Attribute-Based Encryption Scheme Based on SIFF

Tianyu Zhao, Lingbo Wei, Chi Zhang
Key Laboratory of Electromagnetic Space Information, Chinese Academy of Sciences
University of Science and Technology of China
Email: zhaoty@mail.ustc.edu.cn, lingbowei@ustc.edu.cn, chizhang@ustc.edu.cn

***Abstract*—Attribute-Based Encryption (ABE) is a public key encryption scheme that allows users to encrypt and decrypt messages based on user attributes. In this paper, we consider the problem of constructing a ciphertext-policy attribute-based encryption (CP-ABE) scheme in a setting where the attributes distributor is also the owner of messages that are to be encrypted and shared. The CP-ABE scheme we propose bases on the Sibling Intractable Function Family (SIFF) scheme. Compared to the existing ABE schemes, the decryption of our scheme in this setting is quite fast and the ciphertext size is rather small. Our ABE system also provides a high degree of compatibility with the messages that are already encrypted when our system is set up, namely, encrypted messages can be used directly in our scheme without being re-encrypted. We compare the efficiency of our scheme with Bethencourt's work in this paper.**

I. Introduction

Attribute-Based Encryption was first introduced by Sahai and Waters in a landmark work [1] as a more expressive form of encryption where one can encrypt according to the attributes description for receivers instead of the exact identities of all the people who should be able to decrypt the ciphertext. ABE comes in two flavors called Key-Policy ABE (KP-ABE) [2] and Ciphertext-Policy ABE (CP-ABE) [3].

In the KP-ABE schemes [1], [2], [4], [5], ciphertexts and users' secret keys are associated with sets of attributes and access structures respectively. If the attributes that are associated with a ciphertext satisfy the access structure of one user, the user is able to decrypt the ciphertext. A real-world example application of KP-ABE is pay-TV system, in which a user can get his package policy, namely his secret key, to enjoy the TV program whose attributes satisfy the user's key. In KP-ABE, users should maintain a secret key, i.e., an access structure, for each ciphertext.

In the CP-ABE [3], [6], [7], the roles of attributes and access policies are swapped from the KP-ABE described above. Attributes are associated with users' keys and the access structures are associated with ciphertexts. If the attributes of one user satisfy the access structure of a ciphertext, the user can decrypt the ciphertext. In CP-ABE, users just manage their attributes instead of the different keys for different ciphertexts in KP-ABE, which is very convenient for users.

We focus on CP-ABE in this work. We construct a CP-ABE scheme where the attributes distributor is also the owner of messages that are to be encrypted. In this setting, there is only one party who distributes attributes to users and shares messages with users after setting up the system. We would like to claim that under most circumstances, a lot of organizations only care about the attributes distributed by themselves. Compared to a bank employee, a university may be more willing to deal with things from students or professors. For example, if a university wants to share an encrypted message with all the professors from one particular department, professors from other departments should not get the message, neither should professors from other universities nor a bank employee. Many companies offer services to all kinds of consumers whose attributes come from a wide variety of ways. During the services, the companies may focus more on the attributes they distributed to the consumers compared with other attributes. Consider the following example: an online game company wants to give some gift packages to some players who possess some particular attributes. The attributes of the players may be some prominent improvements in the levels, strength, constitution or dexterity of the roles in the game. The online game company may not care too much about whether a player is a professor or a bank employee in real life. The company pays more attention to the achievements that a player achieves in the game than in the real life.

Many companies distribute attributes to their customers. A cloud storage service company may distribute attributes, such as storage space, download and upload speed, to its customers. Taking horizontal competition into consideration, a company may launch promotions a lot according to some special customers' attributes that distributed by itself other than attributes from other companies or from a user's real life. The cloud storage service company may give some rewarding storage space to a user for his purchase of large storage space. However, the company may not give the same reward to one user when he graduates in his real life. The cloud storage service company will set up a new ABE system by itself for security purpose instead of using existing ABE systems. In that way, all the public parameters will be chosen by itself which leads to the setting we mentioned above: (1) each company has its own ABE systems; (2) the attributes distributor is also the owner of the message that are to be encrypted.

