Partially Wildcarded Attribute-based Encryption and Its Efficient Construction

Go Ohtake[1], Yuki Hironaka[1], Kenjiro Kai[1], Yosuke Endo[1], Goichiro Hanaoka[2], Hajime Watanabe[2], Shota Yamada*[2,3], Kouhei Kasamatsu[2,4], Takashi Yamakawa[2,3] and Hideki Imai[2,5]

[1]*Science & Technology Research Laboratories, Japan Broadcasting Corporation, 1-10-11 Kinuta, Setagaya-ku, Tokyo 157-8510, Japan*

[2]*Research Institute for Secure Systems (RISEC), National Institute of Advanced Industrial Science and Technology (AIST), 1-1-1 Umezono, Tsukuba-shi, Ibaraki 305-8568, Japan*

[3]*Graduate School of Frontier Sciences, The University of Tokyo, 5-1-5 Kashiwanoha, Kashiwa-shi, Chiba 277-8561, Japan*

[4]*Security Solution Business Department, NTT Software Corporation, Teisan Kannai Bldg. 209, Yamashita-cho, Naka-ku, Yokohama-shi, Kanagawa 231-8551, Japan*

[5]*Faculty of Science and Engineering, Chuo University 1-13-27 Kasuga, Bunkyo-ku, Tokyo 112-8551, Japan*

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Keywords: Attribute-based Encryption, Ciphertext Policy, Wildcard.

Abstract: Many kinds of ciphertext-policy attribute-based encryption (CP-ABE) schemes have been proposed. In CP-ABE, the set of user attributes is associated with his/her secret key whereas a policy is associated with a ciphertext so that only users whose attributes satisfy the policy can decrypt the ciphertext. CP-ABE may be applied to a variety of services such as access control for file sharing systems and content distribution services. However, CP-ABE costs more for encryption and decryption in comparison with conventional public key encryption schemes since it can handle more flexible policies. In particular, wildcards, which mean that certain attributes are not relevant to the ciphertext policy, are not essential for a certain service. In this paper, we construct a partially wildcarded CP-ABE scheme with a lower decryption cost. In our scheme, the user's attributes are separated into those requiring wildcards and those not requiring wildcards. Our scheme hence embodies a CP-ABE scheme with a wildcard functionality and an efficient CP-ABE scheme without wildcard functionality. We compare our scheme with the conventional CP-ABE schemes and describe a content distribution service as an application of our scheme.

1 INTRODUCTION

1.1 Background

In attribute-based encryption (ABE), the set of user attributes is associated with a secret key or a ciphertext so that only users whose attributes satisfy the policy can decrypt the ciphertext. ABE may be applied to a variety of services, e.g., access control for file sharing systems and content distribution services. The first ABE scheme was proposed as an extension of the identity-based encryption (IBE) scheme called Fuzzy IBE (Sahai and Waters, 2005) and many kinds of ABE schemes have been proposed. ABE scheme is classified into two types: key-policy ABE (KP-ABE) (Goyal et al., 2006; Ostrovsky et al., 2007) and ciphertext-policy ABE (CP-ABE) (Bethencourt et al., 2007; Cheung and Newport, 2007; Emura et al., 2009; Katz et al., 2008; Lewko et al., 2010; Nishide et al., 2008; Okamoto and Takashima, 2010; Waters, 2011). In KP-ABE, ciphertexts are associated with attributes, and users' secret keys are associated with policies. If the attributes satisfy the key policy, the user can decrypt the ciphertext successfully. On the other hand, in CP-ABE, attributes are associated with secret keys and policies are associated with ciphertexts. If the attributes satisfy the ciphertext policy, the user can decrypt the ciphertext successfully. In this paper, we focus on CP-ABE.

Bethencourt, Sahai, and Waters proposed the first CP-ABE scheme (Bethencourt et al., 2007), where ci-

*The seventh author is supported by a JSPS Research Fellowship for Young Scientists.

Ohtake G., Hironaka Y., Kai K., Endo Y., Hanaoka G., Watanabe H., Yamada S., Kasamatsu K., Yamakawa T. and Imai H..
Partially Wildcarded Attribute-based Encryption and Its Efficient Construction.
DOI: 10.5220/0004509303390346
In *Proceedings of the 10th International Conference on Security and Cryptography* (SECRYPT-2013), pages 339-346
ISBN: 978-989-8565-73-0
Copyright © 2013 SCITEPRESS (Science and Technology Publications, Lda.)

phertext policies are expressed by a tree structure including **AND**-gates and **OR**-gates. This scheme allows ciphertext policies to be very expressive, but it has larger costs for encryption and decryption than conventional public key encryption schemes. In contrast, Cheung and Newport proposed an efficient CP-ABE scheme (Cheung and Newport, 2007), where ciphertext policies are compactly expressed by **AND**-gates and three types of attribute values: *positive*, *negative*, and *don't care*. This scheme has much lower costs for encryption and decryption than the scheme in (Bethencourt et al., 2007). However, the expression of ciphertext policies is rather restricted: the size of possible values for each attribute is only one bit. On the other hand, Nishide, Yoneyama, and Ohta proposed a CP-ABE scheme (Nishide et al., 2008) where ciphertext policies are expressed by **AND**-gates and a subset of possible values for each attribute and the corresponding policies are hidden for the purpose of guaranteeing the recipient's anonymity. Both (Cheung and Newport, 2007) and (Nishide et al., 2008) construct an efficient CP-ABE scheme with limited ciphertext policies by using only **AND**-gates. Furthermore, in these schemes, encryptors can use *wildcards* to mean that certain attributes are not relevant to the ciphertext policy. On the other hand, Emura, Miyaji, Nomura, Omote, and Soshi proposed a CP-ABE scheme (Emura et al., 2009) that is more efficient than those of (Cheung and Newport, 2007) and (Nishide et al., 2008) by removing the wildcard functionality. In this scheme, ciphertext policies are expressed by **AND**-gates and one of the possible values for each attribute. This scheme has much lower costs for decryption compared with the scheme presented in (Nishide et al., 2008).

CP-ABE schemes with a wildcard functionality (Cheung and Newport, 2007; Nishide et al., 2008) are effective for services where certain attributes might not be relevant to the ciphertext policy. However, this scheme is functionally redundant if all attributes are relevant. In contrast, the CP-ABE scheme without the wildcard functionality (Emura et al., 2009) has much lower costs for decryption. However, this scheme cannot be applied to services where certain attributes are not relevant to the ciphertext policy.

