RESEARCH ARTICLE OPEN ACCESS

Attribute-based Encryption with Decryption and Revocation Outsource in Cloud Environment

Lixian Liu[1], Guowei Wu[2]
[1,2](Department of computer science , Jinan university,China)

----------**********----------

Abstract:

In order to improve the efficiency of attribute based encryption, according to the work of Li etal. [17] , a new technique is adopted to implement the revocation of attribute-based encryption. Based on the work of Green etal.[16], a ciphertext attribute based encryption (CP-ABE) scheme with outsourced revocation and decryption simultaneously is proposed. After outsourcing complex computation in decryption and key update operation to decryption cloud service provider (D-CSP) and key update cloud service provider (KU-CSP), the proposed construction not only decreases the decryption computation of users, but also not requires the attribute authority to online all the time to provide key update operation for users, which greatly relieves the burden of users and attribute authority.

Keywords — **Attribute-based encryption, CP-ABE, revocation outsource, private key update, decryption outsource.**

----------**********----------

I. INTRODUCTION

With the development of cloud, more and more users store their data in cloud service system, it is widely concerned how to provide an efficient way to control the access ability of data. In traditional encryption system ,such as symmetric and public encryption system, only users who are the designated receiver can recover the encrypted data. However, in most case, the one to one encryption is not practical in some situation, e.g., for the data in the cloud server, many users is available to access. In order to overcome the drawbacks in traditional encryption, Sahai and Waters[1] proposed the notion of attribute based encryption. Two kinds of CP-ABE are be classified: ciphertext-policy attribute based encryption (CP-ABE) and key-policy attribute based encryption (KP-ABE). In CP-ABE[2], the ciphertext is associated with an access structure, and users private key is associated with an attributes set, a user can decrypt the ciphertext if only if that its attributes set associate with private key satisfy the access structure associated with ciphertext. While in KP-ABE, the situation is reversed, the ciphertext is associated with an attributes set, and users private key is associate with an access structure, a user can decrypt if and only if that the access structure associated with user private key is satisfied by the attributes set associated with ciphertext.

Though ABE brings great convenience for data access that provide data store in untrusted server fine-grained access control, it exists the efficiency problem need to be solved. In ABE schemes, the decryption time is linearly grows with the complexity of access structure, and when the private key of a user is leaked ,there should exist a efficient way to revoke the corresponding user. However ,it will make attribute authority work inefficiently if the revocation operation is done by it, especially when a large number of users exist in the system, the attribute authority not only needs to generation private key for all users, but also needs to online all the time so that to update private key for them.

In order to reduce the cost at local, it is desirable to outsource complex computation to a third party, and this research is focused by schorlars. Atallah et.al.[3] firstly considered how to outsource complex scientific computations such as matrix

multiplication and quadrature, however, technique they used can not protect the privacy of data. Atallah and Li consider the problem of computing the edit distance between two sequences and presented an efficient protocol to securely outsource sequence comparison with two servers. Following their work, Benjamin and Atallah [5] solve the problem of secure outsourcing for widely applicable linear algebraic computations, but their method requires expensive computations such as homomorphic encryption, then, Atallah and Frikken[6] adopted a new technique to improve their work. Wang et al.[7] consider the efficient mechanisms for secure outsourcing of linear programming computation. Despite the large mount of work done by researchers, these techniques is not suitable to be directly used to attribute based encryption. In order to reduce the computation costs of exponentiation, previous work utilize server-aided technique[8][9][10], and thus speed up the computation of exponentiation, which will need the ABE system inefficient if these technique are directly used. Another technique might be used is the general outsource method or proxy computation[11][12][13][14][15], and all these techniques are based on homomorphic encryption or interaction prove system. Green et al. introduced the notion of outsourced ABE, which has highly reduced the computation cost of user decryption. However, they just consider the outsource of decryption computation, in a revocable ABE system, the attribute authority not only need to generate private key for all users in the system, but also required to be online all the time so that to update key for users periodically , which brings high burden to him.

In this paper, we propose a new method to provide revocation to ABE system according to Li et al.[17], and present a concrete construction , based on Green et al.[16], that including decryption outsource and revocation outsource. For a resource limited device, a user can efficiently complete the decryption operation, and the attribute authority is allowed to update user private key for users off-line , thus, our construction have greatly relieve the burden of users and attribute authority. The performance at last shows that our construction is indeed efficient.

