A Secure and Efficient Ciphertext Encryption Scheme Based on Attribute and Support Strategy Dynamic Update Via Hybrid Encryption Method

Yanhua Wang[1], Yaqiu Liu[1] and Kun Wang[2]
(Corresponding author: Kun Wang)

College of Information and Computer Engineering, Northeast Forestry University[1]
Harbin 150040, China
College of Information and Electronic Technology, Jiamusi University[2]
148 Xuefu Road, Xiangyang District, Jiamusi City, Heilongjiang Province 154007, China
(Email: wk_116@126.com)

(Received Sept. 21, 2017; revised and accepted Dec. 8, 2017)

Abstract

Traditional ciphertext encryption scheme easily leaks individual data privacy information. Therefore, this paper proposes a secure and efficient ciphertext encryption scheme based on attribute and support strategy dynamic update via hybrid encryption method. The asymmetric encryption algorithm (ASE) is adopted to encrypt the data and the attribute-based encryption algorithm (ABEA) to encrypt the sharing key. The results are stored in cloud with the form of ciphertext, which ensures the security of stored data. Meanwhile, it reduces the number of ciphertext variable parameters, this way saves the energy consumption of physiological sensors. Access strategy of dynamic update technology is based on attribute encryption algorithm which provides an extension features, it is the effective way to achieve fine-grained access control. Dynamic update technology allows data owner to dynamically update ciphertext access information in the clouds. Through comparing the proposed scheme with the state-of-the-art schemes in terms of performance, the experimental results show that the proposed scheme can ensure the security of the stored data. In addition, it has a low communication and computation overhead, which proves that our new scheme has higher performance than other encryption schemes.

Keywords: Asymmetric Encryption; Ciphertext Encryption Scheme; Data Privacy; Energy Consumption; Support Strategy Dynamic Update

1 Introduction

In recent years, how to design an efficient and secure public key encryption scheme is a hot issue. Ciphertext encryption system is a special public key encryption system [1,16]. In a ciphertext-based encryption scheme, any string (such as an E-mail address), address, ID card number *etc.*, can be used as a legitimate public key. The private key of user needs to be generated by the private key generation center. Due to the widely use and flexible application of ciphertext encryption, researchers propose a variety of secure ciphertext encryption schemes, but the common drawback is that the safety certificate is not compact, or difficult to solve the Diffie-Hellman (BDH) problem [6,9,14]. For data storage security privacy protection, Jivanyan [4] proposed to send the user key of encrypted the data through the security channel to ensure the stored data privacy security. To reduce the transmission overhead,

Li [5] proposed a new encrypted storage scheme that used the user's private key to encrypt the data and store it on the server side. The user access server could decrypt the data directly. But the above two programs need to maintain a large number of keys, and it is not easy to control. Zhou [18] proposed a new construction of Ciphertext Policy Attribute-Based Encryption (CP-ABE), named Privacy Preserving Constant CP-ABE (denoted as PP-CP-ABE) that significantly reduced the ciphertext to a constant size with any given number of attributes. Furthermore, PP-CP-ABE leveraged a hidden policy construction such that the recipients' privacy was preserved efficiently. As far as we know, PP-CP-ABE was the first construction with such properties [2]. Furthermore, he developed a Privacy Preserving Attribute-Based Broadcast Encryption (PP-AB-BE) scheme. Compared to existing Broadcast Encryption (BE) schemes, PP-AB-BE was more flexible because a broadcasted message could be encrypted by an expressive hidden access policy, either with or without explicit specifying the receivers. Hu [?] proposed a secure and efficient data communication protocol. After establishing a communication channel by a two-factor authentication, it used CP-ABE technology and signature methods to make protection and privacy certification for the data, but the weak point was that the energy consumption was large. But they has a common problem-low computational efficiency.

In view of the above problems, this paper presents an attribute-based security and efficient ciphertext encryption scheme via hybrid encryption method. The asymmetric encryption algorithm encrypts the data and attribute-based encryption algorithm encrypts the sharing key respectively to ensure the security of the stored data and realize the user's attribute-based access control. The traditional encryption algorithm is improved by reducing the number of ciphertext variable parameters, it has improved communication efficiency and reduced computational overhead.

