorized by the patients, we call them 'indirectly authorized' instead). The only right they have is accessing the personal health record, but not the patient's identity. For the unauthorized persons, neither could be obtained.

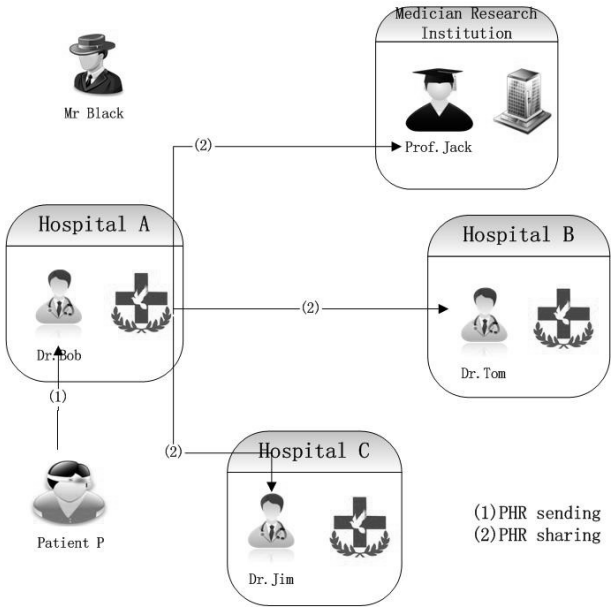


Figure 1: An overview of the M-healthcare cloud computing system

## 2.2 Authorized Accessible Privacy Model (AAPM)

A novel attribute based designated verifier signature scheme (ADVS) is proposed by Zhou *et al.* [13] to realize three levels of security and privacy requirement in distributed m-healthcare cloud computing system which is mainly constituted of the following five algorithms: Setup, Key Extraction, Sign, Verify and Transcript Simulation Generation. Suppose the universe set of attributes is $U$. If and only if $\mathbb{A}(\omega) = 1$ where $\omega$ is selected from $U$, We say an attribute set $\omega$ satisfies a specific access structure $\mathbb{A}$. The five phases are presented as follows.

**Setup.** The algorithm takes $1^l$ as input, where $l$ is the security parameter. It outputs public parameters and $y$ as the master key for the central attribute authority.

**Key Extract.** Assume that a physician requests the attribute keys for an attribute set $\omega_D \in U$. If he is qualified to be issued with $sk_D$ for these attributes, the attribute authority produces $sk_D$ for him.

**Sign.** The patient takes as input his private key $sk_P$, the uniform public key $pk_D$ of the healthcare provider which the physicians work in and a personal healthcare information $m$ to generate a signature $\sigma$. Namely, $\sigma \leftarrow Sign(sk_P, pk_D, m)$.

**Verify.** Suppose that a physician wants to validate the correction of a signature $\sigma$ which contains an access structure $\mathbb{A}$ and owns a subset of attributes $\omega_J \subseteq \omega_D$ satisfying $\mathbb{A}(\omega_J) = 1$, a deterministic verification algorithm can be executed. Once receiving a signature $\sigma$, he uses his attribute private key $sk_D$ and the patient's public key $pk_P$, then outputs the message $m$

and $True$ if the signature is correct, or $\perp$ otherwise. Namely, $\{True, \perp\} \leftarrow Verify(sk_D, pk_P, m, \sigma)$.

**Transcript Simulation Generation.** Through the Transcript Simulation algorithm, the directly authorized physicians who kept the authorized private key $sk_D$ can always produce identical distributed transcripts indistinguishable from the signature which is received from the patient.

## 2.3 PSMPA Design

In this section, we introduce the proposed PSMPA to implement AAPM mentioned above, realizing three different levels of security and privacy requirements. Most of the notations which are useful in our scheme are showed in Table 1 with the corresponding description.