II. PRELIMINARIES

A. bilinear group

Let $\mathcal{G}$ be an algorithm that takes as input a security parameter λ and outputs a tuple $(P, \mathbb{G}, \mathbb{G}_T, e)$, where $\mathbb{G}$ and $\mathbb{G}_T$ are multiplicative cyclic groups of prime order P ,and $e: \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a map satisfy the following properties:

1) ***Bilinearity:*** $e(g^a, h^b) = e(g, h)^{ab}$ for all $g, h \in \mathbb{G}$ and $a, b \in \mathbb{z}_p^*$.

2) ***Nondegeneracy:*** $(g, h) \neq 1_{\mathbb{G}_T}$, where $g, h \neq 1_{\mathbb{G}}$;

3) ***Computable :*** efficient computability for any input pair.We refer to the tuple $(p, \mathbb{G}, \mathbb{G}_T, e)$ as a bilinear group.

B. Access Structures

Definition 1 : Let $\{P_1, \dots, P_n\}$ be a set of parties, A collection $\mathbb{A} \subseteq 2^{\{P_1, \dots, P_n\}}$ is monotone if $\forall B \in \mathbb{A}$ and $B \subseteq C$, then $C \in \mathbb{A}$.

C. .Linear Secret Sharing Schemes (LSSS)

Definition 2 : (Linear Secret-Sharing Schemes (LSSS)): A secret sharing scheme Π over a set of parties P is called linear (over $\mathbb{z}_p$) if

1) The shares for each party form a vector ove $\mathbb{z}_p$.

2) There exists a matrix A with rows and n columns called the share-generating matrix for Π. For all $i = 1, \dots, \ell$, the ith row of A is labeled by a party $\rho(i)$ (ρ is a function from $\{1, \dots, \ell\}$ to P). When we consider the column vector $v = (s, r_2, \dots, r_n)$, where $s \in \mathbb{z}_p$ is the secret to be shared, and $r_2, \dots, r_n \in \mathbb{z}_p$ are randomly chosen, then Av is the vector of l shares of the secret s according to Π . The share $(Av)_i$ belongs to party $\rho(i)$.

It is shown in [33] that every linear secret-sharing scheme according to the above definition also enjoys the linear reconstruction property, defined as follows.Suppose that Π is anLSSS for the access structure $\mathbb{A}$. Let $S \in \mathbb{A}$ be any authorized set, and let $I \subset \{1, \dots, \ell\}$ be defined as $I = \{i | \rho(i) \in S\}$. Then there exists constants $\{\omega_i \in \mathbb{z}_p\}_{i \in I}$ such that, if λ_i are valid shares of any secret s according to Π, then $\sum_{i \in I} \omega_i \lambda_i = s$. Let A_i denotes the ith row of A , we

have $\sum_{i\in I}\omega_i A_i = (1,0,\dots,0)$. These constants ω_i can be found in time polynomial in the size of the share-generation matrix A [33]. Note that, for unauthorized sets, no such constants $\{\omega_i\}$ exist.

III. TECHNIQUE DISCUSSION

In this paper ,we propose a new method to provide revocation to ABE system according to Li et al.[17]. In CP-ABE, the attributes is about users, and there is an attribute that can distinguish all the users, e.g., in an university, attributes set about students is (Num, Major,Grade,Gender), and the student ID "Num" can be utilized to distinguish students, which means that student ID can be used to present the identity of them.

Our CP-ABE support outsourced revocation is based on Li et al.[17], in our construction, every users private key including two parts: one is about user attributes set, and the other is about the time T_i , when a user in the system is revoked, the key update cloud server provider just need to update this part of private key to T_{i+1}, only users obtains the updated key can decrypt.

The construction we present not only provide decryption outsource, but also provide revocation outsource, which is based on Green et al.[16], which makes the user to recover message efficiently and allows the attribute authority to update users private key locally.

For the security of our construction, it not only need to prevent the adversary obtain ciphertext to attack directly, but also need to make sure that the encrypted data is protected from D-CSP and KU-CSP. Besides, it should resistant collusion attack. Two collusion should be considered: one is a user collusion with S-CSP who want to recover message, the other is unrevoked user collusion with KU-CSP who want to generate private key for users that is revoked.

