Attribute Based Proxy Re-encryption with Delegating Capabilities

Xiaohui Liang
Department of Computer Science & Engineering
Shanghai Jiao Tong University
liangxh1207@gmail.com

Zhenfu Cao
Department of Computer Science & Engineering
Shanghai Jiao Tong University
zfcao@cs.sjtu.edu.cn

Huang Lin
Department of Computer Science & Engineering
Shanghai Jiao Tong University
faustlin@sjtu.edu.cn

Jun Shao[*]
Department of Computer Science & Engineering
Shanghai Jiao Tong University
chn.junshao@gmail.com

ABSTRACT

Attribute based proxy re-encryption scheme (ABPRE) is a new cryptographic primitive which extends the traditional proxy re-encryption (public key or identity based cryptosystem) to the attribute based counterpart, and thus empower users with delegating capability in the access control environment. Users, identified by attributes, could freely designate a proxy who can re-encrypt a ciphertext related with a certain access policy to another one with a different access policy. The proposed scheme is proved selective-structure chosen plaintext secure and master key secure without random oracles. Besides, we develop another kind of key delegating capability in our scheme and also discuss some related issues including a stronger security model and applications.

Categories and Subject Descriptors

E.4 [**Data Encryption**]: Public Key Cryptosystems

General Terms

Security, Theory

Keywords

Attribute Based Encryption, Proxy Re-encryption, Key Delegation

1. INTRODUCTION

Proxy re-encryption (PRE) allows a proxy to translate a ciphertext encrypted under Alice's public key into one that can be decrypted by Bob's secret key. Considering an email forwarding scenario: Alice is going on vacation and wishes the others could still open the message in the encrypted email aiming to her. With a PRE system, she could fulfill her will without giving her secret key to either the mail server or Bob.

Compared with the traditional proxy decryption scheme in which users delegate part of his decryption capability to others, PRE also takes an advantage that users don't need to store any additional decryption key, in other words, any decryption would be finished using only his own secret keys. Besides, PRE isolates a proxy authority alone who, though is capable of translating one ciphertext into another one with a different public key, can not obtain any confidential information including the corresponding plaintext or the original secret keys.

In light of these advantages, PRE schemes can be widely embedded into any public key cryptosystem and have many practical applications, including encrypted email forwarding, distributed file system, and the DRM of Apple's iTunes (demonstrated in [2, 7]). In 1998, Blaze, Bleumer and Strauss first introduced the PRE scheme in public key cryptosystem. In 2007, Green and Ateniese also extended the PRE technique in identity based cryptosystem [6] and gave its applications. Meanwhile, another new notion was proposed in 2005 called attribute based cryptosystem [17]. This paper dedicates itself to apply the PRE technique into such system in order to construct an ABPRE scheme and explore its applications.

However, it is not a trivial work to apply proxy re-encryption technique into attribute based system. From a theoretical point of view, proxy re-encryption has its complex properties which would lead to many security issues while attribute based system is very limited in security proof. Therefore, how to cope with these new security issues is a challenging problem. From a practical side, it is clear that a proxy could do the translation without any trust party involved. Moreover, ciphertext translation is extended to many-to-one mapping instead of one-to-one mapping existed in traditional PRE. These requirements enhance the difficulty of our construction and lower the efficiency of the whole system.

[*] Jun Shao currently is a post-doc of College of Information Scienes and Technology in Pennsylvania State University.

ASIACCS '09, March 10-12, 2009, Sydney, NSW, Australia.
Copyright 2009 ACM 978-1-60558-394-5/09/03 ...$5.00.

1.1 Our Contribution

To better understand the concept of ABPRE, we first demonstrate an application scenario of personal information system in a university. In this system, there are some confidential records of the grades of every student. These information is encrypted under the access policy ((AGE $>$ 40) $\wedge$ (Tenure)). The professors who are older than 40 and have a tenure position would receive the secret keys corresponding to "AGE $>$ 40", "Tenure" and thus they are qualified to retrieve the confidential records. Nevertheless, when these professors are on vacation, it is necessary to find some trustworthy substitutes who is able to decrypt these records in time. Therefore, ABPRE allows a qualified professor to freely designate a proxy who can translate these encrypted records to those encrypted with a different access policy (such as administrators with at least 10 years of working experience (Admin) $\wedge \neg$(EXP $<$ 10)). Hence, even if no qualified professor are available, some highly experienced administrator can open the confidential records with the help of the professor's proxy.

A more general relationship between users and ciphertexts are shown in Figure 1. Suppose there are three users and three ciphertext sets existing in this system, where user U_1 is able to decrypt any ciphertext in sets C_1, C_2 encrypted under access structure AS_1, AS_2 and users U_2, U_3 are able to decrypt C_3 corresponding with AS_3. Then, U_1 designates a proxy with a re-key which can be used for translating the ciphertext of C_1, C_2 into one of C_3. In this way, even if U_1 is offline, U_2, U_3 could still obtain the information encrypted in C_1, C_2 with the help of U_1's proxy.

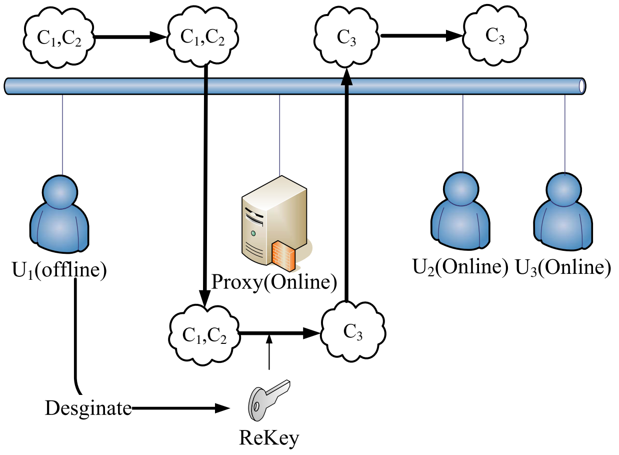

Figure 1: relationships in ABPRE system

In this paper, we propose the first concrete ABPRE scheme which has several properties:

- Unidirectional: the proxy who is designated with the capability for translating ciphertext from $C \rightarrow C'$ does not allow a translation from $C' \rightarrow C$.
- Multi-hop: re-encrypted ciphertext in C' can be re-encrypted again by another valid proxy.
- Non-interactive: re-keys can be generated by a user alone, no trusted third party involved and no interaction is required.
- Master key security: as in Figure **??**, a valid proxy designated by U_1, and two users U_2, U_3 who are able to decrypt U_1's ciphertext with the help from the proxy can not collude to obtain the secret key of U_1.

This scheme can be proven selective-structure chosen plaintext secure in the standard model under Augment Decisional Bilinear Diffie-Hellman assumption. Its master key security can be proved in the standard model under Augment Diffie-Hellman assumption.

1.2 Related Works

As a primitive work of attribute based encryption (ABE), fuzzy identity based encryption (FIBE) [17] aimed to extend the concept of IBE in the sense that one ciphertext could be decrypted by a group of users. Afterwards, access structure are considered to be generalized to a boolean function consisting of threshold gate, AND, OR and NOT operations in the subsequently proposed ABE schemes [4, 8, 9, 11, 15]. Two categories of ABE including ciphertext policy and key policy are presented in [11]. Ostrovsky *et al.* [15] consequently developed a fine-grain non-monotonic access structure in key policy ABE (the user's key corresponds to the access structure). However, to design a more expressive access structure in ciphertext policy ABE (the ciphertext corresponds to the access structure) encounters many obstacles. The first concrete CPABE scheme [4] was proposed by Bethencourt, Sahai and Waters. Their scheme only provides a security argument in the generic group model, although it is an efficient construction (supporting threshold gate). Consequently, The first provable secure CPABE [9], though only permitting AND operation, would well handle positive and negative attributes. Most recently, Goyal *et al.* [10] provide a "bounded" CPABE construction, which supports threshold gates in access tree and can be proved secure in the standard model, however the computational cost of the whole system grow exponentially with the depth of a pre-defined universal tree.

In Eurocrypt 1998, Blaze, Bleumer and Strauss proposed the first concrete scheme (called BBS) of proxy re-encryption [5], in which a proxy was given a re-key that allows it to translate a message m encrypted under pk_1 into an encryption of the same message m under a different public key pk_2. BBS scheme has bidirectional property which means if a re-key could be used for translation from pk_1 to pk_2, then it could also translate ciphertexts in a reverse direction. BBS scheme is also multi-hop, that is, a ciphertext can be re-encrypted from Alice to Bob to Carol and so on. Then, Ateniese *et al.* [2, 3] presented the first unidirectional and single-hop proxy re-encryption scheme. In 2007, Green and Ateniese [12] extended proxy re-encryption in identity based cryptosystem and gave two concrete schemes under CPA and CCA secure notion in random oracle model. At the same year, Canetti and Hohenberger [7] proposed the first CCA secure proxy re-encryption scheme in the standard model, satisfying bidirectional and multi-hop.

Besides, master key security of PRE firstly appeared in [3]. The authors formally gave the security model of master key secure scheme where an adversary (including the proxy and a group of colluding delegatees) can not recover the delegator's secret key. However, most existing proxy re-encryption scheme [5, 13, 7, 12] does not meet this security requirement.

