Role-based Access Control for Body Area Networks Using Attribute-based Encryption in Cloud Storage

Ye Tian[1,2], Yanbin Peng[3], Gaimei Gao[1], Xinguang Peng[1]
(Corresponding author: Xinguang Peng)

College of Computer Science and Technology, Taiyuan University of Technology[1]
No.79,West Yingze Street, Taiyuan, Shanxi 030024, China
(Email: sxgrant@126.com)
Computer Center, Taiyuan Normal University[2]
No.319 DaXue Street, Yuci District Jinzhong, Shanxi 030619, China
Software Development Center, Agricultural Bank of China[3]
NO. 18, Lize Road, Jintang International Finance Building, Fengtai district, Beijing 100073, China
(Received Aug. 5, 2016; revised and accepted Jan. 15, 2017)

Abstract

In order to save storage space, the data collected from body area networks can be stored in a third party. However, this may bring security problems. The common method is encrypting data before outsourcing. In this paper, we design a role-based access control scheme (RACS) used in the cloud. Firstly, we classify the data which are collected from body area networks into different types, and use the ciphertext-policy attribute-based encryption to encrypt them. Secondly, we divide the ciphertext into two parts, one part is stored in cloud, and the other is in the owner. Different users own different attributes, therefore, they only can access the data when their attributes satisfy the corresponding access structure. The security of medical data is assured in this way. Thirdly, we also add the user revocation to prevent the vicious user from obtaining and modifying the data. Lastly, when the emergency happens, users can obtain the temporary key to access medical data, so as to cure the patients in the first time. We analyze the correctness, security, storage and computation overhead of the scheme. The results show that RACS can resist the ciphertext attack and superior to others in the storage space and computation overhead.

Keywords: Access Control; Attribute-based Encryption; Body Area Networks; Cloud Storage

1 Introduction

In recent years, body area networks can be used to monitor illnesses of patients. The sensors which are put in, on and around patient's body can monitor physiological activities of patients continually, for example, the temperature, breathing, arrhythmia and endoscope. This medical treatment is very convenient for the chronic diseases and disables. The communication in body area networks has three layers, intra-BAN communications, inter-BAN communications and beyond-BAN communications [10]. "Intra-BAN communications" refers to the communications between sensors or between sensors and personal devices; "Inter-BAN communications" is the communications between personal devices and one or more access points (APs); in "Beyond-BAN communications" the authorized persons (doctors or nurses) can access medical data through Internet, so they can diagnose the patients according to their states. Database is an important part in the third layer, where the patients' personal information and medical history are stored. These data can be outsourced to the third party, such as the cloud servers. According to the access trees which are defined by the patients and the attributes users owned, users can access special medical data. Cloud is used widely for its powerful storage and convenience. Users can outsource their sensitive data in the cloud [3, 11]. If these data are stored in cloud, most of the cost can be saved. However, the cloud is outside of the owners' control, so personal information and medical data will be exposed to the third party. One of the serious challenges is protecting the confidentiality of the data [9, 14].

For the high value of medical data, the third party is always the attacks targeting of many malicious actions. It is necessary to construct a novel data access control scheme. The method of attribute-based encryption(ABE) before outsourcing is a common method to control medical data. ABE is a one-to-many encryption. If and only if