IV. CONSTRUCTION

A. System decription

In A CP-ABE system contain revocation and decryption outsource, its running process is as in Fig.1. Firstly, the encryptor runs the algorithm $CT = Encrypt(PK, M, T_j, (A, \rho))$ to encrypt data and stores in Storage Cloud Service Provider (S-CSP). When a user with attributes set S request private key to the attribute authority, it runs the key generation algorithm $KeyGen(PK, MSK, RL, TL, S)$ to generate private key $SK_S = (SK_1, SK_{T_j})$ and the corresponding update key UK_{ID} , then , the user runs the $KeyGen_{out}(PK, SK_S)$ algorithm to generate the transformation key TK_{out} which is send to D-CSP along with CT in T_j . Next, D-CSP runs the $Transform_{out}(PK, TK_{out}, CT)$ algorithm to get the partial decrypted ciphertext CT' ,and users who owns the retrieving key can run the algorithm $Decrypt_{out}(PK, RK, CT')$ to recover the message M.

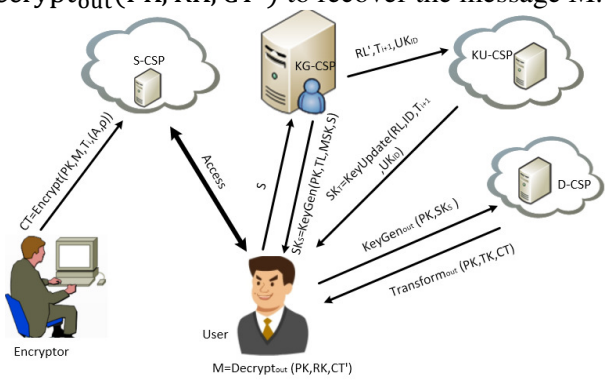

Fig. 1 System algorithm process of our scheme

The user who owns the private key can also directly runs the algorithm $Decrypt(PK, CT, SK_s)$ to recover message M. When a users who owns the attributes $ID_1, .., ID_k$ are revoked from the system, the attribute authority runs the algorithm $Revoke(RL, TL, \{ID_1, .., ID_k\})$ to add these users to revocation list, and send the new revocation list RL' and new period T_{j+1} to KU-CSP which will update private key about time for users who is not in the revocation list RL' .So, only users with private key $SK_S = (SK_1, SK_{T_{j+1}})$ can decrypt the ciphertext in the period T_{j+1} .

B. Construction

1) $\boldsymbol{Setup(\lambda, U)}$: The setup algorithm takes input the security parameter λ , an attribute universe description U, $U = \{0,1\}^*$, and choose a cyclic group $\mathbb{G}$ of prime order P, g is a generator of group $\mathbb{G}$, g is a generator of $\mathbb{G}$, and $F: \{0,1\}^* \to \mathbb{G}$ is a hash function. Then, the algorithm randomly choose α, $a \in \mathbb{Z}_p$. The attribute authority sets the

master secrete key MSK $= \alpha$, and public parameter as PK $= (g, e(g,g)^{\alpha}, g^{a}, F)$.

2) $\boldsymbol{KeyGen(PK, MSK, RL, TL, S)}$:The private key generation algorithm takes as input the public parameter PK , the master secrete key MSK , the revocation list RL , the time list TL and user attributes set S . When a user requests private key , the attribute authority first checks whether the attribute ID $\in$ S exists in the revocation list RL . If exists, it aborts, else it randomly choose $\alpha_1, t', r_{T_j} \in \mathbb{Z}_p^*$, and sets $\alpha_2 = \alpha - \alpha_1$. Then it computes $SK_S = (SK_1 = (K', L', K_x), SK_{T_j} = K'', L'', K_{T_j}))$ as follows:

$K' = g^{\alpha_1} g^{at'}$, $L' = g^{t'}$, $K_x = F(x)^{t'}$, where $x \in S$. $K'' = g^{\alpha_2} g^{ar_{T_j}}$, $L'' = g^{r_{T_j}}$, $K_{T_j} = F(T_j)^{r_{T_j}}$

Finally, it sets the update key as $UK_{ID} = \alpha_2$, and sends the UK_{ID} to KU-CSP.