Traditionally, to use the models for ABE, we usually have assumed that a user can receive his attributes from many organizations and show them to others. However, in the examples above, the attributes and the messages to be encrypted are from the same entity, e.g., the entity both distributes attributes and shares messages. In these scenarios, we can take advantage of this situation to exchange for faster decryption and smaller ciphertext size.

978-1-4799-6664-6/16/$31.00 ©2016 IEEE

A. Our Contribution

In consideration of the simplification of the settings that all existing ABE are built upon, we propose a new large universe CP-ABE scheme with the restriction that the attributes distributor and the message encrypter is a same party. In our setting, the attributes distributor is also the owner of the messages to be encrypted, we denote the party as Authority. The parties who receive attributes are users.

Compared to all the existing ABE schemes, the most significant contribution of our scheme is that the decryption algorithm is very fast. It only takes a user one pairing to find out the encryption key to decrypt a ciphertext so that we encourage the organizations, who use our scheme, to use a symmetric encryption algorithm to encrypt the shared messages. In this way, the decryption process in our scheme will be very fast. Our scheme also provides a high degree of compatibility to existing encrypted messages because we care more about the encryption keys than the encryption scheme that used in our encryption process. Any encryption algorithm can be used in our scheme so that all the already encrypted messages can be delivered to users without being re-encrypted. If the encryption algorithm we use in our scheme is a symmetrical encryption algorithm, the ciphertext size is nearly the plaintext size which is quite a small size compared to other existing ABE schemes. There is no restriction on the number of attributes in our scheme and Authority can add any attributes into the scheme as long as they have a united format. The access structure in our system is a propositional formula which is quite simple and easily to be computed. To change an access structure of a ciphertext, there is only a little simple computation needed.

B. Construction

We first introduce some related works in section II. The preliminaries will be described in section III. Then we propose our scheme and give a description of our scheme in section IV and section V respectively. After that, we discuss some efficiency and tradeoff issues in section VI. Security and performance analysis are presented in section VII. We conclude our work in section VIII.

II. Related Works

ABE was first introduced by Sahai and Waters [1] and the scheme was also called Fuzzy Identity-Based Encryption (FIBE). The access structure in FIBE was limited to expressing threshold access policies. After that, Pirretti et al. [5] developed and evaluated the Sahai-Waters system. Subsequently, Goyal et al. formulated two forms of ABE: KP-ABE and CP-ABE.

Goyal et al. [2] proposed the first KP-ABE scheme which greatly increased the expressibility of the access structures. The access structures of users' private keys consist of **AND**, **OR**, or threshold gates. However, the access structures should be monotonic. Ostrovsky et al. [4] proposed a KP-ABE scheme which allows users' private keys to be expressed in terms of any access formula over attributes by adding **NOT** to existing access structures. Hohenberger and Waters [8] proposed an KP-ABE system with fast decryption. The decryption scheme in Hohenberger's system took only 2 pairings with the tradeoff that the private keys of users' increased. Attrapadung et al. [9], [10] studied the problem of constant-size ciphertext problem.

Bethencourt et al. [3] proposed the first CP-ABE scheme. The access structure in [3] was an access tree which was also a more general expression of **AND** gate or **OR** gate. Kapadia et al. [11] proposed a CP-ABE scheme whose ciphertext policy were hidden. However, their scheme was not collusion-resistant and needed an online semi-trusted server. Nishide et al. solved the problems above by introducing [12]. In Nishide's scheme, even the legitimate decryptor can not get the information about the ciphertext policy more than the fact that he can decrypt the ciphertext. The ABE systems we mentioned above were only proven to be selectively secure, Lewko et al. [13] proposed a fully secure ABE scheme and detailed a CP-ABE scheme. After that, many literatures [14], [15], [16], [17] researched new proof techniques to achieve adaptive and full security.