**Setup.** Assume that $\mathbb{G}_0$ and $\mathbb{G}_1$ are two bilinear groups of prime order $p$ and $g$ is a generator of $\mathbb{G}_0$. Moreover, let $\hat{e} : \mathbb{G}_0 \times \mathbb{G}_0 \to \mathbb{G}_1$ denote a bilinear map. Pick $g_1 \in \mathbb{G}_0$, $y \in \mathbb{Z}_p^*$ at random and compute $g_2 = g^y$. We additionally employ three collision-resistant hash functions: $H_0 : \{0,1\}^* \to \mathbb{G}_0$, $H_1 : \{0,1\}^* \to \mathbb{Z}_p^*$ and $H_2 : \mathbb{G}_1 \to \{0,1\}^{k_{Enc}}$ where $k_{Enc}$ is the length of symmetric key in the secure private key encryption construction chosen by the patient. Then, define the attributes in universe $U$ as elements in $\mathbb{Z}_p$. If $q_x(\cdot)$ is a polynomial related to leaf nodes, a default attribute set from $\mathbb{Z}_p$ with the size of $d_x - 1$ is denoted by $\Psi_x = \{\Psi_1, \Psi_2, \cdots, \Psi_{d_x-1}\}$ in the access tree.

**Key Extract.** The patient choose $b \in \mathbb{Z}_p^*$ and $B = g^b$ as his private key and public key. We define the patient's registered local healthcare provider's uniform private key is $sk^{HP} = hc$ and the corresponding public key is $pk^{HP} = g_1^{hc}$. Both of the keys are shared by each physician working in it. The attribute private key of the physician can be

$$sk_D = (\gamma_i, \delta_i) = ((g_1 H_0(i))^{q_x(i)}, g^{q_x(i)})_{i \in \omega_D \cup \Psi_x},$$

and the public parameters are

$$(p, g, \hat{e}, \mathbb{G}_0, \mathbb{G}_1, H_0, H_1, H_2, g_1, g_2).$$

**Sign.** The signing algorithm produces a signature of the patient's personal health information $m$ which can only be decoded and validated by the directly authorized physicians whose sets of attributes enable to satisfy the patients' requirements. First of all, the patient need to construct a polynomial $q_x(\cdot)$ for each node $x$ in $\Gamma$ of the degree $D_x = d_x - 1$.

Beginning with the root node $R$, the algorithm chooses a random $y \in \mathbb{Z}_p$ and sets $q_R(0) = y$. Then, it chooses $d_R - 1$ other points on the polynomial $q_R$ randomly to define it completely. For any other node $x$, it sets $q_x(0) = q_{parent(x)}(index(x))$ and chooses $d_x - 1$ other points randomly to completely define $q_x(\cdot)$.

To sign a message $m$ with the verification predicate $\Gamma$, for the leaf node $x$ in the access tree $\Gamma$, let the current threshold required for the physician be $k_x$. For the leaf node polynomial $q_x(\cdot)$, the patient randomly selects a default subset $\Psi'_x \subseteq \Psi_x$ with $|\Psi'_x| = d_x - k_x$ and calculates $B_{P_i} = H_0(i)^b$ for $i \in \omega_x^* \cup \Psi'_x$. Then, he can derive the corresponding keys for authentication

$$\begin{aligned} K_{Encp} &= \hat{e}(g_1, g_2)^b, \\ K_{Enc} &= H_2(K_{Encp}), \\ K_{Sig} &= K_{Encp}\hat{e}(pk^{HP}, g_2). \end{aligned}$$

Finally, the patient randomly selects $r_i \in \mathbb{Z}_p^*$ for each $i \in \omega_x^* \cup \Psi'_x$, publishes $g^{r_i} (i \in \omega_x^* \cup \Psi'_x)$ and completes the signature as follows

$$\begin{aligned} \sigma' &= H_1(m \parallel K_{Sig}), \\ C_0 &= E_{pk^{HP}}(B \parallel B_{P_i}), \\ C &= E_{K_{Enc}}(m), \\ \sigma''_i &= \{H_0(i)^{r_i}\}_{i \in \omega_x \cup \Psi'_x}, \\ \sigma''' &= H_0(m)^b, \end{aligned}$$

where $E_{pk^{HP}}(\cdot)$, $E_{K_{Enc}}(\cdot)$ are secure public key and private key encryptions chosen by the patient. At last, he can export the signature $\sigma = (\omega_x^*, C_0, C, \sigma', \sigma''_i, \sigma''')$.