The rest of the paper is organized as follows. Section 2 introduces public key encryption definition and security model. Section 3 outlines the proposed scheme to analyze detailed processes. Experience and security analysis are given in Section 4. Section 5 finally concludes this paper.

2 Public Key Encryption Definition and Security Model

A public key encryption scheme consists of three algorithms.

1) Key generation algorithm ($Setup(k)$). Given secure parameter k, this algorithm outputs public key pk and corresponding private key sk.

2) Encryption algorithm ($Enc(pk, m)$). Given public key pk and plaintext m, this algorithm outputs C.

3) Decryption algorithm ($Dec(sk, C)$). Given private key sk and ciphertext C, this algorithm outputs plaintext M or N. N indicates that it is illegal ciphertext.

IND-CCA2 secure model of public key encryption can be defined by the game between challenger Ch and adversary A.

Stage 1. System building. Challenger Ch runs $Setup$ to generate public-private key pair (pk, sk) and reserve sk. pk is sent to adversary.

Stage 2. Query1. Adversary does a series of decryption queries. When adversary decrypts ciphertext C, challenger runs $Dec(sk, C)$ and sends results to adversary.

Stage 3. Challenge. When adversary decides to finish query1 stage, it will output two equal plaintexts m_1 and m_2. Challenger will randomly select $\alpha \in (0, 1)$ and set challenge ciphertext as $C' = Enc(pk, m_\alpha)$. Then C' will be returned to adversary.

Stage 4. Query2. Adversary does a series of decryption queries. Challenger uses the same method like stage1 to respond. Here, adversary cannot decrypt C'.

Stage 5. Guessing. Finally, adversary outputs the guess α' of α.

The above adversary A is defined as IND-CCA2 adversary. And the advantage of public key encryption is as:

$$Adv_A^{IND-CCA2} = |Pr[\alpha' = \alpha] - \frac{1}{2}|.$$

For any adversary A with probabilistic polynomial time, the advantage $Adv_A^{IND-CCA2}$ of public encryption scheme can be ignored. Therefore, this public encryption scheme is IND-CCA2 security.

3 Secure and Efficient Ciphertext Encryption Scheme

The system mainly involves four basic technologies: access tree [3], bilinear map [8], Shamir secret Sharing [17] and CABE [12].

3.1 Access Tree

Supposing T is an access tree, and each non-leaf node x in T represents a threshold structure described by its child node num_x and a threshold k_x, where num_x represents the number of child nodes of node x. k_x represents the threshold value of x, and $k_x \in [1, num_x]$. When $k_x = 1$, node x denotes a logic OR relation. When $k_x = num_x$, it represents a logic AND relationship. Each leaf node x is described by an attribute and a threshold $k_x = 1$.

Access tree involves several related functions in the practical application. Here is a brief introduction. The function $parent(x)$ represents the parent of node x. The function $att(x)$ represents the associated attribute of return leaf node x. In order to conveniently invoke nodes in the access tree, access tree sorts each node, and the function $index(x)$ represents the index value of the return node x.

3.2 Bilinear Map

Supposing G_0 and G_1 are two $p - order$ multiplicative cyclic groups. g is a generator of G_0 and e is a bilinear map, namely $e : G_0 \times G_0 \to G_1$, then for any $i, j, k \in G_0$ and $a, b \in Z_p$, the map e has the following properties:

1) Bilinear: $e(i^a, j^b) = e(i, j)^{ab}$.

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

3) Polymerizability: $e(i \cdot j, k) = e(i, k) \times e(j, k)$.

If the group operation is highly computable in G_0 and the map $e : G_0 \times G_0 \rightarrow G_1$, then the group is called bilinear. So map e is commutative: $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

3.3 Shamir Secret Sharing

Shamir secret sharing technology can be explained that distributors distribute the sharing key s to n participants $P_1, P_2, \cdots, P_n$, any t participants can restore the key, and less than t participants can not restore the key. The main principle of Shamir secret sharing is to divide the key s into n parts and distribute them to n participants. Randomly selecting the t integers $a_0, a_1, \cdots, a_{t-1}$ from finite field $GF(q)$. Supposing $a_0 = s$, we can form a polynomial $f(x) = s + a_1x + a_2x^2 + \cdots + a_{t-1}x^{t-1}$. Calculating $f(x_i)(i = 1, 2, \cdots, n)$, and then sharing x_i and $f(x_i)$ with the participant P_i. When the collected sub-secrets are greater than or equal to that provided by t participants, we can rebuild the key S by the following formula.