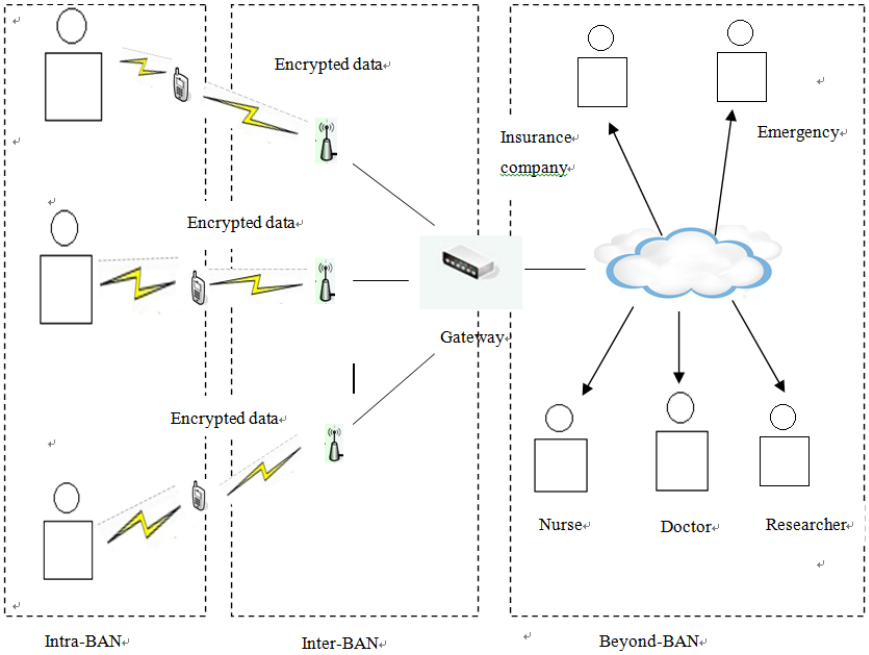

Figure 1: The three-layer architecture communication in body area networks

the attributes the users own satisfy the access tree, they can access the data. So the patients can decide the access policy. That is to say, different users can access different data. The authorized users include family numbers or friends, doctors, pharmacist and researchers. In the ciphertext policy (CP-ABE) schemes, the patients use access tree to encrypt data, and only when the users give the corresponding key they can access the data. CP-ABE schemes always require intensive computing resources to run the encryption and decryption algorithms. However, in body area networks the computing devices are always lightweight such as cell phones and sensors. Outsourcing the heavy computation without exposing the sensitive data contents or keys to the cloud service providers is a good choice to solve the problem.

Sahai and Waters first proposed the attribute-based encryption [12] which was built on the Identity-Based Encryption(IBE). In IBE, the users' key and ciphertext are described by strings. When the number of strings is within the threshold, the users can decrypt the ciphertext. Bethencourt and Cheung proposed the CP-ABE based on ABE [1, 2]. The key of user is associated with some attributes, and the access structure is included in ciphertext. If and only if the users' attributes satisfy the access structure, the users can decrypt ciphertext. Li and Yu focused on the multiple data owner scenario, and divided the users into multiple security domains to reduce the key management complexity [7]. Wei Li et al. conducted a threshold multi-authority CP-ABE access control scheme for public cloud storage [8]. However, the medical data can be accessed by all the users whose attributes satisfy the policy. If we want the users only can access special data, these methods are impossible, because all the data are encrypted in one ciphertext. Wan and Wang proposed the hierarchy attribute-based encryption in cloud [13, 15]. These schemes achieve scalability and inherit flexibility and fine-grained access control, however, the medical data are not partitioned. Therefore, all the data are in the same security level. Zhou incorporated his system into mobile cloud computing scenarios. He put the intensive computation of CP-ABE encryption and decryption to cloud service providers without disclosing their data content and secret keys [16]. Therefore, the cloud services do most of the storage and computation work, alleviating the burden of the sensors. But, he didn't classify the medical data. Li et al. proposed an outsourced ABE system, which supports both secure outsourced key issuing and decryption [6]. Jung et al. proposed an AnonyControl scheme which addressed not only the data privacy but also the user identity privacy [4]. However, they also didn't consider the classification of data.