In particular, the decryption costs of broadcasting services must be as small as possible, since the devices in the user terminals are usually lower in performance than personal computers and it is possible that the decryption process is performed on tamper-resistant devices such as smart cards.

1.2 Our Contributions

In this paper, we propose a partially wildcarded CP-ABE scheme to reduce the decryption cost. (Emura et al., 2009) shows that the presence or absence of wildcard functionality has an influence on the efficiency of the CP-ABE scheme. In our scheme, an user's attribute list is separated into a list of attributes which require wildcards and an list of attributes which do not require wildcards. Our scheme embodies a CP-ABE scheme with a wildcard functionality and an efficient CP-ABE scheme without a wildcard functionality. Our idea is to split the master secret key into two shares by using 2-out-of-2 secret sharing and to use the shares as master secret keys of each CP-ABE scheme. We compare our scheme with conventional CP-ABE schemes and describe a content distribution service as an application of our scheme. For example, if there is only one attribute that requires wildcards among four attributes, our scheme can reduce the decryption cost by 40% in comparison with the conventional CP-ABE schemes.

2 PRELIMINARIES

2.1 Model

A CP-ABE scheme consists of the following four algorithms (Nishide et al., 2008).

Setup(1^k): This algorithm takes the security parameter k as input and generates a public key PK and a master key MK.

KeyGen(MK, L): This algorithm takes MK and an attribute list L as input and generates a secret key SK_L associated with L.

Encrypt(PK, M, W): This algorithm takes PK, a message M, and an ciphertext policy W as input and generates a ciphertext CT.

Decrypt(CT, SK_L): This algorithm takes CT and SK_L associated with L as input. We use the notation $L \models W$ to mean that L satisfies W. If $L \models W$, it returns the message M such that $\mathsf{Decrypt}(\mathsf{Encrypt}(PK, M, W), SK_L) = M$

2.2 Security Definition

We consider the following security game.

Init: The adversary $\mathcal{A}$ chooses the challenge ciphertext policy W and gives it to the challenger $\mathcal{B}$.

Setup: $\mathcal{B}$ runs Setup and gives PK to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ transmits an attribute list L for KeyGen query to $\mathcal{B}$. $\mathcal{B}$ returns SK_L associated with L to $\mathcal{A}$ iff $L \not\models W$.

Challenge: $\mathcal{A}$ transmits two messages M_0 and M_1 to $\mathcal{B}$. $\mathcal{B}$ chooses $b \in \{0,1\}$ at random, generates a ciphertext $CT = \mathsf{Encrypt}(PK, M_b, W)$, and transmits it to $\mathcal{A}$.

Phase 2: Same as **Phase 1**.

Guess: $\mathcal{A}$ outputs a guess b' of b.

The security of the CP-ABE scheme is defined as follows:

Definition 1. *We say that a CP-ABE scheme is selective IND-CPA secure if* $Adv_{\mathcal{A}} = \left|\Pr[b' = b] - \frac{1}{2}\right|$ *is negligible in the above game.*

2.3 Bilinear Maps

Let $\mathbb{G}, \mathbb{G}_T$ be multiplicative cyclic groups of prime order p and g be a generator of $\mathbb{G}$. A bilinear map is a map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ with the following properties:

Bilinear: $e(g^a, g^b) = e(g,g)^{ab} \quad \forall a, b \in \mathbb{Z}_p$

Non-degenerate: $e(g,g) \neq 1$

We say that $\mathbb{G}$ is a bilinear group if the group action in $\mathbb{G}$ can be efficiently computed and there exists a group $\mathbb{G}_T$ and an efficiently computable bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, as above.

2.4 Decisional Bilinear Diffie-Hellman (DBDH) Assumption

Let $z_1, z_2, z_3 \in \mathbb{Z}_p^*$ be chosen at random and $g \in \mathbb{G}$ be a generator. Also, let Z be a random element in $\mathbb{G}_T$. The DBDH assumption is that no probabilistic polynomial-time algorithm can distinguish the tuple $[g, g^{z_1}, g^{z_2}, g^{z_3}, e(g,g)^{z_1 z_2 z_3}]$ from the tuple $[g, g^{z_1}, g^{z_2}, g^{z_3}, Z]$ with a non-negligible advantage.

3 CONVENTIONAL SCHEMES

Here, we describe the CP-ABE algorithm with a wildcard functionality that was proposed in (Nishide et al., 2008) and the algorithm without a wildcard functionality that was proposed in (Emura et al., 2009).

3.1 CP-ABE with Wildcard (Nishide et al., 2008)

In (Cheung and Newport, 2007), each attribute can take two values: 1 (*positive*) and 0 (*negative*), but in (Nishide et al., 2008), each attribute can take two or more values, and each W_i in a ciphertext policy W can be any subset of possible values for each attribute $\mathbb{A}_i$. In this paper, the user's attribute list is simply represented by indices corresponding to the possible values for each attribute. Let $S_i = \{1, 2, ..., n_i\}$ be the set of possible values for $\mathbb{A}_i$ where n_i is the number of possible values for $\mathbb{A}_i$. Then, let $L = [L_1, L_2, ..., L_n]$ be the attribute list where $L_i \in S_i$ and let $W = [W_1, W_2, ..., W_n]$ be the ciphertext policy where $W_i \subseteq S_i$. When the encryptor specifies a wildcard for $\mathbb{A}_i$, it corresponds to specifying $W_i = S_i$ for $\mathbb{A}_i$. The attribute list L satisfies the ciphertext policy W; that is, $L \models W$ iff $L_i \in W_i$ for all $i \in [n]$.

Setup(1^k): Choose multiplicative cyclic groups $\mathbb{G}$ and $\mathbb{G}_T$ of prime order p, a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, and a random generator $g \in \mathbb{G}$. Then, pick w, $a_{i,t}$, $b_{i,t} \in \mathbb{Z}_p^*$, $A_{i,t} \in \mathbb{G}$ at random for $i \in [n]$, $t \in [n_i]$. Compute $Y = e(g,g)^w$ and output the public key $PK = \langle p, \mathbb{G}, \mathbb{G}_T, e, g, Y, \{\{A_{i,t}^{a_{i,t}}, A_{i,t}^{b_{i,t}}\}_{t \in [n_i]}\}_{i \in [n]} \rangle$ and the master key $MK = \langle w, \{\{a_{i,t}, b_{i,t}\}_{t \in [n_i]}\}_{i \in [n]} \rangle$.