1.3 Road Map

The rest of this paper is organized as follows. In the next section, the algorithm definition and security model of ABPRE are first introduced. In Section 3, some preliminaries including bilinear map and complexity assumptions are provided to help the construction and the security proof. Then, the first concrete ABPRE scheme and its security proof are given in Section 4. Section 5 includes some discussions on delegating capability and security model. After that, two applications are shown in Section 6 and finally the conclusion is drawn in Section 7.

2. DEFINITIONS

2.1 ABPRE model

The similar definitions can be found in [7, 12].

Definition 1. *An* ABPRE *scheme is a tuple of probabilistic polynomial time algorithms* (SETUP, KEYGEN, RKGEN, ENC, REENC, DEC).

- $\mathtt{SETUP}(1^k) \rightarrow (pp, mk)$: On input a security parameter 1^k, the setup algorithm SETUP outputs a system public parameter pp and a master key mk.
- $\mathtt{KEYGEN}(S, mk) \rightarrow (usk)$: On input an index set S[1] and a master key mk, the key generation algorithm KEYGEN outputs a secret key usk.
- $\mathtt{RKGEN}(usk, AS) \rightarrow (rk)$: On input a secret key usk and an access structure AS, the re-key generation algorithm RKGEN outputs a re-key rk.
- $\mathtt{ENC}(AS, m) \rightarrow (C)$: On input an access structure AS and a message m, the encryption algorithm ENC outputs a ciphertext C.
- $\mathtt{REENC}(rk, C) \rightarrow (C')$: On input a re-key rk and a ciphertext C, the re-encryption algorithm REENC first checks if the index set in rk satisfies the access structure of C. Then, if check passes, it outputs a re-encrypted ciphertext C'; otherwise, it outputs "reject".
- $\mathtt{DEC}(usk, C) \rightarrow (m)$: On input a secret key usk and a ciphertext C, the decryption algorithm DEC first checks if the index set in usk satisfies the access structure of C. Then, if check passes, it outputs a message m in the message space; otherwise, it outputs "reject".

Correctness. The correctness property has two requirements. For any message m in the message space, the following two equations must hold[2]:

1. $\mathtt{DEC}(\mathtt{KEYGEN}(S, mk), \mathtt{ENC}(AS, m)) = m$;
2. $\mathtt{DEC}(\mathtt{KEYGEN}(S'', mk), \mathtt{REENC}(\mathtt{RKGEN}(\mathtt{KEYGEN}(S', mk), AS''), C)) = m$.

where S satisfies AS, S' satisfies AS', S'' satisfies AS'', mk is a valid master key, C is the ciphertext related with message m and access structure AS'.

Selective-Structure Chosen Plaintext Security. We say that an ABPRE scheme is secure against selective-structure chosen plaintext attack if no probabilistic polynomial time adversary $\mathcal{A}$ has a non-negligible advantage in winning the SS-CPA-ABPRE game.

Init The adversary $\mathcal{A}$ chooses a challenge access structure AS^* and delivers it to the challenger. The challenger runs $\mathtt{SETUP}(1^k)$ and gives $\mathcal{A}$ the resulting system public parameter pp. It keeps the corresponding master-key mk to itself.

Phase 1 The adversary $\mathcal{A}$ issues queries to the oracles:

- **Key generation oracle** $\mathcal{O}_{kg}$: On input an index set I_q, if I_q does not satisfy AS^*, then output a secret key $usk = \mathtt{KEYGEN}(I_q, mk)$; otherwise, output "reject".
- **Rekey generation oracle** $\mathcal{O}_{rkg}$: On input an index set I_q and an access structure AS, if I_q does not satisfy AS^*, then output a re-key $rk = \mathtt{RKGEN}(\mathtt{KEYGEN}(I_q, mk), AS)$; otherwise, output "reject".
- **Re-encryption oracle** $\mathcal{O}_{ree}$: On input an index set I_q, an access structure AS and a ciphertext C, if I_q does not satisfy AS^* and I_q satisfies the access structure of C, then output a ciphertext $C' = \mathtt{REENC}(\mathtt{RKGEN}(\mathtt{KEYGEN}(I_q, mk), AS), C)$; otherwise, output "reject".

Challenge Once the adversary $\mathcal{A}$ decides that Phase 1 is over, it outputs two equal length messages m_0, m_1 from the message space. The challenger chooses $\mu \in \{0, 1\}$ at random and encrypts M_μ with AS^*. Then, the ciphertext C^* is given to the adversary $\mathcal{A}$.

Phase 2 The same as Phase 1.

Guess The adversary $\mathcal{A}$ outputs a guess $\mu' \in \{0, 1\}$ and wins the game if $\mu' = \mu$.

We define an adversary $\mathcal{A}$'s advantage in SS-CPA-ABPRE game as follows:

$$Adv^{\mathcal{O}_{kg},\mathcal{O}_{rkg},\mathcal{O}_{ree}}_{\text{SS-CPA-ABPRE},\mathcal{A}} = |\Pr[\mu' = \mu] - \frac{1}{2}|$$

Master Key Security for ABPRE. An ABPRE scheme has master key security if no probabilistic polynomial time adversary $\mathcal{A}$ has a non-negligible advantage in winning the MKS-ABPRE game.

Init The adversary $\mathcal{A}$ chooses a challenge index set I^* and delivers it to the challenger. The challenger runs $\mathtt{SETUP}(1^k)$ and gives $\mathcal{A}$ the resulting system public parameters pp. It keeps the corresponding master-key mk to it self.

Queries The adversary $\mathcal{A}$ issues queries to the oracles:

- **Key generation oracle** $\mathcal{O}_{kg}$: On input an index set I_q, if $I_q \neq I^*$, output a secret key $usk = \mathtt{KEYGEN}(I_q, mk)$; otherwise, output "reject".
- **Rekey generation oracle** $\mathcal{O}_{rkg}$: On input an index set I_q and an access structure AS, output a re-key $rk = \mathtt{RKGEN}(\mathtt{KEYGEN}(I_q, mk), AS)$.
- **Re-encryption oracle** and **Decryption oracle** are straightforward, since any re-key could be generated from **Rekey generation oracle**.

[1] Similar to [9], each secret key is related with an attribute set indexed by S

[2] S, S', S'' are the index sets and AS, AS', AS'' are the access structures

Output The adversary $\mathcal{A}$ outputs the secret key usk^* for index set I^*.

The advantage of $\mathcal{A}$ in the above game is defined as

$$Adv^{\mathcal{O}_{kg},\mathcal{O}_{rkg}}_{\text{MKS-ABPRE},\mathcal{A}} = \Pr[\mathcal{A} \text{ succeeds}]$$

3. PRELIMINARIES

3.1 Bilinear Map

Most attribute-based encryption schemes including ours are constructed based on bilinear maps [6, 17, 4, 9].

Consider two finite cyclic groups G and G_T having the same prime order p. It is clear that the respective group operation is efficiently computable. Assume that there exists an efficiently computable mapping $e : G \times G \rightarrow G_T$, called a bilinear map or pairing, with the following properties.

- Bilinear: For any $g, h \in G$, and $a, b \in Z_p$, we have $e(g^a, h^b) = e(g, h)^{ab}$.
- Non-degeneracy: $e(g, g) \neq 1$

Note that, $e(*, *)$ is symmetric since $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

3.2 Complexity Assumptions

Definition 2. *Complex Triple Diffie-Hellman (CTDH) Problem. Let $e : G \times G \rightarrow G_T$ be a bilinear map, where G has prime order p and g is a generator of G, random numbers $n, a, b, c, d, R \in Z_p$. Given a tuple $\langle g, n, g_b = g^b, h = g^c, g_d = g^d, g_1 = g^{\frac{c}{b}}, g_2 = g^{bc}, g_3 = g^{ac}, g_4 = g^{abc-Rc}, g_5 = g^{c(R+nd)}, g_6 = g^{\frac{c(R+nd)}{b}} \rangle$ as inputs, output g^{abc}.*

Definition 3. *Augment Diffie-Hellman (ADH) Problem. Let $e : G \times G \rightarrow G_T$ be a bilinear map, where G has prime order p and g is a generator of G, random numbers $a, b \in Z_p$. Given a tuple $\langle g, g^a, g^b, g^{b^2} \rangle$ as inputs, output g^{ab}.*

Definition 4. *We say that the CTDH assumption holds in G if no probabilistic polynomial time adversary is able to output g^{abc} from $\langle g, n, g_b = g^b, h = g^c, g_d = g^d, g_1 = g^{\frac{c}{b}}, g_2 = g^{bc}, g_3 = g^{ac}, g_4 = g^{abc-Rc}, g_5 = g^{c(R+nd)}, g_6 = g^{\frac{c(R+nd)}{b}} \rangle$ with non-negligible advantage, where random numbers $n, a, b, c, d, R \in Z_p$ and generator $g \in G$ are chosen independently and uniformly at random.*

Notice that CTDH problem is harder than ADH problem, since if given ADH problem's inputs $\langle g, g^c, g^{c^2}, g^R \rangle$, we could select $R+nd = 1, R = ab, b = c$ and outputs g^{Rc} from CTDH oracle with inputs $\langle g, n, g^c, g^c, g^{(1-R)n^{-1}}, g, g^{c^2}, g^R, 1, g^c, g \rangle$. (The master key security of our scheme is based on CTDH assumption.)