In this paper, we propose a role-based attribute-based access control scheme (RACS) by extending the CP-ABE. In this scheme, the owners classify the medical data into different parts, and encrypt them by different access policies using different attributes, in this way the users can access data according to their roles. The users can only access parts of the medical data by their roles. Therefore the security of medical data can be guaranteed. For the illegal users, the patients can revoke their access privileges. When the patients are in danger, for example, coma, the paramedics can obtain the temporary key to access the medical data to insure the efficient rescue in the short time. In order to relieve the burden of sensors, parts of the encryption and decryption are put in the cloud. We adopt a method which divides the access structure into

two parts, one part is in the sensors and the other is in the cloud.

The main contributions of this paper are: (1) we divide the medical data into different parts to realize that different users can access different parts, (2) we divide the policy into two parts: one part is stored in the cloud, and the other is in owner, (3) we consider break-glass in the scheme, (4) we analysis the correctness, security and storage and computation overhead of the scheme.

2 Preliminary

2.1 Bilinear Maps

Let G_1 and G_2 be two groups of prime order p, and g be a generator of G_1. A bilinear map is an injective function $e : G_1 \times G_1 \to G_2$ with the following properties:

Bilinearity: $\forall u, v \in G_1$, $a, b \in Z_p$, there is $e(u^a, v^b) = e(u, v)^{ab}$.

Non-degeneracy: $e(g, g) \neq 1$.

Computability: There is an efficient algorithm to compute $e(u, v)$ for each $u \in G_1$ and $v \in G_1$.

2.2 Bilinear Diffie-Hellman Problem (BDHP)

Given two groups G_1 and G_2 with the same prime order p, let $e : G_1 \times G_1 \to G_2$ be a bilinear map and g be a generator of G_1. The objective of BDHP is to compute $e(g, g)^{abc}$ in (G_1, G_2, e) from the given (g, g^a, g^b, g^c), where $a, b, c \in Z_p$.

2.3 Access Tree

Access tree expresses the structure of access control. Let T be an access tree with root r, and $att(x)$ be the attributes associated with the node x. If num_x is the number of children of a node x and k_x is its threshold value, then $0 < num_x \leq k_x$. When $k_x = 1$, the threshold gate is an OR gate; when $k_x = num_x$, it is an AND gate. Denote T_x as the subtree of T rooted at the node x. Hence T_x is the same as T. When the attributes associated with the ciphertext satisfy the owners' access structures, the users can get the medical data. If a set of attributes satisfy the access tree T_x, we denote it as $T_x(\gamma) = 1$. $T_x(\gamma)$ can be computed recursively as follows: When x is a leaf node, $T_x(\gamma) = 1$ if and only if $att(x) = \gamma$. When x is a non-leaf node, evaluate $T'_x(\gamma)$ for all children x' of node x. $T_x(\gamma) = 1$ if and only if at least k_x children return 1.

3 Scheme Model

3.1 Problem Definition

We consider a role-based attribute encryption cloud storage access control scheme in which there are different owners and different users. Figure 2 shows the cloud storage model. Every patient owns his medical data and shares them to users through Internet. The medical data are stored in a third party. The patients have their control rights of medical data entirely. They can produce, manage and delete their data. The center server manages these data. The users can read or write different parts of medical data from the servers according to their attributes. When a user wants to access medical data, he first checks that which access tree his attributes satisfy. For example, if a user's attributes satisfy the access tree of CT_i, he only can obtain CT_i and can't get other parts of CT. The role-based scheme consists of the following components:

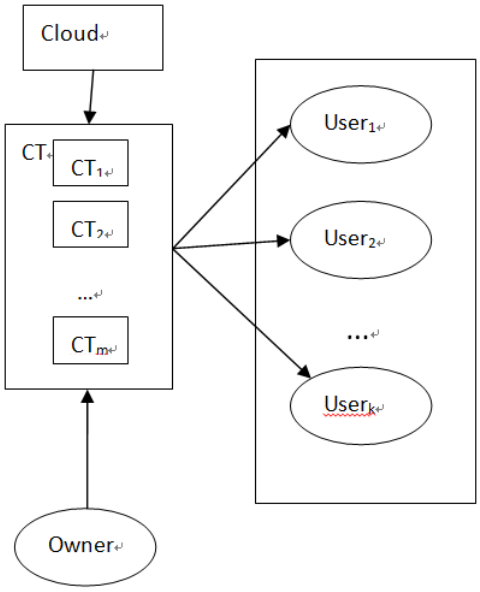

Figure 2: Storage model of medical data

1) The service provider. It is the third party which controls users accessing the outsourcing data and providing outsourcing services.

