Information **2015**, *6*, 704-718; doi:10.3390/info6040704

OPEN ACCESS

information

ISSN 2078-2489

www.mdpi.com/journal/information

Article

Batch Attribute-Based Encryption for Secure Clouds

Chen Yang [1], Yang Sun [2,†] and Qianhong Wu [2,†,*]

[1] China Information Security Research Institute, No. 11 ShangDi Xinxi Road, Beijing 100085, China; E-Mail: yangchenyf@163.com

[2] School of Electronics and Information Engineering, Beihang University, XueYuan Road No.37, Haidian District, Beijing 100191, China; E-Mail: yangsun@buaa.edu.cn

† These authors contributed equally to this work.

* Author to whom correspondence should be addressed; E-Mail: qianhongwu@buaa.edu.cn; Tel./Fax: +86-10-8233-9469.

Academic Editors: Qiong Huang and Guomin Yang

Received: 31 July 2015 / Accepted: 26 October 2015 / Published: 29 October 2015

Abstract: Cloud storage is widely used by organizations due to its advantage of allowing universal access with low cost. Attribute-based encryption (ABE) is a kind of public key encryption suitable for cloud storage. The secret key of each user and the ciphertext are associated with an access policy and an attribute set, respectively; in addition to holding a secret key, one can decrypt a ciphertext only if the associated attributes match the predetermined access policy, which allows one to enforce fine-grained access control on outsourced files. One issue in existing ABE schemes is that they are designed for the users of a single organization. When one wants to share the data with the users of different organizations, the owner needs to encrypt the messages to the receivers of one organization and then repeats this process for another organization. This situation is deteriorated with more and more mobile devices using cloud services, as the ABE encryption process is time consuming and may exhaust the power supplies of the mobile devices quickly. In this paper, we propose a batch attribute-based encryption (BABE) approach to address this problem in a provably-secure way. With our approach, the data owner can outsource data in batches to the users of different organizations simultaneously. The data owner is allowed to decide the receiving organizations and the attributes required for decryption. Theoretical and experimental analyses show that our approach is more efficient than traditional encryption implementations in computation and communication.

Keywords: data privacy; access control; cloud storage; batch attribute-based encryption

1. Introduction

In cloud computing, if a file owner would like to share his file with others, he just stores his data on a cloud server, then his friends/colleagues, including himself, can access the data by visiting the server from any place where there is an Internet connection. In this way, cloud storage makes it possible for people to exploit almost unlimited computation, storage space and information services anytime and anywhere without suffering from complicated local maintenance and management. Due to these advantages, cloud computing is becoming an indispensable part of organizations and individuals.

The security concern of the data owner is believed to be the main obstacle for wide adoption of cloud storage. Especially, the user who do not have access right may collude with the server for illegal access. Encryption is a standard way to protect data security. Public encryption is preferable in organization-oriented applications. In this scenario, one organization can have a cloud account associated with a public key, and anyone can securely upload files to this account by encrypting the files under the public key. Only the one who owes the corresponding secret key of the public key can decrypt the encrypted file, which prevents illegal access from the cloud server or other non-authorized ones. However, the traditional public key cryptosystems suffer from complicated key management in this scenario, since it is unknown who will be allowed to access a specific encrypted file when the system is set up or the file is encrypted. Attribute-based encryption (ABE) [1] is the up-to-date cryptographic concept to address this problem. In this case, the encryptor can specify the attributes required for decryption, and the access policy can be made for the secret key of each requestor. A requestor can decrypt a ciphertext only if the labeled attributes meet the access policy associated with the secret key of the requestor.

An issue in deploying ABE is that all known ABE encryption/decryption operations are time and resource consuming. Furthermore, the existing ABE schemes only allow encryption for the receivers of a single organization. Consider a scenario where the scientists from different universities jointly work on a project. In this case, if a scientist wants to share his or her research progress with the scientists of other universities, then he or she needs to encrypt the files to these organizations repetitively. In [2], an approach is proposed to speed up the decryption with the help of a semi-honest third party. However, to the best of our knowledge, no known work in the public literature has been done to speed up the ABE encryption in our motivating scenario. This situation inspires our work in this paper.

1.1. Our Contribution

In this paper, we investigate the acceleration of ABE encryption where one encrypts for the requestors of different organizations. Our contributions are summarized as follows.

We propose a batch attribute-based encryption (BABE) framework. In this framework, when the different ABE systems are set up, the users of different systems get their keys from the authority of the corresponding system, and these keys are associated with the access policies of the corresponding organizations. One data owner can simultaneously encrypt his or her data for the receivers from different systems in a batch way. The data owner can freely choose the receiving organizations and define separate attributes for the receivers of different organizations. One can decrypt only if his/her organization has been chosen and his/her attributes meet the access policy for his organization.

Following the BABE framework, we propose a concrete BABE scheme. The security of the scheme is formally proven under a well-established computational assumption. Even if an adversary colludes with the cloud server and all non-authorized users, he or she cannot get any useful information of the encrypted message. We conduct experiments in typical system parameters. The experimental results show that our BABE scheme is more efficient than existing ABE implementations. These features render our BABE an efficient solution to securing one-to-many organization-oriented sensitive data sharing in clouds.

1.2. Related Work

Due to its versatility, ABE has attracted extensive research [1,3–5]. The ABE concept was introduced by Sahai and Waters [1]. There are two types of ABE systems, *i.e.*, ciphertext-policy ABE (CP-ABE) and key-policy ABE (KP-ABE). In CP-ABE, the secret key of each user is associated with his/her attributes. When one encrypts for the users of the system, he or she specifies an access policy, which is a collection of attribute sets. Only the user whose attributes match the policy can decrypt. Similarly, in KP-ABE, the secret key of a user is associated with an access policy, and each ciphertext is labeled with a set of attributes. One can decrypt a ciphertext only if the labeled attributes meet the policy associated with the user's secret key. Yu *et al.* [6] proposed a KP-ABE scheme to achieve flexible access control over outsourced data.