**Verify.** Upon obtaining the signature $\sigma$, the physicians working in the patient's registered local healthcare provider can firstly decipher $B \parallel B_{P_i} = D_{sk^{HP}}(C_0)$, where $D_{sk^{HP}}(\cdot)$ is the decryption algorithm of the public key encryption. If the set of attributes kept by the physician satisfies the access tree $\Gamma$, the patient is able to further finish the verification by implementing a recursive algorithm illustrated as follows.

For the leaf node $x$, to testify the signature with the node predicate, that is to prove possessing at least $k_x$ attributes among an attribute set $\omega_x$ with the size of $n_x$, the physician firstly selects a subset $\omega_J \subseteq \omega_D \cap \omega_x^*$ of the size $k_x$, chooses $r'_i \in_R \mathbb{Z}_p^*$ for each $i \in \omega_x^* \cup \Psi'_x$ and computes

$$\begin{aligned} V' &= \prod_{i \in \omega_J \cup \Psi'_x} \gamma_i^{\Delta_{i,\omega_J \cup \Psi'_x}(0)}, \\ V'' &= \prod_{i \in \omega_x^* \cup \Psi'_x} (\sigma''_i)^{r'_i}, \\ V''' &= \prod_{i \in \omega_J \cup \Psi'_x} \hat{e}(B_{P_i}, \delta_i^{\Delta_{i,\omega_J \cup \Psi'_x}(0)} g^{r_i r'_i}), \quad (1) \\ V'''' &= \prod_{i \in \omega_x^* \setminus \omega_J} \hat{e}(B_{P_i}, g^{r_i r'_i}), \quad (2) \end{aligned}$$

Table 1: Notations in the PSMPA

| Notation | Description |
|---|---|
| $d_x$ | Threshold for node $x$ in access tree $\Gamma$ |
| $k_x$ | Number of attributes required to be owned by the patient w.r.t. node $x$ |
| $q_x(\cdot)$ | $D_x = d_x - 1$ -degree polynomial assigned to node $x$ |
| $\Psi_x$ | A default attribute set of size $d_x - 1$ for node $x$ |
| $sk^{HP}$ | Uniform private key of the healthcare center |
| $pk^{HP}$ | Uniform public key of the healthcare center |
| $\omega_D$ | The set of attributes owned by the physician |
| $sk_D$ | Private key of the physician |
| $\omega_x^*$ | Attributes in predicate of node $x$ for physicians |
| $\Psi'_x$ | A subset of default attribute set of size $d_x - k_x$ chosen by the patient |
| $K_{Enc}/K_{Dec}$ | Symmetric key for message encryption/decryption |
| $K_{Sig}$ | Signing key for ADVS |
| $\omega_J$ | The subset of physician's attribute set of size $k_x$ chosen to satisfy the predicate |
| $H_0$,$H_1$,$H_2$ | Hash functions mapping $\{0,1\}^* \to \mathbb{G}_0$, $\{0,1\}^* \to \mathbb{Z}_p^*$ and $\mathbb{G}_1 \to \{0,1\}^{k_{Enc}}$ |

and

$$\begin{aligned} K^x_{Decp} &= \frac{\hat{e}(V'V'', B)}{V'''V''''} \\ &= \frac{\hat{e}(g_1^{q_x}(0)\prod_{i\in\omega_J\cup\Psi'_x} H_0(i)^{q_x(i)\Delta_{i,\omega_J\cup\Psi'_x}(0)+r_i r'_i}, g^b)}{\prod_{i\in\omega_J\cup\Psi'_x}\hat{e}(H_0(i)^b, g^{q_x(i)\Delta_{i,\omega_J\cup\Psi'_x}(0)+r_i r'_i})} \\ &\quad \cdot\frac{\hat{e}(\prod_{i\in\omega^*_x\setminus\omega_J} H_0(i)^{r_i r'_i}, g^b)}{\prod_{i\in\omega^*_x\setminus\omega_J}\hat{e}(H_0(i)^b, g^{r_i r'_i})} \\ &= \hat{e}(g_1^{q_x(0)}, g^b). \end{aligned} \quad (3)$$