2) Owners. The patients who own the medical data define the access policies and outsource them to the service provider after encrypting.

3) Center server. It is an attribute set key institution. It produces the public key and the master key. It also distributes, revokes and renews the users' private key.

4) Users. The persons who want to access medical data. If a user owns the attributes which satisfy the access policy, he can decrypt the corresponding data.

We show an example to illustrate the process of the scheme. Suppose Alice is a patient in hospital A. She creates her medical data file F and divides them into different parts, such as personal information, medical history, physical examination information, and sensitive data. We illustrate it in Figure 3. Alice encrypts them according

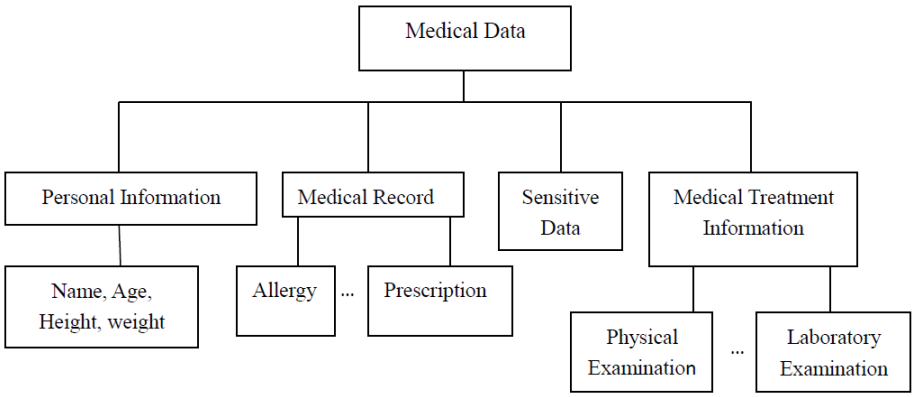

Figure 3: The division of medical data

to the access structures she defines. Different users have different access privileges as they own different attributes. For example, doctors can access all data; insurance company can only access personal information; her friends and researchers can access medical history. Alice also transport a temporary key to the trust center, when the emergency happens, the emergencies can get the key and access the fist-aid data.

3.2 Role-based Access Control Scheme in Cloud Storage (RACS)

Scheme Initialization:

1) Let G be the group with prime order p and g be the generator of G.
2) Choose two random numbers $\alpha, \beta \in Z_p$, the public key is $PK : \{g, h = g^{\beta}, f = g^{y_{\beta}}, e(g,g)^{\alpha}\}$.
3) The master key is $MK : \{\beta, g^{\alpha}\}$.

Key Generation:

1) Choose a random number $\gamma \in Z_p$.
2) For each attribute $j \in S$, choose random numbers $\gamma_j \in Z_p$.
3) Generate the private key: $D = g^{(\alpha+\gamma)/\beta}$, $\forall j \in S : D_j = g^{\gamma} \cdot H(j)^{\gamma_j}$, $D' = g^{\gamma_j}$.

The ciphertext is divided into different parts. For each part, an access tree is constructed. Therefore, different keys are generated to decrypt them. According to users' roles, the key generation algorithm generates their private keys.

