Research Article

Achieving Secure and Efficient Data Access Control for Cloud-Integrated Body Sensor Networks

Zhitao Guan, Tingting Yang, and Xiaojiang Du

1 School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China
2 Department of Computer and Information Science, Temple University, Philadelphia, PA 19122, USA

Correspondence should be addressed to Zhitao Guan; guanzhitao@126.com

Received 29 May 2015; Revised 22 July 2015; Accepted 27 July 2015

1. Introduction

Body sensor network (BSN) emerges recently with rapid development of wearable sensors, implantable sensors, and short range wireless communication, which make pervasive healthcare monitoring and management become increasingly popular [1, 2]. By the body sensor network, health-related data of the patient can be collected and transferred to the healthcare staff in real time, so the patient’s state of health can be under monitoring and precautions can be taken if something bad happened.

In order to enhance the scalability of the body sensor network, some work focuses on combining cloud computing and body sensor network together. As shown in Figure 1, with the support of mobile cloud computing, cloud-integrated body sensor network (C-BSN) can be constructed [3]. In C-BSN, massive local body sensor networks are integrated together and mass data are collected and stored in cloud servers; healthcare staffs will continually monitor their patients’ status and exchange views when it is difficult to make diagnosis; researchers can make data analysis to get some useful results such as regularity of disease development; government agencies also can take measures on disease prevention and control based on data analysis.

However, there are still several problems and challenges in C-BSN [3, 4]. For example, data security and data privacy must be concerned since patient-related data is private and sensitive. In this paper, we propose a secure data access control mechanism MC-ABE (Mask-Certificate Attribute-Based Encryption) for cloud-integrated body sensor networks. A specific signature is designed to mask the plaintext, and then the masked data can be securely outsourced to cloud servers. An authorization certificate composed of the signature and related privilege items is constructed which is used to grant privileges to data receivers. To ensure security, a unique value is chosen to mask the certificate for each data receiver. Thus, the certificate is unique for each user and user revocation can be easily completed by removing the mask value. The analysis shows that proposed scheme can meet the security requirement of C-BSN, and it also has less computation cost and storage cost compared with other popular models.
person, Bob does not want anyone else to know about his health data. However, his information may be leaked in many ways: the cloud operator/administrator may access his data; malicious user may intrude into the cloud server to steal user data; unauthorized DR may exceed to access others’ data. In summary, there are three key problems which need to be solved to ensure the users’ data security and data privacy in C-BSN. Firstly, the cloud is semitrusted; that is, although we outsource the data to the cloud, we still need to prevent cloud operators from accessing the data content; secondly, we must take measures to keep malicious users out of C-BSN system; lastly, it is also important to study how to avoid the unauthorized access of other users.

In this paper, we propose a novel secure access control mechanism MC-ABE to tackle the aforementioned problems. And main contributions of this paper can be summarized as follows:

(i) We construct one specific signature to CP-ABE to mask the plaintext and then realize securely encryption/decryption outsourcing.

(ii) We construct the unique authentication certificate for each visitor, which makes the system achieve more effective control on malicious visitors; in particular, it also leads to a low cost for user revocation.

(iii) We introduce the third-party trust authority to manage above-mentioned signatures and certificates, which can guarantee data security even if the cloud server is semitrusted.

(iv) In C-BSN, processing data in time is quite necessary. Our proposed scheme can meet such requirement. From the section of performance evaluation, our scheme takes less time than other compared methods to do data collecting, data transmission, and data acquisition.

The rest of this paper is organized as follows. Section 2 introduces the related work. Then, in Section 3, some preliminaries are given. Our scheme is stated in Section 4. In Section 5, security analysis is given. In Section 6, the performance of our scheme is evaluated. The paper is concluded in Section 7.