Melissa Chase [18] proposed a multi-authority KP-ABE scheme. In [18], multiple authorities were allowed to monitor attributes and distribute secret keys. A user can decrypt a ciphertext only if he has more than d_k attributes from authority k. The access structure in [18] was more like a form of multiple threshold gates. Chase improved his work in [19] by removing the trusted central authority which made his scheme more usable in practice. Müller et al. [20] proposed a multi-authority CP-ABE scheme. In Müller's scheme, users only held the attributes he got from multiple authorities. Müller's scheme was very convenient for users to manage their keys. Lewko and Waters [21] then proposed a Decentralizing Attribute-Based Encryption system. In Lewko's system, any party can be an authority and there was not any central authority. A party who acts as an authority can simply publish public keys and distribute attributes to relevant users. Lin et al. [22] also considered the similar problem.

Attrapadung and Imai [23] proposed a new cryptosystem called Broadcast ABE for both CP-ABE and KP-ABE. Broadcast ABE can be used to construct ABE system with direct revocation mechanism without affecting any non-revoked users. After that, Sahai et al. [24] proposed a fully secure revocable ABE system by modifying an ABE system due to Lewko et al. [13]. Lewko et al. [25] obtained "unbounded" ABE scheme which has a large attribute universe and no bound on the size of attribute sets used for encryption. Hohenberger and Waters [26] considered the online and offline situation when constructing an attribute-based encryption systems.

III. Preliminaries

Before we set about describing our scheme, we first introduce some preliminaries.

A. Basic Assumptions

Assumption 1 (Discrete Logarithm Assumption): $\mathbb{G}$ is a cyclic group and its order is p. g is a generator of $\mathbb{G}$. Given $h \in \mathbb{G}$, it is difficult to compute $r \in \mathbb{Z}_n$ such that $h = g^r$.

B. Bilinear Maps

Let $\mathbb{G}$ and $\mathbb{G}_t$ be two groups of prime order p and g is a generator of $\mathbb{G}$. We define $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_t$ as a bilinear map between the $\mathbb{G}$ and $\mathbb{G}_t$. We can figure out that e should have the following features:

1) *Bilinear*: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$, $e(u^a, v^b) = e(u, v)^{ab}$.
2) *Non-degenerate*: $e(g, g) \neq 1$.
3) *Computable*: for any $u, v \in \mathbb{G}$, there is an efficient algorithm to compute $e(u, v)$.

C. The Sibling Intractable Function Family (SIFF)

Zheng, Hardjono, and Pieprzyk [27] introduced the notion of sibling intractable function family (SIFF). One of the applications of SIFF is described as "How to manage 1 million passwords for shared devices". We assume that there is a system who has n users. The system wants to share a message m with all the users after he encrypts the message using secret key y. Every user has a different secret key, x_k for user k, for this system. So the system chooses a polynomial:

$$f(x) = x^n + a_1 x^{n-1} + ... + a_i x^{x-i} + ... + a_n.$$

For equation $f(x) = y$, the system uses the n keys of the n users as the roots to work out all the coefficients which we donate as $A = \{a_1, ..., a_n\}$. Then the system can publish A and the ciphertext for m. If a person is one of the n users of the system, he can use $f(x_i)$ to compute the secret key y which can be used to decrypt the ciphertext.

IV. Overall Structure of Our ABE Scheme

A. A Sketch of Our ABE Scheme

In this section, we give a sketch of our attribute-based encryption scheme. The overall structure is illustrated in Fig.1. Our scheme is composed of Authority, storage server and users. Authority distributes attributes to users and shares message with users. If Authority wants to share a message with some people who has some particular attributes, he does not need to know all the identities of the target people. He describes the encrypted message in terms of descriptive attributes and sends the ciphertext and the access structure to a storage server. Uers who has the required attributes can download the encrypted message from the storage server and decrypt the ciphertext.

The access structure in our scheme is a propositional formula which describes the attributes needed to decrypt the ciphertext.

Our attribute-based encryption scheme consists of five fundamental algorithms: ***Setup***, ***AttributeSpace***, ***Encrypt***, ***KeyGen*** and ***Decrypt***.