$$s = f(0) = \sum_{i=1}^{t}(f(x_i) \prod_{j\in[1,t],j\neq i\frac{0-x_j}{x_i-x_j}}).$$

3.4 Ciphertext Attribute-based Encryption

Ciphertext attribute-based encryption (CABE) is a method that associates a user's private key with a set of user attributes and expresses the attribute discrimination condition as an access tree and is deployed in a ciphertext. The user can decrypt the ciphertext only if the attribute set of the user's private key satisfies the attribute judgment condition in the ciphertext.

In general, CABE includes the following four steps:

1) System initialization *Setup*: input a random parameter, output the main key MK and public parameter PK.

2) Message Encryption $CT = Encrypt(PK, M, T)$: input encrypted message M, public parameter PK and access tree T, output ciphertext CT.

3) Key generation $SK = KeyGen(MK, S)$: input attribute set S and the main key MK, output the user's public key PK and private key SK.

4) Ciphertext decryption $M = Decrypt(CT, SK)$: input ciphertext CT and user private key SK. And decrypt ciphertext CT to get the plaintext M.

3.5 Proposed Security and Efficient Ciphertext Encryption Scheme

Because of the excellent characteristics of CABE, it can be used for our ciphertext access control scenario, but CP-ABE belongs to asymmetric encryption algorithm, it is difficult to encrypt a lot of data. So in our scheme, we first use an AES sharing secret key K to encrypt the data, a RSA public key is used for signing and verifying K_{pub}, K_{priv} for K. Then the K is encrypted using the improved CP-ABE to reduce the number of ciphertext variables and ensure that K can only be accessed by legitimate users.

The following is the detailed realizing process for our new scheme.

3.5.1 System Initialization

1) Randomly select a $p - order$ cyclic group G_0, a generator g and a bilinear pair $e : G_0 \times G_0 \rightarrow G_1$. Select a hash function $H : G_0^2 \rightarrow Z_p$ and $(x_1, x_2, x_3 \in G)$. Compute $g_1 = g^x$;

2) Define a Lagrangian coefficient $\Delta_{i,x} : \Delta_{i,x}(x) = \prod_{j\in[1,t],j\neq i\frac{0-x_j}{x_i-x_j}})$, i represents the element in user attribute set S;

3) Randomly select a parameter α and a hash function $H : 0, 1^* \rightarrow 0, 1^l$;

4) Generate the master key $MK = (\alpha, g^\alpha)$ and the parameter $PK = G_0, g, e(g, g)^a$;

5) Use PK and user attribute set S to generate the user private key $SK = (\forall_j \in S : D_j = g^{\frac{a}{H(j)}}$.

3.5.2 Data Encryption

Given a plaintext $M \in G_T$, it first randomly selects $\tau \in Z_p^*$.

1) Physiological sensor SN_i perceives information M_1, generates a hash value $hash(M_1)$ for M_1;

2) SN_i uses K to encrypt information M_1 and get $E_K(M_1, hash(M_1))$, uses K_{priv} to sign K and get $SIG_{K_{priv}}(K)$;

3) Use Algorithm 1 to encrypt K and get $CK = (T, C = Ke(g, g)^{as}$, $y \in Y$, $C_y = g^{H(att(y))qy(0)}$;

4) SN_i sends $E_K(M_1,\ hash(M_1))$ and $\{SIG_{K_{priv}}(K)$, $CK\}$ to PDA, $SN_i \rightarrow PDA$: $\{ID_{SN_i}$, ID_{PDA}, $E_K(M_1,\ hash(M_1))$, $SIG_{K_{priv}}(K)$, $CK\}$;

5) When the SN_i uploads the $i{-}th$ data M_1, performing step1 and step2, sends $E_K(M_i, hash(M_i))$ to PDA: $SN_i \rightarrow PDA : ID_{SN_i}, ID_{PDA}, E_K(M_1, hash(M_1))$;

   In each time period, SN_i only sends the ciphertext data CK and the signature data $SIG_{K_{priv}}(K)$ to PDA, the next transmission is happened when the sharing key K is updated or invalidated.