Definition 5. *Augment Decisional Bilinear Diffie-Hellman (ADBDH) Problem. Let $e : G \times G \rightarrow G_T$ be a bilinear map, where G has prime order p and g is a generator of G, random numbers $a, b, c \in Z_p$. Given a tuple $\langle g, A = g^a, B = g^b, C = g^c, B' = g^{\frac{1}{b}}, Z \rangle$ as inputs, output 1 if $Z = e(g, g)^{abc}$; otherwise, output 0.*

Definition 6. *We say that the ADBDH assumption holds in G if no probabilistic polynomial time adversary is able to distinguish the tuples $\mathcal{D}_{adbdh} = \langle g, g^a, g^b, g^c, g^{\frac{1}{b}}, e(g, g)^{abc} \rangle$ and $\mathcal{D}_{rand} = \langle g, g^a, g^b, g^c, g^{\frac{1}{b}}, e(g, g)^z \rangle$ with non-negligible advantage, where $a, b, c, z \in Z_p$ and a generator $g \in G$ are chosen independently and uniformly at random.*

4. ATTRIBUTE BASED PRE

4.1 Our techniques

Reviewing the development of proxy re-encryption, Green and Ateniese [12] presented a delicate approach to combine the existing identity based scheme with proxy re-encryption using key sharing technique. In their scheme, the proxy's translation can be regarded as the partial decryption of the original ciphertext, while embedding another secret value s into it. However, Alice encrypts this secret value s under Bob's identity and the proxy relay "s" to Bob while learning nothing about it. Eventually, Bob gets s and finishes the remaining decryption steps. In this paper, our scheme combines key sharing technique [12] with the construction of CPABE [9].

4.2 Satisfying an Access Structure

In this scheme, we only consider the access structure consisting of "AND" gates between positive and negative attributes. Denote the index set of all the attributes as $\mathcal{I}$. The access structure can be represented as $\bigwedge +d_i(-d_i)$[3], $i \in \mathcal{I}$. Each user would receive a secret key corresponding to an attribute set $S \subseteq \mathcal{I}$ from the authority. The user successfully decrypts the ciphertext if and only if the following requirements are met:

- if $+d_i$ appears in the access structure, then $i \in \mathcal{S}$;
- if $-d_i$ appears in the access structure, then $i \notin \mathcal{S}$;

4.3 Main Construction

In this section, we present our construction with six algorithms `SETUP`, `KGEN`, `ENC`, `RKGEN`, `REENC`, `DEC`.

At the beginning, `SETUP` algorithm initializes the whole system by outputting public parameters. Users are given the secret keys with any attribute set S from `KGEN`. The encrypter is allowed to develop an access policy which consists of AND gate between positive and negative attributes. To designate a proxy who has translation capability, a user would run `RKGEN` algorithm and obtain rk related with an attribute set S and a new access structure, where translation only succeed if and only if S satisfies the access structure of the ciphertext. Users could also obtain a re-encrypted ciphertext from `REENC` algorithm with input a valid ciphertext and a re-key. Finally, `DEC` algorithm can be used for decrypting the original ciphertext or the re-encrypted one.

`SETUP`(1^k): Generate a bilinear group G of prime order p, with bilinear map $e : G \times G \rightarrow G_T$. Next, it selects elements $y, t_i (1 \leq i \leq 3n)$ in Z_p and two generators g, h of G at random. Let $Y := e(g, h)^y$ and $T_i := g^{t_i}, T'_i := h^{\frac{1}{t_i}}$ for each $1 \leq i \leq 3n$. The public parameter pp includes $\langle e, g, h, Y, \{T_i, T'_i\}_{1 \leq i \leq 3n} \rangle$. The master key mk is

[3] $+d_i$: the positive attribute, $-d_i$: the negative attribute

$\langle y, \{t_i\}_{1 \leq i \leq 3n} \rangle$.

KGEN(S, mk): Let S denote an index set of attributes. It chooses random $r_1, ..., r_n$ from Z_p and sets $r = r_1 + r_2 + ... + r_n$. Compute $\hat{D} = h^{y-r}$, and for each $i \in \mathcal{N}$ ($\mathcal{N} = \{1, 2, ..., n\}$): if $i \in S$, $D_{i,1} = h^{\frac{r_i}{t_i}}, D_{i,2} = h^{\frac{r_i}{t_{2n+i}}}$; otherwise, $D_{i,1} = h^{\frac{r_i}{t_{n+i}}}, D_{i,2} = h^{\frac{r_i}{t_{2n+i}}}$. It outputs a user's secret key $usk = \langle S, (D_{i,1}, D_{i,2})_{i \in \mathcal{N}}, \hat{D} \rangle$.

ENC(m, AS): Let AS denote an access structure. To encrypt a message $m \in G_T$, it selects random $s \in Z_p$ and computes $\tilde{C} = m \cdot Y^s, \hat{C} = g^s, \breve{C} = h^s$. For $i \in \mathcal{N}$: if $+d_i$ appears AS, $C_i = T_i^s$; if $-d_i$ appears AS, $C_i = T_{n+i}^s$; otherwise, $C_i = T_{2n+i}^s$. It outputs $C = \langle AS, \tilde{C}, \hat{C}, \breve{C}, (C_i)_{i \in \mathcal{N}} \rangle$.

RKGEN(usk, AS): Let usk denote a valid secret key consisting of $\langle S, (D_{i,1}, D_{i,2})_{i \in \mathcal{N}}, \hat{D} \rangle$ and AS denote an access structure. It selects random $d \in Z_p$ and set $\mathfrak{D} = g^d, \hat{D}' = \hat{D}$. For $i \in \mathcal{N}$: if $i \in S$, $D'_{i,1} = D_{i,1} \cdot (T'_i)^d, D'_{i,2} = D_{i,2} \cdot (T'_{2n+i})^d$; otherwise, $D'_{i,1} = D_{i,1} \cdot (T'_{n+i})^d, D'_{i,2} = D_{i,2} \cdot (T'_{2n+i})^d$; $\mathbb{C}$ is the ciphertext of $\mathfrak{D}$ under the access structure AS.[4]

It outputs $rk = \langle S, AS, (D'_{i,1}, D'_{i,2})_{i \in \mathcal{N}}, \hat{D}', \mathbb{C} \rangle$.

REENC(rk, C): Let rk denote a valid re-key consisting of $\langle S, AS', (D'_{i,1}, D'_{i,2})_{i \in \mathcal{N}}, \hat{D}', \mathbb{C} \rangle$ and C denote a well-formed ciphertext $\langle AS, \tilde{C}, \hat{C}, \breve{C}, (C_i)_{i \in \mathcal{N}} \rangle$, It checks if S satisfies AS, if not, output $\perp$; otherwise, for $i \in \mathcal{N}$:

- $+d_i$ appears in AS, $E_i = e(C_i, D'_{i,1}) = e(g^{t_i s}, h^{\frac{r_i + d}{t_i}}) = e(g, h)^{s(r_i + d)}$;
- $-d_i$ appears in AS, $E_i = e(C_i, D'_{i,1}) = e(g^{t_{n+i} s}, h^{\frac{r_i + d}{t_{n+i}}}) = e(g, h)^{s(r_i + d)}$;
- Otherwise, $E_i = e(C_i, D'_{i,2}) = e(g^{t_{2n+i} s}, h^{\frac{r_i + d}{t_{2n+i}}}) = e(g, h)^{s(r_i + d)}$;

It then computes $\bar{C} = e(\hat{C}, \hat{D}') \prod_{i \in \mathcal{N}} E_i = e(g^s, h^{y - \sum_{i=1}^n r_i}) \cdot e(g, h)^{nds + s \sum_{i=1}^n r_i} = e(g, h)^{ys + nds}$; output a re-encrypted ciphertext $C' = \langle AS', \tilde{C}, \bar{C}, \breve{C}, \mathbb{C} \rangle$.

Note that $\mathbb{C}$ can be re-encrypted again. Thus, we would obtain $C'_{re} = \langle AS', \tilde{C}, \bar{C}, \breve{C}, \mathbb{C}' \rangle$, where $\mathbb{C}'$ is obtained from the REENC algorithm with the input of another rk' and $\mathbb{C}$. The size of re-encrypted ciphertext increases linearly with the re-encryption times.

DEC(C, usk): Let usk denote a valid secret key $\langle S, (D_{i,1}, D_{i,2})_{i \in \mathcal{N}}, \hat{D} \rangle$. It checks if S satisfies AS, if not, output $\perp$; otherwise, do

1. If C is an original well-formed ciphertext consisting of $\langle AS, \tilde{C}, \hat{C}, \breve{C}, (C_i)_{i \in \mathcal{N}} \rangle$, for $i \in \mathcal{N}$:
   - $+d_i$ appears in AS, $E_i = e(C_i, D_{i,1}) = e(T_i^s, h^{\frac{r_i}{t_i}}) = e(g, h)^{sr_i}$;
   - $-d_i$ appears in AS, $E_i = e(C_i, D_{i,1}) = e(T_{n+i}^s, h^{\frac{r_i}{t_{n+i}}}) = e(g, h)^{sr_i}$;
   - Otherwise, $E_i = e(C_i, D_{i,2}) = e(T_{2n+i}^s, h^{\frac{r_i}{t_{2n+i}}}) = e(g, h)^{sr_i}$;

   It outputs $\frac{\tilde{C}}{e(\hat{C}, \hat{D}) \cdot \prod_{i \in \mathcal{N}} E_i} = \frac{m \cdot e(g,h)^{ys}}{e(g^s, h^{y-r}) \cdot e(g,h)^{sr}} = m$.
2. Else if C is a re-encrypted well-formed ciphertext consisting of $\langle AS', \tilde{C}, \bar{C}, \breve{C}, \mathbb{C} \rangle$, it decrypts $\mathbb{C}$ using usk and obtains $\mathfrak{D} = g^d$. Then, it outputs $\frac{\tilde{C} e(\mathfrak{D}, \breve{C})^n}{\bar{C}} = \frac{m \cdot e(g,h)^{ys} \cdot e(g^d, h^s)^n}{e(g,h)^{ys+nds}} = m$.
3. Else if C is a multi-time re-encrypted well-formed ciphertext, decryption is similar with the above phases.