Setup(κ)**.** The Setup algorithm, which is run by Authority, takes only the security parameter κ as input and outputs the public key PK and the master secret key MK for Authority.

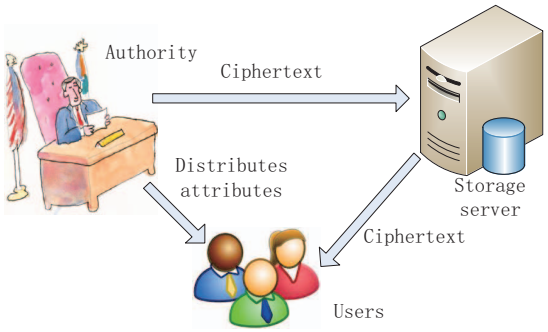

Fig. 1. Overall Structure

AttributeSpace$(attributes, MK)$**.** Authority runs this algorithm. The AttributeSpace algorithm takes attributes and MK as input and outputs the attributes space which contains all the attributes that can be used in our system.

Encrypt(M, MK)**.** The Encrypt algorithm takes the master key MK and the message M which is to be encrypted as input and it outputs the ciphertext message CM and the access propositional formula pf for CM.

KeyGen(MK, S)**.** This algorithm takes as input the master key MK and an attribute set S. It outputs the secret key SK for a user who owns the attribute set S.

Decrypt(CM, pf, SK)**.** The decrypt algorithm takes the ciphertext message CM, the access propositional formula pf and the secret key SK of users as input, it outputs the message M or false.

According the description above, we can see that Authority distributes attributes to all users including himself. We have to assume that Authority is trusted. However, in our setting, all the messages are from Authority, therefore it is not very important whether we assume Authority is trusted. What really matters is if users are honest. In our scheme, we can ensure that users can not compute attributes by themself and they can not collude.

V. Detail of Our Method

A. Setup(κ).

Let $\mathbb{G}_0$ be a bilinear group of prime order p and g is a generator of $\mathbb{G}_0$. $\mathbb{G}_1$ is also a group of order p. We donate a bilinear map $e : \mathbb{G}_0 \times \mathbb{G}_0 \to \mathbb{G}_1$. Authority generates all the parameters to set up our ABE system. Authority first chooses a random number MK as his master key. After that Authority chooses two hash functions $h(\cdot)$ and $h_2(\cdot)$. $h(\cdot)$ will be used by Authority to describes attributes and $h_2(\cdot)$ will be used by a user to decrypt a ciphertext. The public parameters are

$$(\mathbb{G}_0, \mathbb{G}_1, g, p, e, h(\cdot), h_2(\cdot)).$$

The master secret key of Authority is MK. Then Authority publishes all the public parameters and stores his secret key.

B. AttributeSpace($attributes, MK$).

Authority formulates uniform formats for every attributes that can be used in our scheme. For attributes a_1, he computes

$h(a_1||MK)$ which we donate as ha_1. If the attributes that can be used in our system are

$$(a_1, a_2, ..., a_{m'})$$

the attribute space in our system are

$$(ha_1, ha_2, ..., ha_{m'}).$$

However, Authority can add new attributes to our system whenever needed.

C. Encrypt(M, MK).

If Authority wants to share message M with some people who have particular attributes, he first lists all the attributes that are associated with M. We assume that the list is

$$Att = \{a_1, a_2, ..., a_q\}.$$

Authority than describes how the attributes in set Att can be combined to decrypt the encrypted message in an access propositional formula. If the access propositional formula is

$$pf = pf'_1 \vee pf'_2 \wedge ... \vee pf'_{n'}$$

where $pf'_1, pf'_2, ..., pf'_{n'}$ are sub-formula of pf. It is easy for Authority to transform pf to

$$pf = pf_1 \vee pf_2 \vee ... \vee pf_n$$

where, for $1 \leq i \leq n$, pf_i only contains "$\wedge$" and attributes. The purpose we transform pf into a "all $\wedge$" form is that a user who has all the attributes descried in any pf_i can decrypt the ciphertext. For every access propositional sub-formula pf_i, if

$$pf_i = a_{i1} \wedge a_{i2} \wedge ... \wedge a_{ii'},$$

Authority computes

$$r'_i = (i' \times MK + ha_{i1} + ... + ha_{ii'}) \times (ha_{i1} + ... + ha_{ii'}).$$