KeyGen(MK, L): Let $L = [L_1, L_2, ..., L_n]$ be the attribute list for the user who will obtain the corresponding secret key. Pick random values $s_i \in \mathbb{Z}_p^*$ for $i \in [n]$, set $s = \sum_{i=1}^n s_i$, and compute $D_0 = g^{w-s}$. Then, pick random values $\lambda_i \in \mathbb{Z}_p^*$ for $i \in [n]$ and compute $\{D_{i,0}, D_{i,1}, D_{i,2}\} = \{g^{s_i} \cdot (A_{i,L_i})^{a_{i,L_i} b_{i,L_i} \lambda_i}, g^{a_{i,L_i} \lambda_i}, g^{b_{i,L_i} \lambda_i}\}$. Output the secret key $SK_L = \langle D_0 \cdot \prod_{i=1}^n D_{i,0}, \{D_{i,1}, D_{i,2}\}_{i \in [n]} \rangle$ associated with L.

Encrypt(PK, M, W): Let $W = [W_1, W_2, ..., W_n]$ be a ciphertext policy and $M \in \mathbb{G}_T$ be a message. Pick a random value $r \in \mathbb{Z}_p^*$ and compute $\tilde{C} = M \cdot Y^r$ and $C_0 = g^r$. Then, execute the following process for all $i \in [n]$: pick random values $r_{i,t} \in \mathbb{Z}_p^*$ for $t \in [n_i]$. If $t \in W_i$, compute $\{C_{i,t,1}, C_{i,t,2}\} = \{(A_{i,t}^{b_{i,t}})^{r_{i,t}}, (A_{i,t}^{a_{i,t}})^{r - r_{i,t}}\}$. If $t \notin W_i$, let $\{C_{i,t,1}, C_{i,t,2}\}$ be random values in $\mathbb{G}$. Output the ciphertext $CT = \langle \tilde{C}, C_0, \{\{C_{i,t,1}, C_{i,t,2}\}_{t \in [n_i]}\}_{i \in [n]} \rangle$.

Decrypt(CT, SK_L): Check whether the attribute list L for the user satisfies the ciphertext policy W. If $L \models W$, output the message,

$$M = \frac{\tilde{C} \cdot \prod_{i=1}^n e(C_{i,L_i,1}, D_{i,1}) \cdot e(C_{i,L_i,2}, D_{i,2})}{e(C_0, D_0 \cdot \prod_{i=1}^n D_{i,0})}.$$

3.2 CP-ABE without Wildcard (Emura et al., 2009)

In (Emura et al., 2009), each attribute can take two or more values, and each W_i in a ciphertext policy W can

be any one of the possible values for attribute $\mathbb{A}_i$. As a result, the encryptor cannot specify a wildcard.

In this paper, an user's attribute list is simply represented by indices corresponding to possible values for each attribute. Let $S_i = \{1,2,...,n_i\}$ be a set of possible values for $\mathbb{A}_i$ where n_i is the number of the possible values for $\mathbb{A}_i$. Then, let $L = [L_1, L_2, ..., L_n]$ be the attribute list where $L_i \in S_i$ and let $W = [W_1, W_2, ..., W_n]$ be the ciphertext policy where $W_i \in S_i$. The attribute list L satisfies the ciphertext policy W, that is, $L \models W$ iff $L_i = W_i$ for all $i \in [n]$.

Setup(1^k): Choose multiplicative cyclic groups $\mathbb{G}$ and $\mathbb{G}_T$ of prime order p, a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, and random generators $g, h \in \mathbb{G}$. Then, pick $y, t_{i,j} \in \mathbb{Z}_p$ at random for $i \in [n]$ and $j \in [n_i]$. Compute $Y = e(g,h)^y$ and $T_{i,j} = g^{t_{i,j}}$. Output the public key $PK = \langle p, \mathbb{G}, \mathbb{G}_T, e, g, h, Y, \{\{T_{i,j}\}_{j\in[n_i]}\}_{i\in[n]}\rangle$ and the master key $MK = \langle y, \{\{t_{i,j}\}_{j\in[n_i]}\}_{i\in[n]}\rangle$.

KeyGen(MK, L): Let $L = [L_1, L_2, ..., L_n]$ be the attribute list for the user who will obtain the corresponding secret key. Pick a random value $r \in \mathbb{Z}_p$ and output the secret key $SK_L = \langle h^y \cdot (g^{\sum_{i\in[n]} t_{i,L_i}})^r, g^r\rangle$ associated with L.

Encrypt(PK, M, W): Let $W = [W_1, W_2, ..., W_n]$ be a ciphertext policy and $M \in \mathbb{G}_T$ be a message. Pick a random value $s \in \mathbb{Z}_p$ and compute $C_1 = M \cdot Y^s$, $C_2 = g^s$, and $C_3 = (\prod_{i\in[n]} T_{i,W_i})^s = (g^{\sum_{i\in[n]} t_{i,W_i}})^s$. Output the ciphertext $CT = \langle W, C_1, C_2, C_3\rangle$.

Decrypt(CT, SK_L): Check whether the attribute list L for the user satisfies the ciphertext policy W. If $L \models W$, output the message

$$M = \frac{C_1 \cdot e(C_3, g^r)}{e(C_2, h^y \cdot (g^{\sum_{i\in[n]} t_{i,L_i}})^r)}.$$

4 PROPOSED SCHEME

We propose a partially wildcarded CP-ABE scheme to reduce the decryption cost.

4.1 Overview of Proposed Scheme

In (Emura et al., 2009), it is found that a more efficient CP-ABE scheme than the scheme in (Nishide et al., 2008) can be constructed by removing the wildcard functionality. Hence, the presence or absence of the wildcard functionality has an influence on the efficiency of the CP-ABE scheme. In our scheme, the user's attribute list is separated into a list of attributes requiring wildcards and a list of attributes not requiring wildcards. Our scheme thus embodies schemes with and without the wildcard functionality. Generally, a CP-ABE scheme without a wildcard functionality has a smaller cost than one with a wildcard functionality. Therefore, the larger the number of attributes not requiring a wildcard functionality is, the smaller the total cost of the CP-ABE scheme will be.

However, combining two schemes with and without the wildcard functionality is not trivial. When the secret key corresponding to the attributes requiring wildcards and the secret key corresponding to the attributes not requiring wildcards are generated, they are associated with each other by using a random number in order to prevent collusion attacks. Furthermore, the ciphertext size is reduced by the encryption algorithms sharing another random number.