Correctness. The correctness is easily observable.

4.4 Security Proof for ABPRE

THEOREM 1. *If the ADBDH assumption holds in (G, G_T), then* ABPRE *scheme is selective-structure chosen plaintext secure in the standard model.*

PROOF. We now reduce SS-CPA security of ABPRE to the augment decisional Bilinear Diffie-Hellman (ADBDH) assumption.

Suppose an adversary $\mathcal{A}$ win the SS-CPA-ABPRE game with non-negligible advantage ε. A simulator $\mathcal{S}$ can be constructed to distinguish $\mathcal{D}_{adbdh}$ from $\mathcal{D}_{rand}$ with non-negligible advantage $\frac{\varepsilon}{2}$.

We first let the challenger set the groups G and G_T with an efficient bilinear map e and a generator g. The challenger flips a fair binary coin ν, outside of $\mathcal{S}$'s view. If $\nu = 1$, the challenger sets $\langle g, A, B, C, B', Z \rangle \in \mathcal{D}_{adbdh}$; otherwise it sets $\langle g, A, B, C, B', Z \rangle \in \mathcal{D}_{rand}$.

Init In this phase, $\mathcal{S}$ receives a challenge access structure AS^*, and notes I_+^*, I_-^* the index set of positive and negative attributes separately. Then, $\mathcal{S}$ selects $k, \alpha_i, \beta_i, \gamma_i$ at random from Z_p for $i \in \mathcal{N}$ and generates the public key $Y = e(A, B)^k, h = g^k$. Then, $\mathcal{S}$ outputs the public parameters are:

- $i \in I_+^*$, $T_i = g^{\alpha_i}, T_{n+i} = B^{\beta_i}, T_{2n+i} = B^{\gamma_i}, T'_i = g^{\frac{k}{\alpha_i}}, T'_{n+i} = B'^{\frac{k}{\beta_i}}, T'_{2n+i} = B'^{\frac{k}{\gamma_i}}$;
- $i \in I_-^*$, $T_i = B^{\alpha_i}, T_{n+i} = g^{\beta_i}, T_{2n+i} = B^{\gamma_i}, T'_i = B'^{\frac{k}{\alpha_i}}, T'_{n+i} = g^{\frac{k}{\beta_i}}, T'_{2n+i} = B'^{\frac{k}{\gamma_i}}$;
- Otherwise, $T_i = B^{\alpha_i}, T_{n+i} = B^{\beta_i}, T_{2n+i} = g^{\gamma_i}, T'_i = B'^{\frac{k}{\alpha_i}}, T'_{n+i} = B'^{\frac{k}{\beta_i}}, T'_{2n+i} = g^{\frac{k}{\gamma_i}}$.

Phase 1: $\mathcal{A}$ makes several queries to the key generation oracle $\mathcal{O}_{kg}$, the re-key generation oracle $\mathcal{O}_{rkg}$ and the re-encryption oracle $\mathcal{O}_{ree}$.

- $\mathcal{A}$ makes a query to $\mathcal{O}_{kg}$ with an index set I_q. According to the security game, if I_q satisfies AS^*, it outputs $\perp$. Otherwise, $\mathcal{S}$ queries $usk = \langle I_q, (D_{i,1}, D_{i,2})_{i \in \mathcal{N}}, \hat{D} \rangle$ from the oracle in Appendix B[5] and outputs usk.
- $\mathcal{A}$ makes a query to $\mathcal{O}_{rkg}$ with an index set I_q and an access structure AS. According to the security game, if I_q satisfies AS^*, it outputs $\perp$. Otherwise, $\mathcal{S}$ submits I_q to $\mathcal{O}_{kg}$ and obtains a secret key $usk = \langle I_q, (D_{i,1}, D_{i,2})_{i \in \mathcal{N}}, \hat{D} \rangle$. $\mathcal{S}$ executes the following steps:

[4] Though $\mathfrak{D}$ is not in the message space, we could divide it into several parts and construct a coding map from each part to an element of G_T

[5] Since the key generation oracle is similar with the counterpart in [9], we put this part in Appendix B.

1. Select $d \in Z_p$ at random and set $\mathfrak{D} = g^d, \hat{D}' = \hat{D}$;
2. For $i \in \mathcal{N}$:
    - $i \in I_q$, $D'_{i,1} = D_{i,1} \cdot (T'_i)^d, D'_{i,2} = D_{i,2} \cdot (T'_{2n+i})^d$;
    - Otherwise, $D'_{i,1} = D_{i,1} \cdot (T'_{n+i})^d, D'_{i,2} = D_{i,2} \cdot (T'_{2n+i})^d$;
3. $rk = \langle I_q, AS, (D'_{i,1}, D'_{i,2})_{i \in \mathcal{N}}, \hat{D}', \mathbb{C}\rangle$, where $\mathbb{C}$ is the ciphertext of $\mathfrak{D}$ under the access structure AS.

- $\mathcal{A}$ makes a query to $\mathcal{O}_{ree}$ with an index set I_q, an access structure AS' and a ciphertext $C = \langle AS, \tilde{C}, \hat{C}, \breve{C}, (C_i)_{i\in\mathcal{N}}\rangle$. According to the security game, if I_q satisfies AS^*, it outputs $\perp$. If I_q does not satisfy AS, it outputs $\perp$. Then, $\mathcal{S}$ submits (I_q, AS') to the re-key generation oracle and obtains

  $$rk = \langle I_q, AS', (D'_{i,1}, D'_{i,2})_{i\in\mathcal{N}}, \hat{D}', \mathbb{C}\rangle$$

  $\mathcal{S}$ uses rk to re-encrypt the ciphertexts C. For $i \in \mathcal{N}$:

    - $+d_i$ appears in AS, $E_i = e(C_i, D'_{i,1}) = e(g^{t_i s}, h^{\frac{r_i+d}{t_i}}) = e(g,h)^{s(r_i+d)}$;
    - $-d_i$ appears in AS, $E_i = e(C_i, D'_{i,1}) = e(g^{t_{n+i} s}, h^{\frac{r_i+d}{t_{n+i}}}) = e(g,h)^{s(r_i+d)}$;
    - Otherwise, $E_i = e(C_i, D'_{i,2}) = e(g^{t_{2n+i} s}, h^{\frac{r_i+d}{t_{2n+i}}}) = e(g,h)^{s(r_i+d)}$;

  It then computes $\bar{C} = e(\hat{C}, \hat{D}') \prod_{i\in\mathcal{N}} E_i = e(g^s, h^{y-\sum_{i=1}^n r_i}) \cdot e(g,h)^{nds+s\sum_{i=1}^n r_i} = e(g,h)^{ys+nds}$. Finally, it outputs the re-encrypted ciphertext

  $$C_{re} = \langle AS', \tilde{C}, \bar{C}, \breve{C}, \mathbb{C}\rangle$$

Challenge: $\mathcal{A}$ submits two message M_0 and M_1 of equal length. $\mathcal{S}$ generates challenge ciphertext:

$$C^* = \langle \tilde{C} = M_\mu \cdot Z^k, C, C^k, (C^{\alpha_i})_{i\in I^*_+}, (C^{\beta_i})_{i\in I^*_-}, (C^{\gamma_i})_{i\notin I^*_+\cup I^*_-}\rangle.$$

Phase 2: Same as Phase 1.
Guess: $\mathcal{S}$ outputs $\nu' = 1$ to indicate that it was given a tuple from $\mathcal{D}_{adbdh}$ if $\mathcal{A}$ gives a correct guess $\mu' = \mu$; otherwise output $\nu' = 0$ to indicate that it was given a tuple from $\mathcal{D}_{rand}$.
Let us compute the success probability of $\mathcal{S}$:

In the case of $\nu = 0$ the adversary gains no information about μ. Therefore, we have $\Pr[\mu \neq \mu' | \nu = 0] = \frac{1}{2}$. Since the simulator guesses $\nu' = 0$ when $\mu \neq \mu'$, we have $\Pr[\nu' = \nu | \nu = 0] = \Pr[\nu' = 0 | \nu = 0] = \frac{1}{2}$.

In the case of $\nu = 1$, the adversary gets a valid ciphertext of m_μ. By definition, the adversary has ε to guess the correct μ', and thus $\Pr[\mu = \mu' | \nu = 1] = \frac{1}{2} + \varepsilon$. Since the simulator guesses $\nu' = 1$ when $\mu = \mu'$, we have $\Pr[\nu' = \nu | \nu = 1] = \Pr[\nu' = 1 | \nu = 1] = \frac{1}{2} + \varepsilon$.