In a regular ABE system, a single authority generates the secrets of all of the users in the system. This incurs the so-called single-point problem, that is all of the parties have to fully trust the single authority, who can read all of the messages encrypted for any users. Furthermore, no new user can join the system if the authority is occasionally unavailable. To mitigate this drawback, the notion of multi-authority ABE (MA-ABE) has been proposed. In this notion, multiple authorities jointly generate the secret keys for the users in the system. No single authority can read the messages encrypted for the users. New users can still be enrolled if some of the authorities are available. An encryptor can choose, for each authority, a number d_k and a set of attributes; he or she can then encrypt a message, such that a user can decrypt only if he or she has at least d_k of the given attributes from each authority k [7]. Chase and Chow improved the MA-ABE in [8] by exchanging a shared secret among the authorities during the system setup process. Their scheme is time consuming and needs heavy communication and computation. Yang *et al.* [9] designed an access control framework for multi-authority systems and proposed a secure multi-authority access control scheme for cloud storage. Li *et al.* [10] proposed an MA-ABE scheme for mobile cloud computing by introducing a cloud-based semi-trusted-authority between the mobile user and the attribute authorities.

ABE schemes require time-consuming computations in encryption and decryption, which prevent them from being widely deployed in cloud computing applications for enforcing fine-grained access control. Several works have been proposed to speed up ABE. Attrapadung *et al.* proposed ABE schemes [11,12] that only need a constant number of bilinear maps in decryption, albeit with the constraint of a bound on the maximum number of attributes. Hohenberger and Waters [2] introduced a method that needs a constant number of bilinear maps in decryption without the bound constraint on the attribute number. In contrast, no public work has been reported to accelerate the encryption process in ABE.

Fiat proposed the first batch signature scheme [13]. Since then, a large body of references has focused on the batch verification of signatures [14–17]. It is unclear whether these technologies can be employed to accelerate ABE encryption in our motivating scenario. Still in a single system-oriented setting, multi-receiver encryption and signcryption [18] have been proposed to reduce the encryption complexity. Baek, Safavi-Naini and Susilo proposed an efficient multi-receiver identity-based encryption scheme [19] in multi-cast encryption applications. All of these schemes are considered in a single organization-oriented scenario. A trivial application of these schemes to multiple organizations needs to repeat the basic encryption operation for each organization, which implies growing overheads linear with the number of organizations. In this paper, we propose to encrypt for the different organizations in a batch way and to provide a more efficient solution to the motivating applications.

1.3. Paper Organization

The rest of this paper is organized as follows. Section 2 presents a BABE framework and the adversary model. We propose a concrete BABE scheme in Section 3, together with detailed security and efficiency analyses. Section 4 concludes the paper.

2. Modeling BABE

2.1. The System Framework

We propose a BABE framework for cloud storage, as illustrated in Figure 1. The framework consists of four types of parties: the data owner, the cloud server, the data requestor/consumer and the authority. The data owner would like to share his or her data with the data requestors managed by different authorities. Each authority generates a secret key for the data requestor in its management domain according to the consumer's attributes (or the corresponding access policies). The data owner encrypts his or her file for the data requestors of different organizations, *i.e.*, the users managed by different authorities, by specifying the intended authorities and the attributes that the data requestors should have in the according authorities' management domains. Then, the data owner uploads the encrypted data to the cloud storage server. When a data requestor requests for the cloud server, the server responds with the corresponding ciphertext. The requestor can decrypt if the attributes labeled with the ciphertext meet the access policy associated with the secret key of the requestor; else, the requestor should get no useful information of the original data.

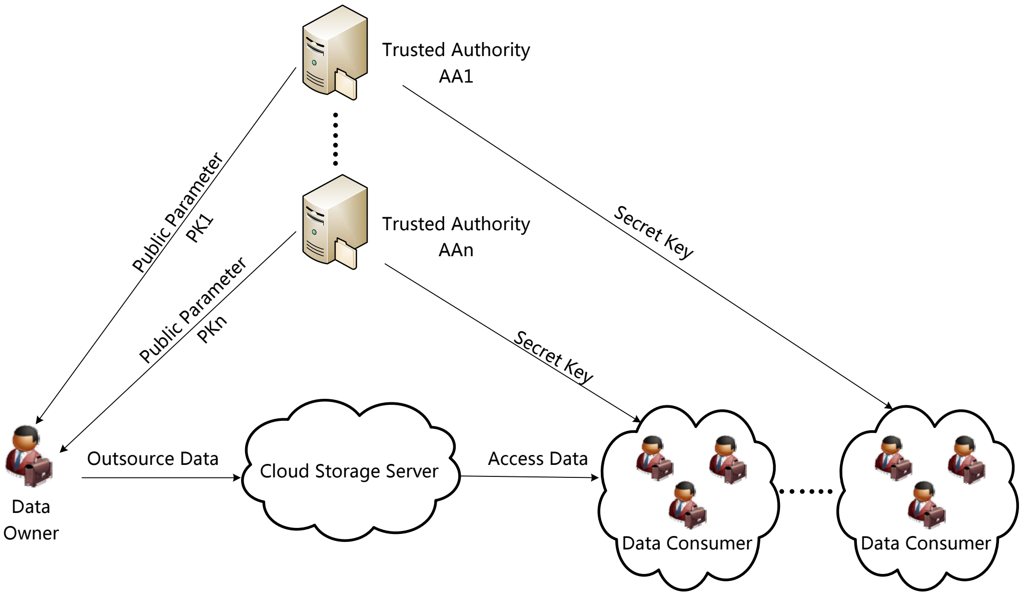

Figure 1. Batch attribute-based encryption (BABE) framework.

2.2. Defining BABE

A BABE scheme consists of four polynomial-time algorithms: *Setup*, *Encrypt*, *KeyGeneration* and *Decrypt*. Let $\mathbb{I}$ represent the collection of underlying ABE subsystems indexed from one to $|\mathbb{I}|$. Denote that the i-th underlying ABE subsystem has master public key PK_i and master secret key MK_i for $i \in \mathbb{I}$. The algorithms work as follows.

