Fine-Grained Data Sharing Supporting Attribute Extension in Cloud Computing

Yinghui Zhang[1,2]

[1] **National Engineering Laboratory for Wireless Security,**
Xi'an University of Posts and Telecommunications, Xi'an 710121, P.R. China

[2] **State Key Laboratory of Information Security, Institute of Information Engineering,**
Chinese Academy of Sciences, Beijing 100093, P.R. China

Abstract

Attribute-based encryption (ABE) can be used for implementing fine-grained data sharing in cloud computing. However, most of the existing ABE schemes cannot realize attribute extension and provable security simultaneously. In this paper, we propose a fine-grained attribute-based data sharing system based on a hybrid encryption mechanism. A rigorous security proof indicates that the proposed scheme is selective-secure under the decisional bilinear Diffie-Hellman assumption. In particular, the proposed data sharing scheme can efficiently support attribute extension and allow AND-gate access policies with multiple attribute values and wildcards. Extensive simulation results indicate that the proposed scheme is extremely suitable for data sharing in cloud computing.

Keywords: *Data Sharing, Attribute-Based Encryption, Attribute Extension, Cloud Computing.*

1. Introduction

As a promising computing paradigm, cloud computing has the advantage that it offers users unlimited computation and storage ability at favorable costs. Although the advantages of cloud computing are desirable, data security issues have impeded users from purchasing such services. Traditional access control methods are not suitable for cloud computing in that it requires users to fully trust the storage server.

As a highly promising public key primitive, attribute-based encryption (ABE) [1] realize one-to-many encryption and it is suitable for realizing fine-grained data sharing in cloud computing. There are two kinds of ABE schemes, that is key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE). In CP-ABE, users can apply attribute secret keys from the attribute center based on their own attributes. During the encryption phase, encryptors can specify access policies themselves and then encrypt data files. A decryptor can recover files from ciphertexts only if his/her attributes satisfy the underlying access policy. All these desirable properties make CP-ABE extremely suitable for data sharing in cloud computing.

However, to the authors' knowledge, state-of-the-art CP-ABE schemes fail to achieve provable security and support attribute extension, simultaneously. In this paper, we address the problem by proposing a fine-grained attribute-based data sharing system based on a hybrid encryption mechanism, where the public encryption is a new CP-ABE construction. The proposed scheme is proven secure under the decisional bilinear Diffie-Hellman assumption and it can support attribute extension. Simulation results indicate that the proposed scheme is extremely suitable for data sharing in cloud computing.

The rest of this paper is organized as follows. In Section 2, we review the previous ABE schemes. Section 3 gives some cryptographic preliminaries. In Section 4, we present the system architecture of data sharing in cloud computing and formalize the security model of CP-ABE. The proposed data sharing scheme is detailed in Section 5. Security results and performance analysis are discussed in Section 6. Finally, we conclude this work in Section 7.

2. Related Work

Sahai and Waters [1] introduced ABE as a fuzzy version of identity-based encryption. Since then, a plenty of researches have been done on ABE schemes. Ostrovsky *et al.* [2] proposed the first KP-ABE system supporting non-monotone key policies. The first CP-ABE scheme is constructed by Bethencourt *et al.* [3], but the security proof is given in the generic group model. To address this issue, Cheung and Newport [4] proposed another CP-ABE scheme that is proven secure in the standard model. Li [5] proposed attribute-based proxy re-encryption scheme with matrix access policies. In practical applications, users' attribute may update frequently and hence attribute revocation mechanism is important. There are three kinds of revocation mechanisms, that is the timed rekeying mechanism [6], the indirect revocation [7] and the direct revocation [8-9]. Note that the direct revocation is most desirable because it does not require users to update attribute secret keys periodically. In the scheme [9], the authors formalize the notion of ciphertext-policy ABE supporting flexible and direct revocation, and present a concrete scheme which enjoys desirable properties of no secret key update, partial ciphertext update and constant-size ciphertext. Privacy protection is indispensable for users of cloud computing platforms [10]. To further preserve users' attribute privacy, anonymous ABE schemes [11-12] were proposed, where access polices are not directly disclosed in ciphertexts. In particular, a novel technique called match-then-decrypt was introduced in [12] to improve the decryption efficiency. Most of the above ABE schemes suffer a severe drawback that the ciphertext length linearly grows with the number of attributes the user has to hold for successful decryption. In order to address this problem, many researchers focus on ABE with constant-size ciphertexts [13-16]. Marwaha *et al.* [17] pointed that it is feasible to apply encryption algorithm to realize data security and privacy protection in cloud computing. There are also many works proposed to make further improvements on ABE, such as key-evolving attribute-based signcryption [18], and outsourced ABE [19]. However, most of the existing CP-ABE schemes fail to achieve provable security and support attribute extension, simultaneously.