3) $\boldsymbol{Encrypt\left(PK, M, T_j, (A, \rho)\right)}$: The encryption algorithm takes as input the public parameter, message M , time period T_j , LSSS access structure (A, ρ) , A is an $\ell \times n$ matrix and the function ρ assocaites each row of A to an attribute $\rho(i)$. First, it choose a random vector $\vec{v} = (s, y_2, \dots, y_n) \in \mathbb{Z}_p^n$, where s is the secrete to be shared among the shares. For $i = 1$ to ℓ , it calculates $\lambda_i = \vec{v} \cdot A_i$,where A_i is the vector corresponding to the ith row of A . Then, the algorithm randomly choose $r_1, \dots r_\ell, r'_1, \dots, r'_\ell \in \mathbb{Z}_p^{2\ell}$, and the ciphertext $CT = (C, C', C_i, D_i, C'_i, D'_i)$ is computed as follow:

$$C = M \cdot e(g,g)^{\alpha s}, \quad C' = g^{s}$$
$$C_i = g^{a\lambda_i} F(\rho(i))^{-r_i}, \quad D_i = g^{r_i}$$
$$C'_i = g^{a\lambda_i} F(T_j)^{-r'_i}, \quad D'_i = g^{r'_i}, \quad \text{where } i \in \{1, \dots, \ell\}$$

4) $\boldsymbol{Decrypt(PK, CT, SK_s)}$: The decryption algorithm takes input the public parameter PK , the ciphertext CT in the time period T_j ,private key SK_S. If the attributes set S does not satisfy the access structure (A, ρ) , it abort. Suppose that the attributes set S satisfy the access structure (A, ρ) and let $I \subset \{1,2 \dots, \ell\}$ be defined as $I = \{i: \rho(i) \in S\}$. Then let $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ be a set of constants such that if λ_i are valid shares of any secret s according to A, then $\sum_{i \in I} \omega_i \lambda_i = s$. It first computes E:

$$\frac{e(C', K')e(C', K'')}{\prod_{i \in I} e(C_i^{\omega_i}, L') e(D_i^{\omega_i}, K_{\rho(i)})) \cdot \prod_{i \in I} e(C_i'^{\omega_i}, L'') e(D_i'^{\omega_i}, K_{T_j})}$$
$$= \frac{e(g^s, g^{\alpha_1}) e\left(g^s, g^{at'}\right) e(g^s, g^{\alpha_2}) e(g^s, g^{ar_{T_j}})}{e(g,g)^{at's} \cdot e(g,g)^{ar_{T_j}s}}$$
$$= e(g^s, g^{\alpha_1})\, e(g^s, g^{\alpha_2}) = e(g,g)^{s\alpha}$$

then , it recover the message $M = C/E$。

5) $\boldsymbol{KeyGen_{out}(PK, SK_S)}$: The transformation key generation algorithm takes as input the public parameter PK , private key SK_S . It randomly choose $z \in \mathbb{Z}_p^*$,the transformation key $TK_{out} = (TK', TL', TK_x, TK'', TL'', TK_{T_j})$ is computes as follows:

$$TK' = K'^{1/z}, \quad TL' = L'^{1/z}, \quad TK_x = K_x^{1/z}$$
$$TK'' = K''^{1/z}, \quad TL'' = L''^{1/z}, \quad TK_{T_j} = K_{T_j}^{1/z}$$

and sets the retrieving key as RK $= z$.

6) $\boldsymbol{Transform_{out}(PK, TK_{out}, CT)}$: The transformation algorithm takes as input the public parameter PK , the transformation key TK_{out} and the ciphertext CT .Suppose that the attributes set S satisfy the access structure (A, ρ) and let $I \subset \{1,2 \dots, \ell\}$ be defined as $I = \{i: \rho(i) \in S\}$. Then let $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ be a set of constants such that if λ_i are valid shares of any secret s according to A, then $\sum_{i \in I} \omega_i \lambda_i = s$. It first computes $\tilde{C}$:

$$\frac{e(C', TK')e(C', TK'')}{\prod_{i \in I} e(C_i^{\omega_i}, TL') e(D_i^{\omega_i}, TK_{\rho(i)})) \cdot \prod_{i \in I} e(C_i'^{\omega_i}, TL'') e(D_i'^{\omega_i}, TK_{T_j})}$$
$$= \frac{e(g^s, g^{\alpha_1})^{\frac{1}{z}} e\left(g^s, g^{at'}\right)^{\frac{1}{z}} e(g^s, g^{\alpha_2})^{\frac{1}{z}} e\left(g^s, g^{ar_{T_j}}\right)^{\frac{1}{z}}}{e(g,g)^{\frac{at's}{z}} \cdot e(g,g)^{\frac{ar_{T_j}s}{z}}}$$
$$= e(g^s, g^{\alpha_1})^{1/z}\, e(g^s, g^{\alpha_2})^{1/z} = e(g,g)^{s\alpha/z}$$

then, output the transferred ciphertext $CT' = (\tilde{C}, \hat{C} = C)$.