We now consider the recursive case when $x$ is a non-leaf node. The verification algorithm will proceed as follows. For all nodes $z$ that are children of $x$, it calls the same verification algorithm with respect to itself and stores the corresponding partial output as $F_z$. Let $\mathbb{S}_x$ be an arbitrary $k_x$ -sized set of child nodes $z$ such that $F_z \neq \perp$. If no such set exists, the node will not be satisfied and the function will return $\perp$. Then, the physicians can compute

$$\begin{aligned} K^x_{Decp} &= \hat{e}(F_x, B) = \hat{e}(\prod_{z\in\mathbb{S}_x} F_z^{\Delta_{i,\mathbb{S}'_x}(0)}, g^b) \\ &\quad (i = index(x)\ and\ \mathbb{S}'_x = \{index(z)\} : z \in \mathbb{S}_x) \\ &= \hat{e}(\prod_{z\in\mathbb{S}_x} g_1^{q_z(0)\Delta_{i,\mathbb{S}'_x}(0)}, g^b) \\ &= \hat{e}(\prod_{z\in\mathbb{S}_x} g_1^{q_{parent(z)}(index(z))\Delta_{i,\mathbb{S}'_x}(0)}, g^b) \\ &= \hat{e}(\prod_{z\in\mathbb{S}_x} g_1^{q_x(i)\Delta_{i,\mathbb{S}'_x}(0)}, g^b) = \hat{e}(g_1^{q_x(0)}, g^b). \end{aligned}$$

Until now, we have defined the verification function for each node in the access tree $\Gamma$. By utilizing the recursive algorithm defined above, the physicians can complete verification by simply calling the function on the root node $R$ of the access tree $\Gamma$. Finally, the directly authorized physician computes

$$\begin{aligned} K_{Decp} &= \hat{e}(F_R, B) = \hat{e}(g_1^{q_R(0)}, g^b) \\ &= \hat{e}(g_1, g_2)^b, \quad (4) \\ K_{Dec} &= H_2(K_{Decp}), m = D_{K_{Dec}}(C), \quad (5) \end{aligned}$$

and verifies whether both

$$\begin{aligned} \hat{e}(g, \sigma''') &= \hat{e}(B, H_0(m)), \quad (6) \\ H_1(m \parallel K_{Decp}\hat{e}(g_1, g_2)^{hc}) &= H_1(m \parallel K_{Sig}) \\ &= \sigma', \quad (7) \end{aligned}$$

hold, where $D_{K_{Dec}}(\cdot)$ is the decryption algorithm for the private key encryption. If Equations (6) and (7) hold simultaneously, the physician outputs $True$; otherwise, outputs $\perp$.

**Transcript Simulation.** Once receiving the medical consultation or research, the directly authorized physician creates a protected session secret $SS_j$ which is unique to each consultation $j$ made for each patient. Next, he can output the transcript simulation $\sigma_T$ which is broadcasted to indirectly authorized physicians by operating the following procedures. Firstly, he computes $K^T_{Decp} = K^{H_1(SS_j)}_{Decp}$, $K^T_{Dec} = H_2(K^T_{Decp})$ to encrypt a specific message $m$ to $C_T$ and computes $\sigma'_T = H_1(m \parallel K^T_{Decp}\hat{e}(pk^{HP'}, g_2)) = H_1(m \parallel K^T_{Sig})$, in which $pk^{HP'}$ is the public key of the hospital which the indirectly authorized physician works in. After that, he can compute $B_T = B^{H_1(SS_j)}$, $B^T_{P_i} = B^{H_1(SS_j)}_{P_i}$ and encrypt them as $C^T_0 = E_{pk^{HP'}}(B_T \parallel B^T_{P_i})$. In the end, he calculates $\sigma'''_T = (\sigma''')^{H_0(SS_j)}$ and generates the transcript simulation as $\sigma_T = (\omega^*_x, C^T_0, C_T, \sigma'_T, \sigma''_i, \sigma'''_T)$ which is indistinguishable from the original signature $\sigma$ for the indirectly authorized physician.

# 3 Attacks Against the PSMPA Scheme

Through analysis and discussion, two primary flaws can be found in the patient self-controllable multi-level privacy-preserving cooperative authentication scheme (PSMPA).