6) Then it calculates $C_0 = e(g_1, x_3)^r M$, $C_1 = g^r$ and $C_2 = (x_1^w, x_2^\tau, x_3)^r$. Here, $w = H(C_0, C_1)$. Finally, it outputs ciphertext CK.

Algorithm 1. $Encrypt(PK, K, T)$

- Input: public parameter PK, sharing key K and access tree T with root node R.
- Output: ciphertext CK.

Step 1. Set a polynomial q_x with $d_x = k_x - 1$ power for each node in the access tree.

Step 2. Select a random number $s \in Z_p$ and set $s = q_R(0)$, meanwhile, select d_R random numbers $a_i \in Z_p$. Define the polynomial q_R for root node R.

Step 3. Set $q_x(0) = q_{parent}(x)(index(x))$ for any other nodes x in the access tree, and select d_x random numbers from Z_p to define q_x.

Step 4. Let Y be the set of leaf nodes, and the ciphertext constructed by access tree T is $CK = (T, C = Ke(g,g)^{as}, y \in Y : C_y = g^{H(att(y))q_y(0)})$.

Step 5. Output ciphertext CK.

3.5.3 Support Strategy Dynamic Update

Support strategy dynamic update (SSDU) contains attribute authority, cloud server, data owner and data user.

- Attribute authority. In the system, it has multiple attribute authorities, each attribute authority does not depend on other attributes authorities. It is only responsible for managing their domain attribute of users. Attribute authority is responsible for generating the public and private key pair. And according to the domain users, they generate their own private key respectively.
- Cloud server. It is responsible for storing data and providing file access service for the data users. Cloud server is responsible for updating old ciphertext access strategy tasks at the same time. After updating, the completion of the new ciphertext is equivalent to the ciphertext directly generated by the new access strategy.
- Data owner. It is responsible for access strategy and using these strategies to encrypt data. Data owner is responsible for generating policy to update the key at the same time, and requesting a cloud server to update the old access strategy.
- Data user. The system generates an unique identity for each data user.

In SSDU, there are seven polynomial time algorithms: *GlobalSetup*, *AuthoritySetup*, *SKeyGen*, *Encrypt*, *Decrypt*, *UKeyGen* and *CTUpdata*.

- $GlobalSetup(\lambda) \rightarrow GP$. Select a random security parameter λ as input and output the public parameter GP.
- $AuthoritySetup(GP, AID) \rightarrow (SK_{AID}, PK_{AID})$. In the system, each attribute authority performs this algorithm to complete initialization operation. Use public parameter GP and unique identity AID to generate private key and public key (SK_{AID}, PK_{AID}).
- $SKeyGen(GID, GP, S_{GID,AID}, SK_{AID})$. AID produces attribute private key SK_{AID} for GID. SK_{AID}, GID, $S_{GID,AID}$ and GP is as input. $SK_{GID,AID}$ is as output.
- $Encrypt(PK, GP, m, A) \rightarrow CT$. Public parameter GP, access structure A and public key set PK are used to encrypt message m. Output ciphertext CT.
- $Decrypt(CT, GP, SK_{GID,AID} \rightarrow m$. GP and $SK_{GID,AID}$ are used to decrypt ciphertext CT. If the attribute of user private key can satisfy the access structure in ciphertext CT, then the encryption algorithm can correctly decrypt ciphertext and output message m.
- $UKeyGen(PK, EnInfo(m), A, A') \rightarrow UK_m$. Data owner executes the cryptographic strategy update key generation algorithm. Public key set PK, enciphered message $EnInfo(m)$ of message m, old access structure A and new access structure A' are as input. Then it generates ciphertext update key UK_m.
- $CTUpdata(CT, UK_m) \rightarrow CT'$. Ciphertext update algorithm is performed by the cloud server. It uses UK_m to update CT and outputs the ciphertext CT' related to new access structure A'.

3.5.4 Decryption Stage

Given ciphertext $C = (\tau, C_0, C_1, C_2)$, computing $w = H(C_0, C_1)$ and verifying $e(C_1, x_1^w, x_2^\tau, x_3) = e(g, C_2)$. Plaintext will be recovered with private key $M \leftarrow \frac{C_0}{e(C_1, x_3)^x}$.