3. Preliminaries

In this section, we review some cryptographic backgrounds and describe access policies.

3.1 Bilinear Pairing

Let $\mathbb{G}$ be a cyclic multiplicative group of a prime order p, $g \in_R \mathbb{G}$ be a generator, and $\mathbb{G}_T$ be a cyclic multiplicative group of the same order, whose identity we denote as 1. We call $\hat{e}$ a bilinear pairing if $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a map with the following properties:

- Bilinear: $\hat{e}(g^a, g^b) = \hat{e}(g,g)^{ab}$ for all $a, b \in \mathbb{Z}_p^*$.
- Non-degenerate: There exists $g_1, g_2 \in \mathbb{G}$ such that $\hat{e}(g_1, g_2) \neq 1$.
- Computable: $\hat{e}(g_1, g_2)$ can be efficiently computed for all $g_1, g_2 \in \mathbb{G}$.

3.2 Complexity Assumption

The Decisional Bilinear Diffie-Hellman (DBDH) Assumption: Let $\mathbb{G}$ be a cyclic multiplicative group of a prime order p and $g \in_R \mathbb{G}$ be a generator. Let a, b, c be random elements in $\mathbb{Z}_p$. The DBDH assumption is that no probabilistic polynomial-time algorithm can distinguish the tuple $[g^a, g^b, g^c, \hat{e}(g,g)^{abc}]$ and $[g^a, g^b, g^c, \hat{e}(g,g)^z]$ with non-negligible advantage.

3.3 Access Policy

The proposed scheme supports AND gate access policies. Formally, given an attribute list $L = [L_1, L_2, \cdots, L_n]$ and an access policy $W = [W_1, W_2, \cdots, W_n]$, we say $L \models W$ if $L_i = W_i$ or $W_i = *$ for $1 \leq i \leq n$, otherwise we say $L \not\models W$. The symbol $*$ in W means that the corresponding attribute is not cared.

4. System Architecture and Security Model

In this section, we present the system architecture of data sharing in cloud computing and give the security model.

4.1 System Architecture

The system architecture is shown in Figure 1. The attribute center generates system public parameters and a master secret key. Both the encryptor and the decryptor apply attribute secret keys from the attribute center. The encryptor encrypts files and sends ciphertexts to the cloud storage server for sharing. The decryptor downloads ciphertexts and recovers corresponding files based on his/her attribute secret key.

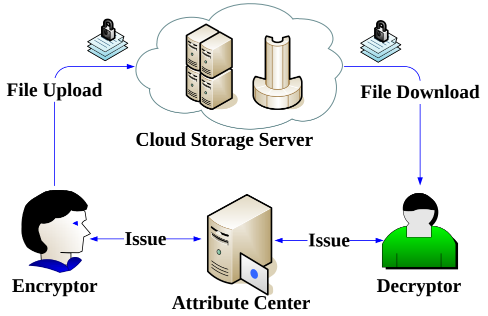

Fig.1 The system architecture of data sharing.

4.2 Security Model

The proposed scheme uses the following security model called indistinguishability against selective ciphertext-policy and chosen-message attacks IND-sCP-CPA.

Init: The adversary $\mathcal{A}$ commits to a challenge access policy W^*.

Setup: The challenger $\mathcal{S}$ chooses a sufficiently large security parameter λ, and runs the $\mathbf{Setup}$ algorithm to get a master key SK and the corresponding system public key PK. It gives PK to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ issues a polynomially bounded number of key generation queries: $\mathcal{A}$ submits an attribute list L, if

$L \not\models W^*$, $\mathcal{S}$ gives $\mathcal{A}$ the secret key SK_L and outputs the symbol $\perp$ otherwise.