## 3.1 Collusion Attack

The indirectly authorized physicians who satisfy the attributes requirement are capable of colluding with the directly authorized physicians who work in the patient's registered hospital and don't satisfy the attribute requirements.

In the proposed scheme, the above two kinds of members can exchange information and get what they shouldn't have the right to get. The former can recognize the patient's authentic identity and the latter can obtain the healthcare data. The analysis and collusion attack are presented as follows.

In the sign phase, the authentic identity of the patient $B$ is encrypted by the public key of his registered local healthcare provider just as $C_0 = E_{pk^{HP}}(B \parallel B_{P_i})$. Since all of the physicians working in the patient's registered hospital possess the private key $sk^{HP}$ of the healthcare provider which they work in, they are able to decipher the authentic identity $B$ by computing $B \parallel B_{P_i} = D_{sk^{HP}}(C_0)$, no matter whether they satisfy the access tree or not.

After the decryption, the directly authorized physicians can share $B \parallel B_{P_i}$ and the corresponding signature $\sigma = (\omega_x^*, C_0, C, \sigma', \sigma_i'', \sigma''')$ with the dishonest indirectly authorized physicians who satisfy the access tree. With their own attribute private key $sk_D = (\gamma_i, \delta_i)$ and the value of $B \parallel B_{P_i}$, the indirectly authorized physicians enable to calculate the value of $V'''$ and $V''''$ just like Equations (1) and (2) in the PSMPA, when $x$ represents a leaf node in the access tree. Then they also can compute Equation (3).

Finally, the indirectly authorized physicians implement the same operations as the primitive paper to compute $K_{Decp}$, $K_{Dec}$ and the healthcare information $m$ easily by computing Equations (4) and (5).

In this way, the indirectly authorized physicians are able to share the healthcare data $m$ with the directly authorized physicians, and they also get the authentic identity $B$ illegitimately from the latter. The collusion attack succeeds after the cooperation.

## 3.2 Forgery Attack

In the transcript simulation phase, because of the flawed design, the directly authorized physicians are equipped with the ability of deceiving the indirectly authorized ones through sharing fake healthcare data, while the latter do not notice that.

The public key infrastructure (PKI) is utilized to issue the certificate for user's public key in the paper. The PKI requires that if a patient wants to get a public key certificate from certificate authority (CA), he must pass the identity verification. In the transcript simulation phase, the directly authorized physicians randomize the patient's authentic identity by an exponent arithmetic $B^{H_1(SS_j)}$ in order to protect the patient's privacy. The blinded identity is certain to fail to get the corresponding certificate from CA. Now that the patient's public/private key pair $B_T = B^{H_1(SS_j)}$ and $b_T = b(H_1(SS_j))$ is fake completely, the directly authorized physician enables to simulate a forged signature for any healthcare data $m^*$ he likes with a fake identity $B^*$ and cheat the indirectly authorized physician as follows.

1) Signature Generation:

   a. The dishonest directly authorized physician randomly selects $b^* \in \mathbb{Z}_p^*$ as a nonexistent patient's private key and computes the corresponding public key $B^* = g^{b^*}$.

   b. A suit of new secret values will be produced with the help of the fake private and public key pair $B^*/b^*$ through computing

$$\begin{aligned} K_{Encp}^* &= \hat{e}(g_1, g_2)^{b^*}, \\ K_{Enc}^* &= H_2(K_{Encp}^*), \\ K_{Sig}^* &= K_{Encp}^* \hat{e}(pk^{HP'}, g_2), \end{aligned}$$

   where $HP'$ denotes the public key of the healthcare provider which the indirectly authorized physician works in.

   c. The forged signature can be computed as follows.

$$\begin{aligned} \sigma'^* &= H_1(m^* \parallel K_{Sig}^*), \\ C_0^* &= E_{pk^{HP'}}(B^* \parallel B_{P_i}^*), \\ C^* &= E_{K_{Enc}^*}(m^*), \\ \sigma_i''^* &= \{H_0(i)^{r_i}\}_{i \in \omega_x^* \cup \Psi_x'}, \\ \sigma'''^* &= H_0(m^*)^{b^*}, \end{aligned}$$

   where $B_{p_i}^* = H_0(i)^{b^*}$ for each $i \in \omega_x^* \cup \Psi_x'$. Then, the forged signature will be $\sigma^* = (\omega_x^*, C_0^*, C^*, \sigma'^*, \sigma_i''^*, \sigma'''^*)$.