7) $\boldsymbol{Decrypt_{out}(PK, RK, CT')}$: Decryption algorithm takes as input the public parameter PK ,the retrieving key RK, the transferred ciphertext CT′, and computes $\hat{C} / \tilde{C}^{z} = M$.

8) $\boldsymbol{Revoke(RL, TL, \{ID_1, .., ID_k\})}$: If users with attributes $ID_1, .., ID_k$ are revokes during the

period T_j , the attribute authority update the revocation list as $RL' = RL \cup \{ID_1, .., ID_k\}$, and update the time list as TL that associated to the period T_{j+1}. Finally, it sends RL' and T_{j+1} to KU-CSP.

9) $\boldsymbol{KeyUpdate(PK, RL, ID, T_{j+1}, UK_{ID})}$:When a user with attribute ID request private key, the attribute authority first checks whether ID exists in the revocation list RL, if exists, it will not update private key for him, else KU-CSP search the item ID, UK_{ID}) in the table UL, and randomly choose $r_{T_{j+1}} \in \mathbb{Z}_p$. It computes private key $SK_{T_{j+1}} = (K'', L'', K_{T_{j+1}})$ in period T_{j+1} as follows :

$$K'' = g^{\alpha_2} g^{ar_{T_{j+1}}}, \quad L'' = g^{r_{T_{j+1}}},$$
$$K_{T_{j+1}} = F(T_{j+1})^{r_{T_{j+1}}}$$

It is easy to verify that our construction satisfy the correctness.

V. SECURITY ANALYSIS

A. Direct attack

Since the decryption requires the user attributes satisfy the access structure, so that it to find the constans $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ such that $\sum_{i \in I} \omega_i \lambda_i = s$, and recover the session key $e(g,g)^{\alpha s}$. When a adversary with attribute S' obbtain the CT , it can not recover the message M since $S' \notin A$.

Besides, we should make sure that the D-CSP can not recover the message through TK_{out} or update key. This can be guarantee since the recover key RK is private, and KU-CSP can only generate private key about time by the update key, so, it can not generrate the final private key used to decrypt which need the private must secrete key to generate.

B. Collusion attack

Two kinds of attacks may be exist in our scheme:

1) Collusion of user and D-CSP

Our scheme resistant this kinds of attack by randomly choose the recover key, so that a user who get the trasferred ciphertxt from D-CSP can just recover the corresponding message by his retrieving key and can not recover message of other users.

2) Collusion of unrevoked user and KU-CSP

In out shceme, the master secrete key α used to generate private key for user U_1 is splited randomly , e.g., the master secrete key α is splited to α_1, α_2 , where $\alpha = \alpha_1 + \alpha_2$, and conpute the private key about attribtues and private key about time throug α_1, α_2 respectivily. The the master secrete key α may be splited as α_1', α_2', where$\alpha = \alpha_1' + \alpha_2'$, when generate private key for U_2 , and conpute the private key about attribtues and private key about time throug α_1, α_2 respectivily. Consider the case that U_2 is revoked and, KU-CSP want to generate private key by collusion with the unrevoked user U_1 , thus it has the two parts of private key conputed through α_1 , α_2' . But this private key can not used to decrypt due to less of complement master secrete key and can not recover the session key.

VI. PERFORMENCE

In order to assess the performance of our outsourced CPABE scheme presented above, we implement our scheme in software based on Charm [34] using a 224-bit MNT elliptic curve from the Stanford Pairing-Based Crypto library [35]. Although we presented our scheme in the above in the symmetric bilinear group setting, the MNT curve in Charm requires that the scheme be implemented in asymmetric groups with a pairing of the form $(\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T)$. Therefore, we translate our scheme to the asymmetric setting in the implementation.

All our performance measurements are conducted on two dedicated hardware platforms: a 3.20 GHz Intel dual core platform with 4 GB of RAM running Linux Kernel 3.13.0, and a 1.6 GHz ARM-based MEIZU III with 2.0 GB of RAM running Android 4.2.