2. Related Work

Recently, various techniques have been proposed to address the problems of data security and data privacy in C-BSN. In [5], Sahai and Waters proposed the Attribute-Based Encryption (ABE) to realize access control on encrypted data. In ABE, the ciphertext’s encryption policy is associated with a set to fail attributes, and the data owner can be foreseen later data is encrypted. One year later, Goyal et al. proposed a new type of ABE, Key-Policy Attribute-Based Encryption (KP-ABE) [6]. In KP-ABE, the ciphertext’s encryption policy is also associated with a set of attributes, but the attributes are organized into a tree structure (named access tree). The benefit of this approach is that more flexible access control strategy can be got and fine-grained access control can be realized. However, data owner was short of entire control over the encryption policy; that is, he cannot decide who can access the data and who cannot. To solve this problem, Bethencourt et al. proposed CP-ABE (Ciphertext-Policy Attribute-Based Encryption) [7], in which data owner constructed the access tree together with visitors’ identity information. The user can decrypt the ciphertext if and only if attributes in his private key match the access tree. So, in CP-ABE, data owner can configure more flexible access policy. In [8], Yu et al. tried to achieve secure, scalable, and fine-grained access control in cloud environment. Their proposed scheme is based on KP-ABE and combines with the other two techniques, proxy reencryption and lazy reencryption. It is proved that the
The proposed scheme can meet the security requirement in cloud quite well. Similarly, Wang et al. proposed an access control scheme based on CP-ABE, which is also secure and efficient in cloud environment [9].

In [10], Ahmad et al. proposed a multitoken authorization strategy to remedy the weaknesses of the authorization architecture in mobile cloud. It reduces the probability of unauthorized access to the cloud data and service when malicious activity happened; for example, IdM (Identity Management Systems) are compromised, network links are eavesdropped, or even communication tokens are stolen. In [11], Yadav and Dave presented an access model based on CP-ABE which could provide the remote integrity check by the way of augmenting secure data storage operations. To reduce computation overhead and achieve secure encryption/decryption outsourcing, the access tree is divided into two parts: one part is encrypted by the data owner and the other part is encrypted by the cloud sever. So a portion of computation overhead was transferred from data owner to cloud sever. The similar method is also adopted in the work of Zhou and Huang [12]. In addition to the access tree division, Zhou and Huang also propose an efficient data management model to balance communication and storage overhead to reduce the cost of data management operations. In [13], Li et al. presented a low complexity multiauthority attribute-based encryption scheme for mobile cloud computing which uses masked shared-decryption-keys to ensure the security of decryption outsourcing and adopts multiauthorities for authorization to enhance security assurance. The above schemes are based on CP-ABE, in which complex bilinear map calculation is performed. In [14], Yao et al. proposed a novel access control mechanism, in which data operation privileges are granted based on authorization certificates. The advantage of such mechanism is that the computation cost can be decreased remarkably, since there is no bilinear map calculation. And the disadvantage is that lots of operations need to be handled by data owner, such as privilege designation, and then it demands that the data owner must know all information about the visitors. In [15], the authors considered the problem of patient self-controlled access privilege to highly sensitive Personal Health Information. They proposed a Secure Patient-Centric Access Control scheme which allows data requesters to have different access privileges based on their roles and then assigns different attribute sets to them. However, they took the cloud server as trusted, and their algorithm is not suitable for large scale C-BSN.

**3. Preliminaries**

**3.1. Notations.** The notations used in MC-ABE are listed as follows.

**Notations in MC-ABE.** Consider the following:

- **DO:** data owner,
- **DR:** data requester/receiver,
- **ESP:** encryption service provider,
- **DSP:** decryption service provider,
- **SSP:** storage service provider,
- **TA:** trust authority,
- **SetS:** setup server,
- **PK:** public key,
- **MK:** master key,
- **SK:** secret key,
- **DO:** data owner,
- **DR:** data requester/receiver,
- **ESP:** encryption service provider,
- **DSP:** decryption service provider,
- **SSP:** storage service provider,
- **TA:** trust authority,
- **SetS:** setup server,
- **PK:** public key,
- **MK:** master key,
- **SK:** secret key,
- **M:** plaintext,
- **CT:** ciphertext,
- **T:** access tree,
- **MM:** masked plaintext,
- **Cert:** authorization certificate,
- **MValue:** mask value,
- **MCert:** masked certificate.

**DO** and **DR** are cloud users. **ESP** is cloud server that can help **DO** do data encryption. **SSP** is cloud storage server. **DSP** is the server that is responsible for data decryption. **TA** is the third-party trust authority. **SetS** is the setup server whose responsibility is to generate **PK** and **MK**.

**PK** and **MK** are parameters that are used for data encryption/decryption. **SK** is held by **DR** which is used to decrypt ciphertext, which is generated using **PK** and **MK**. The data is plaintext before encryption, denoted as **M**, and **CT** is the ciphertext of **M**. **T** is the access policy (access tree). **MM** is the masked plaintext; in MC-ABE, the plaintext will be masked to **MM** by a signature before being encrypted to achieve “double protection.” **Cert** is the authorization certificate (see Section 4.2.1 for details). **MValue** is used to mask **Cert** to generate **MCert** (see Section 4.2.2 for details).