Setup$(\lambda, U_i) \longrightarrow (PK_i, MK_i)$. This is a randomized algorithm run by an authority AA_i. It takes as inputs the security parameter λ and an attribute universe U_i, which defines the set of allowed attributes in different underlying ABE subsystems. The authority AA_i outputs the public parameters PK_i and the mater keys MK_i.

Encrypt$(\vec{PK}, \vec{M}, \vec{S}) \longrightarrow \vec{CT}$. Assume that the data owner wants to share data M_i with the data requestors of the systems $i \in \psi \subseteq \mathbb{I}$. This is a randomized algorithm that takes as inputs the public parameters vector $\vec{PK}$, a message vector $\vec{M}$, which contain different messages M_i, and a series of attribute sets $\vec{S}$, which contains attribute sets $S_i, i \in \psi$. It outputs $|\psi|$ ciphertext $CT_i, i \in \psi$ associated with the attribute sets $S_i, i \in \psi$.

KeyGen$(MK_i, A_i) \longrightarrow$ SK. Assume that a data requestor belongs to the i-th subsystem. Then, MK_i is used to generate the secret key SK. This is a randomized algorithm that takes as inputs the master secret key MK_i and an access structure A_i and outputs a secret key SK.

Decrypt $(SK, CT_i) \longrightarrow M_i$. The decryption algorithm takes as inputs a secret key SK associated with the access structure A_i and a ciphertext CT_i associated with attribute set S_i. It outputs the message M_i if S_i satisfies A_i or the error message $\perp$ otherwise.

2.3. The Adversary Model

For sensitive files stored on the cloud, we mainly consider the secrecy of the files. In this scenario, both the cloud server and users may be potential attackers. Specifically, the cloud server (or the staff who manages the cloud server) may be curious and access the private files of the users without being known by the users. Furthermore, some unauthorized users may collude, even with the cloud server, to illegally access the files. However, we have to make a minimum trust assumption that the cloud server will honestly follow the scheme to execute the procedures and return the service to the users. Else, the scheme will not work and makes no sense in practice. We require that the attacker in such a situation cannot get any useful information of the outsourced data.

Now we formally define the security model of a BABE scheme. Let U_i be the attribute universe of the i-th underlying ABE scheme. All of the underlying ABE schemes share the security parameter λ. Each underlying ABE scheme is managed by an authority AA_i. $\mathbb{I}$ is the index set of the underlying ABE schemes. Formally, we define the security model for BABE via a game between a challenger and an adversary.

Initialize. The adversary declares the different underlying ABE schemes upon which he or she wants to be challenged, indexed by $\psi \subseteq \mathbb{I}$. The adversary also declares the attribute set $S_i^*, i \in \psi$ of different ABE schemes upon which he or she wished to be challenged upon.

Setup. The challenger runs the *Setup* algorithm to set up the BABE system and sends the public parameters, $PK_i, i \in \mathbb{I}$, to the adversary.

Phase 1. For the i-th underlying ABE subsystem, the challenger initializes for $i \in \mathbb{I}$ an empty table T_i, an empty set D_i and an integer counter $j_i = 0$. The adversary can repeatedly make the following queries:

Create(A_i): The challenger sets $j_i = j_i + 1$. It runs the key generation algorithm on A_i to obtain the secret key SK and stores it in table T_i, forming the entry (j_i, A_i, SK).

Corrupt(κ, i): The adversary queries for the secret key corresponding to the κ-th entry in Table i. If there exists such an entry in table T_i, the challenger obtains the entry (κ, A_i, SK), sets $D_i = D_i \bigcup \{A_i\}$ and returns the secret key SK to the adversary. Else, it returns $\perp$.

Challenge. The adversary submits two equal-length messages, vector $\vec{M}_0$ and $\vec{M}_1$, with each vector having $|\psi|$ message $M_{0,i}, i \in \psi$ and $M_{1,i}, i \in \psi$. In addition, the adversary gives a set of attributes $S_i^*, i \in \psi$, such that for all $A_i \in D_i, i \in \psi$, the set $S_i^*, i \in \psi$ does not satisfy the access structure $A_i, i \in \psi$. The challenger chooses $b \in \{0, 1\}$ randomly and encrypts $\vec{M}_b$ under $S_i^*, i \in \psi$. The resulting ciphertext $CT_i^*, i \in \psi$ is given to the adversary.

Phase 2. Phase 1 is repeated with the restriction that the adversary cannot obtain any secret key that can be used to decrypt the challenge ciphertext. This means that it cannot issue a *Corrupt* query that would result in an access structure A_i that S_i^* satisfies being added to $D_i \in \psi$. Of course, the adversary cannot issue a decryption query on the challenge ciphertext $CT_i^*, i \in \psi$.

Guess. The adversary outputs a guess bit b' of b. The output of the experiment is one if and only if $b = b'$.

The advantage of the adversary $\mathcal{A}$ in this game is defined as $Pr[b' = b] - \frac{1}{2}$.

Definition 1. A BABE scheme is secure if all polynomial time adversaries have at most a negligible advantage in the above game.

3. The Proposed BABE Scheme

3.1. Mathematical Background

We briefly review the basic mathematical concepts used in our schemes.

Definition 2. (Access structure [20]). Consider a set of parties $P = \{P_1, P_2, \cdots, P_n\}$. A collection $\mathbb{A} \subseteq 2^{\{P_1, P_2, \cdots, P_n\}}$ is said to be monotone if for $\forall B, C$, we have that $B \in \mathbb{A}$ and $B \subseteq C$, then $C \in \mathbb{A}$. An access structure (respectively, monotone access structure) is a collection (respectively, monotone collection) $\mathbb{A}$ of non-empty subsets of $\{P_1, P_2, \cdots, P_n\}$, *i.e.*, $\mathbb{A} \subseteq 2^P \backslash \emptyset$. The authorized sets represent the sets in $\mathbb{A}$, and the sets not in $\mathbb{A}$ are called the unauthorized sets.