2) Signature Verification:

   a. After receiving the signature $\sigma^*$, the indirect authorized physician firstly utilizes the healthcare provider's secret key $sk^{HP'}$ to decipher the patient's identity information by calculating $B^* \parallel B_{P_i}^* = D_{sk^{HP'}}(C_0^*)$.

   b. The indirectly authorized physician will be able to decipher the healthcare information $m^*$ and verify the correction of the signature according to the other procedures in the verification phase of PSMPA.

Through the verification, the indirectly authorized physician is convinced that $m^*$ is the healthcare information he desires and $B^*$ is the corresponding patient's authentic identity without being aware of being cheated.

In this case, the correction of transcript simulation is entirely dependent on the honesty of the directly authorized physicians. Unfortunately, the probability of this honesty guarantee is negligible in practice such that this type of data sharing mechanism is unrealistic.

# 4 Possible Solution

In this section, we provide a possible solution to avoid the above two attacks. In order to ensure that each patient has full control over his identity information and personal health information, we leverage CP-ABE proposed in [1] and ABS proposed in [10] as the encryption primitive and the signature primitive in our possible solution. Our scheme realizes the same three levels of security and privacy requirement as the PSMPA scheme. All the members are also classified into three categories: the directly authorized physicians in the local hospital, the indirectly authorized physicians in the remote hospital and the unauthorized persons. We generally describe the possible solution and discuss its security in the following.

Before encrypting the PHRs, the patients divide the PHRs into patient's identity information $m_1$ and personal health information $m_2$. To achieve the goal of access control, the CP-ABE scheme in [1] is brought in. The patients can use two different access tree $T_1$ and $T_2$ to encrypt $m_1$ and $m_2$ into $CT_1$ and $CT_2$ respectively. The set of leaf nodes in $T_2$ does not contain the attribute of the hospital where the physician works, while the access tree $T_1$ contains. The patient can define the root node of $T_1$ as an "AND" gate with two children: one is $T_2$ and the other is a leaf that is associated with the attribute of the hospital where the physician works. For example, if patient P is registered in hospital A, he can specify the attribute "hospital=A" as the leaf node of the root. In this way, only the directly authorized physicians working in hospital A whose attributes satisfy $T_2$ are able to decrypt the ciphertexts $(CT_1, CT_2)$ and get the plaintext $(m_1, m_2)$ simultaneously, while the indirectly authorized physicians working in other hospital whose attributes satisfy $T_2$ only can decrypt $CT_2$ and get $m_2$. The unauthorized persons whose attributes can not satisfy $T_2$ will obtain nothing. Through constructing the two different access tree, we realize the fine-grained access control to patient's identity information and personal health information.

As we all know, encryption offers confidentiality and signature provides authenticity, one can perform encryption and signing sequentially to achieve this. Once receiving the PHRs uploaded by someone, the storage server in hospital must check their authenticity. Traditional digital signature can undertake this task, but the patient's identity will be exposed to the ones who are not desired by the patient. In [10], Rao *et al.* constructed a key-policy ABS scheme with constant-size signature to achieve signer privacy. A valid ABS attests to the fact that "a single user, whose attributes satisfy the predicate, endorsed the message" and provides the public verifiability. The public just knows the signature comes from people who satisfy certain criteria like that they should possess some specific attributes. In our possible solution, the patients leverage ABS to sign the ciphertexts $CT_1||CT_2$ before generated from CP-ABE and output the corresponding signature $\sigma$. Finally, the patients produce the tuple $(CT_1, CT_2, \sigma)$ and upload it to the storage server in the local hospital. Receiving the tuple, the storage server executes the verify algorithm of ABS. If the signature passes the validation, the server stores the tuple. Otherwise, the server rejects it. Because of the utilization of ABS, the new PHRs sharing scheme achieves the function of anonymous authentication successfully.