In (Nishide et al., 2008), the authors achieve recipient anonymity by hiding the subset W_i for each $\mathbb{A}_i$ specified in the ciphertext policy of the **AND**-gate of all the attributes. However, ciphertext policies must be revealed in certain services. For example, in a content distribution service, users must know what attributes are required for playing content. In this paper, we construct a modified CP-ABE scheme by removing recipient anonymity from the CP-ABE scheme in (Nishide et al., 2008) and use the modified scheme as a CP-ABE scheme with a wildcard functionality. Moreover, we combine it with the CP-ABE scheme without the wildcard functionality in (Emura et al., 2009).

4.2 Proposed Scheme

Let $\hat{n}$ be the number of attributes $\hat{\mathbb{A}}_i$ which require wildcards and $\check{n}$ be the number of attributes $\check{\mathbb{A}}_i$ which do not require wildcards. Moreover, let $\hat{S}_i = \{1,2,\ldots,\hat{n}_i\}$ be the set of possible values for attribute $\hat{\mathbb{A}}_i$ and $\check{S}_i = \{1,2,\ldots,\check{n}_i\}$ be the set of possible values for attribute $\check{\mathbb{A}}_i$. The user's attribute list L is separated into one list $\hat{L} = \{\hat{L}_1, \hat{L}_2, ..., \hat{L}_{\hat{n}}\}$ which requires wildcards, where $\hat{L}_i \in \hat{S}_i$, and another list $\check{L} = \{\check{L}_1, \check{L}_2, ..., \check{L}_{\check{n}}\}$ which does not require wildcards, where $\check{L}_i \in \check{S}_i$. Also, the ciphertext policy W is separated into a ciphertext policy $\hat{W} = \{\hat{W}_1, \hat{W}_2, ..., \hat{W}_{\hat{n}}\}$ which requires wildcards, where $\hat{W}_i \subseteq \hat{S}_i$, and a policy $\check{W} = \{\check{W}_1, \check{W}_2, ..., \check{W}_{\check{n}}\}$ which does not require wildcards, where $\check{W}_i \in \check{S}_i$.

Setup(1^k, $\hat{n}$, $\check{n}$, $\{\hat{n}_i\}_{i\in[\hat{n}]}$, $\{\check{n}_i\}_{i\in[\check{n}]}$): Choose multiplicative cyclic groups $\mathbb{G}$ and $\mathbb{G}_T$ of prime order p, a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, and a random generator $g \in \mathbb{G}$. Then, pick a random value $w \in \mathbb{Z}_p^*$, and compute $Y = e(g,g)^w$. Also, pick a random value $A_{i,j} \in \mathbb{G}$ for all $i \in [\hat{n}]$ and $j \in [\hat{n}_i]$, and pick a random value $T_{i,j} \in \mathbb{G}$ for all $i \in [\check{n}]$ and $j \in [\check{n}_i]$. Output the public key $PK = \langle p, \mathbb{G},$

$\mathbb{G}_T$, e, g, Y, $\{\{A_{i,j}\}_{j\in[\hat{n}_i]}\}_{i\in[\hat{n}]}$, $\{\{T_{i,j}\}_{j\in[\check{n}_i]}\}_{i\in[\check{n}]}$ $\rangle$ and the master key $MK = w$.

KeyGen(PK, MK, $\hat{L}$, $\check{L}$): Pick a random value $\xi \in \mathbb{Z}_p^*$ and a random value $s_i \in \mathbb{Z}_p^*$ for all $i \in [\hat{n}]$. Set $s = \sum_{i=1}^{\hat{n}} s_i$ and compute $D = g^{w-s+\xi}$. After that, pick a random value $\lambda_i \in \mathbb{Z}_p^*$ for all $i \in [\hat{n}]$ and compute $\{D_{i,0}, D_{i,1}\} = \{g^{s_i} \cdot A_{i,\hat{L}_i}^{\lambda_i}, g^{\lambda_i}\}$. Also, pick a random value $u \in \mathbb{Z}_p^*$ and compute $\{D'_1, D'_2\} = \{g^{-\xi} \cdot (\prod_{i\in[\check{n}]} T_{i,\check{L}_i})^u, g^u\}$. Output the secret key $SK_{[\hat{L},\check{L}]} = \langle\ \hat{L}, \check{L}, D \cdot \prod_{i\in[\hat{n}]} D_{i,0} \cdot D'_1, \{D_{i,1}\}_{i\in[\hat{n}]}, D'_2\ \rangle$ associated with the user's attribute list $\hat{L}$ and $\check{L}$.

Encrypt(PK, M, $\hat{W}$, $\check{W}$): Pick a random value $r \in \mathbb{Z}_p^*$ and compute $C_1 = M \cdot Y^r$, $C_2 = g^r$, $C_3 = (\prod_{i\in[\check{n}]} T_{i,\check{W}_i})^r$. Then, compute $C_{i,j} = A_{i,j}^r$ for all $i \in [\hat{n}]$ and $j \in \hat{W}_i$. Output the ciphertext $C_{[\hat{W},\check{W}]} = \langle\ \hat{W}, \check{W}, C_1, C_2, C_3, \{C_{i,j}\}_{i\in[\hat{n}], j\in\hat{W}_i}\ \rangle$.

Decrypt($SK_{[\hat{L},\check{L}]}$, $C_{[\hat{W},\check{W}]}$): If $\hat{L} \models \hat{W}$ and $\check{L} \models \check{W}$, output the message,

$$M = C_1 \cdot \frac{\prod_{i\in[\hat{n}]} e(C_{i,\hat{L}_i}, D_{i,1}) \cdot e(C_3, D'_2)}{e(D \cdot \prod_{i\in[\hat{n}]} D_{i,0} \cdot D'_1, C_2)}.$$

4.3 Security Proof

Theorem 1. *Our scheme is selective IND-CPA secure if the DBDH assumption holds in* $\mathbb{G}$.