Definition 3. (Linear secret sharing scheme [20]). Let P be a set of parties. Let W be an $l \times m$ matrix. Let f be a function that maps a row to a party. A secret sharing scheme Π for access structure $\mathbb{A}$ over a set of parties P is a linear secret-sharing scheme (LSSS) in $\mathbb{Z}_p$ and is represented by (W, f) if it consists of two efficient algorithms:

Share(W, f)**:** This takes as input $s \in \mathbb{Z}_p$, which is to be shared. It chooses $r_2, r_3, \cdots, r_n$, and let $\vec{v} = (s, r_2, \cdots, r_n)$. It outputs $W\vec{v}$ as the vector of the shares. The share $\omega_i = W_i\vec{v}$ belongs to party $f(i)$, where W_i is the i-th row of W.

Recon(W, f)**:** This takes as input an access set $S \in A$. Let $I = \{i | f(i) \in S\}$. It outputs a set of constants $(i, \mu_i)_{i \in I}$, such that $\sum_{i \in I} \mu_i \cdot \omega_i = s$.

In a KP-ABE scheme, a ciphertext is associated with a set of attributes S, and each secret key corresponds to an access structure $\mathbb{A}$. One can decrypt if the attribute set S is authorized in the access structure $\mathbb{A}$ $(i.e., S \in \mathbb{A})$.

Our scheme is built on bilinear groups briefly defined as follows.

Definition 4. (Bilinear group). Let G and G_T be two multiplicative cyclic groups of prime order p. Let g be a generator of G and e be a bilinear map, $e : G \times G \longrightarrow G_T$. The bilinear map e has the following properties:

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

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

We say that G is a bilinear group if the group operation in G and the bilinear map $e : G \times G \longrightarrow G_T$ are both efficiently computable. Notice that the map e is symmetric since $e(g^a, h^b) = e(g, h)^{ab} = e(g^b, h^a)$.

3.2. The Proposal

We first give an outline of our BABE scheme. The scheme is proposed to accelerate the encryption process when one needs to encrypt for the users from different organizations in a cloud storage setting where ABE is used to enforce fine-grained access control. The data owner chooses the data he or she wants to share with the data requestors of the different systems. Then, he or she encrypts the data and sends the encrypted data to the cloud server. Authorized data requestors will be able to decrypt the encrypted data by using the secret key given by the corresponding authority. Using the traditional approach, the owner needs to encrypt each data item for the receivers of the different organizations repetitively, which is inefficient in practice. We suggest to encrypt in a batch way and propose a BABE scheme managing to solve this problem.

For $i \in \mathbb{I}$, we describe different subsystems with different universes U_i of different attributes. The attributes are indexed by integers $1, \cdots, |U_i|$. We use $|U_i|$ to represent the number of the attribute universe U_i. The number of the attributes may be different in distinct subsystems.

Setup. This algorithm, run by the authority AA_i, takes as inputs a system parameter λ and an attribute universe U_i. It generates a bilinear group G of prime order p, with generator $g \in G$. It selects random values $h_{i,1}, \cdots h_{i,|U_i|} \in G$. It further chooses a distinct $\alpha_i \in \mathbb{Z}_p$ for each i. Finally, for each i, it outputs the public key and master secret key as:

$$PK_i = (G, p, g, e(g,g)^{\alpha_i}, h_{i,1}, \cdots, h_{i,|U|}), MK_i = (PK_i, \alpha_i)$$

Encrypt. The data owner chooses the organizations that he or she wants to communicate with, denoted by $\psi \subseteq \mathbb{I}$. Then, he or she asks the authority AA_i,$i \in \psi$ for the public key PK_i and chooses a series of attributes $\vec{S}$, where $S_i, i \in \psi$, a series of messages $\vec{M} \in G_T$. The data owner chooses a random $s \in \mathbb{Z}_p$. The ciphertext component used by i-th subsystem is $CT_i = (S_i, C_i, C', C_{i,x})$ where:

$$C_i = M_i \cdot e(g,g)^{\alpha_i s}, C' = g^s, \{C_x = h_{i,x}^s\}_{x \in S_i}$$

After the encryption, the data owner sends the encrypted data to the cloud server.

KeyGen. For each i, the authority AA_i takes as inputs the master secret key MK_i and an LSSS access structure A_i represented by (W, f). Let W be an $l \times m$ matrix. The function f is associated with the rows of W. Let β denote the set of distinct attributes appearing in the access structure matrix W, *i.e.*, $\beta = \{d : \exists \tau \in [1,l], f(\tau) = d\}$. This algorithm chooses a random vector $\vec{v} = (\alpha_i, y_2, \cdots, y_m) \in Z_n$. Here, these values will be used to share the master secret α_i, and $y_2, \cdots, y_m$ are random numbers. For $\tau = 1$ to l, it calculates $\lambda_\tau = \vec{v} \cdot W_\tau$, where W_τ is the vector corresponding to the τ-th row of W. The algorithm chooses random integers $r_1, \cdots r_l \in \mathbb{Z}_p$. It outputs the secret key SK as:

$$(D_1 = g^{\lambda_1} \cdot h_{i,f(1)}^{r_1}, R_1 = g^{r_1}, \forall d \in \beta \setminus f(1), Q_{1,d} = h_{i,d}^{r_1})$$

$$\vdots$$

$$(D_l = g^{\lambda_l} \cdot h^{r_l}_{i,f(l)}, R_l = g^{r_l}, \forall d \in \beta \setminus f(l), Q_{l,d} = h^{r_l}_{i,d})$$

In the above, $\beta \setminus x$ means the set β excluding x.

Decrypt. The data requestor requests the encrypted data from the cloud server. The requestor runs the decryption algorithm with his/her secret key $(SK = PK_i, (D_1, R_1, Q_{1,d}), \cdots, (D_l, R_l, Q_{l,d}))$ for access structure (W, f) and the ciphertext $CT_i = (S_i, C_i, C', \{C_x\}_{x \in S_i})$ for set S_i. Let W be an $l \times m$ matrix. The function f is associated with some rows of W. If S_i does not satisfy the access structure, the decryption algorithm outputs $\perp$.