In the new scheme, not all the physicians working in the same hospital as the patient P can recognize P's actual identity, except the ones whose attributes satisfy $T_2$. Therefore, the collusion attack described in Section 3 does not exist in our scheme. Furthermore, since the PHRs received by indirectly authorized physicians derive from the patients directly instead of the directly authorized physicians, our scheme also does not suffer from the forgery attack as PSMPA.

In this section, we provide a possible solution to avoid the above two attacks. To ensure that each patient has full control over his identity information and personal health information, we leverage CP-ABE proposed in [1] and ABS proposed in [10] as the encryption primitive and the signature primitive in our possible solution. Our scheme realizes the same three levels of security and privacy requirement as the PSMPA scheme. As shown in Figure 1, All the members are also classified into three categories: the directly authorized physicians such as Bob in the local healthcare provider, the indirectly authorized physicians such as Jack, Tom and Jim in the remote healthcare providers and the unauthorized persons such as Black. We generally describe the possible solution which is consisted of five phases and discuss its security in the following.

**Setup.** The algorithm takes $1^l$ as input, where $l$ is the security parameter. It outputs public parameters and $y$ as the master key for the central attribute authority. This algorithm is the same as the setup algorithm in the PSMPA scheme.

**Key Extract.** As the ABS and CP-ABE involved, both patients and physicians request their own attribute keys for an attribute set in this algorithm. If someone is qualified to be issued with $sk_D$ for some attributes, the attribute authority produces $sk_D$ for him.

**Encrypt-Sign.** Before encrypting the PHRs, the patients firstly divide the PHRs into patient's identity information $m_1$ and personal health information $m_2$. Secondly, they choose two different access tree $T_1$ and

$T_2$ as the corresponding access policy of the plaintexts $m_1$ and $m_2$. The set of leaf nodes in $T_2$ does not contain the attribute of the hospital where the physician works, while the access tree $T_1$ contains. The patient can define the root node of $T_1$ as an "AND" gate with two children: one is $T_2$ and the other is a leaf that is associated with the attribute of the hospital where the physician works. For example, if patient P is registered in hospital A, he can specify the attribute "hospital=A" as the leaf node of the root. Taking the two access tree and public parameters as input, the encryption algorithm encrypt $m_1$ and $m_2$ into $CT_1$ and $CT_2$ respectively.

Finally, to provide authenticity, the patients need to claim that they possess some specific attributes which the healthcare provider requires. Taking the corresponding attribute keys and public parameters as input, the signing algorithm signs the ciphertexts $CT_1||CT_2$ and outputs the signature $\sigma$. In this way, a tuple $(CT_1, CT_2, \sigma)$ can be constructed and uploaded to the storage server in the local hospital.

**Verify.** Once receiving the PHRs uploaded by someone, the storage server in hospital executes the verify algorithm of ABS and decides whether the signer possesses the attributes as they claimed in the signature $\sigma$. If the signature $\sigma$ passes the validation, the server stores the tuple. Otherwise, the server rejects it.

**Decrypt.** When the physicians issue a request to the server, it returns the corresponding tuple. Receiving the tuple, the directly authorized physicians working in local healthcare provider whose attributes satisfy $T_2$ decrypt the ciphertexts $(CT_1, CT_2)$ and get the plaintexts $(m_1, m_2)$ simultaneously through executing the decrypt algorithm of CP-ABE, while the indirectly authorized physicians working in remote healthcare provider whose attributes satisfy $T_2$ only can decrypt $CT_2$ to get $m_2$ using their attribute keys. The unauthorized persons whose attributes can not satisfy $T_2$ will obtain nothing.

In our scheme, we treat the CP-ABE proposed in [1] as the encryption primitive. For purpose of realizing collusion-resistance, Bethencourt *et al.* [1] embeds independently chosen secret shares into the ciphertext such that the attacks can not combine their attribute keys to satisfy the access tree. Thus, not all the physicians working in the local healthcare provider can recognize the patient's actual identity, except the ones whose attributes satisfy $T_2$. However, in the PSMPA scheme, the fact that all the directly authorized physicians working in the local healthcare provider can decrypt the patient's identity causes the collusion attack. Therefore, our new scheme can resist the collusion attack between the directly authorized physicians and the indirectly authorized physicians.