**3.2. Basics**

**3.2.1. Bilinear Pairing.** Let **G**₁ and **G**₂ be two multiplicative cyclic groups of prime order **p**. Let **g** be a generator of **G**₁ and let **e** be a bilinear map, **e**: **G**₁ × **G**₁ → **G**₂. For **a, b ∈ Z**ₚ, the bilinear map **e** has the following properties [3, 10]:

1. **Bilinearity:** for all **u, v ∈ G**₁, we have **e**(**u**<sup>a</sup>, **v**<sup>b</sup>) = **e**(**u**, **v**)<sup>ab</sup>.
2. **Nondegeneracy:** **e**(**g**, **g**) ≠ 1.
3. **Being symmetric:** **e**(**g**<sup>a</sup>, **g**<sup>b</sup>) = **e**(**g**, **g**<sup>b</sup>) = **e**(**g**<sup>b</sup>, **g**<sup>a</sup>).

**3.2.2. Discrete Logarithm (DL) Problem**

**Definition 1** (discrete logarithm (DL) problem). Let **G** be a multiplicative cyclic group of prime order **p** and let **g** be its generator, for all **a ∈ Z**ₚ, given **g, g**<sup>a</sup> as input, output **a**.

The DL assumption holds in **G** if it is computationally infeasible to solve the DL problem in **G** [17].
3.3. Ciphertext-Policy Attribute-Based Encryption (CP-ABE)

3.3.1. Access Structure. Let \( P = \{P_1, P_2, \ldots, P_n\} \) be the universal set [18]. Each element in \( P \) is an attribute, that is, the descriptive identification information of a visitor [19]. An access structure is a collection (resp., monotone collection) \( T \) of nonempty subsets of \( P \). For example, \( P = \{\text{Beijing, Shanghai, No. 1 Middle School, No. 2 Middle School, student, teacher, administrator}\} \). Visitor 1 has the attributes set \( A = \{\text{Beijing, No. 1 Middle School, student}\} \) [20].

The access structure in CP-ABE is the tree structure, which is named access tree [2]. For the access tree \( T \), the leaf nodes are associated with descriptive attributes; each interior node is a relation function, such as AND (\( n \) of \( n \)), OR (1 of \( n \)), and \( n \) of \( m \) (\( m > n \)).

Each DR has a set of attributes, which are associated with DR’s SK. If DR’s attributes set satisfies the access tree, the encrypted data can be decrypted by DR’s SK.

3.3.2. Working Process. In CP-ABE, the plaintext is encrypted with a symmetric key, and then the key is shared in the access tree. In the process of decryption, if DR’s SK satisfies the access tree, then DR gets the shared secret and the data can be recovered.

3.4. Assumptions. In this work, we make the following assumptions.

Assumption 1 (service providers (ESP, DSP, and SSP) are semitrusted). That is, they will follow our proposed protocol in general but try to find out as much secret information as possible. And the information may be accessed illegally by internal malicious employees or external attackers. In particular, although ESP and DSP undertake most of the computing cost, they do not have enough information to deduce the plaintext.

Assumption 2 (SetS and TA are trusted). On no conditions will they leak information about data and related keys.

In order to deduce more information about encrypted data, service providers might combine their information to perform collusion attack. In our scheme, collusions between service providers are taken into consideration.

4. MC-ABE

4.1. Overview. Our proposed scheme MC-ABE is shown in Figure 2. Seven algorithms are included in MC-ABE: Setup, Encrypt\(_{DO}\), Encrypt\(_{ESP}\), KeyGen, CerGen, Decrypt\(_{DSP}\), and Decrypt\(_{DR}\).

For data outsourcing, DO encrypts \( M \) with algorithm Encrypt\(_{DO}\), in which signature is used to mask \( M \). Then ESP encrypts \( T \) with algorithm Encrypt\(_{ESP}\) to finish the encryption. The encrypted data is stored in SSP.

For data access, when DR requests data from SSP, the request is sent to TA after verification. TA chooses a unique value to the mask certificate for DR. Then, using the attributes set of DR, TA computes SK with algorithm KeyGen. After that, SK is sent to DSP and the certificate is sent to DR. At the same time, SSP sends the CT to DSP. With SK and CT, DSP can do decryption and get \( M \) that is masked by signature. Once DR receives the certificate, he decrypts the masked certificate with his unique value (TA sends the unique value to this DR when the first authorized request occurred. It will be used in the following requests until this DR is revoked) to get the certificate. Using the certificate, DR can decrypt the masked \( M \) with signatures in the certificate.