Challenge: Once $\mathcal{A}$ decides that $\mathbf{Phase\ 1}$ is over, it outputs two messages M_0 and M_1 of equal length. $\mathcal{S}$ randomly chooses a bit $b \in \{0,1\}$, computes $CT_{W^*} = \mathbf{Encrypt}(PK, M_b, W^*)$ and sends CT_{W^*} to $\mathcal{A}$.

Phase 1: The same as $\mathbf{Phase\ 1}$.

Guess: $\mathcal{A}$ outputs a guess bit $b' \in \{0,1\}$ and wins the game if $b' = b$. The advantage of $\mathcal{A}$ is defined as:

$$\mathbf{Adv}_{\text{CP-ABE}}^{\text{IND-sCP-CPA}}(\mathcal{A}) = |\Pr[b' = b] - \frac{1}{2}|.$$

5. Fine-Grained Data Sharing System

In this section, we describe the proposed fine-grained data sharing system.

5.1 System Initialization

The attribute center chooses a security parameter λ and runs the following $\mathbf{Setup}$ algorithm to generate a system public parameter PK and a master secret key MK.

$\mathbf{Setup}(1^\lambda)$: Let $\mathbb{G}, \mathbb{G}_T$ be cyclic multiplicative groups of prime order p , and $\hat{e}: \mathbb{G} \times \mathbb{G} \ \rightarrow \mathbb{G}_T$ be a bilinear map. Assume there are n attributes in universe and the universal attribute set is $\mathcal{U} = \{\omega_1, \omega_2, \cdots, \omega_n\}$. Suppose each attribute has multiple values and the multi-value set for ω_i is $S_i = \{v_{i,1}, v_{i,2}, \cdots, v_{i,n_i}\}$. The attribute center chooses $y \in_R \mathbb{Z}_p$ and $g_2 \in_R \mathbb{G}$. For each attribute ω_i where $1 \leq i \leq n$, the attribute center also chooses $\{T_{i,t} \in_R \mathbb{G}\}_{1 \leq t \leq n_i}$ and $\{a_i, b_i \in_R \mathbb{Z}_p\}$. Next it computes $g_1 = g^y, Y = \hat{e}(g_1, g_2)$ and $\{A_i = g^{a_i}, B_i = g^{b_i}\}_{1 \leq i \leq n}$. Finally, the system public key is published as

$$PK = \langle \hat{e}, g, g_1, g_2, Y, \{\{T_{i,t}\}_{1 \leq t \leq n_i}, A_i, B_i\}_{1 \leq i \leq n} \rangle,$$

and the master key is $MK = \langle y, \{a_i, b_i\}_{1 \leq i \leq n} \rangle$.

5.2 User Registration

When a new user applies to join the data sharing system, the attribute center generates an attribute secret key based on the $\mathbf{KeyGen}$ algorithm.

$\mathbf{KeyGen}(PK, MK, L)$: Let $L = [L_1, L_2, \cdots, L_n]$ be the attribute list for the user who obtains the corresponding attribute secret key. The attribute center chooses $r \in_R \mathbb{Z}_p$, $\{r'_i, r''_i \in_R \mathbb{Z}_p\}_{1 \leq i \leq n}$, and computes $D_0 = g_2^{y-r}$. If $L_i = v_{i,k_i}$, set $D_i = (g_2^r T_{i,k_i}^{r'_i}, g^{r'_i})$, else if $L_i = \neg\omega_i$, set $D_i = (g_2^r A_i^{r'_i}, g^{r'_i})$. The attribute center computes $F_i = (g_2^r B_i^{r''_i}, g^{r''_i})$. Finally, the attribute secret key is $SK_L = \langle L, D_0, \{D_i, F_i\}_{1 \leq i \leq n} \rangle$.