If S_i satisfies the access structure, we let $I \subseteq \{1, 2, \cdots l\}$ be a set of indices and $\{\omega_\tau\}_{\tau \in I} \in \mathbb{Z}_p$ be a set of constants, such that $f(\tau) \in S_i$ and $\Sigma_{\tau \in I} \omega_\tau \cdot W_\tau = (1, 0, 0 \cdots 0)$ for all $\tau \in I$. Note that ω_τ can be found in polynomial time.

We represent the attributes satisfying the access structure as $\gamma = \{x : \exists \tau \in I, f(\tau) = x\}$. I is the set of indices corresponding to the rows used to decrypt the ciphertext, and γ is the set of distinct attributes associated with these rows. In fact, there are multiple I's that satisfy the constraints above. Typically, the smaller the size of I, the better. Observe that $\gamma \subseteq S_i$, where S_i is the attribute set used to encrypt the ciphertext M_i and $\gamma \subseteq \beta$, the set of attributes used to create the secret key.

To recover the value $e(g, g)^{\alpha_i s}$, we define a function v, which transforms a set of attributes into an element of G in the way $v(\gamma) = \prod_{x \in \gamma} h_{i,x}$. For each $\tau \in I$, the data requestor computes the value $D'_\tau = D_\tau \cdot \prod_{x \in \gamma / f(\tau)} Q_{\tau,x} = g^{\lambda_\tau} f(\gamma)^{r_\tau}$. For the ciphertext, the data requestor computes the value $L = \prod_{x \in \gamma} C_x = \prod_{x \in \gamma} h^s_{i,x} = f(\gamma)^s$.

Finally, the data requestor recovers the value $e(g, g)^{\alpha_i s}$ by computing:

$$\begin{aligned}
& e(C', \prod_{\tau \in I} D'^{\omega_\tau}_\tau)/e(\prod_{\tau \in I} R^{\omega_\tau}_\tau, L) \\
&= e(g^s, \prod_{\tau \in I} g^{\lambda_\tau \omega_\tau} v(\gamma)^{r_\tau \omega_\tau})/e(\prod_{\tau \in I} g^{r_\tau \omega_\tau}, v(\gamma)^s) \\
&= e(g, g)^{\alpha_i s} \cdot e(g, v(\gamma))^{s \sum_{\tau \in I} r_\tau \omega_\tau}/e(g, v(\gamma))^{s \sum_{\tau \in I} r_\tau \omega_\tau} \\
&= e(g, g)^{\alpha_i s}
\end{aligned}$$

Note that $C_i = M_i \cdot e(g, g)^{\alpha_i s}$. Hence, the data requestor can obtain M_i from C_i. From the computation process, it can be seen that the decryption algorithm requires only two pairing operations.

3.3. Security Analysis

The security of the proposal is based on the hardness of the q-decision bilinear Diffie–Hellman exponent (q-DBDHE) problem defined below.

Definition 5. (q-Decision bilinear Diffie–Hellman exponent [21]). In (G, G_T), the q-DBDHE problem states that, given:

$$(g, g^x, g^{(x^2)}, \cdots g^{(x^q)}, g^{(x^{q+2})}, \cdots g^{(x^{2q})}, h, T)$$

where $x \in \mathbb{Z}_p, g, h \in G$ and $T \in G_T$, decide whether $T = e(g,h)^{x^{(q+1)}}$ or T is a random element of G_T.

The q-DBDHE assumption states that any probabilistic polynomial time (PPT) algorithm has only negligible advantage in solving the above q-DBDHE problem, where the advantage is defined as $|Pr[T = e(g,h)^{x^{(q+1)}}] - Pr[T = R]|$.

The security of the proposed BABE scheme is guaranteed by the following claim.

Theorem 1. *(Security). The BABE scheme is selectively secure against chosen plaintext attacks under the $|U| - DBDHE$ assumption in G.*

Proof. Let $|U|$ be the maximum of all of the $|U_i|$'s for security parameter λ. Assume that each attribute is indexed by a unique integer between one and $|U_i|$. Suppose there exists a PPT adversary $\mathcal{A}$ who can distinguish the encrypted message vectors with non-negligible probability under the selective security experiment [22]. Then, we show that there exists a PPT adversary $\mathcal{B}$ that can break the $|U| - DBDHE$ assumption in G, which contradicts the hardness of the $|U| - DBDHE$ problem.

Init: $\mathcal{A}$ chooses the underlying ABE schemes on which he or she wants to be challenged. For each sub-system, $\mathcal{A}$ outputs the attribute set $S_i^*, i \in [1,n]$ for the challenge ciphertext.

Setup: The challenger $\mathcal{B}$ is given the tuple:

$$(G, p, g, g^s, g^a, g^{(a^2)}, \cdots g^{(a^{|U|})}, g^{(a^{|U|+2})}, \cdots g^{(a^{2|U|})}, T)$$

for security parameter λ. The challenger is required to solve the $|U| - DBDHE$ challenge, *i.e.*, to answer whether $T = e(g,g)^{a^{|U|+1}s}$ or a random element in G_T. The challenger randomly chooses $\alpha_i', z_{i,1}, \cdots, z_{i,|U_i|} \in \mathbb{Z}_p, i \in [1,n]$ and sets $e(g,g)^{\alpha_i} = e(g,g)^{\alpha_i'} \cdot e(g^a, g^{a^{|U|}})$, which means that $\alpha_i = \alpha_i' + a^{|U|+1}$. For $x \in [1, |U|]$, compute:

$$h_{i,x} = g^{z_{i,x}}, x \in S_i^*$$
$$h_{i,x} = g^{z_{i,x}} g^{a^x}, x \notin S_i^*$$

The public parameter PK_i for the i-th subsystem is $(G, p, g, e(g,g)^{\alpha_i}, h_{i,1}, \cdots, h_{i,|U_i|})$ and is sent to the adversary $\mathcal{A}$.