Proof. Let $\mathcal{A}$ be an adversary interested in thwarting our scheme. We build an algorithm $\mathcal{B}$ that solves the DBDH problem in $\mathbb{G}$ by using $\mathcal{A}$. Pick random values α, β, γ in $\mathbb{Z}_p^*$ and compute $g_1 = g^\alpha$, $g_2 = g^\beta$, and $g_3 = g^\gamma$. Then, pick a random bit $\delta \in \{0,1\}$. If $\delta = 1$, set $R = e(g,g)^{\alpha\beta\gamma}$. If $\delta = 0$, let R be a random value in $\mathbb{G}_T$. $\mathcal{B}$ takes as input $\langle g, g_1, g_2, g_3, R\rangle$ and proceeds as follows:

Init: $\mathcal{A}$ chooses the challenge ciphertext policies $\hat{W}^* = (\hat{W}_1^*, \hat{W}_2^*, ..., \hat{W}_{\hat{n}}^*)$ and $\check{W}^* = (\check{W}_1^*, \check{W}_2^*, ..., \check{W}_{\check{n}}^*)$ and gives them to $\mathcal{B}$.

Setup: $\mathcal{B}$ receives $\hat{W}^*$ and $\check{W}^*$. It computes the public key as follows: $\mathcal{B}$ computes $Y = e(g_1, g_2)$. After that, it picks random values $a_{i,j}$ in $\mathbb{Z}_p^*$ for all $i \in [\hat{n}]$ and $j \in [\hat{n}_i]$. If $j \in \hat{W}_i^*$, it computes $A_{i,j} = g^{a_{i,j}}$. If $j \notin \hat{W}_i^*$, it computes $A_{i,j} = g_1^{a_{i,j}}$. Moreover, $\mathcal{B}$ picks random values $b_{i,j}$ in $\mathbb{Z}_p^*$ for all $i \in [\check{n}]$ and $j \in [\check{n}_i]$. If $j = \check{W}_i^*$, it computes $T_{i,j} = g^{b_{i,j}}$. If $j \neq \check{W}_i^*$, it computes $T_{i,j} = g_1^{b_{i,j}}$. Finally, it returns $PK = \langle g, Y, \{\{A_{i,j}\}_{j\in[\hat{n}_i]}\}_{i\in[\hat{n}]}, \{\{T_{i,j}\}_{j\in[\check{n}_i]}\}_{i\in[\check{n}]}\rangle$ to $\mathcal{A}$.

Phase 1: When $\mathcal{A}$ transmits the attribute lists $\hat{L} = (\hat{L}_1, \hat{L}_2, ..., \hat{L}_{\hat{n}})$ and $\check{L} = (\check{L}_1, \check{L}_2, ..., \check{L}_{\check{n}})$ for the KeyGen query to $\mathcal{B}$, $\mathcal{B}$ returns the corresponding secret key as follows: If $(\hat{L} \models \hat{W}^*) \wedge (\check{L} \models \check{W}^*)$, $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$. Here, the query can be classified into three types. A type 1 query satisfies $(\hat{L} \not\models \hat{W}^*) \wedge (\check{L} \not\models \check{W}^*)$; a type 2 query satisfies $(\hat{L} \models \hat{W}^*) \wedge (\check{L} \not\models \check{W}^*)$; and a type 3 query satisfies $(\hat{L} \not\models \hat{W}^*) \wedge (\check{L} \models \check{W}^*)$.

- Type 1 or Type 2: $\mathcal{B}$ picks a random value ξ' in $\mathbb{Z}_p^*$. After that, it picks a random value s_i in $\mathbb{Z}_p^*$ for all $i \in [\hat{n}]$. It computes $s = \sum_{i=1}^{\hat{n}} s_i$ and $D = g^{s-\xi'}$. It then picks random values λ_i in $\mathbb{Z}_p^*$ for all $i \in [\hat{n}]$ and computes $D_{i,0} = g^{s_i} \cdot A_{i,\hat{L}_i}^{\lambda_i}$ and $D_{i,1} = g^{\lambda_i}$. In Type 1 and Type 2, $\check{L} \not\models \check{W}^*$ is satisfied. Therefore, $\mathcal{B}$ can set $\sum_{i\in[\check{n}]} t_{i,\check{L}_i} = T_1 + T_2\alpha$, where $T_2 \neq 0$ (The probability of $T_2 = 0$ is negligible and it has no influence on the security proof, so we will omit this case. See (Emura et al., 2009).). $\mathcal{B}$ can compute T_1 and T_2 by using $\{b_{i,j}\}_{i\in[\check{n}], j\in[\check{n}_i]}$. It picks a random value u' in $\mathbb{Z}_p^*$ and computes $D'_1 = g^{\xi'} \cdot g_1^{u'} \cdot g^{\frac{T_1 u'}{T_2}} \cdot g_2^{-\frac{T_1}{T_2}}$ and $D'_2 = g^{\frac{u'}{T_2}} \cdot g_2^{-\frac{1}{T_2}}$.
- Type 3: $\mathcal{B}$ picks random values ξ and u in $\mathbb{Z}_p^*$ and computes $D'_1 = g^{-\xi} \cdot (\prod_{i\in[\check{n}]} T_{i,\check{L}_i})^u$ and $D'_2 = g^u$. In Type 3, $\hat{L} \not\models \hat{W}^*$ is satisfied. Therefore, there exists the index k such that $\hat{L}_k \notin \hat{W}_k$. $\mathcal{B}$ picks random values s_i in $\mathbb{Z}_p^*$ for all $i \in [\hat{n}]\backslash k$. Then, it picks random values λ_i in $\mathbb{Z}_p^*$ and computes $D_{i,0} = g^{s_i} \cdot A_{i,\hat{L}_i}^{\lambda_i}$ and $D_{i,1} = g^{\lambda_i}$. It picks random values s'_k and λ'_k in $\mathbb{Z}_p^*$ for the index k such that $\hat{L}_k \notin \hat{W}_k$ and computes $D = g^{-s'_k - \sum_{i\in[\hat{n}]\backslash k} s_i + \xi}$, $D_{k,0} = g_1^{\lambda'_k} \cdot g^{s'_k}$, and $D_{k,1} = g^{\frac{\lambda'_k}{a_{k,\hat{L}_k}}} \cdot g_2^{\frac{1}{a_{k,\hat{L}_k}}}$.

Finally, $\mathcal{B}$ computes $SK_{[\hat{L},\check{L}]} = \langle\ \hat{L}, \check{L}, D \cdot \prod_{i\in[\hat{n}]} D_{i,0} \cdot D'_1, \{D_{i,1}\}_{i\in[\hat{n}]}, D'_2\ \rangle$ and returns $SK_{[\hat{L},\check{L}]}$ to $\mathcal{A}$.

Lemma 1. *$SK_{[\hat{L},\check{L}]}$ is distributed identically to that in the real IND-CPA game.*

We will prove Lemma 1 after showing the advantage of $\mathcal{B}$.