5.3 File Upload

Before uploading a file $\mathcal{F}$ to cloud storage servers, a data owner has to encrypt $\mathcal{F}$ with K based on a symmetric encryption mechanism to obtain a ciphertext CT_0. Then the data owner defines an access policy W for $\mathcal{F}$, and runs the following $\mathbf{Encrypt}$ algorithm to get a ciphertext CT_W of K. At last, he/she sets $CT_{\mathcal{F}} = \{CT_0, CT_W\}$ as the final ciphertext and uploads it to the cloud storage server.

$\mathbf{Encrypt}(PK, M, W)$: To encrypt a message $M \in \mathbb{G}_T$ under an access policy $W = [W_1, W_2, \cdots, W_n]$, an encryptor chooses $\{s_i \in_R \mathbb{Z}_p\}_{1 \leq i \leq n}$, sets $s = \sum_{i=1}^n s_i$, and computes $\widetilde{C} = M \cdot Y^s$, $C_0 = g^s$. Then for $1 \leq i \leq n$, the encryptor computes $C_{i,1} = g^{s_i}$, $C_{i,2} = T_{i,k_i}^{s_i}$ if $W_i = v_{i,k_i}$, $C_{i,2} = A_i^{s_i}$ if $W_i = \neg\omega_i$, and $C_{i,2} = B_i^{s_i}$ if $W_i = *$. Finally, the ciphertext of M with respect to W is $CT_W = \langle W, \widetilde{C}, C_0, \{C_{i,1}, C_{i,2}\}_{1 \leq i \leq n} \rangle$.

5.4 File Download

If a user wants to access a file, he/she first downloads the ciphertext $CT_{\mathcal{F}} = \{CT_0, CT_W\}$. Then performs

$$K = \mathsf{Decrypt}(PK, CT_W, SK_L),$$

and retrieves the file from K and CT_0 based on the symmetric decryption.

$\mathbf{Decrypt}(PK, CT_W, SK_L)$: A decryptor checks whether $L \models W$ or not. If $L \models W$, the ciphertext CT_W can be decrypted as follows. If $W_i \neq *$, set $D'_i = D_{i,1}$ and $D''_i = D_{i,2}$, else if $W_i = *$, set $D'_i = F_{i,1}$ and $D''_i = F_{i,2}$. Then the plaintext is recovered as

$$M = \frac{\widetilde{C} \prod_{i=1}^n \hat{e}(C_{i,2}, D''_i)}{\hat{e}(C_0, D_0) \prod_{i=1}^n \hat{e}(C_{i,1}, D'_i)}.$$

6. Security and Performance Analysis

In this section, the proposed scheme is proved to be secure, and the performance analysis is given.

6.1 Consistency

Suppose $L \models W$ and $L_i = v_{i,k_i}$, we have

$$\begin{aligned} & \frac{\widetilde{C} \prod_{i=1}^n \hat{e}(C_{i,2}, D''_i)}{\hat{e}(C_0, D_0) \prod_{i=1}^n \hat{e}(C_{i,1}, D'_i)} \\ = & \frac{M \cdot Y^s \prod_{i=1}^n \hat{e}(T_{i,k_i}^{s_i}, g^{r'_i})}{\hat{e}(g^s, g_2^{y-r}) \prod_{i=1}^n \hat{e}(g^{s_i}, g_2^r T_{i,k_i}^{r'_i})} \\ = & \frac{M \cdot Y^s}{\hat{e}(g^s, g_2^{y-r}) \prod_{i=1}^n \hat{e}(g^{s_i}, g_2^r)} = M. \end{aligned}$$

6.2 Security Result

In the proposed construction, files are encrypted based on a hybrid encryption mechanism [20]. To be specific, files are protected by a symmetric key, which is encrypted based on a ciphertext-policy attribute-based encryption scheme. Hence, we need to prove the security of CP-ABE.

Theorem 1. *The proposed CP-ABE scheme is secure in the IND-sCP-CPA model under the DBDH assumption without random oracles.*

Proof. Suppose there exists an adversary $\mathcal{A}$ that can attack the proposed scheme with advantage ϵ. We build a simulator $\mathcal{S}$ that can solve the DBDH problem with advantage $\frac{\epsilon}{2}$. The DBDH challenger flips a fair binary coin μ outside of $\mathcal{S}$'s view. If $\mu = 0$, the challenger sets $[A, B, C, Z] = [g^a, g^b, g^c, \hat{e}(g,g)^{abc}]$. Otherwise, it sets $[A, B, C, Z] = [g^a, g^b, g^c, \hat{e}(g,g)^z]$ for random a, b, c, z. The simulation proceeds as follows:

$\mathbf{Init}$: The simulator $\mathcal{S}$ runs $\mathcal{A}$ and receives a challenge policy W^*.