Phase 1: For system i, the challenger $\mathcal{B}$ initializes an empty table T_i, an empty set D_i and a counter j_i = 0. The adversary $\mathcal{A}$ can query the challenger $\mathcal{B}$, and the challenger $\mathcal{B}$ responds as follows:

(1). *Create*(A_i): The challenger $\mathcal{B}$ sets $j_i = j_i + 1$. The access structure A_i is represented by (W, f), where W is an $l \times m$ matrix. Let K_i be the set of the rows in which the attributes are in S_i^* and K_i' be the rows in which the attributes are not in $S_i^*(i.e., K_i' = [1,l]/K_i)$. Define an m-element vector $\vec{v}$ over $\mathbb{Z}_p$ with the first element of $\vec{v}$ being one, for all $\tau \in K_i$, $\vec{v}W_\tau = 0$. W_τ is the vector row τ of the matrix W. Note that when S_i does not satisfy W, it must be the case that $(1, 0, \cdots, 0)$ is not in the span of rows W_τ for $\tau \in K_i$. With the property of LSSS, the vector $\vec{v}$ can be found in polynomial time. In addition, in order to share the secret α_i, $\mathcal{B}$ generates a secret key with the vector $\alpha_i\vec{v}$ by noting $v_1 = 1$. $\mathcal{B}$ computes $(\alpha_i\vec{v})W_\tau$ regarding share λ_τ for $\tau \in [1,l]$.

The challenger $\mathcal{B}$ randomly chooses r_1',,$\cdots$,$r_l' \in Z_q$ and $y_2, \cdots, y_m \in \mathbb{Z}_p$ to construct a new vector $\vec{v_{new}} = (0, y_2, y_3, \cdots, y_m)$. Then, it computes $\lambda_\tau' = \vec{v_{new}}W_\tau$ for $\tau \in [1,l]$.

For $\tau \in K_i$, compute $D'_\tau = g^{\lambda'_\tau} \cdot h_{i,f(\tau)}^{r'_\tau}, R'_\tau = g^{r'_\tau}, Q'_{\tau,x} = h_{i,x}^{r'_\tau}, \forall x \in \beta \setminus f(\tau)$.

For $\tau \in K'_i$, compute $c_\tau = \vec{v}W_\tau$. Set that $\lambda_\tau = c_\tau \cdot \alpha_i = c_\tau(\alpha'_i + a^{|U|+1})$. Set $R_\tau = g^{-c_\tau a^{|U|+1-f(\tau)}}$, which means $r_\tau = -c_\tau a^{|U|+1-f(\tau)}$. Then, set $R'_\tau = g^{-c_\tau a^{|U|+1-f(\tau)}} g^{r'_\tau}$, and compute:

$$\begin{aligned}
D'_\tau &= g^{c_\tau \alpha'_i} \cdot R_\tau^{z_{i,\tau}} \cdot g^{\lambda'_\tau} \cdot h_{i,f(\tau)}^{r'_\tau} \\
&= g^{c_\tau \alpha'_i} \cdot g^{-z_{i,\tau} c_\tau a^{|U|+1-f(\tau)}} \cdot g^{\lambda'_\tau} \cdot h_{i,f(\tau)}^{r'_\tau} \\
&= g^{c_\tau \alpha'_i} \cdot g^{-z_{i,\tau} c_\tau a^{|U|+1-f(\tau)}} \cdot g^{\lambda'_\tau} \cdot h_{i,f(\tau)}^{r'_\tau} \cdot g^{c_\tau a^{|U|+1}} \cdot g^{-c_\tau a^{|U|+1}} \\
&= g^{c_\tau \alpha_i} \cdot (g^{z_\tau})^{r_\tau} \cdot (g^{a^{f(\tau)}}) \cdot g^{\lambda'_\tau} \cdot h_{i,f(\tau)}^{r'_\tau} \\
&= g^{c_\tau \alpha_i} \cdot h_{i,f(\tau)}^{r_\tau} \cdot g^{\lambda'_\tau} \cdot h_{i,f(\tau)}^{r'_\tau}
\end{aligned}$$

Next, for all $x \in \beta \setminus f(\tau)$, $Q'_{\tau,x} = h_{i,x}^{r_\tau + r'_\tau}$ is computed as follows:

$$\begin{aligned}
&h_{i,x}^{r_\tau + r'_\tau} = g^{z_{i,x}(r_\tau + r'_\tau)} = g^{-z_{i,x} c_\tau a(|U|+1)-f(\tau)} g^{z_{i,x} r'_\tau}, x \in S_i^* \\
&h_{i,x}^{r_\tau + r'_\tau} = g^{z_{i,x}(r_\tau + r'_\tau)} (g^{a^x})^{(r_\tau + r'_\tau)} = g^{-z_{i,x} c_\tau a(|U|+1)-f(\tau)} g^{-c_\tau a(|U|+1)-f(\tau)+x} \cdot (g^{z_{i,x}} g^{a^x})^{r'_\tau}, x \notin S_i^*
\end{aligned}$$

After the process above, the keys are valid with access policy (W, f). From the view of $\mathcal{A}$, their distribution is identical to that of the keys generated by the *KeyGen* algorithm. Hence, the secret keys requested by the adversary is perfectly simulated.

(2). *Corrupt*(κ, i): The challenger $\mathcal{B}$ obtains the entry $(\kappa; A_i; SK)$ if the κ-th entry appears in table T_i. Set $D_i = D_i \bigcup \{A_i\}$. It then sends SK to adversary $\mathcal{A}$. If no such entry exists, then return $\perp$.

Challenge: $\mathcal{A}$ outputs two equal-length message vectors $\vec{M}_0, \vec{M}_1$, each of which contains n messages in the message space of the corresponding subsystem. The challenger $\mathcal{B}$ chooses a random bit b. It constructs and sends to $\mathcal{A}$ the challenge ciphertext:

$$CT_i^* = (C_i^* = M_b \cdot T \cdot e(g^{\alpha'_i}, g^s), C' = g^s, \forall x \in S^*, C_{i,x}^* = (g^s)^{z_{i,x}})$$

Phase 2: $\mathcal{B}$ responds to $\mathcal{A}$'s queries in the same manner as in Phase 1, except that it refuses to answer any *Corrupt* query that would result in an access structure A_i, which S_i satisfies.