Then Authority chooses a random number R and computes

$$r_i = h_2(e(g, g)^{r'_i}||R)$$

After Authority gets $r_1, r_2, ..., r_n$ for every pf_i for $i \in [1, n]$, he chooses a key y to encrypt message M using any secure encryption scheme, of course, a symmetrical encryption scheme is suggested. We assume that the encrypted message is $E(M)$. Authority then chooses a polynomial using the SIFF scheme:

$$f(x) = x^n + a_1 x^{n-1} + ... + a_i x^{n-i} + ... + a^n$$

where $r_1, r_2, ..., r_n$ are n solutions of equation $f(x) - y = 0$. Authority works out all the coefficients $co[j] = a_j$ of $f(x)$ for $1 \leq j \leq n$.

Finally, Authority can put $CM = (E(M), \{co[j]\}, R, pf)$, where $j \in [1, n]$, to a storage server. Anyone who wants to try to decrypt $E(M)$ can get CM from the storage server.

D. KeyGen(MK, S).

If a user's attribute set is

$$S = \{s_1, s_2, ..., s_k\}$$

, Authority computes

$$(g^{\{[MK+hs_i]/t\}}, g^{t \times hs_i})$$

for $i \in [1, k]$ after he randomly chooses t which is used to distinguish different users. Then Authority sends them to the user. The secret key of the user is

$$((g^{\{[MK+hs_1]/t\}}, g^{t \times hs_1}), ..., (g^{\{[MK+hs_k]/t\}}, g^{t \times hs_k})).$$

E. Decrypt(CM, SK).

After a user gets CM form the storage server, he first finds out which attributes in S can be used to decrypt CM from pf. If he has all the attributes in pf_k which we donate as

$$Sa[j] = \{s_{j1}, s_{j2}, ..., s_{jj'}\},$$

the user then computes

$$g_1 = g^{\{[MK+hs_{j1}]/t\}} \times ... \times g^{\{[MK+hs_{jj'}]/t\}}$$

and

$$g_2 = g^{t \times hs_{j1}} \times ... \times g^{t \times hs_{jj'}}$$

and after that he computes $root = h_2(e(g_1, g_2)||R)$ which is $h_2(e(g, g)^d||R)$ where

$$d = \{j' \times MK + hs_{j1} + ... + hs_{jj'}\} \times \{hs_{j1} + ... + hs_{jj'}\}$$

If $pf_k = s_{j1} \wedge s_{j2} \wedge ... \wedge s_{jj'}$, then $root$ is one of the solutions of $f(x) - y = 0$, namely, $f(root) = y$. The user can get y and decrypt $E(M)$.

VI. Efficiency and Tradeoffs

The main feature of our CP-ABE scheme is that the decryption algorithm only takes one pairings and $2n$ multiplications, where n is the attributes number in a sub-formula in our access propositional formula, to work out the key y. If the encryption algorithm that used in our ***Encrypt()*** process is a symmetric encryption algorithm, compared to a public encryption algorithm, the decryption time is quite short. The decryption scheme in the milestone work [3] requires two pairings for each leaf in the ciphertext's access tree and one exponentiation for each node along a path from such a leaf to the root. To our knowledge, the fast decryption KP-ABE scheme proposed by Hohenberger and Waters [8] is the fastest decryption scheme and it requires two pairings and two exponentiations per row used. However, their work is a key-policy attribute-based encryption scheme.