In addition, if a DR is revoked, TA will mark the DR as “revoked” and this DR’s unique mask value will be invalid. No certificate will be granted to this DR any more.

4.2. Two Important Notions

4.2.1. Authorization Certificate (Cert). The authorization certificate is introduced in MC-ABE to grant data privileges for DR. As shown in Structure of Authorization Certificate, it includes five items that are privilege related information. DO provides the certificate related information to TA, and then TA constructs the unique authorization certificate for each authorized DR.

Structure of Authorization Certificate

- File ID list \( (f_1, f_2 \ldots) \)
- Valid Period (From the start time to the end time)
- Signature \( (\{\text{sign}_{f_1}, \{\text{sign}_{f_2}\} \ldots) \)
- Privilege \( (\{p_{f_1}, \{p_{f_2}\} \ldots) \)
- PK, MK
Figure 3: Algorithms’ implementation in MC-ABE.

4.2.2. Mask Value (MValue). To achieve fine-grained access control over DR, the mask value is introduced in MC-ABE. The mask value is maintained by TA. For each DR, TA sets a unique mask value for him. The mask value is used to blind the authorization certificate before the certificate is sent to DR. Thus, each DR receives its own unique blinded certificate since the mask value is unique. In the following, the process is described in detail.

After TA receives a data access request, it checks DRID firstly. If the requester is a new user, TA generates a random number \( t_{DRID} \in \mathbb{Z}_p \) and inserts it into the mask value table.

Otherwise, if this DRID already exists in mask value table and the item of revocation is “N” (initial value of this item is “N.” Only at the time when the DRID is revoked will this item be set as “Y”), TA invokes algorithm CerGen to compute the masked certificate (see Table 1).

Algorithm (CerGen(\( t_{DRID}, PK \)) \( \rightarrow \) MCert). Construct a certificate Cert as Structure of Authorization Certificate shows. MCert is the masked Cert.

Then, compute as follows:

\[
\begin{align*}
\text{MValue} &= g^{t_{DRID}} \\
\text{MCert} &= \text{Cert} \cdot e(g^\beta, g^\alpha_{\text{DRID}}) = \text{Cert} \cdot e(g^\alpha, g^\theta_{\text{DRID}}). \tag{1}
\end{align*}
\]

If DR is a new user, MValue and MCert will be sent to him. Otherwise, send MCert to the DR.

<table>
<thead>
<tr>
<th>DRID</th>
<th>Mask value</th>
<th>Revocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DR1</td>
<td>MValue_{DR1}</td>
<td>N</td>
</tr>
<tr>
<td>DR2</td>
<td>MValue_{DR2}</td>
<td>Y</td>
</tr>
<tr>
<td>DR3</td>
<td>MValue_{DR3}</td>
<td>N</td>
</tr>
</tbody>
</table>

DRID: ID of DR.
Mask value: unique mask value for each DR.
Revocation: revocation mark. “Y” means this DR is revoked. “N” means this DR is authorized.

4.3. Scheme Description. The whole process of MC-ABE is shown in Figure 3. In this section, we describe each step in detail.

4.3.1. Data Outsourcing. In C-BSN, DO usually uses mobile devices that lack computing power and storage space. To reduce the encryption overhead of DO, the encryption process is divided into two parts: Encrypt_{DO} and Encrypt_{ESP}. Encrypt_{DO} is the encryption algorithm implemented by DO and Encrypt_{ESP} is carried out by ESP. Since ESP is semitrusted, we introduce the signature in Encrypt_{DO} to mask M. In general, there are three steps for data outsourcing.

Firstly, SetS generates PK and MK.

Algorithm 2 (Setup \( \rightarrow \) PK, MK). SetS performs the algorithm. Let \( G_0 \) be a multiplicative cyclic group of prime order \( p \) and let \( g \) be its generator, and four random numbers \( \alpha, \beta, \epsilon, \theta \in \mathbb{Z}_p \) (further details in [7]). Consider

\[
\begin{align*}
PK &= (G_0, g, h = g^\beta, e(g, g)^\alpha, g^\epsilon, g^\theta) \tag{2} \\
MK &= (\beta, g^\alpha).
\end{align*}
\]

Secondly, DO performs the first step of data encryption.
Algorithm 3 (Encrypt\textsubscript{DO}(PK, M, K) → MM). DO implements the algorithm. PK is got from SetS; M is DO’s plaintext; MM is masked M; K is the set of operation privileges, and k is one of the elements in K.