Experimental Setup:In a CP-ABE scheme, the decrption time is depends on the complexity of access structure. In oreder to implement the our scheme, we genenrate an access structure $(A_1 \text{ AND } A_2 \text{ AND } \ldots \text{AND } A_N)$, where A_i is an atribtue. besides, we also generate 100 kinds of access structure with N increase from 1to 100. In In each case we generate a corresponding decryption key that contains the N attributes for decryption.

Disscuss: The performance of our scheme is showed below. In standard CP-ABE, the decryption

time grows with the complexty of access structure linearly. With the grow of attributes number, the decryption time linearly. When there are 100 attributes, the decryption time is 5sec and 35sec on Intel and ARM (Fig.2). The time of outsourced decryption linearly grows with the complexty of access structure(Fig.3), and the time of local decryption is constant, which just needs 0.03sec when the attributes number is also 100(Fig.4).The last picature shows the update time for users in the system , its computation time is constant (Fig.5), our scheme not only allows attribute authority to update key off-line but also remove some computation operation.

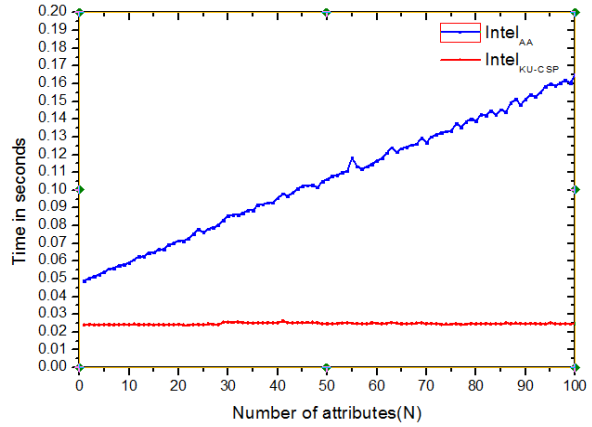

Fig.2 Standard decryption time

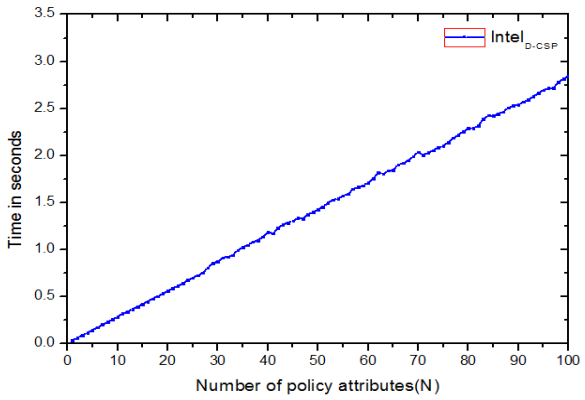

Fig.3 Outsourced Decryption Time

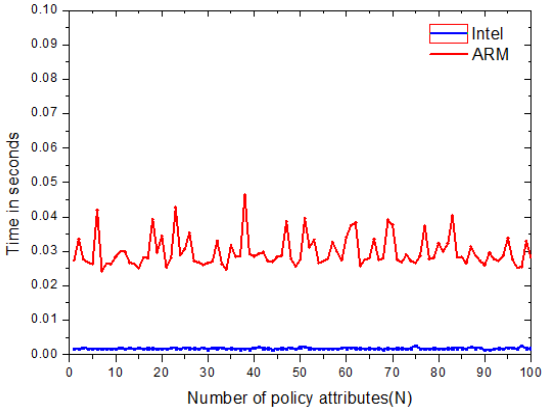

Fig.4 Local Decryption Time

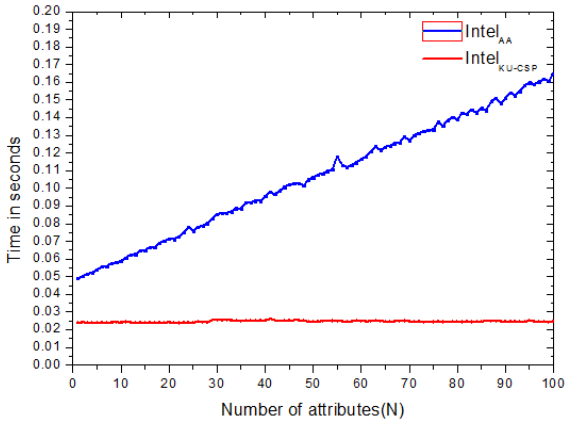

Fig.5 Private key generation and update time

VII. CONCLUSION

In this paper , we proposed a CP-ABE with revocation and decrytion outsource, which not only decrease the costs of users, but also allows the attribute authority to update key for users off-line, thus, our scheme relieve the burden of users and attribute ahthority. The performance shows that outsource the complexty computation outside can highly decrease the resource costs at local , which is efficient and practical.