Guess: Finally, $\mathcal{A}$ outputs a bit b'. If $b = b'$, then $\mathcal{B}$ outputs that $T = e(g, g)^{a^{|U|+1}s}$, else it outputs that T is a random element.

3.4. Efficiency Analysis

In this section, we analyze the performance of our BABE scheme. It can be seen that our approach is more efficient than encrypting data for different organizations one by one using the underlying Hohenberger-Waters ABE (HW-ABE) [2] scheme. In the encryption process, the number of exponentiations grows with the number of attributes. Comparing to the plain implementation of the HW-ABE scheme, our approach is more efficient because C' is the same for all of the ciphertexts. Hence, it can be calculated once and then be reused. Assume that the data owner wants to share data with the users of n organizations simultaneously, and the average number of the attributes used in encryption is m. Our scheme needs $(m + 1)n + 1$ exponentiations, while the HW-ABE scheme needs $(m + 2)n$ exponentiations. From the comparison, the improved efficiency is dominated by the average number of the attributes used by the data owner. We can save $n - 1$ exponentiations in the encryption process. The computation overhead for the data requestors is also very small. They need only two pairings and a number of multiplications. This is the same as the HW-ABE scheme [2].

We encoded our BABE and performed experiments on a PC with a 2.80 GHz CPU and 3 GB RAM, using the C programming language in the GMP (https://gmplib.org/) and PBC (https://crypto.stanford.edu/pbc/) library. For the encryption algorithms: (1) we generated n different 1024 bit-long random numbers as exponents; (2) we implemented the batch algorithm and normal algorithm, respectively. We executed Steps 1 and 2 100 times to eliminate experimental noise. We set the values of n ranging from 10 to 100 to see the performance improvement due to the batch algorithms.

Figure 2 show the time to compute n ciphertext in the HW-ABE and BABE with the average attribute number $m = 4$ or $m = 8$. It can be seen that the time consumed in both BABE and HW-ABE grows linearly with the number of messages to be encrypted. However, the batch encryption takes about 15% less time for $m = 4$ or 10% less time for $m = 8$ than the normal encryption, which confirms the theoretical analysis.

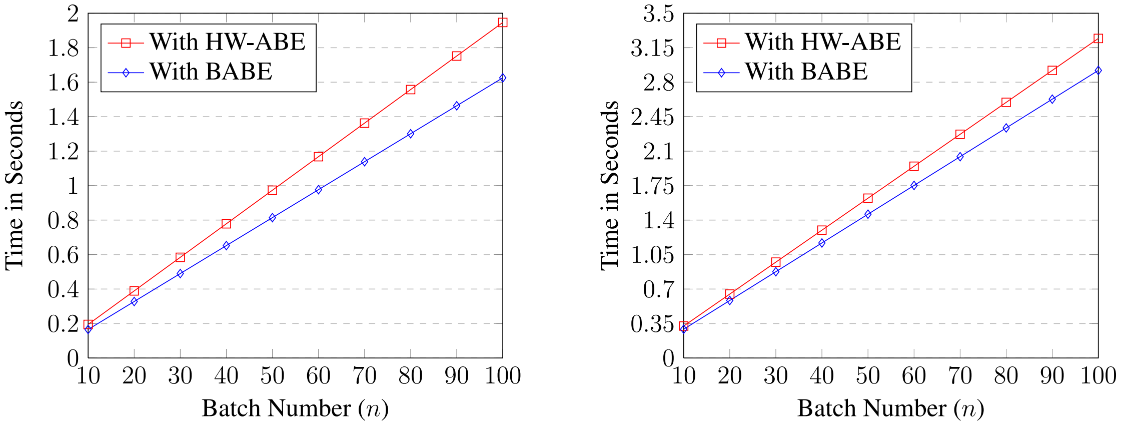

Figure 2. The encryption performance comparison for producing multiple ciphertexts with BABE/HW-ABE ($m = 4, m = 8$).

4. Conclusions

In this paper, we proposed BABE to accelerate the ABE encryption for the users of multiple organizations. Our approach makes the data owner able to share his or her data with the data requestors of different systems simultaneously. We formally proved the security of our scheme and compared it to previous work. The analyses and comparisons show that our scheme has desirable high performance. It is efficient to secure data to be outsourced to clouds.

Acknowledgments

This paper is partially supported by the National Key Basic Research Program (973 Program) under Project 2012CB315905, by the Natural Science Foundation under Projects 61370190, 61272501, 61173154, 61402029 and 61003214, by the Beijing Natural Science Foundation through Projects 4132056 and 4122041, by the Natural Science Foundation of Guangdong Province through Projects 2014A030313517 and S2012020011071, the Fundamental Research Funds for the Central Universities, the Research Funds (No. 14XNLF02) of Renmin University of China, the Open Research Fund of the State Key Laboratory of Information Security (Institute of Information Engineering), the Chinese Academy of Sciences and the Open Research Fund of Beijing Key Laboratory of Trusted Computing.