The overall advantage of the simulator to output a correct $\nu' = \nu$ is $\Pr[\nu = \nu'] - \frac{1}{2} = \Pr[\nu = \nu', \nu = 0] + \Pr[\nu = \nu', \nu = 1] - \frac{1}{2} = \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot (\frac{1}{2} + \varepsilon) - \frac{1}{2} = \frac{\varepsilon}{2}$ □

Theorem 2. *If the CTDH assumption holds in* G, G_T, *then* ABPRE *scheme has master key security.*

Proof. The simulator $\mathcal{S}$ receives a tuple $\langle g, n, g_b = g^b, h = g^c, g_d = g^d, g_1 = g^{\frac{c}{b}}, g_2 = g^{bc}, g_3 = g^{ac}, g_4 = g^{abc-Rc}, g_5 = g^{c(R+nd)}, g_6 = g^{\frac{c(R+nd)}{b}}\rangle$ and a challenge index set I^*. To output g^{abc}, the simulator $\mathcal{S}$ does follows:
Init: $\mathcal{S}$ selects elements $\alpha_i, \beta_i, \gamma_i \in Z_p$ for $i \in \mathcal{N}$, and let $Y := e(g,h)^{ab} = e(g_b, g_3)$. Then, it generates system public keys as follows:

- If $i \in I^*$, $T_i = g^{\alpha_i}, T_{n+i} = g_b^{\beta_i}, T_{2n+i} = g_b^{\gamma_i}, T'_i = h^{\frac{1}{\alpha_i}}, T'_{n+i} = (g_1)^{\frac{1}{\beta_i}}, T'_{2n+i} = (g_1)^{\frac{1}{\gamma_i}}$;
- Else $i \notin I^*$, $T_i = g_b^{\alpha_i}, T_{n+i} = g^{\beta_i}, T_{2n+i} = g_b^{\gamma_i}, T'_i = (g_1)^{\frac{1}{\alpha_i}}, T'_{n+i} = h^{\frac{1}{\beta_i}}, T'_{2n+i} = (g_1)^{\frac{1}{\gamma_i}}$;

$\mathcal{S}$ outputs $pp := \langle e, g, h, Y, (T_i, T_{n+i}, T_{2n+i}, T'_i, T'_{n+i}, T'_{2n+i})_{i\in\mathcal{N}}\rangle$.
Key generation oracle: $\mathcal{A}$ makes a query to key generation oracle with an index set $I_q \subseteq \mathcal{N}$ where $I_q \neq I^*$. Therefore, there must exist an index j so that $(j \in I_q) \wedge (j \notin I^*)$ or $(j \notin I_q) \wedge (j \in I^*)$. W.l.o.g, we analyze the case of $(j \notin I_q) \wedge (j \in I^*)$.

For every $i \in \mathcal{N}$, $\mathcal{S}$ chooses $r'_i \in Z_p$ at random and implicitly sets r_i in two ways:

$$\begin{cases} r_i = br'_i, & \text{if } i \neq j; \\ r_j = ab + br'_j, & \text{otherwise;} \end{cases}$$

Thus, we have $r = \sum_{i=1}^n r_i = ab + \sum_{i=1}^n r'_i \cdot b$; $\hat{D}$ can be computed as $\hat{D} = h^{y-r} = h^{-\sum_{i=1}^n r'_i \cdot b} = \prod_{i=1}^n \frac{1}{g_2^{r'_i}}$;

- For $i \in I_q, i \neq j$: if $i \in I^*$, $D_{i,1} = h^{\frac{r_i}{\alpha_i}} = g_2^{\frac{r'_i}{\alpha_i}}, D_{i,2} = h^{\frac{r_i}{b\gamma_i}} = h^{\frac{r'_i}{\gamma_i}}$; else $i \notin I^*$, $D_{i,1} = h^{\frac{r_i}{b\alpha_i}} = h^{\frac{r'_i}{\alpha_i}}, D_{i,2} = h^{\frac{r_i}{b\gamma_i}} = h^{\frac{r'_i}{\gamma_i}}$;
- For $i \notin I_q, i \neq j$: if $i \in I^*$,$D_{i,1} = h^{\frac{r_i}{b\beta_i}} = h^{\frac{r'_i}{\beta_i}}, D_{i,2} = h^{\frac{r_i}{b\gamma_i}} = h^{\frac{r'_i}{\gamma_i}}$; else $i \notin I^*$, $D_{i,1} = h^{\frac{r_i}{\beta_i}} = g_2^{\frac{r'_i}{\beta_i}}, D_{i,2} = h^{\frac{r_i}{b\gamma_i}} = h^{\frac{r'_i}{\gamma_i}}$;
- For $i = j$, $D_{i,1} = g^{\frac{r_i}{b\beta_i}} = h^{\frac{ab+br'_i}{b\beta_i}} = g_3^{\frac{1}{\beta_i}} h^{\frac{r'_i}{\beta_i}}$, $D_{i,2} = g^{\frac{r_i}{b\gamma_i}} = h^{\frac{ab+br'_i}{b\gamma_i}} = g_3^{\frac{1}{\gamma_i}} h^{\frac{r'_i}{\gamma_i}}$.

It outputs a secret key $usk = \langle I_q, (D_{i,1}, D_{i,2})_{i\in\mathcal{N}}, \hat{D}\rangle$.

Rekey generation oracle: $\mathcal{A}$ makes a query to key generation oracle with an index set I_q and an access structure AS, if $I_q \neq I^*$, obtain usk from $\texttt{KGEN}(I_q)$, and generate $rk = \texttt{RKGEN}(usk, AS)$; else $I_q = I^*$, do

- Select $j \in I^*$ and $r'_i, r \in Z_p$ at random for each $i \in \mathcal{N} \setminus \{j\}$;
- Implicitly set $r'_j = r + R + nd - \sum_{i=1, i\neq j}^n r'_i$ and $r_i = r'_i - d$ for $i \in \mathcal{N}$;
- Compute $\hat{D}' = h^{y-\sum_{i=1}^n r_i} = h^{y-\sum_{i=1}^n (r'_i - d)} = h^{ab-R-r} = g_4 h^{-r}$;
- For $i \in \mathcal{N}$:
    - If $i \neq j, i \in I^*$,$D'_{i,1} = h^{\frac{r_i+d}{\alpha_i}} = h^{\frac{r'_i}{\alpha_i}}, D'_{i,2} = h^{\frac{r_i+d}{b\gamma_i}} = g_1^{\frac{r_i+d}{\gamma_i}}$;

- Else if $i \neq j, i \notin I^*$,$D'_{i,1} = h^{\frac{r_i+d}{\beta_i}} = h^{\frac{r'_i}{\beta_i}}, D'_{i,2} = h^{\frac{r_i+d}{b\gamma_i}} = g_1^{\frac{r_i+d}{\gamma_i}}$;
- Else if $i = j$, $D'_{i,1} = h^{\frac{r_j+d}{\alpha_j}} = h^{\frac{r'_j}{\alpha_j}} = h^{\frac{r+R+nd-\sum_{i=1,i\neq j}^{n} r'_i}{\alpha_j}} = g_5^{\frac{1}{\alpha_j}} h^{\frac{r-\sum_{i=1,i\neq j}^{n} r'_i}{\alpha_j}}$, $D'_{i,2} = h^{\frac{r_j+d}{b\gamma_j}} = g_1^{\frac{r'_j}{\gamma_j}} = g_1^{\frac{r+R+nd-\sum_{i=1,i\neq j}^{n} r'_i}{\gamma_j}} = g_6^{\frac{1}{\gamma_j}} g_1^{\frac{r-\sum_{i=1,i\neq j}^{n} r'_i}{\gamma_j}}$;

- Select $t \in Z_p$ at random. For $i \in \mathcal{N}$, if $i \in I^*$, $D''_{i,1} = D'_{i,1} \cdot (T'_i)^t, D''_{i,2} = D'_{i,2} \cdot (T'_{2n+i})^t$;
  Otherwise, $D''_{i,1} = D'_{i,1} \cdot (T'_{n+i})^t, D''_{i,2} = D'_{i,2} \cdot (T'_{2n+i})^t$;
- Output $rk = \langle I^*, AS, (D''_{i,1}, D''_{i,2})_{i\in\mathcal{N}}, \hat{D}', \mathbb{C}\rangle$, where $\mathbb{C}$ is the ciphertext of $g_d \cdot g^t$ under the access structure AS.

Re-encryption oracle and **Decryption oracle** are straightforward, since any re-key could be correctly generated from **Rekey generation oracle**.

Output: $\mathcal{A}$ outputs a secret key usk^* for I^* including $\langle S, (D_{i,1}, D_{i,2})_{i\in\mathcal{N}}, \hat{D}\rangle$. If it is a valid secret key, usk^* should satisfy the following equation:

$$e(g, \hat{D}) \prod_{i\in I^*} e(T_i, D_{i,1}) \prod_{i\notin I^*} e(T_{n+i}, D_{i,1}) = e(g, h)^y;$$

Thus, $\mathcal{S}$ outputs $\hat{D} \cdot \prod_{i\in I^*} D_{i,1}^{\alpha_i} \cdot \prod_{i\notin I^*} D_{i,1}^{\beta_i} = h^y = g^{abc}$ and solves CTDH problem. □

5. DISCUSSION AND EXTENSION

5.1 Delegating capability

Key delegating capability is another important issue in cryptosystem. Benefits from delegation are: first, the user is able to act as an authority to delegate one's partial decryption capability to the others, and thus the system would accommodate more users without increasing the authority's workload; second, if users can be organized as a hierarchical structure, it counts as a more efficient system because the authority only needs to generate secret keys for the high level users.