Challenge: $\mathcal{A}$ transmits two messages M_0 and M_1 to $\mathcal{B}$. $\mathcal{B}$ picks a random bit $\eta \in \{0,1\}$ and computes $C_1 = M_\eta \cdot R$, $C_2 = g_3$, and $C_3 = g_3^{\sum_{i\in[\check{n}]} b_{i,\check{W}_i^*}}$. It then computes $C_{i,j} = g_3^{a_{i,j}}$ for all $i \in [\hat{n}]$ and $j \in \hat{W}_i^*$. It returns the challenge ciphertext $C_{\hat{W}^*,\check{W}^*} = \langle\ \hat{W}^*, \check{W}^*, C_1, C_2, C_3, \{C_{i,j}\}_{i\in[\hat{n}], j\in\hat{W}_i^*}\ \rangle$ to $\mathcal{A}$.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Lemma 2. *If $R = e(g,g)^{\alpha\beta\gamma}$, $C_{\hat{W}^*,\check{W}^*}$ is distributed identically to the challenge ciphertext in the real IND-CPA game. Otherwise, $\mathcal{A}$ can obtain no information about η.*

this Lemma can be proved easily since $A_{i,j} = g^{a_{i,j}}$ for all $i \in [\hat{n}]$, $j \in \hat{W}_i^*$ and $T_{i,j} = g^{b_{i,j}}$ for all $i \in [\check{n}]$ if $j = \check{W}_i^*$. Hence, we will omit the proof.

Phase 2: Same as Phase 1.

Guess: $\mathcal{A}$ outputs η'. $\mathcal{B}$ outputs $\delta' = 1$ if $\eta = \eta'$. Otherwise, $\mathcal{B}$ outputs $\delta' = 0$.

$\mathcal{B}$ outputs $\delta' = 1$ iff $\mathcal{A}$ can predict the value of η. When $R = e(g,g)^{\alpha\beta\gamma}$, $\mathcal{B}$ completely simulates the IND-CPA game for $\mathcal{A}$. In contrast, when R is a random element in $\mathbb{G}_T$, the value of η is information-theoretically hidden, so the probability that $\mathcal{A}$ can predict the value of η is $1/2$. Hence, $\Pr[\delta' = 1|\delta = 1] = Adv_{\mathcal{A}}^{CPA}(k) + \frac{1}{2}$ and $\Pr[\delta' = 1|\delta = 0] = \frac{1}{2}$. That is, the advantage of $\mathcal{B}$ solving the DBDH problem is as follows:

$$\left|\Pr[\delta' = 1|\delta = 1] - \Pr[\delta' = 1|\delta = 0]\right| = Adv_{\mathcal{A}}^{CPA}(k)$$

If $Adv_{\mathcal{A}}^{CPA}(k)$ is non-negligible, $\mathcal{B}$ has non-negligible advantage for the DBDH problem, which contradicts the DBDH assumption. Therefore, in the selective IND-CPA game for our scheme, the advantage of $\mathcal{A}$ is negligible. Hence, Theorem 1 holds. □

Proof. To prove Lemma 1, we show that $SK_{[\hat{L},\check{L}]}$ generated by $\mathcal{B}$ satisfies the following equations:

$$SK_{[\hat{L},\check{L}]} = \langle \hat{L}, \check{L}, D'', \{D_{i,1}\}_{i\in[\hat{n}]}, D'_2 \rangle \quad (1)$$

$$\text{where } D = g^{w-s+\xi} \quad (2)$$

$$D_{i,0} = g^{s_i} \cdot A_{i,\hat{L}_i}^{\lambda_i} \text{ for all } i \in [\hat{n}] \quad (3)$$

$$D_{i,1} = g^{\lambda_i} \text{ for all } i \in [\hat{n}] \quad (4)$$

$$D'_1 = g^{-\xi} \cdot g^{u\cdot\sum_{i\in[\hat{n}]} t_{i,\hat{L}_i}} \quad (5)$$

$$D'_2 = g^{u} \quad (6)$$

$$D'' = D \cdot \prod_{i\in[\hat{n}]} D_{i,0} \cdot D'_1 \quad (7)$$

where $s = \sum_{i=1}^{\hat{n}} s_i$ and $w = \alpha\beta$.

- Type 1 or Type 2: By setting $-\xi' = \alpha\beta + \xi$, it becomes obvious that D, $D_{i,0}$, and D'_1 respectively satisfy equations (2), (3), and (4). These computations do not require $g^{\alpha\beta}$, so $\mathcal{B}$ can easily compute them. Next, D'_1 and D'_2 can be computed as follows by setting $-\xi = \alpha\beta + \xi'$ and $u' = \beta + uT_2$.

$$\begin{aligned} D'_1 &= g^{\xi'} \cdot g_1^{u'} \cdot g^{\frac{T_1 u'}{T_2}} \cdot g_2^{-\frac{T_1}{T_2}} \\ &= g^{\xi'} \cdot g^{\alpha u'} \cdot g^{T_1 \cdot \frac{u'-\beta}{T_2}} \\ &= g^{\xi'} \cdot g^{\alpha(\beta+uT_2)} \cdot g^{uT_1} \\ &= g^{(\alpha\beta+\xi')} \cdot g^{u(T_1+T_2\alpha)} \\ &= g^{-\xi} \cdot g^{u\cdot\sum_{i\in[\hat{n}]} t_{i,\hat{L}_i}} \end{aligned}$$

$$\begin{aligned} D'_2 &= g^{\frac{u'}{T_2}} \cdot g_2^{-\frac{1}{T_2}} \\ &= g^{\frac{\beta+uT_2}{T_2}} \cdot g^{-\frac{\beta}{T_2}} \\ &= g^{u} \end{aligned}$$

where $\sum_{i\in[\check{n}]} t_{i,\check{L}_i} = T_1 + T_2\alpha$ from the conditions of Type 1 and Type 2. ξ' and u' are uniformly and randomly chosen, so ξ and u are also uniformly and randomly distributed, and D'_1 and D'_2 satisfy the above equations. Therefore, D'_1 and D'_2 satisfy equations (5) and (6). Hence, $SK_{[\hat{L},\check{L}]}$ generated by $\mathcal{B}$ satisfies equation (1), and Lemma 1 holds.