$\mathbf{Setup}$: To generate a system public key, $\mathcal{S}$ sets $g_1 = A$, $g_2 = B$ and computes $Y = \hat{e}(A, B)$. $\mathcal{S}$ chooses $\{t_{i,j}\}_{1 \le j \le n_i}, a_i, b_i \in_R \mathbb{Z}_p$ and does the following:

✧ If $W_i^* = v_{i,k_i}$, then $T_{i,k_i} = g^{t_{i,k_i}}$, $T_{i,j} = B^{t_{i,j}}$ for $j \neq k_i$, $A_i = B^{a_i}$, $B_i = B^{b_i}$;

✧ If $W_i^* = \neg\omega_i$, then $T_{i,j} = g^{t_{i,j}}$, $A_i = g^{a_i}$, $B_i = B^{b_i}$;

✧ If $W_i^* = *$, then $T_{i,j} = B^{t_{i,j}}$, $A_i = B^{a_i}$, $B_i = g^{b_i}$.

Finally, $\mathcal{S}$ sends $\mathcal{A}$ the system public key $PK = \langle \hat{e}, g, g_1, g_2, Y, \{\{T_{i,t}\}_{1 \le t \le n_i}, A_i, B_i\}_{1 \le i \le n} \rangle$.

$\mathbf{Phase1}$: $\mathcal{S}$ answers $\mathcal{A}$'s secret key queries as follows:

✧ $\mathcal{O}_{KeyGen}(L)$: Suppose $\mathcal{A}$ submits $L = [L_1, L_2, \cdots, L_n]$ in order to obtain a corresponding secret key. Only the case where $L \not\models W^*$ is taken into consideration. $\mathcal{S}$ chooses $r' \in \mathbb{Z}_p$ and sets $r = a + r'$. Then $D_0 = B^{-r'} = g_2^{a-r}$. $\mathcal{A}$ computes D_i and F_i as follows: If $L_i = v_{i,k_i}$,$\mathcal{S}$ chooses $\beta \in \mathbb{Z}_p$ and computes $r_i' = \frac{\beta - a}{t_{i,k_i}}$, then sets $D_i = (B^{\beta + r'}, g^{\frac{\beta}{t_{i,k_i}}} A^{-\frac{1}{t_{i,k_i}}})$=$(g_2^r T_{i,k_i}^{r_i'}, g^{r_i'})$; If $L_i = \neg\omega_i$,$\mathcal{S}$ chooses $\hat{r}_i \in \mathbb{Z}_p$ and sets $D_i = (B^{\beta + r'} (T_{i,k_i}^{r_i'})^{-1} A_i^{\hat{r}_i}, g^{\hat{r}_i})$=$(g_2^r A_i^{\hat{r}_i}, g^{\hat{r}_i})$. In addition, $\mathcal{S}$ chooses $\delta \in \mathbb{Z}_p$ and computes $r_i'' = \frac{\delta - a}{b_i}$, then sets

$$F_i = (B^{\delta + r'}, g^{\frac{\delta}{b_i}} A^{-\frac{1}{b_i}})=(g_2^r B_i^{r_i''}, g^{r_i''}).$$