For k ∈ K, we choose a random number ν\textsubscript{k} ∈ Z\textsubscript{p} and then compute the signature:

\[
\text{signature}_{k} = e\left(g^{\nu_{k}}, g^{\nu_{k}}\right) = e\left(g, g\right)^{\nu_{k}}. \tag{3}
\]

For simplicity, let \nu denote the set of ν\textsubscript{k} : \nu = { ν\textsubscript{k} | k ∈ K}; signature denotes the set of signature\textsubscript{k} : signature = {signature\textsubscript{k} | k ∈ K}.

Choose a random number s ∈ Z\textsubscript{p}; then

\[
\text{MM} = \overline{C} = M \cdot e\left(g, g\right)^{as} \quad \text{signature} \tag{4}
\]

Lastly, ESP performs the last step of data encryption.

Algorithm 4 (Encrypt\textsubscript{ESP}(PK, s, T, MM) [7, 11] → CT). ESP implements the algorithm. The access tree T is encrypted from the root node to leaf nodes. For each node x in T, choose a polynomial q\textsubscript{x}.

For node x, consider the following:

\( k_{x} \): it denotes the threshold value of x.
\( d_{x} \): it denotes the degree of q\textsubscript{x} and \( d_{x} = k_{x} - 1 \).
parent\textsubscript{x}: a function returns the parent node of x.
num\textsubscript{x}: it is the number of child nodes of x. For a child node y, y is uniquely identified by an index number index\textsubscript{y}, and \( 1 ≤ \text{index}(y) ≤ \text{num}_{x} \). Consider

\[
q_{x}(0) = q_{\text{parent}(x)}(\text{index}(x)). \tag{5}
\]

For root node R, \( q_{R}(0) = s \). Choose \( d_{R} \) other points randomly to completely define q\textsubscript{R}. For any other node x in T, let \( q_{x}(0) = q_{\text{parent}(x)}(\text{index}(x)) \), and choose \( d_{x} \) other points randomly to completely define q\textsubscript{x}.

Y is the set of leaf nodes in T. Compute as follows:

\[
C = h^{t} \quad \forall y \in Y : C_{y} = g^{q_{y}(0)}, C'_{y} = H\left(\text{att}(y)\right)^{q_{y}(0)}. \tag{6}
\]

Then,

\[
\text{CT} = \left\{ \left[T, \overline{C} = M \cdot e\left(g, g\right)^{as} \cdot e\left(g, g\right)^{ev}, C = h^{t}, \forall y\right] \right\}, \tag{7}
\]

\( e \cdot Y : C_{y} = g^{q_{y}(0)}, C'_{y} = H\left(\text{att}(y)\right)^{q_{y}(0)} \}
\]

CT is stored in SSP. Detailed communication information is shown in Figure 4.

4.3.2. Data Request. When a DR requests data from SSP, TA generates SK and a certificate for DR. Most of decryption cost is taken by DSP but DSP cannot get M. Based on the effort of DSP, DR finishes the last step of decryption and gets M. Similar to data outsourcing, there are also three steps for data outsourcing.

Firstly, TA generates SK for DR.

Algorithm 5 (KeyGen(MK, S) → SK). S is the attributes set of DR.

We generate a random number \( r \in Z_{p} \) and then generate the random number \( r_{j} \in Z_{p} \) for each \( j \in S \). Compute as follows:

\[
\text{SK} = \left(D = g^{(a + r)/\beta}, \forall j \in S : D_{j} = g^{r_{j} \cdot H\left(j\right)^{t}}, D'_{j} = g^{r_{j}}\right). \tag{8}
\]

Then, TA sends SK to DSP.

Secondly, DSP performs the first step of data decryption: decrypt the access tree in CT to get MM.

Algorithm 6 (Decrypt\textsubscript{ESP}(SK, CT) → MM). When x is a leaf node, let \( i = \text{att}(x) \). Function \text{att}(x) denotes the attribute associated with the leaf node x in the tree.

If \( i \in S \),

\[
\text{DecryptNodeL}(\text{CT}, \text{SK}, x) = \frac{e(D_{i}, C_{x})}{e(D'_{i}, C'_{x})} = e\left(g^{r_{i} \cdot H\left(i\right)^{t}}, g^{q_{i}(0)}\right) \tag{9}
\]

Otherwise,

\[
\text{DecryptNodeL}(\text{CT}, \text{SK}, x) = \perp. \tag{10}
\]

When x is an interior node, call the algorithm DecryptNodeNL(CT, SK, x).