1) The user obtains the ciphertext $\{ID_{SN_i}$, $E_K(M_1$, $hash(M_1)$, $\cdots$, $E_K(M_n$, $hash(M_n)))\}$ and $SIG_{K_{priv}}(K), CK$ of SN_i from the PDA;

2) The user uses Algorithm 2 to decrypt CK and get K;

3) The user uses K_{pub} to verify the correctness of $SIG_{K_{priv}}(K)$, if it is correct, then it performs the next step, otherwise returns to the first step;

4) User uses K to decrypt $E_K(M_1$, $hash(M_1)$, $E_K(M_2$, $hash(M_2)$, $\cdots$, $E_K(M_n$, $hash(M_n)))$ and get M_1', M_2', $\cdots$, M_{n1}', $hash(M_1)$, $hash(M_2)$, $\cdots$, $hash(M_n)$;

5) User verifies batched the correct of information $M_1', M_2', \cdots, M_{n1}'$, if $hash(M_i') = hash(M_i), \cdots, hash(M_n), (1 \leq i \leq n)$, then the information acquired by users is valid, otherwise give up it;

Algorithm 1 is an improved algorithm for CP-ABE encryption. And Algorithm 2 is the corresponding decryption algorithm. In Algorithm 1, calculating one Tate pair needs time $O(|p|)$, outputting ciphertext CK needs to calculate two Tate pairs, so the time complexity of the whole algorithm is $O(2|p|)$. While the CP-ABE encryption algorithm needs to calculate four Tate pairs with a time complexity $O(4|p|)$.

Algorithm 2. $Decrypt(CK, SK)$.

- Input: ciphertext $CT = T, C, C_y$, user privacy key $SK = \forall j \in S : D_j$ and user attribute set S.
- Output: Sharing key K.

Step 1. If node x is a leaf node, executing $DecryptNode(CT, SK, x)$ to get $e(D_i, C_x) = e(g^{\frac{a}{H(i)}}, g^{H(att(x)q_x(0))}) = e(g,g)^{aq_x(0)}$;

Step 2. If node x is a non-leaf node, it calls $DecryptNode(CT, SK, y)$ for each child node y to obtain $e(g,g)^{aq_x(0)}$;

Step 3. Executing the formula $F_X = \prod_{y\in s_x} e(g,g)^{aq_x(i)\cdot\Delta_{i,s'_x(0)}}$ for all child nodes y and get $e(g,g)^{aq_x(0)}$;

Step 4. Repeat step 1, 2, 3, until $e(g,g)^{aq_R(0)}$ of the root node R of access tree T is obtained;

Step 5. Calculating $\frac{C}{e(g,g)^{aq_R(0)}}$ to get K;

Step 6. Output the sharing key K.

4 Experiment and Analysis

In this section, we compare our new scheme (abbreviated to AHEM) with state-of-the-art schemes PPCPA [13], FEAC [11], EABE [10]. Our experimental equipment is Intel i5-4200U 2.30GHz processor, 8G memory, 64-bit Windows 8.1 operating system.

Because the PDA has a wealth of energy resources, but physiological sensor energy resources are limited. The experiment mainly analyzes calculation overhead of data encryption operation and transmission data communication costs for the physiological sensor. Since the size of information is directly related to the communication overhead, we begin to analyze the information size.

4.1 Information Size

Any physiological sensors SN_i needs to upload two types of data for PDA: one is the encryption and signature data $E_K(M, hash(M), SIG_{K_{priv}}(K)$, and the other is ciphertext data $CK = T, C, C_y$. Therefore, the information size of our new scheme is:

$$T^1 = |ID_{SN_i}| + |ID_{PDA}| + |T| + |C| + |C_y| + |SIG_{K_{priv}}(k)| + |E_K(M, hash(M))|.$$

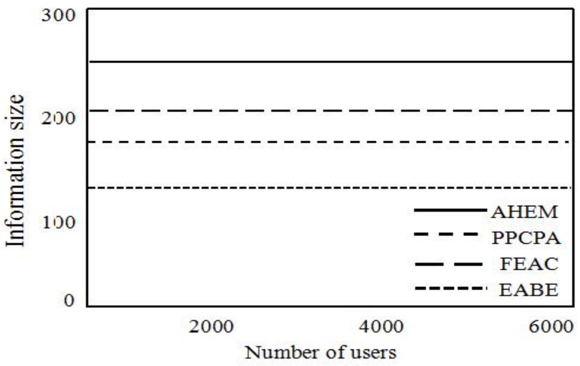

Figure 1: Relation between information size and users' number

At the same time, we also give the information size of PPCPA, FEAC, EABE as shown in following equations respectively.