$\mathbf{Challenge}$: $\mathcal{A}$ submits two challenge messages M_0 and M_1. $\mathcal{S}$ chooses $\{s_i \in_R \mathbb{Z}_p\}_{\le i \le n-1}$ and sets $s_n = c - \sum_{i=1}^{n-1} s_i$. For $1 \le i \le n-1$, $\mathcal{S}$ sets $C_{i,1} = g^{s_i}$. In addition, for $i = n$, $\mathcal{S}$ sets

$$C_{n,1} = C \cdot \prod_{i=1}^{n-1} g^{-s_i} = g^{c - \sum_{i=1}^{n-1} s_i} = g^{s_n},$$

and does the following: if $W_i^* = v_{i,k_i}$, then $C_{i,2} = T_{i,k_i}^{s_i}$; if $W_i^* = \neg\omega_i$, then $C_{i,2} = C_{i,1}^{a_i}$; if $W_i^* = *$, then $C_{i,2} = C_{i,1}^{b_i}$. Finally $\mathcal{S}$ sets $\widetilde{C} = M_\nu \cdot Z$, $C_0 = C$, and sends $\mathcal{A}$ the ciphertext

$$CT_{W^*} = \langle W^*, \widetilde{C}, C_0, \{C_{i,1}, C_{i,2}\}_{1 \le i \le n} \rangle,$$

where $\nu \in \{0, 1\}$ is a random coin.

$\mathbf{Phase2}$: The same as $\mathbf{Phase1}$.

$\mathbf{Guess}$: $\mathcal{A}$ outputs a guess bit ν'. If $\nu = \nu'$, $\mathcal{S}$ outputs $\mu' = 0$ to indicate that it was given a DBDH tuple. Otherwise, it will output $\mu' = 1$ to indicate that it was given a random tuple. In the case $\mu = 1$, $\mathcal{A}$ gains no information about ν, and we have $\Pr[\nu \neq \nu' | \mu = 1] = 1/2$. Because $\mathcal{S}$ guesses $\mu' = 1$ when $\nu \neq \nu'$, we have $\Pr[\mu' = \mu | \mu = 1] = 1/2$. In the case $\mu = 0$, $\mathbf{Adv}_{\mathcal{A}} = \epsilon$ by definition. Hence we know $\Pr[\nu = \nu' | \mu = 0] = 1/2 + \epsilon$. Since $\mathcal{S}$ guesses $\mu' = 0$ when $\nu = \nu'$, we have $\Pr[\mu' = \mu | \mu = 0] = 1/2 + \epsilon$. Therefore, $\mathcal{S}$ can break the DBDH assumption with an advantage $\mathbf{Adv}_{\mathrm{DBDH}}$ as below:

$$\begin{aligned} & \mathbf{Adv}_{\mathrm{DBDH}} \\ = \; & \frac{1}{2}\Pr[\mu' = \mu | \mu = 0] + \frac{1}{2}\Pr[\mu' = \mu | \mu = 1] - \frac{1}{2} \\ = \; & \frac{1}{2}(\frac{1}{2} + \epsilon) + \frac{1}{2}\frac{1}{2} - \frac{1}{2} = \frac{1}{2}\epsilon. \end{aligned}$$ ■

6.3 Attribute Extension

In the proposed scheme, a new value can be added to the system after the setup phase. If the encryptor chooses a ciphertext policy associated with new attributes, the corresponding ciphertexts cannot be decrypted based on previous attributes. In order to successfully recover messages, the decryptor has to apply new attribute secret keys from the attribute center. In fact, in the proposed scheme, the encryptor computes $\widetilde{C} = M \cdot Y^s$ and the secret exponent s is split to a total of n secrets, which are used to generate other ciphertext components $C_{i,1} = g^{s_i}$ and $C_{i,2}$. Obviously, the construction requires decryptors have the attribute secret key components for all the attributes appeared in the access policy. Suppose after the setup phase, there are four attributes $\omega_1, \omega_2, \omega_3, \omega_4$ in the data sharing system. A user receives his/her attribute secret key SK_L with $L = [L_1, L_2, L_3, L_4] = [0, 1, 0, 1]$. Then a new attribute ω_5 is added to the system. Suppose an access policy $W = [W_1, W_2, W_3, W_4] = [0, *, 0, *, 1]$ is chosen by the encryptor. It requires the legitimate decryptor to have the value 1 for the new attribute ω_5. Hence, the above user with SK_L fails to recover messages because he/she cannot reconstruct the secret exponent s for decryption. Accordingly, the proposed data sharing scheme can support attribute extension.