For all of the children z of node x, call DecryptNodeL(CT, SK, z), and the output is \( F_{z} \). Let \( S_{x} \) be \( k_{x} \) (the threshold value of interior node) random set and let \( F_{z} \neq \perp \). If no such set exists, the function cannot be satisfied, so return \( \perp \).
Choose $v'_k \in \mathbb{Z}_p$ randomly,

$$r_k = v'_k/v_k$$

Signature $\rightarrow$ signature$'$

Algorithm 7 (Decrypt$_{DR}$($M \cdot$ signature, MCert) $\rightarrow$ $M$). DR retrieves Cert to get related signatures:

$$\frac{MCert}{e(g^{\theta_{\text{DRID}}}, g^{\theta_{\text{DRID}}})} = \frac{\text{Cert} \cdot \left(e(g, g)^{\theta_{\text{DRID}}}ight)^{\theta_{\text{DRID}}}}{e(g, g)^{\theta_{\text{DRID}}}} = \text{Cert.}$$

(14)

Then, DR gets $M$ with the signature:

$$M \cdot \text{signature} = M.$$  
(15)

4.3.3. User Revocation. An invalid DR is a DR who is thought to be malicious or whose certificate is expired. The invalid DR should be revoked from the authorized access list. In MC-ABE, we can remove the MValue record in Table 1 to revoke DR. Firstly, TA modifies the revoked DR's "Revocation" item from "N" to "Y" in mask value table. Secondly, current signature must be updated to a new one (signature updating is shown in Figure 5). After these two steps, the invalid DR is revoked. When he requests new data, he will be taken as new comer (the signature is updated, and he does not have the new one), and TA will refuse his request since he is marked as revoked. For valid DR, they will get the new signature and access the system as usual.

5. Security Analysis

5.1. Encryption and Decryption Outsource. In CP-ABE, both data encryption and data decryption are only done by the cloud users. Meanwhile, in MC-ABE, data encryption is done by DO and the cloud server collaboratively, and data decryption is undertaken by DR and the cloud server together. $M$ is masked by DO before it is sent to ESP. DO and authorized DR can get $M$. ESP and DSP can get MM (Masked $M$), but they cannot deduce $M$ from MM.
Theorem 8. The security in encryption and decryption in MC-ABE is not weaker than that of CP-ABE.

Proof. In algorithm Encrypt\(_{ESP}\), ESP encrypts the access tree \(T\) with the parameters \(s, T\), and \(MM\). Consider

\[
\overline{C} = M \cdot e(g, g)^{αs} \cdot \text{signature} = M \cdot e(g, g)^{αs} \cdot e(g, g)^{εs}.
\]

Using PK and \(s\), ESP can get \(e(g, g)^{αs}\); what ESP gets is \(M \cdot e(g, g)^{εs}\).

The encrypted data in CP-ABE is \(\overline{C} = M \cdot e(g, g)^{αs}\); both of \(α\) and \(s\) are random; let \(z = α \cdot s\); \(z\) is also random; then \(\overline{C} = Me(g, g)^{αz}\) is equal to \(Me(g, g)^{εz}\). According to security proof in [7], the structure of \(\overline{C} = M \cdot e(g, g)^{αs}\) is secure to prevent the adversary from deducing \(M\). Thus, \(Me(g, g)^{εz}\) in our scheme is secure. That is to say, ESP cannot deduct \(M\) with \(Me(g, g)^{εz}\), and encryption outsourcing is secure in MC-ABE.

For DSP, it can decrypt CT using SK and get the masked \(M = M \cdot \text{signature}\). The information DSP gets is the same as ESP. So, in MC-ABE, data decryption outsourcing is also secure since it is similar to data encryption outsourcing.

5.2. Certificate. From the above statement, the signature is vitally important to the security of our scheme. Since the signature is an item of the certificate, the security of the signature relies on the certificate. Each DR has his unique masked certificate; DR can retrieve his certificate only by his own MValue. In the following, we prove that malicious DR cannot get MCert without the right MValue.

Theorem 9. MCert cannot be decrypted without the right MValue.

\(DR_1\) has \(MCert_1 = Cert_1 \cdot \text{MValue}_1 = Cert_1 e(g, g)^{αt_{DR_1}}\); 
\(DR_2\) wanted to retrieve \(Cert_1\) without \(e(g, g)^{αt_{DR_2}}\).