- Type 3: It is obvious that D'_1 and D'_2 satisfy equations (5) and (6) since their computations are similar to equations (5) and (6) without $g^{\alpha\beta}$. As a result, it is obvious that $\{D_{i,0}, D_{i,1}\}_{i\in[\hat{n}]\setminus k}$, where k is an index wherein $\hat{L}_k \notin \hat{W}_k$, satisfies equations (3) and (4) since their computations are similar to equations (3) and (4) without $g^{\alpha\beta}$. D, $D_{k,0}$, and $D_{k,1}$ can be computed as follows by setting $-s'_k = \alpha\beta - s_k$, $\lambda'_k = a_{k,\hat{L}_k}\lambda_k - \beta$.

$$\begin{aligned} D &= g^{-s'_k - \sum_{i\in[\hat{n}]\setminus k} s_i + \xi} \\ &= g^{\alpha\beta - s_k - \sum_{i\in[\hat{n}]\setminus k} s_i + \xi} \\ &= g^{w-s+\xi} \end{aligned}$$

$$\begin{aligned} D_{k,0} &= g_1^{\lambda'_k} \cdot g^{s'_k} \\ &= g^{\alpha(-\beta + a_{k,\hat{L}_k}\lambda_k)} \cdot g^{s'_k} \\ &= g^{-\alpha\beta + s'_k} \cdot g_1^{a_{k,\hat{L}_k}\lambda_k} \\ &= g^{s_k} \cdot A_{k,\hat{L}_k}^{\lambda_k} \end{aligned}$$

$$\begin{aligned} D_{k,1} &= g^{\frac{\lambda'_k}{a_{k,\hat{L}_k}}} \cdot g_2^{\frac{1}{a_{k,\hat{L}_k}}} \\ &= g^{\frac{\lambda'_k + \beta}{a_{k,\hat{L}_k}}} \\ &= g^{\lambda_k} \end{aligned}$$

s'_k is uniformly and randomly chosen, so s_k is also uniformly and randomly distributed and D, $D_{k,0}$, and $D_{k,1}$ satisfy the above equations. There-

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Table 1: Comparison of our scheme and conventional CP-ABE schemes. (See Section 5.1 for the notation.)

	Modified scheme of (Nishide et al., 2008)	(Emura et al., 2009)	Our scheme
$\|PK\|$	$(n_s+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(n_s+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(n_s+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$
$\|SK\|$	$(n+1)\|\mathbb{G}\|$	$2\|\mathbb{G}\|$	$(\theta n+2)\|\mathbb{G}\|$
$\|CT\|$	$(m_s+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$2\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(\hat{m}_s+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$
Enc	$(m_s+1)M_{\mathbb{G}}+M_{\mathbb{G}_T}$	$(n+1)M_{\mathbb{G}}+M_{\mathbb{G}_T}$	$(\theta n+\check{m}_s+1)M_{\mathbb{G}}+M_{\mathbb{G}_T}$
Dec	$(n+1)P$	$2P$	$(\theta n+2)P$
Wildcard	yes (for all attributes)	no	yes (for partial attributes)
Assumption	DBDH		

fore, D, $D_{k,0}$, and $D_{k,1}$ respectively satisfy equations (2), (3), and (4). Hence, $SK_{[\hat{L},\check{L}]}$ generated by $\mathcal{B}$ satisfies equation (1), and Lemma 1 holds. □

5 PERFORMANCE

5.1 Cost Comparison

Table 1 compares our schemes with the modified scheme of (Nishide et al., 2008) and the scheme presented in (Emura et al., 2009). In this table, $|PK|$ denotes the size of the public key, $|SK|$ the size of the secret key, $|CT|$ the size of the ciphertext, Enc the encryption cost, and Dec the decryption cost. $|\mathbb{G}|$ and $|\mathbb{G}_T|$ denote the size of the elements in $\mathbb{G}$ and $\mathbb{G}_T$, respectively. n is the number of attributes $\mathbb{A}$, $\hat{n}$ the number of attributes $\hat{\mathbb{A}}$ that require wildcards, and $\check{n}$ the number of attributes $\check{\mathbb{A}}$ that do not require wildcards. θ denotes the proportion of attributes which require wildcards, and $0 \leq \theta \leq 1$. $n_s = \sum_{i=1}^{n} n_i$, where n_i denotes the number of possible values for attribute $\mathbb{A}_i$. $\hat{n}_s = \sum_{i=1}^{\hat{n}} \hat{n}_i$, where $\hat{n}_i$ denotes the number of possible values for attribute $\hat{\mathbb{A}}_i$ which require wildcards, and $\check{n}_s = \sum_{i=1}^{\check{n}} \check{n}_i$, where $\check{n}_i$ is the number of possible values for attribute $\check{\mathbb{A}}_i$ which do not require wildcards. $m_s = \sum_{i=1}^{n} m_i$, where m_i means the number of attribute values in a policy W_i such that $m_i \leq n_i$. $\hat{m}_s = \sum_{i=1}^{\hat{n}} \hat{m}_i$, where $\hat{m}_i$ denotes the number of attribute values in a policy $\hat{W}_i$ that require wildcards and $\hat{m}_i \leq \hat{n}_i$. $\check{m}_s = \sum_{i=1}^{\check{n}} \check{m}_i$, where $\check{m}_i$ denotes the number of attribute values in a policy $\check{W}_i$ that do not require wildcards and $\check{m}_i \leq \check{n}_i$. $M_{\mathbb{G}}$ and $M_{\mathbb{G}_T}$ denote modulo exponentiation in $\mathbb{G}$ and $\mathbb{G}_T$, respectively. P denotes a pairing computation on an elliptic curve.

Figure 1 compares the processing times for decryption. The number of attributes is the horizontal axis and the processing time for decryption is the vertical axis. We assume that the processing time for one pairing computation is 10 (msec) (Zhang et al., 2008). The graphs for our scheme correspond to the case of $\theta = 0.2$ and that of $\theta = 0.8$. Figure 1 clearly

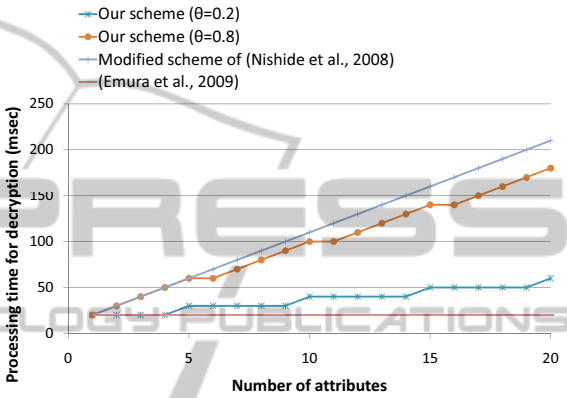

Figure 1: Processing time for decryption in CP-ABE schemes.

shows that the processing time for decryption in our scheme is short when the proportion of attributes requiring wildcards is small.