Encryption before Outsourcing:

For the patients, the privacy is an important issue. For example, a patient may don't want some users to know that he has certain diseases. We can divide the data M into N parts, $M = \{M_1, M_2, \cdots, M_N\}$, and encrypt them with corresponding access trees. After encryption, the ciphertext is constructed as $CT = \{CT_1, CT_2, \cdots, CT_N\}$. $CT_k (k = 1, 2, \cdots, N)$ indicates one part of ciphertext. It can be decrypted by users whose attributes satisfy the corresponding access trees. In this way, we can realize role-based access control.

In order to alleviate the heavy computation, parts of encryption and decryption are moved to cloud. An access tree is divided into two parts: $T = T_{CLOUD} \wedge T_{DO}$. T_{CLOUD} is one part of access tree which is controlled in the cloud and T_{DO} is the other part which is controlled by data owner. To relieve the computation overhead in the Inter-BAN, T_{DO} usually has a small number of attributes. Most of the computation is performed in T_{CLOUD} which is stored in cloud. We illustrate an access tree in Figure 4.

For the T_{CLOUD}, the process is as follows: Choose a polynomial q_x for each node x, and set the degree $d_x = k_x - 1$ (k_x is the threshold of x). For the root node R in the tree, choose a random number $s \in Z_p$ and set $q_k(0) = s$. For other nodes, set $q_x(0) = q_{parent(x)(index(x))}$ and choose randomly other d_x nodes to define q_x.

For the T_{DO}, the process is as follows:

1) Encrypt $(q_R(2), T_{DO})$ and $CT_{DO} = \{\forall y \in Y_2 : C_y = g^{q_y(0)}, C'_y = H(att(y))^{q_y(0)}\}$.
2) Computer $\widetilde{C} = Me(g,g)^{\alpha s}$ and $C = h^s$, where M is the message.
3) Send CT_{DO}, $\widetilde{C}$, C to the cloud.

On receiving the message from data owner, the cloud server generates the following ciphertext:

$$\begin{aligned} CT = \{&T_{ESP} \wedge T_{DO}; \widetilde{C} = Me(g,g)^{\alpha s}; C = h^s; \\ &\forall y \in Y_{ESP} \cup Y_{DO} : C_y = g^{q_y(0)}, \\ &C'_y = H(att(y))^{q_y(0)}\}. \end{aligned}$$

Data Decryption:

When a user wants to access CT_k, the center server

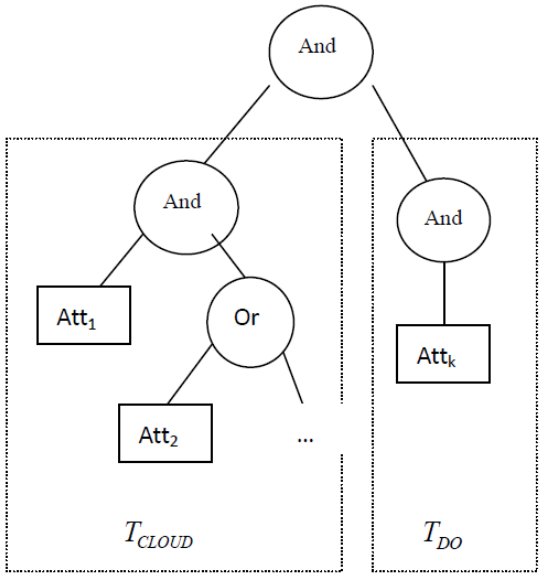

Figure 4: An Access Tree of CT_k

first checks whether his attributes satisfy the corresponding access tree. If it is satisfied, the decryption is handed over to the cloud. The user sends SK_k to the cloud, and requests the cloud provider to send the ciphertext. When x is a leaf node, $i = att(x)$, the decryption process is as follows: if $i \in S$,

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

If $i \notin S$, $DecryptNode(CT, SK, x) = \perp$;

The recursion is processed as follows: $\forall y$ is the child of x. It calls $DecryptNode(CT, SK, y)$ and stores the output as F_y. Let S_x be an arbitrary k_x-sized set of child node y, the computation is processing in cloud as follows:

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

Where $i = index(z)$ and $S'_x = \{index(z) : z \in S_x\}$. Finally, the recursive algorithm returns $A = e(g,g)^{rs}$.