Many related studies [1, 4, 11, 14] show its significance to extend user's delegating capability. This concept in attribute based encryption was first mentioned in paper [4] in which users re-randomize the secret key and output delegation keys which have similar decryption capability to the secret keys received directly from the authority. Goyal *et al.* also designated a method to extend their KPABE scheme with delegating capability. In this section, we present a new key delegation (KD) scheme from the previous ABPRE and also give a complete security proof in the standard model (shown in Appendix C.). The delegation algorithm in our scheme outputs a delegation key with some particular restrictions specified by the users, different from those directly obtained from the authority. In the following, we explain such particular restrictions to one valid user's secret key.

Each key delegation procedure involves a secret key usk, an index set I_c and op as inputs. op indicates two different policies on delegation. If $op = -$, usk' could only decrypt a subset of ciphertexts corresponding to usk, subject to the requirement that no attributes in I_c appears in the access structure no matter in a positive or a negative way. Else $op = +$, usk' could decrypt a subset of ciphertexts corresponding to usk, subject to the requirement that each attributes in I_c appears in the access structure no matter in a positive or a negative way.

According to the above explanation of delegation, we let the user's secret key be re-written in another form $usk = \langle S, I_+, I_-, (D_{i,1})_{i\in I_+}, (D_{i,2})_{i\in I_-}, (D_{i,1}, D_{i,2})_{i\in\mathcal{N}\setminus(I_+\cup I_-)}, \hat{D}\rangle$, where $I_+ = I_- = \emptyset$. Each delegation algorithm would first do a check: if ($op = -$ and $I_c \cap I_+ \neq \emptyset$) or ($op = +$ and $I_c \cap I_- \neq \emptyset$), it outputs $\perp$; else, $I_- \to I_- \cup I_c$ or $I_+ \to I_+ \cup I_c$.

Now, we give the details of the algorithm `DELEGATE` in our ABPRE scheme, `SETUP`, `KGEN`, `ENC`, `DEC` remain the same.
`DELEGATE`(usk, I_c, op): Let usk denote a valid user's secret key consisting of $\langle S, I_+, I_-, (D_{i,1})_{i\in I_+}, (D_{i,2})_{i\in I_-}, (D_{i,1}, D_{i,2})_{i\in\mathcal{N}\setminus(I_+\cup I_-)}, \hat{D}\rangle$. It first selects random $r'_1, ..., r'_n \in Z_p$. Then, if $op = -$

- $I_+ \cap I_c \neq \emptyset$, it outputs $\perp$;
- otherwise, it computes $\hat{D}' = \hat{D} \cdot (g^{\sum_{i=1}^{n} r'_i})^{-1}, \{D'_{i,2} = D_{i,2} \cdot (T'_{2n+i})^{r'_i}\}_{i\notin I_+}, \{D'_{i,1} = D_{i,1} \cdot (T'_i)^{r'_i}\}_{i\notin I_-\cup I_c, i\in S}$, $\{D'_{i,1} = D_{i,1} \cdot (T'_{n+i})^{r'_i}\}_{i\notin I_-\cup I_c, i\notin S}$. It outputs a new secret key as $usk = \langle S, I_+, I_- \cup I_c, (D_{i,1})_{i\in I_+}, (D_{i,2})_{i\in I_-\cup I_c}, (D_{i,1}, D_{i,2})_{i\in\mathcal{N}\setminus(I_+\cup I_-\cup I_c)}, \hat{D}\rangle$.

if $op = +$

- $I_- \cap I_c \neq \emptyset$, it outputs $\perp$;
- otherwise, it computes $\hat{D}' = \hat{D} \cdot (g^{\sum_{i=1}^{n} r'_i})^{-1}$, $\{D'_{i,2} = D_{i,2} \cdot (T'_{2n+i})^{r'_i}\}_{i\notin I_+\cup I_c}$, $\{D'_{i,1} = D_{i,1} \cdot (T'_i)^{r'_i}\}_{i\notin I_-, i\in S}$, $\{D'_{i,1} = D_{i,1} \cdot (T'_{n+i})^{r'_i}\}_{i\notin I_-, i\notin S}$. It outputs a new secret key as $usk = \langle S, I_+ \cup I_c, I_-, (D_{i,1})_{i\in I_+\cup I_c}, (D_{i,2})_{i\in I_-}, (D_{i,1}, D_{i,2})_{i\in\mathcal{N}\setminus(I_+\cup I_-\cup I_c)}, \hat{D}\rangle$.

THEOREM 3. *If the ADBDH assumption holds in (G, G_T), then the* ABPRE *scheme with key delegation algorithm is selective-structure chosen plaintext secure in the standard model.*

PROOF. The security model is presented in Appendix A, and the proof can be seen in Appendix C. □

5.2 Stronger Security Model

Until now, we prove our scheme selective-structure chosen plaintext secure and master key secure. In this section, we would consider a stronger chosen plaintext secure model. The adversary in Section 2.1 is not allowed to query from $\mathcal{O}_{rkg}$ and $\mathcal{O}_{ree}$ related with I_q, where I_q satisfies the challenge access structure AS^*. Actually, a stronger security model should only refuse the queries which make the adversary trivially decrypt the challenge ciphertext. That means, for any I_q satisfied AS^*, the adversary would query a re-key related with (I_q, AS') and query a re-encryption with (I_q, AS', C^*) only if he queries no secret keys that could decrypt ciphertext corresponding to the access structure AS'.

To strengthen our scheme under chosen ciphertext security notion is a challenging open problem. Notice that re-encryption algorithm probably varies the format of valid ciphertexts. Therefore, it would be very difficult to simulate the decryption oracle. Here, we give the ideal definitions on decryption query.

- In Phase 1: no constraint to the adversary.
- In Phase 2: the adversary is allowed to query any re-encrypted ciphertext except the challenge ciphertext C^* and those which can be re-encrypted from C^*.

Since the current construction of ABE scheme are very complicated and the security model is very limited, we conclude three improvements of security model as our future works, i.e., to allow any non-trivial queries in $\mathcal{O}_{rkg}$ and $\mathcal{O}_{ree}$, to provide decryption oracle in phase 1 and phase 2, and to eliminate the selective phase at the beginning of the security game.

6. APPLICATION

As observed in [9, 11, 16], ABE is suitable for many application scenarios which are related to secure management of sensitive data according to user's requirements, such as audit log, medical information in hospital and confidential data of government. Meanwhile, PRE is another interesting primitive proposed recently. In this section, we demonstrate the applicability of ABPRE with multi-targeted re-encryption and selective attribute delegation.

6.1 Multi-Targeted Re-encryption

We describe a new re-encryption scenario that we call *multi targeted re-encryption* in this section. Consider a scenario in which a large school periodically sends updating schedule to all the professors. Those files are encrypted under some access policies concerning with a professor's gender (Male, $\neg$Male), experience (($>$ 5 years), $\neg$($>$ 5 years)), department (Computer, $\neg$(Computer)), specialty (Network, $\neg$(Network)). The school manager probably constructs an encrypted email with access structure (Male) $\wedge$ ($>$ 5 years) $\wedge$ (Computer) for informing these professors the location of recent conferences.

Suppose there is a male doctor Jack, with seven years of work experience in a computer department. He is qualified to open the encrypted email since the attributes set of his secret key satisfies the access structure. But considering the situation when he is sick or he goes out for travel, he wishes someone pass this important information to the others without bothering him. By employing the proposed ABPRE scheme, he could set a new access policy on others such as ((Male) $\wedge$ (Computer) $\wedge$ (Network)). Then, he generates a re-key and delivers it to his proxy who can translate his ciphertexts to those with the new access policy. In other words, the information related to conferences could now be read by any male professor with specialty network in computer department.

6.2 Selective Attribute Delegation

Consider a scenario in which Jack is the senior faculty of a large university. Jack's secret key is related with attribute set (Gender:Male), (Experience:>5 years) and (Department:Computer). If the access structure includes some information about gender, we call this ciphertext a gender sensitive ciphertext; otherwise, it is called a gender non-sensitive ciphertext. Jack is able to decrypt the ciphertexts with access policy of gender sensitive ((Male) $\wedge$ (>5 years) $\wedge$ (Computer)) and gender non-sensitive ((>5 years)$\wedge$(Computer)). As shown in Figure 2, Jack could delegate his partial decryption capability to Lucy and Paul as: Lucy can only decrypts gender sensitive ciphertexts; Paul can only decrypts gender non-sensitive ciphertexts. Lucy can also delegate his partial decryption capability to Lily and Simon concerning with department: Lily can only decrypts department sensitive ciphertexts; Simon can only decrypts department non-sensitive ciphertexts.