5.2 Application

A content distribution service is a potential application of our scheme. Let us assume that users have four attributes: residence, membership, contract information, and gender. First, the user's attributes are classified according to the need of wildcards as follows:

(Attribute with wildcard)

$\hat{\mathbb{A}}_1$: residence

$\hat{S}_1 = \{1,2,...,47\} = \{Hokkaido, Aomori, ..., Okinawa\}$

(Attribute without wildcard)

$\check{\mathbb{A}}_1$: membership

$\check{S}_1 = \{1,2\} = \{general, premium\}$

$\check{\mathbb{A}}_2$: contract information

$\check{S}_2 = \{1,2\} = \{payer, non\text{-}payer\}$

$\check{\mathbb{A}}_3$: gender

$\check{S}_3 = \{1,2\} = \{male, female\}$

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Our scheme can realize a regionally restricted content distribution service. In Japan, there are 47 prefectures, so we assign them to possible values for $\hat{\mathbb{A}}_1$. We also allow two kinds of membership, *general* and *premium*, as possible values for $\check{\mathbb{A}}_1$, two kinds of contract information, *payer* and *non-payer*, as possible values for $\check{\mathbb{A}}_2$, and two genders, *male* and *female*, as possible values for $\check{\mathbb{A}}_3$. For example, when a service provider encrypts a piece of content with the policy $\hat{W}_1 = \{$*Tokyo*, *Kanagawa*, *Saitama*, *Chiba*, *Gunma*, *Tochigi*, *Ibaraki*$\}$, which means the Kanto region, a user who has the attribute $\hat{L}_1 =$ *Tokyo* can decrypt the content but a user who has the attribute $\hat{L}_1 =$ *Osaka* cannot decrypt the content. In this case, $n = 4$ and $\theta = 0.25$. Therefore, the decryption cost is $5P$ in the modified scheme of (Nishide et al., 2008) and $3P$ in our scheme, respectively, which means that our scheme can reduce the decryption cost by 40% in comparison with the modified scheme of (Nishide et al., 2008).

For attribute $\check{\mathbb{A}}_1$, a service provider must encrypt content with either $\check{W}_1 =$ *general* or $\check{W}_1 =$ *premium*. If the service provider allows both *general* members and *premium* members to decrypt a content, they must transmit two corresponding ciphertexts to users (For the other attributes $\check{\mathbb{A}}_2$ and $\check{\mathbb{A}}_3$, the service provider must do the same as the above.). If the number of possible values for an attribute is large, the wildcard functionality is effective. On the other hand, if the number of possible values for an attribute is small, the service provider should employ such a trivial scheme rather than use wildcards to reduce total costs. That is, the service provider should transmit as many ciphertexts as possible attribute values to users.

Table 2 is a numerical comparison of the schemes described in Table 1. Several parameters are set according to the above content distribution service: $|\mathbb{G}| = 176$ (bits), $|\mathbb{G}_T| = 1056$ (bits), $M_{\mathbb{G}} = 5$ (msec), $M_{\mathbb{G}_T} = 8$ (msec), $P = 10$ (msec), $n = 4$, $\hat{n} = 1$, $\check{n} = 3$, $\theta = 0.25$, $n_s = 53$, $m_s = 10$, $\hat{n}_s = 47$, $\hat{m}_s = 7$, $\check{n}_s = 6$, and $\check{m}_s = 3$. As shown in Table 2, our scheme is more efficient than the modified scheme of (Nishide et al., 2008).

Table 2: Numerical comparison of our scheme and conventional CP-ABE schemes. M-NYO08 denotes the modified scheme of (Nishide et al., 2008) and EMNOS09 denotes the scheme in (Emura et al., 2009).

	M-NYO08	EMNOS09	Ours
$\|PK\|$ (bits)	10,560	10,736	10,560
$\|SK\|$ (bits)	880	352	528
$\|CT\|$ (bits)	2,992	1,408	2,640
Enc (msec)	63	33	33
Dec (msec)	50	20	30

6 CONCLUSIONS

We proposed a partially wildcarded CP-ABE scheme. We compared our scheme with conventional CP-ABE schemes and described a content distribution service as an application of our scheme. The result shows that our scheme can reduce the decryption cost in comparison with the conventional CP-ABE schemes.

REFERENCES

Bethencourt, J., Sahai, A., and Waters, B. (2007). Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security and Privacy*, pages 321–334.

Cheung, L. and Newport, C. (2007). Provably secure ciphertext policy abe. In *ACMCCS'07*, pages 456–465.

Emura, K., Miyaji, A., Nomura, A., Omote, K., and Soshi, M. (2009). A ciphertext-policy attribute-based encryption scheme with constant ciphertext length. In *ISPEC*, pages 13–23.

Goyal, V., Pandey, O., Sahai, A., and Waters, B. (2006). Attribute-based encryption for fine-grained access control of encrypted data. In *ACMCCS*, pages 89–98.

Katz, J., Sahai, A., and Waters, B. (2008). Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *Eurocrypt*, pages 146–162.

Lewko, A., Okamoto, T., Sahai, A., Takashima, K., and Waters, B. (2010). Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In *Eurocrypt*, pages 62–91.

Nishide, T., Yoneyama, K., and Ohta, K. (2008). Attribute-based encryption with partially hidden encryptor-specified access structures. In *ACNS*, pages 111–129.

Okamoto, T. and Takashima, K. (2010). Fully secure functional encryption with general relations from the decisional linear assumption. In *Crypto*, pages 191–208.

Ostrovsky, R., Sahai, A., and Waters, B. (2007). Attribute-based encryption with non-monotonic access structures. In *ACMCCS*, pages 195–203.

Sahai, A. and Waters, B. (2005). Fuzzy identity-based encryption. In *Eurocrypt*, pages 457–473.

Waters, B. (2011). Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *PKC*, pages 53–70.

Zhang, Y., Kanayama, N., and Okamoto, E. (2008). Java implementation of pairing on elliptic curves over $\mathbb{F}_{2^m}$ (in japanese). In *CSS*, pages D2–1.