Furthermore, since the PHRs received by indirectly authorized physicians derive from the patients directly instead of the directly authorized physicians and we do not hide the patient's actual identity by randomizing it, our scheme does not suffer from the forgery attack as PSMPA. In summary, the PHRs sharing scheme proposed above can be regarded as a possible solution for the PSMPA scheme.

# 5 Conclusions

In this paper, we discuss two important flaws in the patient self-controllable multi-level privacy-preserving cooperative authentication scheme. Exploiting collusion attack and forgery attack, we specify that the scheme doesn't possess the feature of identity privacy as they have claimed and there exists a flawed design during the transcript simulation. In the end, we establish an improved PHRs sharing scheme as a remedy solution through incorporating CP-ABE and ABS. A concrete description of the proposed scheme will be given in the future work.

# Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61003230, Grant 61370026, and Grant 61202445, in part by the Fundamental Research Funds for the Central Universities under Grant ZYGX2013J073, and in part by the Applied Basic Research Program of Sichuan Province under Grant 2014JY0041.

# References

[1] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *IEEE Symposium on Security and Privacy (SP'07)*, pp. 321–334, Berkeley, USA, 2007.

[2] M. Chase and S. S. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS'09)*, pp. 121–130, Chicago, USA, 2009.

[3] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS'06)*, pp. 89–98, Alexandria, USA, 2006.

[4] L. Guo, C. Zhang, J. Sun, and Y. Fang, "Paas: A privacy-preserving attribute-based authentication system for ehealth networks," in *IEEE 32nd International Conference on Distributed Computing Systems (ICDCS'12)*, pp. 224–233, Macau, China, 2012.

[5] C. C. Lee, P. S. Chung, and M. S. Hwang, "A survey on attribute-based encryption schemes of access control in cloud environments.," *International Journal of Network Security*, vol. 15, no. 4, pp. 231–240, 2013.

[6] M. Li, S. Yu, K. Ren, and W. Lou, "Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings," in *6th International ICST Conference on Security and Privacy in Communication Networks*, pp. 89–106, Singapore, 2010.

[7] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 131–143, 2013.

[8] Q. Li, H. Xiong, F. Zhang, and S. Zeng, "An expressive decentralizing kp-abe scheme with constant-size ciphertext.," *International Journal of Network Security*, vol. 15, no. 3, pp. 161–170, 2013.

[9] X. Liu, J. Ma, J. Xiong, and G. Liu, "Ciphertext-policy hierarchical attribute-based encryption for fine-grained access control of encryption data.," *International Journal of Network Security*, vol. 16, no. 4, pp. 351–357, 2014.

[10] Y. S. Rao and R. Dutta, "Efficient attribute-based signature and signcryption realizing expressive access structures," *International Journal of Information Security*, pp. 1–29, 2015.

[11] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *The 29th Conference on Computer Communications (INFOCOM'10)*, pp. 1–9, San Diego, USA, 2010.

[12] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS'10)*, pp. 261–270, Beijing, China, 2010.

[13] J. Zhou, X. Lin, X. Dong, and Z. Cao, "Psmpa: Patient self-controllable and multi-level privacy-preserving cooperative authentication in distributed m-healthcare cloud computing system," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 6, pp. 1693–1703, 2014.

**Yang Zhao** is a Ph.D. Candidate at the School of Computer Science and Engineering, University of Electronic Science and Technology of China. His research interests are in the area of networking security and e-commerce protocol.

**Feng Yue** received his B.S. degree in the School of International Education, Henan University of Science and Technology (HAUST) in 2008. He is currently pursing his M.S. degree in the School of Computer Science and Engineering, (UESTC). His research interests include: cryptography and information security.

**Songyang Wu** is an associate professor at The Third Research Institute of Ministry of Public Security, China. Vice director. He received his Ph.D. Degree in computer Science from TongJi University, China in 2011. His current research interests are in information security, cloud computing and digital forensics.

**Hu Xiong** is an associate professor at University of Electronic Science and Technology of China (UESTC). He received his Ph.D degree from UESTC in 2009. His research interests include: cryptography and network security.

**Zhiguang Qin** is the dean and professor in the School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC). He received his PH.D. degree from UESTC in 1996. His research interests include: information security and computer network.