Another feature is that the ciphertext in our ABE scheme requires very little storage space. The ciphertext in our scheme is $CM = (E(M), \{co[j]\}, R, pf)$. If Authority uses a symmetric encryption algorithm, such as AES , in our ***Encrypt()*** process, the size of $E(m)$ equals the message M. $\{co[j]\}, R$ and pf are just some parameters which is fairly small. So the size of ciphertext CM in our scheme roughly equals the

size of the message M. The ciphertext size in [3] is quite huge compared to our ciphertext size. The ciphertext in [3] contains the encrypted message, an access tree and two big prime number for every leaf node in the access tree. Our scheme also has a high degree compatibility with existing ciphertext. In our encryption algorithm, the major work is to deal with the key y other than the encryption scheme so that Authority can use any encryption scheme he wants. The ciphertexts encrypted before our system was set up can be used in our system directly without being re-encrypted.

The attributes in our scheme is quite flexible. The number of attributes is not limited. Authority can add any attributes to our scheme whenever needed even after the setup of the system. If Authority wants to add more attributes or attribute combinations to existing access propositional formulas, he only have to re-calculate the polynomial $f(x)$. However, if Authority wants to reduce the attributes or attributes combinations from existing access propositional formulas, he should re-encrypt the message first in case that someone skips the key-calculating process after he already got the key before the access propositional formulas were changed.

However, tradeoffs exists in our system. In our setting, Authority distributes attributes and sends message to users. We narrow down the normal CP-ABE situations to exchange the fast decryption and small storage space. In fact, our ABE scheme can also be used in the normal CP-ABE circumstance. Authority distributes attributes to users and he can distributes any attribute to himself, so we have to assume that Authority is totally trusted. If a user wants to share a message with some users with particular attributes, he can sends the message to Authority and Authority can do the rest work for him. However, the workload of Authority will be very high.

VII. Security and Performance Analysis

A. Security Analysis

We assume that Authority is trusted since he distributes attributes to users including himself. The only concern is that users may be dishonest.

A user may want to distribute attributes, which do not belong to him, to himself. For attribute a, the target form of a he wants to get is $(g^{[MK+ha]/t}, g^{t\times ha})$. The user does not know MK, so he can not distribute attribute a to himself. He may already get other attributes form Authority and we assume that he has got $(g^{[MK+hb]/t_1}, g^{t_1\times hb})$, $(g^{[MK+hc]/t_1}, g^{t_1\times hc})$ for attribute b and c respectively. Even though, basing on the Discrete Logarithm assumption, he can not get $([MK + hb]/t_1, t_1 \times hb)$ from the information of attributes b and $([MK + hc]/t_1, t_1 \times hc)$ from attribute c. So he can not get MK which leads that he can not compute $(g^{[MK+ha]/t}, g^{t\times ha})$ even he knows $g, h(\cdot), a$ and t (t can be chosen randomly).

Two or more users may want to collude together. Consider the situation where Alice has attributes (a, b), Bob has an attribute c and an access propositional formula of a ciphertext is $pf = a \wedge b \wedge c$. If Alice and Bob want to get the message from the ciphertext, they may collude together. However, the expressions of Alice's attributes are $((g^{[MK+ha]/t}, g^{t\times ha}), (g^{[MK+hb]/t}, g^{t\times hb}))$ and Bob's attribute expression is $(g^{[MK+hc]/t_1}, g^{t_1\times hc})$. If they two run the decryption algorithm using their attributes, they get $h_2(e(g,g)^d||R)$ where

$$d = (\frac{2MK + ha + hb}{t} + \frac{MK + hc}{t_1}) \times (t \times (ha + hb) + t_1 \times hc)$$

rather than

$$d = (3MK + ha + hb + hc) \times (ha + hb + hc).$$

That's the reason we use a random number t to distinguish users.