Key Update:

When users need to be revoked or attributes are changed, the owner can update the users' privileges through trusted center. Suppose there is a user revocation, the key is updated as follows:

The trust center chooses $s' \in Z_p$ randomly and a key K'_{λ_i} which is different to original K_{λ_i}, then encrypt the ciphertext again.

1) $C' = Me(g,g)^{\alpha(s+s')}$.
2) $C_i = g^{q_i(0)+s'}$, $C'_i = (H(\lambda_i)^{q_i(0)+s'})K'_{\lambda_i}$.
3) $\forall y \in Y/\{i\} : C_y = g^{q_y(0)+s'}$, $C'_y = (H(\lambda_y)^{q_y(0)+s'})^{K_{\lambda_y}}$.

Break-glass:

When a patient is in emergency, the first-aiders need to access medical data temporarily. They prove their privileges from emergency response department, and get the patient's emergency key to decrypt the medical data. After the emergency treatment, the patient computes an emergency key once again.

1) Choose a random number $\eta \in Z_p$.
2) Generate the private key $D = g^{(\alpha+\eta)/\beta}$.

4 Scheme Analysis

4.1 Correctness

The decryption process starts from the root of tree. We observe $DecryptNode(CT, SK, r) = e(g,g)^{rq_R(0)} = e(g,g)^{rs}$ if and only if the attributes satisfy the access tree.

$$\begin{aligned} M' &= \frac{C'}{e((C,D)/A)} \\ &= \frac{Me(g,g)^{\alpha s}}{e(h^s, g^{(\alpha+\gamma)/\beta})/e(g,g)^{rs}} \\ &= \frac{Me(g,g)^{\alpha s}}{e(g^{\beta}, g^{(\alpha+\gamma)/\beta})/e(g,g)^{rs}} \\ &= \frac{Me(g,g)^{\alpha s}}{e(g,g)^{(\alpha+\gamma)}/e(g,g)^{rs}} \\ &= M. \end{aligned}$$

4.2 Security Analysis

Theorem 1. *RACS is secure against the collusion attack. In RACS, each attribute is assigned with a random number. For the users, all the private keys are generated based on these random numbers. The access tree is divided into two parts, T_{CLOUD} and T_{DO}. One is stored in cloud and the other in owners. The polynomials in the cloud is set by random numbers. Therefore, even the users collude together, they can't decrypt the part which is in cloud. Thus, it is impossible for multiple users collude together to decrypt the ciphertext.*

Table 1: Comparison of storage overhead

Scheme	Public key	Master key	Private key	Cloud storage
EDRS [5]	3	2	$3\|n_i\|+1$	$2\|S\|+3+\|T\|$
FH-CP-ABE [15]	$3\|p\|_1+\|p\|_2$	$Z_p+\|p\|_1$	$(2\|n_i\|+1)p$	$(2S+k)p_1+(jS+k)p_2$
RACS	4	2	$2\|n_i\|+1$	$2\|S\|+2+\|T\|$

Table 2: Comparison of computation overhead

Scheme	Setup	KeyGen	Encrypt	Decrypt
EDRS [5]	$3E+1e$	$(\|U\|\|n_i\|+\|U\|)M+(4\|U\|\|n_i\|+2\|U\|)E$	$1M+(2\|S\|+2)E$	$O(\|S\|+2)M+(2\|S\|+1)e$
FH-CP-ABE [15]	$2E+1e$	$(\|U\|\|n_i\|+1)M+(3\|U\|\|n_i\|+1)E$	$1M+(2\|S\|+2)E+2e$	$O(\|S\|+2)M+(2\|S\|+1)e$
RACS	$3E+1e$	$(\|U\|\|n_i\|+1)M+(\|U\|\|n_i\|+1)E$	$1M+(2\|S\|+2)E$	$O(\|S\|+2)M+(2\|S\|+1)e$

Theorem 2. *RACS with outsourced decryption is secure against chosen-plaintext attack in selective model under DBDH assumption.*

Proof. We now describe the security model of RACS by the following game between a challenger and an adversary.