Proof. \(DR_1\) forged \(\text{MValue}_1' = e(g, g)^{αt_{DR_1}}\), to get \(Cert_1\):

\[
\text{Cert}_1 = \frac{MCert_1}{\text{MValue}_1'} = \text{Cert}_1 \cdot \frac{\text{MValue}_1}{\text{MValue}_1'}
\]

(17)

In other words, if the forged \(\text{MValue}_2'\) is right, we must have \(t_{DR_1} = t_{DR_1}'\) to solve the DL problem. The DL problem is computationally infeasible; thus, \(\text{MValue}\) is difficult to be forged and MCert cannot be decrypted without the right MValue.

5.3. Collusion. Service providers might collude with each other to combine their information to deduce \(M\). In the above statement, ESP and DSP hold similar information to retrieve \(M\). If ESP colluded with DSP, they could get \(M \cdot \text{signature}\). We have given the security proof of \(M \cdot \text{signature}\) in Theorem 8. Thus, MC-ABE is quite qualified for anticollusion.

SSP is a semitrusted server, which stores CT. If SSP colluded with ESP and DSP, it provides no useful information to deduce \(M\). So, MC-ABE can defend against collusion among SSP, ESP, and DSP.

5.4. Revocation. If a DR is revealed to be malicious, he will be revoked from the authorized user list. We update the signature encrypted in CT; after that, as shown in the following, the revoked DR cannot get authorized data any more:

Revoked signature held by DR: signature = \(e(g, g)^{εs}\).

Updated signature: signature' = \(e(g, g)^{εs}\).

Masked \(M' = M \cdot \text{signature}' = Me(g, g)^{εs}\).

Masked \(M'/\text{signature}' = Me(g, g)^{εs} / e(g, g)^{εs} = Me(g, g)^{εs} / e(g, g)^{εs} = Me(g, g)^{1/(εs - εs)}\).

It is the same with the proof of Theorem 9. MC-ABE is secure in revocation.

6. Performance Evaluation

In this section, we numerically analyze the communication and computation cost of MC-ABE. We also give the simulation results in detail.

6.1. Numerical Analysis

6.1.1. Computation Cost

Setup. The setup procedure includes defining multiplicative cyclic group and generating PK and MK that will be used in encryption and key generation. There are four exponentiation operations and one pairing operation in setup procedure. Time complexity of the procedure is \(O(1)\). The computation cost has nothing to do with the number of attributes.

\(Encrypt_{ESP}\). In this procedure, DO is responsible for generating signature and masking \(M\). Two operations are included in signature computation, which are random number generation and bilinear map computation. And operations performed in mask computation include random number generation and three multiplication operations. Thus, it needs to do two exponentiation operations, two multiplication operations, and one pairing operation for each file. But if more privileges are permitted at the same time, more signatures will be computed. For each privilege, computation cost is fixed, so the total cost is proportional to the number of privileges.

\(Encrypt_{SSP}\). ESP encrypts the access tree in this procedure. The computation cost is proportional to the number of attributes in the tree. If the universal attributes set in \(T\) is \(I\) (\(|I|\) denotes the total number of attributes in set \(I\)), for each element in \(I\), it needs two exponentiation operations; totally, the computation complexity is \(O(|I|)\).

KeyGen. This procedure is carried out to generate SK for DR. Computation cost is proportional to the number of...
attributes in SK. For each attribute, two pairing operations and one multiplication operation are needed. If the universal attributes set is $S$ ($|S|$ is the total number of attributes in set, $|S| \leq |I|$), the time complexity of SK computation is $O(|S|)$.

*CerGen.* In this procedure, we construct the certificate and mask it. Items in certificate are denoted by DO. TA needs to do one exponentiation operation, one multiplication operation, and one pairing operation. Computation cost is fixed; the computation complexity is $O(1)$.

*Decrypt$_{DS}$.* In this procedure, DSP decrypts the ciphertext. The main overhead is incurred at the decryption of every attribute. The cost is proportional to the number of attributes in the access tree. Thus, the complexity is $O(|I|)$.

*Decrypt$_{DR}$.* In this procedure, DR gets $M$ from the masked $M$ by a divide operation. Thus, the complexity is $O(1)$.

### 6.1.2. Storage Cost

Compared to CP-ABE, more storage cost is incurred in MC-ABE because the certificate and the unique value are introduced. As shown in Table 2, the items in certificates are related to data access privileges, so the storage space of the certificate is proportional to the number of the documents (data). For each DR, one record is kept in mask value table (Table 1). Thus, the storage space for mask value table is proportional to the number of DR. Since the items in mask value table are quite simple, the total storage cost is not heavy.