6.4 Performance Analysis

In this section, we compare the previous CP-ABE scheme [11] denoted as NYO08 and the proposed scheme in terms of encryption cost and decryption cost in Figure 2 and Figure 3, respectively.

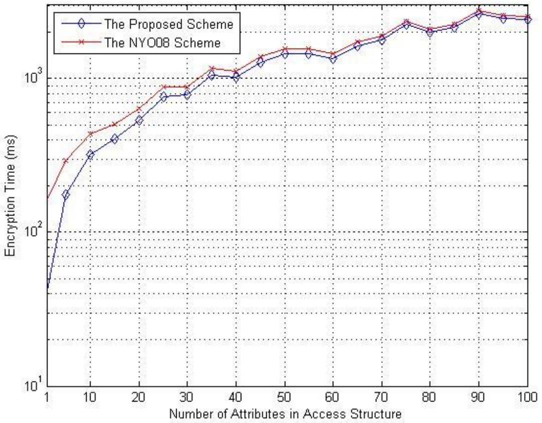

Fig.2 Comparison of encryption cost.

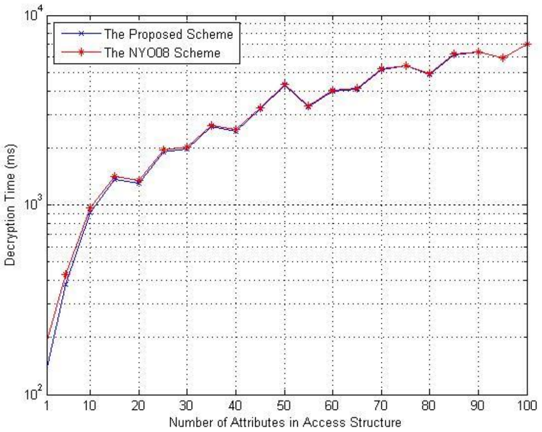

Fig.3 Comparison of decryption cost.

Both schemes can support attribute extension. It's noted that the vertical axis is log scale. For the sake of precision, the simulation experiments are performed based on the Stanford Pairing-Based Crypto library of version 0.5.14 [21] and a Linux machine with 3.30 GHz 8 Intel Xeon(R) E3-1230 V2 CPU and 7.5 GB of RAM. In our experiments, we consider the worst case of access structures, which ensures that all the ciphertext components are involved in decryption. In fact, 100 distinct access structures are generated in the form of $(W_1 \wedge W_2 \wedge \cdots \wedge W_t)$ with t increasing from 1 to 100, where each component W_i is required for decryption. For each access policy, the experiment is repeated 50 times and the average values are used as the final result. Obviously, as for encryption efficiency, the proposed is more efficient that the NYO08 scheme. Roughly speaking, both schemes have the same decryption efficiency. It is worth noting that only the proposed scheme is provably secure.

7. Conclusions

For the sake of data security in cloud computing, a fine-grained data sharing scheme is proposed. The proposed scheme can simultaneously achieve provable security and support attribute extension. It is proved to be secure in the selective-policy model under the DBDH assumption. Performance analysis is made to show that the proposed scheme is suitable for realizing data sharing in cloud computing.

Acknowledgements

We are grateful to the referees for their invaluable suggestions. This work is supported by the National Natural Science Foundation of China (Grant Nos. 61402366, 61472472, and 61272037), the Natural Science Basic Research Plan in Shaanxi Province of China (Grant No. 2015JQ6236), the Ministry of Industry and Information Key Project Fund (Grant No. 2013ZX03002004), and the Natural Science Foundation of Shaanxi Province (Grant No. 2013JZ020).

References

[1] A. Sahai, and B. Waters, "Fuzzy Identity-based Encryption", EUROCRYPT'05, 2005, pp. 557-557.

[2] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based Encryption with Non-monotonic Access Structures", in Proc. of the ACM conference on Computer and Communications Security (CCS'07), 2007, pp. 195-203.

[3] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy Attribute-based Encryption", in Proc. of IEEE Symposium on Security and Privacy (SP'07), 2007, pp. 321-334.

[4] L. Cheung, and C. Newport, "Provably Secure Ciphertext Policy ABE," in Proc. of the ACM conference on Computer and Communications Security (CCS'07), 2007, pp. 456-465.