REFERENCES

[1] A. Sahai and B. Waters. Fuzzy identity-based encryption[C]//Proc. of EUROCRYPT'05. Aarhus, Denmark: Springer, 2005: 457–473.

[2] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute based encryption[C]//Proc. of IEEE Symposium on Security and Privacy, California, USA: IEEE Computer Society, 2007: 321–334.

[3] M. Jakobsson and S. Wetzel. Secure server-aided signature generation[C] //Proc. of 4th International Workshop on Practice and Theory in Public Key Cryptography, Cheju Island, Korea: Springer : 2001, 383–401.

[4] M. J. Atallah and J. Li. Secure outsourcing of sequence comparisons[J]. Int. J. Inf. Sec., 2005, 4(4): 277–287.

[5] David Benjamin and Mikhail J. Atallah. Private and cheating-free outsourcing of algebraic computations[C]//Proc. of Sixth Annual Conference on Privacy, Security and Trust, Fredericton, Canada : IEEE, 2008: 240–245.

[6] Mikhail J. Atallah and Keith B. Frikken. Securely outsourcing linear algebra computations[C]//Proc. of ACM Symposium on Information, Computer and Communications Security, Beijing, China: ACM,2010: 48–59.

[7] Cong Wang, Kui Ren, and Jia Wang. Secure and practical outsourcing of linear programming in cloud computing[C]//Proc. of 30th IEEE International Conference on Computer Communications, Shanghai, China: IEEE,2011: 820–828.

[8] Kemal Bicakci and Nazife Baykal. Server assisted signatures revisited[C]//Proc. of The Cryptographers' Track at the RSA Conference 2004 , San Francisco, USA : Springer, 2004: 143–156.

[9] Markus Jakobsson and Susanne Wetzel. Secure server-aided signature generation[C]//Proc. of 4th International Workshop on Practice and Theory in Public Key Cryptography, Cheju Island, Korea : Springer, 2011: 383–401, 2001.

[10] Susan Hohenberger and Anna Lysyanskaya. How to securely outsource cryptographic computations[C]//Proc. of Theory of Cryptography, Second Theory of Cryptography Conference, Cambridge,USA : Springer ,2005, 264–282.

[11] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: interactive proofs for muggles[C]//Proc.of the 40th Annual ACM Symposium on Theory of Computing , Victoria, Canada: ACM, 2008,113–122.

[12] Craig Gentry. Fully homomorphic encryption using ideal lattices.[C]// the 41st Annual ACM Symposium on Theory of Computing, Bethesda, USA: ACM , 2009:169–178.

[13] Rosario Gennaro, Craig Gentry, and Bryan Parno. Non-interactive verifiable computing:Outsourcing computation to untrusted workers[C]//Proc.of the CRYPTO, Santa Barbara, USA: Springer ,2010: 465–482.

[14] Kai-Min Chung, Yael Tauman Kalai, Feng-Hao Liu, and Ran Raz. Memory delegation[C]//Proc.of the CRYPTO, Santa Barbara, USA: Springer ,2011, 151–168.

[15] Craig Gentry and Shai Halevi. Implementing gentry's fully-homomorphic encryption scheme[C] //Proc.of the CRYPTO , Tallinn, Estonia: Springer, 2011: 129–148.

[16] Matthew Green, Susan Hohenberger, and Brent Waters. Outsourcing the decryption of ABE ciphertexts[C]// Proc. of the 20th USENIX Security Symposium, San Francisco, USA: USENIX Association, 2011.

[17] J. Li, J. Li, X. Chen, C. Jia, and W. Lou. Identity-based encryption with outsourced revocation in cloud computing[J]. IEEE Trans. Computers, 2015, 64(2): 425–437.

[18] Joseph A. Akinyele, Christina Garman, Ian Miers, Matthew W. Pagano, Michael Rushanan, Matthew Green, and Aviel D. Rubin. Charm: a framework for rapidly prototyping cryptosystems[J]. J. Cryptographic Engineering, 2013, 3(2):111–128.