Users may get some useful information from the polynomial associated with a ciphertext. $f(x)$ is a polynomial of one variable, if a user get it without knowing the exact value of the combination of associated attributes, he has countless choices so that the chance he guess the right answer is negligible. In consideration of the easiness to solve a polynomial equation with one unknown, a user may get other information from $f(x)$. If the access propositional formula of a ciphertext is $pf = (a \wedge b) \vee (c \wedge d)$ and a user has attributes a and b. The user can use a and b to get the key y which was used to encrypt the message. Then the user can solve $f(x) = y$, which is very easy, to get the expression of $c \wedge d$. So the user can use this expression to get some information that requires attributes c and d. We introduce a random number R to stop that from happening. For every ciphertext, they have a different R so that if a user get $h_2(e(g,g)^k||R)$, where $k = (2MK + hc + hd)(hc + hd)$, he can not solve out the key from other ciphertext, whose random number is R', which requires attributes c and d. Although the user knows R, R' and $h_2(e(g,g)^k||R)$, he can not get $e(g,g)^k$ because $h_2(\cdot)$ is a one way hash function.

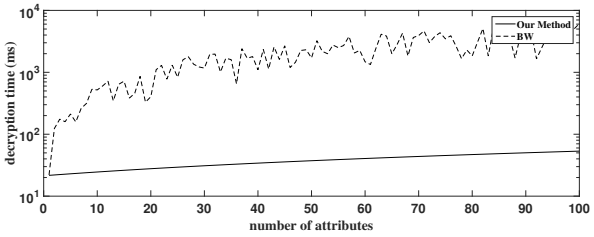

Fig. 2. Decryption time

B. Performance Analysis

We narrow down the occasion of ABE system to a setting where the attributes distributor is also the message owner, namely, Authority shares message with users after he distributes attributes to users. We have discussed the efficiency of our system, one of the significant feature is that the decryption speed in our system is quite fast, it only requires one pairing and some multiplication to work out y. The other feature is that the storage space that the ciphertext needed roughly equals the size of the message. In this subsection, we give a comparison between our work and Bethencourt's work (BW) [3].

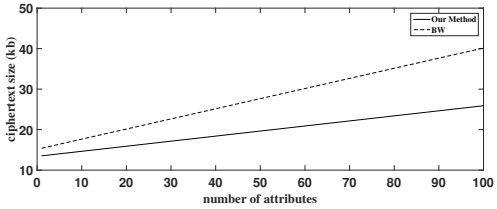

Fig. 3. Ciphertext size

We encrypted a random file whose size is 13.4 kb using both BW scheme and our scheme. Then in Fig. 2, we present the decryption time of our work and BW scheme. The number of attributes indicates that the attributes number in the access propositional sub-formula of our scheme and the attributes that actually used in the decryption process of BW scheme. Fig. 3 shows the ciphertext size of our work and BW scheme. In fact, the larger the message to be encrypted, the better the result of our scheme will be.

VIII. Conclusion

In this paper, we propose a ciphertext-policy attribute-based encryption scheme that can be used in a setting where the authority distributes attributes to users as well as sharing messages with them. The decryption scheme in our CP-ABE system is very fast and the ciphertext in our system is quite small. When encrypting a message using our scheme, we care more about the key than the encryption algorithm. In fact, to share a message with users, Authority actually share the encryption key. So any encryption scheme can be used in our system and of course symmetric encryption method is suggested. Considering the above situation, our scheme also have a very good compatibility with the message that have already been encrypted when our system is set up. We implement our system and give a comparison of our scheme and Bethencourt's work [3].

ACKNOWLEDGEMENT

This work was supported in part by the Natural Science Foundation of China (NSFC) under Grants 61202140 and 61328208, by the Program for New Century Excellent Talents in University under Grant NCET-13-0548, and by the Fundamental Research Funds for the Central Universities under Grand WK2101020006.

References

[1] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology–EUROCRYPT 2005*. Springer, 2005, pp. 457–473.

[2] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on computer and communications security*. Acm, 2006, pp. 89–98.

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

[4] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proceedings of the 14th ACM conference on Computer and communications security*. ACM, 2007, pp. 195–203.

[5] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters, "Secure attribute-based systems," in *Proceedings of the 13th ACM conference on Computer and communications security*. ACM, 2006, pp. 99–112.