$$\begin{aligned} T^2 &= |ID_{SN_i}| + |ID_{PDA}| + |T| + |\tilde{C}| + |C_y| \\ &\quad + |C| + |AES(K, M)| + |\sigma| + |C'|. \\ T^3 &= |ID_{SN_i}| + |ID_{PDA}| + |2T| + |\tilde{C}| + |C_y| \\ &\quad + |2C| + |AES(K, M)| + |3C'|. \\ T^4 &= |ID_{SN_i}| + |ID_{PDA}| + |3T| + |2C_y| \\ &\quad + |5C| + |AES(K, M)| + |4\sigma| + |C'|. \end{aligned}$$

We set the ID_{SN_i} of physiological sensor SN_i and the ID_{PDA} of the PDA is 1byte, the size of T is 4bytes, size of $SIG_{K_{priv}}(K)$ is 40bytes, the size of $E_K(M, hash(M)$ is 16bytes. C and C_y are variable. In our scheme, the bilinear map e uses Tate pairs. The elliptic curve E is defined as $q-order$ 160-bit prime in F_p, groups G_1 and G_2. According to the literature [7,15], in order to provide a security level equivalent to 1024-bit RSA, if the group G_2 is defined as a $q-order$ subgroup in the multiplicative group of finite fields $F^*_{p^2}$, then p is a 64bytes prime. In addition, in the finite field $F^*_{p^3}$, p is 42.5bytes and p is 20bytes in the finite field $F^*_{p^6}$. For uniform comparison, we take $|p|$=64bytes. Therefore, the information size of PPCPA is 175bytes, the information size of FEAC is 201 bytes, the information size of the EABE is 131bytes (m=64). Figure 1 shows the relationship between the size of the information and the number N of users. From the curve we can see that the new scheme's information size and the number of users are independent of each other.

4.2 Communication Overhead

Assuming that during time period, the physiological sensor SN_i transforms n data to PDA. Total communication costs AHEM, PPCPA, FEAC, EABE are shown as following equations respectively.

$$\begin{aligned} T^1_n &= |ID_{SN_i}| + |ID_{PDA}| + |T| + |C| + |C_y| \\ &\quad + |SIG_{K_{priv}}(k)| + |E_K(M, hash(M))| \times n. \end{aligned}$$

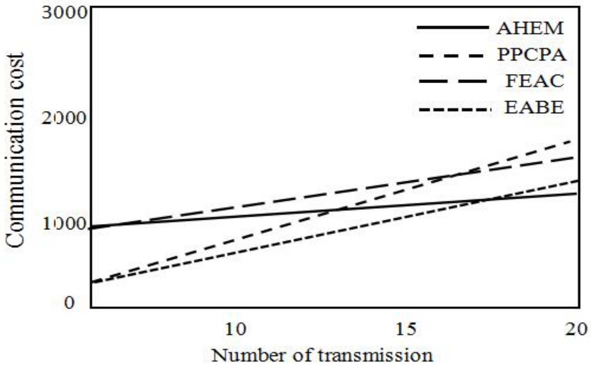

Figure 2: Relation between communication cost and data transmission number

$$
\begin{aligned}
T_n^2 &= |ID_{SN_i}| + |ID_{PDA}| + |T| + |\tilde{C}| + |C_y| \\
&\quad +|C| + |AES(K,M)| \times n + |\sigma| + |C'|. \\
T_n^3 &= |ID_{SN_i}| + |ID_{PDA}| + |2T| + |C| \\
&\quad +|SIG_{K_{priv}}(k)| \times n + |E_K(M, hash(M))|. \\
T_n^4 &= |ID_{SN_i}| + |ID_{PDA}| + |\tilde{C}| + |C_y| \times n \\
&\quad +|1.5C| + |AES(K,M)| + |\sigma| + |2C'|.
\end{aligned}
$$

Figure 2 shows the relationship between the communication overhead and the number of data transmissions n. It can be seen from the curve that when the number of n increases, the communication overhead of all schemes increases. The growth rate of the EABE is the largest, and the growth rate of AHEM is the smallest.