Author Contributions

Chen Yang mainly designed the research. Yang Sun mainly conducted the research and analyzed the data. Qianong Wu mainly wrote the paper. All authors have read and approved the final manuscript.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Sahai, A.; Waters, B. Fuzzy identity-based encryption. In *Advances in Cryptology—EUROCRYPT 2005*, Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, 22–26 May 2005; Springer: Berlin/Heidelberg, Germany, 2005; Lecture Notes in Computer Science, Volume 3494; pp. 457–473.
2. Hohenberger, S.; Waters, B. Attribute-based encryption with fast decryption. In *Public-Key Cryptography–PKC 2013*, Proceedings of the 16th International Conference on Practice and Theory in Public-Key Cryptography, Nara, Japan, 26 February–1 March 2013; Springer: Berlin/Heidelberg, Germany, 2013; Lecture Notes in Computer Science, Volume 7778; pp. 162–179.
3. Goyal, V.; Pandey, O.; Sahai, A.; Waters, B. Attribute-based encryption for fine-grained access control of encrypted data. In Proceedings of the 13th ACM Conference on Computer and Communications Security, Alexandri, VA, USA, 30 October–3 November 2006; pp. 89–98.
4. Bethencourt, J.; Sahai, A.; Waters, B. Ciphertext-policy attribute-based encryption. In Proceedings of the SP'07 IEEE Symposium on Security and Privacy, Berkeley, CA, USA, 20–23 May 2007; pp. 321–334.
5. Waters, B. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *Public Key Cryptography (PKC) 2011*, Proceedings of the 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, 6–9 March 2011; Springer: Berlin/Heidelberg, Germany, 2011; Lecture Notes in Computer Science, Volume 6571; pp. 53–70.
6. Yu, S.; Wang, C.; Ren, K.; Lou, W. Achieving secure, scalable, and fine-grained data access control in cloud computing. In Proceedings of the IEEE INFOCOM, San Diego, CA , USA, 14–19 March 2010; pp. 1–9.
7. Chase, M. Multi-authority attribute based encryption. In *Theory of Cryptography*, Proceedings of the 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, 21–24 February 2007; Springer: Berlin/Heidelberg, Germany, 2007; Lecture Notes in Computer Science, Volume 4392; pp. 515–534.

8. Chase, M.; Chow, S.S. Improving privacy and security in multi-authority attribute-based encryption. In Proceedings of the 16th ACM Conference on Computer and Communications Security, Chicago, IL, USA, 9–13 November 2009; pp. 121–130.
9. Yang, K.; Jia, X. Attributed-based access control for multi-authority systems in cloud storage. In Proceedings of the 2012 IEEE 32nd International Conference on Distributed Computing Systems (ICDCS), Macau, China, 18–21 June 2012; pp. 536–545.
10. Li, J.; Huang, Q.; Chen, X.; Chow, S.S.; Wong, D.S.; Xie, D. Multi-authority ciphertext-policy attribute-based encryption with accountability. In Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, Hong Kong, China, 22–24 March 2011; pp. 386–390.
11. Attrapadung, N.; Libert, B.; de Panafieu, E. Expressive key-policy attribute-based encryption with constant-size ciphertexts. In *Public Key Cryptography–PKC 2011*, Proceedings of the 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, 6–9 March 2011; Springer: Berlin/Heidelberg, Germany, 2011; Lecture Notes in Computer Science, Volume 6571; pp. 90–108.
12. Attrapadung, N.; Herranz, J.; Laguillaumie, F.; Libert, B.; de Panafieu, E.; Ràfols, C. Attribute-based encryption schemes with constant-size ciphertexts. *Theor. Comput. Sci.* **2012**, *422*, 15–38.
13. Fiat, A. Batch rsa. In *Advances in Cryptology CRYPTO 89 Proceedings*; Springer: Berlin/ Heidelberg, Germany, 1990; Lecture Notes in Computer Science, Volume 435; pp. 175–185.
14. Bellare, M.; Garay, J.A.; Rabin, T. Fast batch verification for modular exponentiation and digital signatures. In *Advances in Cryptology EUROCRYPT 98*, Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques Espoo, Finland, 31 May–4 June 1998; Springer: Berlin/Heidelberg, Germany, 1998; Lecture Notes in Computer Science, Volume 1403; pp. 236–250.
15. Zhang, C.; Lu, R.; Lin, X.; Ho, P.H.; Shen, X. An efficient identity-based batch verification scheme for vehicular sensor networks. In Proceedings of the IEEE INFOCOM 2008, the 27th Conference on Computer Communications, Phoenix, AZ, USA, 13–18 April 2008.
16. Ferrara, A.L.; Green, M.; Hohenberger, S.; Pedersen, M.Ø. Practical short signature batch verification. In *Topics in Cryptology—CT-RSA 2009*, Proceedings of the The Cryptographers' Track at the RSA Conference 2009, San Francisco, CA, USA, 20–24 April 2009; Springer: Berlin/Heidelberg, Germany, 2009; Lecture Notes in Computer Science, Volume 5473; pp. 309–324.
17. Camenisch, J.; Hohenberger, S.; Pedersen, M.Ø. Batch verification of short signatures. In *Advances in Cryptology—EUROCRYPT 2007*, Proceedings of the 26th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Barcelona, Spain, 20–24 May 2007; Springer: Berlin/Heidelberg, Germany, 2007; Lecture Notes in Computer Science, Volume 4515; pp. 246–263.

18. Duan, S.; Cao, Z. Efficient and provably secure multi-receiver identity-based signcryption. In *Information Security and Privacy*, Proceedings of the 11th Australasian Conference, ACISP 2006, Melbourne, Australia, 3–5 July 2006; Springer: Berlin/Heidelberg, Germany, 2006; Lecture Notes in Computer Science, Volume 4058; pp. 195–206.
19. Baek, J.; Safavi-Naini, R.; Susilo, W. Efficient multi-receiver identity-based encryption and its application to broadcast encryption. In *Public Key Cryptography-PKC 2005*, Proceedings of the 8th International Workshop on Theory and Practice in Public Key Cryptography, Les Diablerets, Switzerland, 23–26 January 2005; Springer: Berlin/Heidelberg, Germany, 2005; Lecture Notes in Computer Science, Volume 3386; pp. 380–397.
20. Beimel, A. Secure Schemes for Secret Sharing and Key Distribution. Ph.D. Thesis, Technion-Israel Institute of Technology, Haifa, Israel, 1996.
21. Boneh, D.; Boyen, X.; Goh, E.J. Hierarchical identity based encryption with constant size ciphertext. In *Advances in Cryptology–EUROCRYPT 2005*, Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, 22–26 May 2005; Springer: Berlin/Heidelberg, Germany, 2005; Lecture Notes in Computer Science, Volume 3494; pp. 440–456.
22. Boneh, D.; Gentry, C.; Waters, B. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *Advances in Cryptology–CRYPTO 2005*, Proceedings of the 25th Annual International Cryptology Conference, Santa Barbara, CA, USA, 14–18 August 2005; Springer: Berlin/Heidelberg, Germany, 2005; Lecture Notes in Computer Science, Volume 3621; pp. 258–275.