[6] L. Cheung and C. Newport, "Provably secure ciphertext policy abe," in *Proceedings of the 14th ACM conference on Computer and communications security*. ACM, 2007, pp. 456–465.

[7] D. Lubicz and T. Sirvent, "Attribute-based broadcast encryption scheme made efficient," in *Progress in Cryptology–AFRICACRYPT 2008*. Springer, 2008, pp. 325–342.

[8] S. Hohenberger and B. Waters, "Attribute-based encryption with fast decryption," in *Public-Key Cryptography–PKC 2013*. Springer, 2013, pp. 162–179.

[9] N. Attrapadung, B. Libert, and E. De Panafieu, "Expressive key-policy attribute-based encryption with constant-size ciphertexts," in *Public Key Cryptography–PKC 2011*. Springer, 2011, pp. 90–108.

[10] N. Attrapadung, J. Herranz, F. Laguillaumie, B. Libert, E. De Panafieu, and C. Ràfols, "Attribute-based encryption schemes with constant-size ciphertexts," *Theoretical Computer Science*, vol. 422, pp. 15–38, 2012.

[11] A. Kapadia, P. P. Tsang, and S. W. Smith, "Attribute-based publishing with hidden credentials and hidden policies." in *NDSS*, vol. 7. Citeseer, 2007, pp. 179–192.

[12] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures," in *Applied cryptography and network security*. Springer, 2008, pp. 111–129.

[13] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Advances in Cryptology–EUROCRYPT 2010*. Springer, 2010, pp. 62–91.

[14] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Public Key Cryptography–PKC 2011*. Springer, 2011, pp. 53–70.

[15] A. Lewko and B. Waters, "New proof methods for attribute-based encryption: Achieving full security through selective techniques," in *Advances in Cryptology–CRYPTO 2012*. Springer, 2012, pp. 180–198.

[16] Y. Rouselakis and B. Waters, "Practical constructions and new proof methods for large universe attribute-based encryption," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 2013, pp. 463–474.

[17] T. Okamoto and K. Takashima, "Fully secure functional encryption with general relations from the decisional linear assumption," in *Advances in Cryptology–CRYPTO 2010*. Springer, 2010, pp. 191–208.

[18] M. Chase, "Multi-authority attribute based encryption," in *Theory of cryptography*. Springer, 2007, pp. 515–534.

[19] M. Chase and S. S. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proceedings of the 16th ACM conference on Computer and communications security*. ACM, 2009, pp. 121–130.

[20] S. Müller, S. Katzenbeisser, and C. Eckert, "Distributed attribute-based encryption," in *Information Security and Cryptology–ICISC 2008*. Springer, 2009, pp. 20–36.

[21] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Advances in Cryptology–EUROCRYPT 2011*. Springer, 2011, pp. 568–588.

[22] H. Lin, Z. Cao, X. Liang, and J. Shao, "Secure threshold multi authority attribute based encryption without a central authority," *Information Sciences*, vol. 180, no. 13, pp. 2618–2632, 2010.

[23] N. Attrapadung and H. Imai, "Conjunctive broadcast and attribute-based encryption," in *Pairing-Based Cryptography–Pairing 2009*. Springer, 2009, pp. 248–265.

[24] A. Sahai, H. Seyalioglu, and B. Waters, "Dynamic credentials and ciphertext delegation for attribute-based encryption," in *Advances in Cryptology–CRYPTO 2012*. Springer, 2012, pp. 199–217.

[25] A. Lewko and B. Waters, "Unbounded hibe and attribute-based encryption," in *Advances in Cryptology–EUROCRYPT 2011*. Springer, 2011, pp. 547–567.

[26] S. Hohenberger and B. Waters, "Online/offline attribute-based encryption," in *Public-Key Cryptography–PKC 2014*. Springer, 2014, pp. 293–310.

[27] Y. Zheng, T. Hardjono, and J. Pieprzyk, "The sibling intractable function family (siff): notion, construction and applications," *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 76, no. 1, pp. 4–13, 1993.