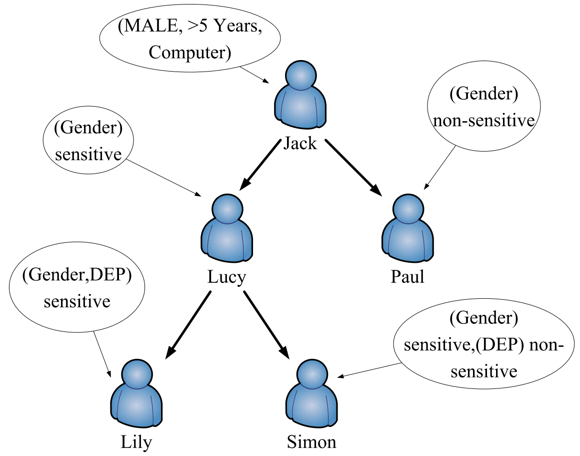

Figure 2: An example for selective attribute delegation

7. CONCLUSION

Proxy re-encryption was proposed to enable users the capability to designate a proxy who could translate their ciphertexts to others. This primitive has been well embedded into traditional cryptosystems including public key and id-based settings. Researchers also explores its applications including encrypted email forwarding, distributed file system, and the DRM of Apple's iTunes.

Combining the proxy re-encryption technique with recently introduced attribute based cryptosystem, the first attribute based proxy re-encryption scheme was proposed in this paper. All the advantages of PRE scheme can be inherited into access control environment. The security model of ABPRE was defined for the first time and our scheme can be proved selective-structure chosen plaintext secure and master key secure in the standard model assuming that ADBDH problem and ADH problem are hard to solve. Moreover, another kind of key delegation algorithm is developed in the ABPRE scheme, providing more delegating capability to each valid user.

The future work includes how to design a more delicate ABPRE scheme with higher efficiency and stronger security. Besides, it remains an open problem to construct an ABPRE scheme that has a reduction based on a more natural assumption.

8. ACKNOWLEDGMENTS

This work is supported by the National Nature Science Foundation of China (No.60673079 and No.60773086), National 973 Program (No.2007CB311201), and Research Fund for the Doctoral Program of Higher Education (No.20060248008).

9. REFERENCES

[1] Michel Abdalla, Eike Kiltz, and Gregory Neven. Generalized key delegation for hierarchical identity-based encryption. In *ESORICS*, pages 139–154, 2007.

[2] Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. In *NDSS*, 2005.

[3] Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Trans. Inf. Syst. Secur.*, 9(1):1–30, 2006.

[4] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security and Privacy*, pages 321–334, 2007.

[5] Matt Blaze, Gerrit Bleumer, and Martin Strauss. Divertible protocols and atomic proxy cryptography. In *EUROCRYPT*, pages 127–144, 1998.

[6] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In *CRYPTO*, pages 213–229, 2001.

[7] Ran Canetti and Susan Hohenberger. Chosen-ciphertext secure proxy re-encryption. In *ACM Conference on Computer and Communications Security*, pages 185–194, 2007.

[8] Melissa Chase. Multi-authority attribute based encryption. In *TCC*, pages 515–534, 2007.

[9] Ling Cheung and Calvin Newport. Provably secure ciphertext policy abe. In *ACM Conference on Computer and Communications Security*, pages 456–465, 2007.

[10] Vipul Goyal, Abhishek Jain, Omkant Pandey, and Amit Sahai. Bounded ciphertext policy attribute based encryption. In *ICALP (2)*, pages 579–591, 2008.

[11] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM Conference on Computer and Communications Security*, pages 89–98, 2006.

[12] Matthew Green and Giuseppe Ateniese. Identity-based proxy re-encryption. In *ACNS*, pages 288–306, 2007.

[13] Anca-Andreea Ivan and Yevgeniy Dodis. Proxy cryptography revisited. In *NDSS*, 2003.

[14] Masahiro Mambo and Eiji Okamoto. Proxy cryptosystems: Delegation of the power to decrypt ciphertexts. *IEICE Transactions Fundamentals*, E80-A(1):54–63, 1997.

[15] Rafail Ostrovsky, Amit Sahai, and Brent Waters. Attribute-based encryption with non-monotonic access structures. In *ACM Conference on Computer and Communications Security*, pages 195–203, 2007.

[16] Matthew Pirretti, Patrick Traynor, Patrick McDaniel, and Brent Waters. Secure attribute-based systems. In *ACM Conference on Computer and Communications Security*, pages 99–112, 2006.

[17] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, pages 457–473, 2005.

APPENDIX

A. SS CPA FOR KD SCHEME

We say that our key delegation (KD) scheme is secure against selective-structure chosen plaintext attack if no probabilistic polynomial time adversary $\mathcal{A}$ has a non-negligible advantage in winning the following game.

Init The adversary $\mathcal{A}$ chooses a challenge access structure AS^* and delivers it to the challenger. The challenger runs $\texttt{SETUP}(1^k)$ and gives $\mathcal{A}$ the resulting system public parameters *pp*. It keeps the corresponding master-key *mk* to itself.
Phase 1 The adversary $\mathcal{A}$ issues queries to the oracles:

- **Key generation oracle** $\mathcal{O}_{kg}$: On input an index set I_q, if I_q does not satisfy AS^*, then output a secret key $usk = \texttt{KGEN}(I_q, mk)$; otherwise, output ”reject”.
- **Delegation oracle** $\mathcal{O}_{delegate}$, On input an index sets (I_q, I_+, I_-)[6], an index set I_c and a symbol *op*, if $op = -$, $I_c \cap I_+ = \emptyset$ and $(I_q, I_+, I_- \cup I_c)$ does not satisfy AS^*, then output a secret key usk' for $(I_q, I_+, I_- \cup I_c)$; else if $op = +$, $I_c \cap I_- = \emptyset$ and $(I_q, I_+ \cup I_c, I_-)$ does not satisfy AS^*, then output a secret key usk' for $(I_q, I_+ \cup I_c, I_-)$; else output ”reject”.

Challenge Once the adversary $\mathcal{A}$ decides that Phase 1 is over, it outputs two equal length messages m_0, m_1 from the message space. The challenger chooses $\mu \in \{0, 1\}$ at random and encrypts M_μ with AS^*. Then, the ciphertext C^* is given to the adversary $\mathcal{A}$.
Phase 2 The same as Phase 1.
Guess The adversary $\mathcal{A}$ outputs a guess $\mu' \in \{0, 1\}$ and wins the game if $\mu' = \mu$.

We define $\mathcal{A}$'s advantage in attacking SS-CPA-KD game as follows:

$$Adv^{\mathcal{O}_{kg},\mathcal{O}_{delegate}}_{\text{SS-CPA-KD},\mathcal{A}} = |\Pr[\mu' = \mu] - \frac{1}{2}|$$

B. KEY GENERATION ORACLE

This is the key generation oracle similar with the one from paper [9]:

$\mathcal{A}$ makes a query to key generation oracle with an index set $I_q \subseteq \mathcal{N}$ where the secret key of index set I_q can not satisfy the access structure AS^* of challenge ciphertext. Therefore, there must exists an index $(j \in I_q) \wedge (j \in I_-^*)$ or $(j \notin I_q) \wedge (j \in I_+^*)$. W.l.o.g., we analyze the case of $(j \notin I_q) \wedge (j \in I_+^*)$.

For every $i \in \mathcal{N}$, $\mathcal{S}$ chooses $r'_i \in Z_p$ at random and sets r_i in two ways:

$$\begin{cases} r_i = br'_i & \text{if } i \neq j \\ r_j = ab + br'_j & \text{otherwise} \end{cases}$$

Thus, we have

$$r = \sum_{i=1}^{n} r_i = ab + \sum_{i=1}^{n} r'_i \cdot b$$

$\hat{D}$ can be computed as

$$\hat{D} = g^{y-r} = g^{-\sum_{i=1}^{n} r'_i \cdot b} = \prod_{i=1}^{n} \frac{1}{B^{r'_i}}$$

[6] A secret key for (I_q, I_+, I_-) can decrypt the ciphertexts with access structure which satisfied by I_q, relates with every attribute of index in I_+ and does not relate any attribute of index in I_-.

- For $i \in I_q, i \neq j$
  - $i \in I_+^*$,

    $$D_{i,1} = g^{\frac{r_i}{\alpha_i}} = B^{\frac{r_i'}{\alpha_i}}, D_{i,2} = g^{\frac{r_i}{b\gamma_i}} = g^{\frac{r_i'}{\gamma_i}}$$

  - $i \in I_-^*$,

    $$D_{i,1} = g^{\frac{r_i}{b\alpha_i}} = g^{\frac{r_i'}{\alpha_i}}, D_{i,2} = g^{\frac{r_i}{b\gamma_i}} = g^{\frac{r_i'}{\gamma_i}}$$

  - $i \notin I_+^* \cup I_-^*$

    $$D_{i,1} = g^{\frac{r_i}{b\alpha_i}} = g^{\frac{r_i'}{\alpha_i}}, D_{i,2} = g^{\frac{r_i}{\gamma_i}} = B^{\frac{r_i'}{\gamma_i}}$$

- For $i \notin I, i \neq j$
  - $i \in I_+^*$,

    $$D_{i,1} = g^{\frac{r_i}{b\beta_i}} = g^{\frac{r_i'}{\beta_i}}, D_{i,2} = g^{\frac{r_i}{b\gamma_i}} = g^{\frac{r_i'}{\gamma_i}}$$

  - $i \in I_-^*$,

    $$D_{i,1} = g^{\frac{r_i}{\beta_i}} = B^{\frac{r_i'}{\beta_i}}, D_{i,2} = g^{\frac{r_i}{b\gamma_i}} = g^{\frac{r_i'}{\gamma_i}}$$

  - $i \notin I_+^* \cup I_-^*$

    $$D_{i,1} = g^{\frac{r_i}{b\beta_i}} = g^{\frac{r_i'}{\beta_i}}, D_{i,2} = g^{\frac{r_i}{\gamma_i}} = B^{\frac{r_i'}{\gamma_i}}$$

- For $i = j$, $D_{i,1} = g^{\frac{r_i}{b\beta_i}} = g^{\frac{ab+br_i'}{b\beta_i}} = A^{\frac{1}{\beta_i}} g^{\frac{r_i'}{\beta_i}}$
  $D_{i,2} = g^{\frac{r_i}{b\gamma_i}} = g^{\frac{ab+br_i'}{b\gamma_i}} = A^{\frac{1}{\gamma_i}} g^{\frac{r_i'}{\gamma_i}}$

Finally, it outputs a secret key $usk = \langle I_q, (D_{i,1}, D_{i,2})_{i \in \mathcal{N}}, \hat{D} \rangle$.