[5] K. Li, "Matrix Access structure Policy used in Attribute-Based Proxy Re-encryption", International Journal of Computer Science Issue, Vol. 9, Issue 6, No 2, 2012, pp. 119-127.

[6] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based Encryption with Efficient Revocation", in Proc. of the ACM conference on Computer and Communications Security (CCS'08), 2008, pp. 417-426.

[7] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute Based Data Sharing with Attribute Revocation", in Proc. of the ACM

Symposium on Information Computer and Communications Security (ASIACCS’10), 2010, pp. 261-270.

[8] Y. Zhang, X. Chen, J. Li, H. Li, and F. Li, “FDR-ABE: Attribute-based Encryption with Flexible and Direct Revocation”, in Proc. of the International Conference on Intelligent Networking and Collaborative Systems (INCoS’13), 2013, pp. 38-45.

[9] Y. Zhang, X. Chen, J. Li, H. Li, F. Li, “Attribute-based Data Sharing with Flexible and Direct Revocation in Cloud Computing”, KSII Transactions on Internet & Information Systems, Vol. 8, No. 11, 2014, pp. 4028-4049.

[10] D. Zheng, Q. Zhao, and Y. Zhang, “A Brief Overview on Cryptography”, Journal of Xi'an University of Posts and Telecommunication, Vol. 18, No. 6, 2013, pp. 1-10. (in Chinese)

[11] T. Nishide, K. Yoneyama, and K. Ohta, “ABE with Partially Hidden Encryptor-Specified Access Structure”, in Proc. of Applied Cryptography and Network Security (ACNS’08), 2008, pp. 111-129.

[12] Y. Zhang, X. Chen, J. Li, D. S. Wong, and H. Li, “Anonymous Attribute-based Encryption Supporting Efficient Decryption Test”, in Proc. of the ACM Symposium on Information, Computer and Communications Security (ASIACCS’13), 2013, pp. 511-516.

[13] C. Chen, Z. Zhang, and D. Feng, “Efficient Ciphertext Policy Attribute-based Encryption with Constant-size Ciphertext and Constant Computation-cost”, ProvSec'11, 2011, pp. 84-101.

[14] Y. Zhang, D. Zheng, X. Chen, J. Li, H. Li, “Computationally Efficient Ciphertext-policy Attribute-based Encryption with Constant-size Ciphertexts”, ProvSec'14, 2014, pp. 259-273.

[15] A. Ge, R. Zhang, C. Chen, C. Ma, and Z. Zhang, “Threshold Ciphertext Policy Attribute-based Encryption with Constant Size Ciphertexts”, ACISP’12, 2012, pp. 336-349.

[16] Y. Zhang, D. Zheng, J. Li, and H. Li, “Attribute Directly-revocable Attribute-based Encryption with Constant Ciphertext Length”, Journal of Cryptologic Research, Vol. 1, No. 5, 2014, pp. 465-480. (in Chinese)

[17] M. Marwaha, and R. Bedi, “Applying Encryption Algorithm for Data Security and Privacy in Cloud Computing”, International Journal of Computer Science Issue, Vol. 10, Issue 1, No 1, 2013, pp. 367-370.

[18] Y. Zhang, X. Chen, and H. Li, “Key-evolving Hierarchical ID-based Signcryption”, The Computer Journal, Vol. 56, No. 10, 2013, pp. 1228-1248.

[19] J. Li, X. Chen, J. Li, C. Jia, J. Ma, and W. Lou, “Fine-grained Access Control System Based on Outsourced Attribute-based Encryption”, ESORICS'13, 2013, pp. 592-609.

[20] E. Fujisaki, and T. Okamoto, “Secure Integration of Asymmetric and Symmetric Encryption Schemes”, CRYPTO’99, 1999, pp. 537-554.

[21] B. Lynn, “The Stanford Pairing Based Crypto Library”, http://crypto.stanford.edu/pbc.

Yinghui Zhang received his Ph.D degree in Cryptography from Xidian University at 2013. Currently, he works at Xi'an University of Posts and Telecommunications. His research interests are in the areas of wireless network security, cloud security and cryptography.