Init: Assume there is an adversary A with attributes set W breaks the proposed scheme. We can build a simulator S that uses A as a sub-algorithm to solve the DBDH problem. The challenger S chooses a fair binary coin $\mu=\{0,1\}$, $a,b,c\in Z_p$. If $\mu=0$, S is given $(A,B,C,Z)=(g^a,g^b,g^c,e(g,g)^{abc})$; otherwise it sets z as a random number, S is given $(A,B,C,Z)=(g^a,g^b,g^c,e(g,g)^z)$. The challenger S runs A and receives a challenge attributes set W from A and sends public key: $PK:\{g,h=g^\beta,f=g^{1/\beta},e(g,g)^\alpha\}$ to A.

Phase 1: In this phase, A repeatedly makes private key requests for W. The center server gives two different private keys to A.

$$SK: \quad (D=g^{(\alpha+\gamma)/\beta},\forall j\in W: D_j=g^r\cdot H(j)^{r_j}, D'_j=g^{r_j})$$

$$SK: \quad (D=g^{(\alpha+\gamma')/\beta},\forall j\in W: D_j=g^{r'}\cdot H(j)^{r'_j}, D'_j=g^{r'_j})$$

j is the attribute in W, r,r',r_j,r'_j are the random numbers in Z_p. Challenge: The adversary A submits two messages M_0, M_1 to challenger, and get the challenge ciphertext as follows:

$$CT^* = (T,C'=M_b\cdot e(g,g)^{\alpha S},C=h_1^s=S, C'=h_2^S,\forall y\in Y: C_y=g^{q_y(0)})$$

The challenger returns CT^* to A.

Phase 2: A queries the questions that is not queried in Phase 1. The challenger answers like in Phase 1.

A outputs a guess b' of b, if $b'=b$, S outputs $\mu'=0$ to indicate that it is given a DBDH-tuple; otherwise, it outputs $\mu'=1$ to indicate it is given a random 4-tuple.

□

4.3 Storage Analysis

We compare the RACS with other schemes according to the size of public key, private key, ciphertext for one data content. As the part of the ciphertext in RACS is stored and computed in cloud, we only consider the part which is at local. The results are shown in Table 1. Let $|p|_1$ denotes the size of an element in G_i, $|p|_2$ denotes the size of an element in G_2, k be the hierarchical files, N_0 be the number of owners, N_μ be the number of users, n be the number of attributes, n_i be the set of attributes belonging to user μ_i, S be the set of attributes which are used to specify the access policy in the ciphertext.

4.4 Computation Analysis

We evaluate the energy consumption on computation of RACS. The analytical results of each scheme in terms of computation are summarized in Table 2. Each of them is based on the entire computation at each phase.

5 Conclusion

In this paper, we propose a novel role-based access control scheme using CP-ABE in cloud. As the cloud servers are partially trustworthy, we don't put all the medical data to cloud. Patients have full control of their own privacy through encrypting their medical data to allow

fine-grained access. We enhance the CP-ABE so as to relieve the storage and computation overhead in body area networks. In order to protect the privacy of patients, we divide medical data into different parts. The RACS permits users to access different parts according to their roles (professional roles, qualifications, and affiliations) and greatly reduce the complexity of key management. We design RACS to encrypt the medical data. Furthermore, we add user revocation and break-glass to handle privacy and emergency problems. Through comparison, we show that our scheme is efficient than other schemes.

References

[1] J. Bethencourt, A. Sahai, B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of IEEE Symposium on Security and Privacy*, pp. 321–334, California, USA, May 2007.