C. PROOF OF THEOREM 3.

We now reduce SS-CPA security of our KD scheme to the Augment Decisional Bilinear Diffie-Hellman (ADBDH) problem.

Suppose an adversary $\mathcal{A}$ can win the SS-CPA-KD game with non-negligible advantage ε. We construct a simulator $\mathcal{S}$ who can distinguish the ADBDH tuple from a random tuple with non-negligible advantage $\frac{\varepsilon}{2}$.

We first let the challenger set the groups G and G_T with an efficient bilinear map e and a generator g. The challenger flips a fair binary coin ν, outside of $\mathcal{S}$'s view. If $\nu = 1$, the challenger sets $\langle g, A, B, C, B', Z \rangle \in \mathcal{D}_{adbdh}$; otherwise it sets $\langle g, A, B, C, B', Z \rangle \in \mathcal{D}_{rand}$.

Init In this phase, $\mathcal{S}$ receives an access structure AS^* of challenge ciphertext, and notes I_+^*, I_-^* the index sets of positive and negative attributes separately. $\mathcal{S}$ computes $Y = e(A, B)^k, h = g^k$, and selects $\alpha_i, \beta_i, \gamma_i$ at random for $i \in \mathcal{N}$. Then, $\mathcal{S}$ outputs the public parameter

$$pp := \langle e, g, h, Y, (T_i, T_{n+i}, T_{2n+i}, T_i', T_{n+i}', T_{2n+i}')_{i \in \mathcal{N}} \rangle$$

as:

- $i \in I_+^*$, $T_i = g^{\alpha_i}, T_{n+i} = B^{\beta_i}, T_{2n+i} = B^{\gamma_i}, T_i' = g^{\frac{k}{\alpha_i}}, T_{n+i}' = B'^{\frac{k}{\beta_i}}, T_{2n+i}' = B'^{\frac{k}{\gamma_i}}$;
- $i \in I_-^*$, $T_i = B^{\alpha_i}, T_{n+i} = g^{\beta_i}, T_{2n+i} = B^{\gamma_i}, T_i' = B'^{\frac{k}{\alpha_i}}, T_{n+i}' = g^{\frac{k}{\beta_i}}, T_{2n+i}' = B'^{\frac{k}{\gamma_i}}$;
- Otherwise, $T_i = B^{\alpha_i}, T_{n+i} = B^{\beta_i}, T_{2n+i} = g^{\gamma_i}, T_i' = B'^{\frac{k}{\alpha_i}}, T_{n+i}' = B'^{\frac{k}{\beta_i}}, T_{2n+i}' = g^{\frac{k}{\gamma_i}}$.

Phase 1: $\mathcal{A}$ could make several queries according to the game descriptions.

- $\mathcal{A}$ makes a query to key generation oracle with an index set I_q, where I_q does not satisfy AS^*. $\mathcal{S}$ receives $usk = \langle I_q, (D_{i,1}, D_{i,2})_{i \in \mathcal{N}}, \hat{D} \rangle$ from key generation oracle in Appendix B, and outputs augment secret key $usk = \langle I_q, \emptyset, \emptyset, (D_{i,1}, D_{i,2})_{i \in \mathcal{N}}, \hat{D} \rangle$.
- $\mathcal{A}$ makes a query to delegation oracle with index sets (I_q, I_+, I_-), an index set I_c and *op*, where $I_c \cap (I_+ \cup I_-) = \emptyset$. If I_q doesn't satisfy AS^*, $\mathcal{S}$ obtains usk from $\texttt{KGEN}(I_q, mk)$.
  - If $op = -$, outputs $\texttt{DELEGATE}(\texttt{DELEGATE}(\texttt{KGEN}(I_q, mk)$ $, I_+, +), I_- \cup I_c, -)$;
  - otherwise, outputs $\texttt{DELEGATE}(\texttt{DELEGATE}(\texttt{KGEN}(I_q, mk)$ $, I_+ \cup I_c, +), I_-, -)$.

  If I_q satisfies AS^*, we explain the case of $op = -$ ($op = +$ is the same), $\mathcal{S}$ should output a secret key for $(I_q, I_+, I_- \cup I_c)$, where $(I_q, I_+, I_- \cup I_c)$ does not satisfy AS^*. There must exist $j \in \mathcal{N}$, $(j \in I_+^*) \wedge (j \in I_q) \wedge (j \in (I_- \cup I_c))$ or $(j \in I_-^*) \wedge (j \notin I_q) \wedge (j \in (I_- \cup I_c))$ or $(j \notin (I_+^* \cup I_-^*)) \wedge (j \in I_+)$. W.l.o.g., we analyze the case of $(j \in I_+^*) \wedge (j \in I_q) \wedge (j \in (I_- \cup I_c))$.

  For every $i \in \mathcal{N}$, $\mathcal{S}$ chooses $r_i' \in Z_p$ at random and sets r_i in two ways:

  $$\begin{cases} r_i = br_i' & \text{if } i \neq j \\ r_j = ab + br_j' & \text{otherwise} \end{cases}$$

  Thus, we have

  $$r = \sum_{i=1}^{n} r_i = ab + \sum_{i=1}^{n} r_i' \cdot b$$

  The $\hat{D}$ component of the secret key can be computed as

  $$\hat{D} = h^{y-r} = h^{-\sum_{i=1}^{n} r_i' \cdot b} = \sum_{i=1}^{n} \frac{1}{B^{kr_i'}}$$

  - For $i \in I_q, i \neq j$
    - $i \in I_+^*$,

      $$D_{i,1} = h^{\frac{r_i}{\alpha_i}} = B^{\frac{kr_i'}{\alpha_i}}, D_{i,2} = h^{\frac{r_i}{b\gamma_i}} = h^{\frac{r_i'}{\gamma_i}}$$

    - $i \in I_-^*$,

      $$D_{i,1} = h^{\frac{r_i}{b\alpha_i}} = h^{\frac{r_i'}{\alpha_i}}, D_{i,2} = h^{\frac{r_i}{b\gamma_i}} = h^{\frac{r_i'}{\gamma_i}}$$

    - $i \notin I_+^* \cup I_-^*$

      $$D_{i,1} = h^{\frac{r_i}{b\alpha_i}} = h^{\frac{r_i'}{\alpha_i}}, D_{i,2} = h^{\frac{r_i}{\gamma_i}} = B^{\frac{kr_i'}{\gamma_i}}$$

  - For $i \notin I_q, i \neq j$
    - $i \in I_+^*$,

      $$D_{i,1} = h^{\frac{r_i}{b\beta_i}} = h^{\frac{r_i'}{\beta_i}}, D_{i,2} = h^{\frac{r_i}{b\gamma_i}} = h^{\frac{r_i'}{\gamma_i}}$$

    - $i \in I_-^*$,

      $$D_{i,1} = h^{\frac{r_i}{\beta_i}} = B^{\frac{kr_i'}{\beta_i}}, D_{i,2} = h^{\frac{r_i}{b\gamma_i}} = h^{\frac{r_i'}{\gamma_i}}$$

- $*$ $i \notin I_+^* \cup I_-$

$$D_{i,1} = h^{\frac{r_i}{b\beta_i}} = h^{\frac{r'_i}{\beta_i}}, D_{i,2} = h^{\frac{r_i}{\gamma_i}} = B^{\frac{kr'_i}{\gamma_i}}$$

- For $i = j$, $D_{i,2} = h^{\frac{r_i}{b\gamma_i}} = h^{\frac{ab+br'_i}{b\gamma_i}} = A^{\frac{k}{\gamma_i}} g^{\frac{kr'_i}{\gamma_i}}$

Finally, output a new secret key as $usk = \langle I_q, I_+, I_- \cup I_c, (D_{i,1})_{i \in I_+}, (D_{i,2})_{i \in I_- \cup I_c}, (D_{i,1}, D_{i,2})_{i \in \mathcal{N} \setminus (I_+ \cup I_- \cup I_c)}, \hat{D} \rangle$.

Challenge: $\mathcal{A}$ submits two message M_0 and M_1 of equal length. $\mathcal{S}$ generates challenge ciphertext:

$$C^* = \langle AS^*, M_\mu{\cdot}Z, C, C^k, (C^{\alpha_i})_{i \in I_+^*}, (C^{\beta_i})_{i \in I_-^*}, (C^{\gamma_i})_{i \notin I_+^* \cup I_-^*} \rangle.$$

Phase 2: Same as Phase 1.

Guess: $\mathcal{S}$ outputs $\nu' = 1$ if $\mathcal{A}$ give a correct guess $\mu' = \mu$; otherwise output $\nu' = 0$.