4.3 Calculation Overhead

At the 64-bit Intel i5-4200U processor with running speed 2.30GHz, calculating one Tate needs about 61.03ms. In addition, the certification of a 160-bit ECDSA signature takes about 18.48ms. Note that we omit the computational overhead of hash operation and symmetric encryption operation. In that they have a significantly lower computational cost. Assuming that the physiological sensor SN_i transmits n data to PDA during the time period. In the EABE scheme, SN_i encrypts the data, and the computational overhead is three Tate pairs. So the total computational cost is $3 \times 61.03 = 181.09ms$. In the FEAC scheme, the calculation is five Tate pairs with a total computational overhead $5 \times 61.03 = 305.15ms$. The computational overhead of PPCPA scheme is mainly generated by two ECDSA signatures, with a total computational cost 36.96nms. The computational overhead of the AHEM scheme is generated primarily by an ECDSA signature with a total computational effort of 18.48nms.

Figure 3 shows the relationship between the computational cost and the number of data transmissions n. From the curve, our scheme has a low computational overhead and is not affected by the number of data transfers, since it only needs to perform a Tate operation on n data transmission. But to illustrate the performance of proposed algorithm, we also perform an experiment with low cost equipment. Results are as Figure 4.

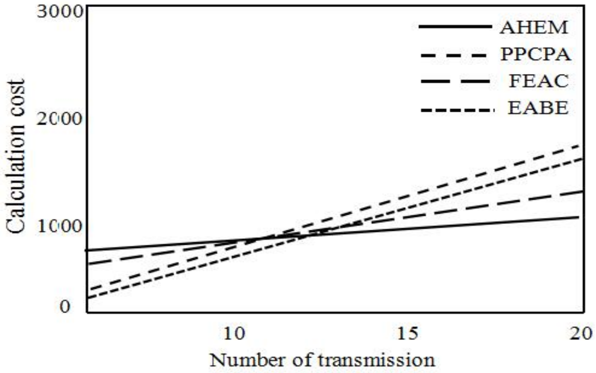

Figure 3: Relation between calculation cost and data transmission number with high cost equipment

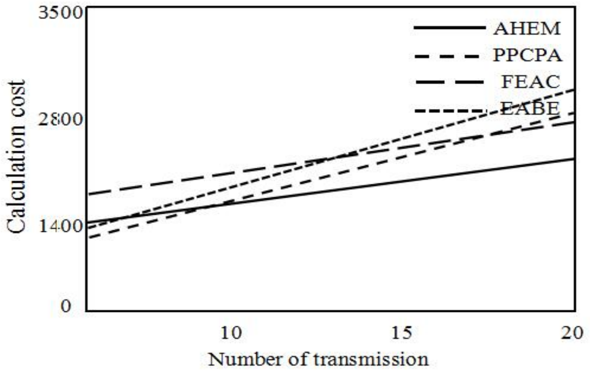

Figure 4: Relation between calculation cost and data transmission number with low cost equipment

5 Conclusion

In this paper, a secure and efficient ciphertext encryption scheme based on attribute and support strategy dynamic update via hybrid encryption method is proposed. The new scheme encrypts the data through asymmetric encryption technology, which guarantees the security of the stored data, and associates the user key with a set of attributes. Associating the sharing key with a set of attribute discrimination criteria, the user can decrypt the ciphertext only if the attribute discrimination condition is satisfied avoiding the cost of distributing the sharing key for each user. Finally, experiments for the proposed scheme, the results show that our new scheme has very low computational and communication overhead. In the future work, we will carry out the proposed program, so as to further improve the effectiveness of privacy protection.

6 Acknowledgments

This work was supported by the Independent Innovation Fund for Young Teachers (2572015BB24); This work was also supported by the National Natural Science Foundation of China (31370565) and School-level Project (JMSUJCMS2016-005).

References

[1] J. S. Chen, C. Y. Yang, M. S. Hwang, "The capacity analysis in the secure cooperative communication system," *International Journal of Network Security*, vol. 19, no. 6, pp. 863-869, Nov. 2017.

[2] 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.

[3] P. Hu, H. Y. Gao, "A key-policy attribute-based encryption scheme for general circuit from bilinear maps," *International Journal of Network Security*, vol. 19, no. 5, pp. 704-710, 2017.