### 6.2. Simulation Results

To evaluate the performance of MC-ABE, we develop simulation codes based on CP-ABE toolkit [21]. We make a comparison between MC-ABE and other two popular models (CP-ABE and PP-CP-ABE [11]) in four aspects: computation cost for data encryption, computation cost for key generation, computation cost for data decryption, and computation cost for user revocation.

1. **Computation Cost for Data Encryption.** Most of the computation cost in encryption is incurred for the encryption of the access tree, which is proportional to the number of the leaf nodes. In CP-ABE, data encryption is done by DO. In PP-CP-ABE, data encryption/decryption is outsourced to service providers; the access tree was divided into two parts: one part is encrypted by DO and the other part is encrypted by ESP. In MC-ABE, the access tree is encrypted by ESP. In Figure 6(a), the computation cost of three different schemes is compared. x-axis indicates the number of leaf nodes in $T$ (the access tree), and y-axis indicates time to encrypt $M$ (computation cost). For $x$, ten values are selected evenly (10, 20, ..., 100). For each $x$ value, we run simulation codes 10 times and take the average value of the results as the final result. It is shown that MC-ABE has better performance than the other two ones. In PP-CP-ABE, the number of leaf nodes in DO's subtree will change with different tree division. So, for simplicity, we set the number of DO's subtrees to be half of the number of the whole leaves. As shown in Figure 6(b), we also show confidence interval to assess the results in Figure 6(a) (only results about DO's computation cost in MC-ABE are given, since the results in PP-CP-ABE and CP ABE are consistent with MC-ABE). In Figure 6(b), it is shown that all average results lie in the confidence interval.

2. **Computation Cost for Key Generation.** Same with simulation about data encryption, we also take the average value of key generation cost as the final result. As shown in Figure 6(c), the average value is very close to lower bound and upper bound of the confidence interval, so we also list source data of the simulation results in Table 3. It shows that all average results lie in the confidence interval, so the simulation result is confident. From the results, we can get that the computation cost will grow with the number of attributes in private key. The algorithm KeyGen is implemented by TA, so there is no cost for DO.

3. **Computation Cost for Data Decryption.** In MC-ABE, most of the computation cost has been shifted to DSP, so the computation cost of DR is constant. The comparison results are shown in Figure 6(d).

4. **Computation Cost for User Revocation.** In MC-ABE, user revocation simplified for the signature is introduced. When user revocation happens, the revoked DR's "Revocation" item in mask value table is set as "Y"; his new data request will not be responded to; his former signature encrypted in ciphertext will be also changed. It needs one multiplication operation and one exponentiation operation for the above operations. The simulation results are as shown in Figure 6(e).

### 7. Conclusion

The C-BSN is one promising technology that can change people's healthcare experiences greatly. However, how to keep
Figure 6: (a) DO’s computation cost for data encryption in CP-ABE, PP-CP-ABE, and MC-ABE. In PP-CP-ABE, part of encryption computation is transferred to cloud sever to reduce DO’s cost. In MC-ABE, more efforts are made to reduce computation cost undertaken by DO. (b) Computation cost of DO (the 95% confidence interval assuming random data with normal distribution is shown). (c) Computation cost of key generation (the 95% confidence interval assuming random data with normal distribution is shown). (d) Computation cost of DR in CP-ABE and MC-ABE. Similar to ESP in MC-ABE, DSP also undertakes most of the computation in decryption. The cost is proportional to attributes number in private key. (e) Computation cost for user revocation. With the authorization certificate in MC-ABE, revocation cost can be reduced obviously.
data security and data privacy in C-BSN is an important and challenging issue since the patients’ health-related data are quite sensitive. In this paper, we propose a novel encryption outsourcing scheme MC-ABE that meets the requirements of data security and data privacy in C-BSN. In MC-ABE, one specific signature is constructed to mask the plaintext; the unique authentication certificate for each visitor is constructed; the third-party trust authority to manage above-mentioned signatures and certificates is also introduced. By security analysis, we prove that MC-ABE can meet the security requirement of C-BSN. And, by performance evaluation, it shows that MC-ABE has less computation cost and storage cost compared with other popular models. In future work, we plan to explore the possibility of improving the scalability of MC-ABE.

Conflict of Interests
The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments
This work is partially supported by Natural Science Foundation of China under Grant no. 61402171, Central Government University Foundation under Grant no. JB2014075, and US Army Research Office under Grant no. WF911NF-14-1-0518.

References