[2] L. Cheung, C. Newport, "Provably secure ciphertext policy ABE," in *Proceedings of 14th ACM Conference on Computer and Communications Security*, pp. 456–465, New York, USA, Oct. 2007.

[3] P. S. Chung, C. W. Liu, and M. S. Hwang, "A study of attribute-based proxy re-encryption scheme in cloud environments," *International Journal of Network Security*, vol. 16, no. 1, pp. 1–13, 2014.

[4] T. Jung, X. Y. Li, Z. Wan, M. Wan, "Control cloud data access privilege and anonymity with fully anonymous attribute based encryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, pp. 190–199, 2015.

[5] D. Y. Koo, J. Hur, H. Yoon, "Secure and efficient data retrieval over encrypted data using attribute-based encryption in cloud storage," *Computers and Electrical Engineering*, vol. 39, no. 1, pp. 34–46, 2013.

[6] J. Li, X. Huang, J. Li, X. Chen, X. Yang, "Secure outsourcing attribute-based encryption with checkability," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 8, pp. 2201–2210, 2014.

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

[8] W. Li, K. Xue, Y. Xue, J. Hong, "TMACS: A robust and verifiable threshold multi-authority access control system in public cloud storage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 5, pp. 1484–1496, 2016.

[9] C. W. Liu, W. Fu Hsien, C. C. Yang, M. S. Hwang, " A survey of attribute-based access control with user revocation in cloud data storage," *International Journal of Network Security*, vol. 18, no. 5, pp. 900–916, 2016.

[10] C. Min, S. Gonzalez, V. Athanasios, "Body area networks: A survey," *Mobile Networks and Applications*, vol. 16, no. 2, pp. 171–193, 2011.

[11] A. Mosa, H. M. El-Bakry, S. M. Abd El-Razek, S. Q. Hasan, "A proposed E-government framework based on cloud service architecture," *International Journal of Electronics and Information Engineering*, vol. 5, no. 2, pp. 93–104, 2016.

[12] A. Sahai, B. Waters, "Fuzzy identity based encryption," in *Proceedings of 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 457–473, Aarhus, Denmark, May 2005.

[13] Z. G. Wan, J. Liu, R. Deng, "HASBE: A hierarchical attribute-based solution for flexible and scalable access control in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 743–754, 2012.

[14] Z. Wang, Y. Lu, G. Sun, "A policy-based deduplication mechanism for securing cloud storage," *International Journal of Electronics and Information Engineering*, vol. 2, no. 2, pp. 70–79, 2015.

[15] S. Wang, J. Zhou, J. K. Liu, J. Yu, J. Chen, W. Xie, "An efficient file hierarchy attribute-based encryption scheme in cloud computing," *IEEE Transactions on Information Froensics and Security*, vol. 11, no. 6, pp. 1265–1277, 2016.

[16] Z. B. Zhou, *On Efficient and Scalable Attribute Based Security Systems*, Arizona State University, Arizona, 2011.

Biography

Ye Tian received her M.E degree from Taiyuan university of technology, China, in 2006. She is currently a ph.D.student in Taiyuan university of technology and an associate professor in Taiyuan Normal University. Her current research interests are information security and wireless body area networks.

Yanbin Peng received the Master degree in Engineering in computer technology in 2016 from the Taiyuan University of Technology, Taiyuan, China. She is currently a software engineer at Software Development Center, Agricultural Bank of China. Her research interests include network and information security.

Gaimei Gao received the M.E from Taiyuan University of Science and technology, China, in 2007. She is currently a ph.D.student in Taiyuan university of technology. Her research interests are Database and Information Security.

Xinguang Peng received the D.E from Beijing Institute of technology, China in 2004. He is now a professor and doctoral supervisor in college of computer science and technology, Taiyuan University of Technology, Taiyuan, China. His research interests include information security and trusted computing.