[4] M. S. Hwang, S. M. Chen and C. Y. Liu, "Digital signature with message recovery based on factoring and discrete logarithm," *IETE Journal of Research*, vol. 62, no. 3, pp. 415-423, Sep. 2016.

[5] H. Li, S. L. Yin, C. Zhao and L. Teng, "A proxy re-encryption scheme based on elliptic curve group," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 8, no. 1, pp. 218-227, Jan. 2017.

[6] L. H. Li, S. F. Tzeng, M. S. Hwang, "Generalization of proxy signature based on discrete logarithms", *Computers & Security*, vol. 22, no. 3, pp. 245–255, 2003.

[7] H. J. Liu, Y. H. Chen, H. Tian, T. Wang and Y. Q. Cai, " Attribute-based secure and efficient ciphertext encryption scheme," *Journal of Chinese Computer Systems*, vol. 38, no. 8, pp. 1708-1711, 2017.

[8] J. Liu, S. L. Yin, H. Li and L. Teng, "A density-based clustering method for k-anonymity privacy protection," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 8, no. 1, pp. 12-18, Jan. 2017.

[9] E. J. L. Lu, M. S. Hwang, and C. J. Huang, "A new proxy signature scheme with revocation", *Applied Mathematics and Computation*, vol. 161, no. 3, PP. 799-806, Feb. 2005.

[10] Y. Mao, Y. Zhang, M. R. Chen, Y. Li and Y. Zhan, "Efficient attribute-based encryption schemes for secure communications in cyber defense," *Intelligent Automation & Soft Computing*, vol. 22, no. 3, pp. 1-7, 2016.

[11] R. Y. Sreenivasa, R. Dutta, "Fully secure bandwidth-efficient anonymous ciphertext-policy attribute-based encryption," *Security & Communication Networks*, vol. 8, no. 18, pp. 4157-4176, 2016.

[12] W. Susil, Y. Jiang, Y. Mu and F. Guo, "Ciphertext-policy attribute-based encryption supporting access policy update and its extension with preserved attributes," *International Journal of Information Security*, pp. 1-16, 2017.

[13] G. S. Tamizharasi, B. Balamurugan, H. A. Gaffar, "Privacy preserving ciphertext policy attribute based encryption scheme with efficient and constant ciphertextsize," in *International Conference on Inventive Computation Technologies*, pp. 1-5, 2017.

[14] L. Teng, H. Li, "A high-efficiency discrete logarithm-based multi-proxy blind signature scheme," *International Journal of Network Security*, v0. 20, no. 5, 2018.

[15] C. H. Wei, M. S. Hwang, A. Y. H. Chin, "A secure privacy and authentication protocol for passive RFID tags," *International Journal of Mobile Communications*, vol. 15, no. 3, pp. 266-277, 2017.

[16] N. I. Wu, M. S. Hwang, "Development of a data hiding scheme based on combination theory for lowering the visual noise in binary images," *Displays*, vol. 49, pp. 116-123, Sep. 2017.

[17] S. L. Yin, L. Teng and J. Liu, "Distributed searchable asymmetric encryption," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 4, no. 3, pp. 684-694, 2016.

[18] Z. Zhou, D. Huang and Z. Wang, "Efficient privacy-preserving ciphertext-policy attribute based encryption and broadcast encryption," *IEEE Transactions on Computers*, vol. 64, no. 1, pp. 126-138, 2016.

Biography

Yanhua Wang received the B.Eng. degree from Northeast Forestry University, Harbin, 150040 China in 2013. Now, she is working for a doctor degree in College of Information and Computer Engineering, Northeast Forestry University. Her research interests include Multimedia Security, Network Security, artificial intelligence and Data Mining.

Yaqiu Liu obtained his Ph.D. degree in Information Science and Engineering from Northeast Forestry University. Yaqiu Liu is a full professor of the College of Information and Computer Engineering at Northeast Forestry University. He is also a doctor's supervisor. He has research interests in wireless networks, cloud computing, social networks and quantum cryptography. Prof. Liu had published more than 50 papers on the above research fields.

Kun Wang received the B.Eng. degree from Jiamusi University, Jiamusi 154007 China in 2010. Now, she is working for a doctor degree in College of Information and Computer Engineering, Northeast Forestry University. Her research interests include Multimedia Security, Network Security and Data Mining.