9-2015

Attribute based broadcast encryption with short ciphertext and decryption key

Tran Viet Xuan PHUONG

Guomin YANG
Singapore Management University, gmyang@smu.edu.sg

Willy SUSILO

Xiaofeng CHEN

Attribute Based Broadcast Encryption with Short Ciphertext and Decryption Key

Tran Viet Xuan Phuong[1(✉)], Guomin Yang[1], Willy Susilo[1], and Xiaofeng Chen[2]

[1] Centre for Computer and Information Security Research, School of Computing and Information Technology, University of Wollongong, Wollongong, Australia
tvxp750@uowmail.edu.au, {gyang,wsusilo}@uow.edu.au

[2] State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an, People's Republic of China
xfchen@xidian.edu.cn

Abstract. Attribute Based Broadcast Encryption (ABBE) is a combination of Attribute Based Encryption (ABE) and Broadcast Encryption (BE). It allows a broadcaster (or encrypter) to broadcast an encrypted message that can only be decrypted by the receivers who are within a predefined user set *and* satisfy the access policy specified by the broadcaster. Compared with normal ABE, ABBE allows direct revocation, which is important in many real-time broadcasting applications such as Pay TV. In this paper, we propose two novel ABBE schemes that have distinguishing features: the first scheme is key-policy based and has short ciphertext and constant size decryption key; and the second one is ciphertext-policy based and has constant size ciphertext and short decryption key. Both of our schemes allow access policies to be expressed using AND-gate with positive, negative, and wildcard symbols, and are proven secure under the Decision n-BDHE assumption without random oracles.

Keywords: Attribute based encryption · Broadcast encryption · AND-gate · Wildcard

1 Introduction

Broadcast encryption (BE), introduced by Berkovits [1] and Fiat and Naor [2], is a very useful tool for securing a broadcast channel. In a traditional BE scheme, a broadcaster can specify a subset of privileged users (out of the user universe) as the legitimate receivers of a message. Due to the practicality of broadcast encryption in real-world applications, many BE schemes have been proposed in various settings since its introduction (e.g., [3–9]).

Attribute Based Encryption (ABE), first introduced by Sahai and Waters [10], allows an encrypter to embed a fine-grained access policy into the ciphertext when encrypting a message. There are two types of ABE. In a Ciphertext Policy (CP) ABE system, each user secret key is associated with a set of user attributes, and every ciphertext is associated with an access policy. A ciphertext can be

G. Pernul et al. (Eds.): ESORICS 2015, Part II, LNCS 9327, pp. 252–269, 2015.
DOI: 10.1007/978-3-319-24177-7_13

decrypted by a secret key if and only if the attributes associated with the secret key satisfy the access policy in the ciphertext. Key Policy (KP) ABE is the dual form of CP-ABE, where attributes are used in the encryption process, and access policies are used in the user secret key generation. ABE systems can provide fine-grained access control of encrypted data, and has been extensively studied in recent years (e.g., [11–16]).

Since ABE gives a one-to-many relationship between a ciphertext and the corresponding valid decryption keys, it can be considered as a natural broadcast encryption where the legitimate decryptors are defined by the access policies (CP-ABE) or the attributes (KP-ABE) associated with the ciphertext. As pointed out in [11,17], ABE is useful in some broadcasting systems, such as Pay TV, which require dynamic and flexible access control. For example, the broadcasting company can specify an access policy ((Location: City A) AND (Age: >18)) when generating an encrypted data stream for a TV program, and the access policy may be changed to ((Location: City A) AND (Age: *)) (here '*' denotes the wildcard symbol, meaning "don't care") for the next program. However, one drawback of using ABE for broadcasting is that the cost of revoking a user (e.g., those fail to pay the subscription fee for Pay TV) is very high, since the secret keys of all the other non-revoked users must be updated.

Attribute Based Broadcast Encryption (ABBE) is a combination of ABE and BE. Specifically, in a CP-ABBE scheme, a user secret key SK is associated with a user identity (or index) ID and a set of user attributes L, and a ciphertext CT generated by the broadcaster is associated with a user list S and an access policy W. The ciphertext CT can be decrypted using SK if and only if L satisfies W (denoted by $L \models W$) and $ID \in S$. KP-ABBE is the dual form of CP-ABBE where the positions of the attributes and the access policy are swapped. We can see that similar to normal ABE, ABBE also allows fine-grained and flexible access control. On the other hand, ABBE can provide *direct revocation*, which is difficult or expensive to achieve in normal ABE systems. Direct revocation means the broadcaster can directly exclude some revoked users without affecting any non-revoked users, and ABBE can easily achieve this by removing the revoked users from the receiver set S. As highlighted in [17,18], direct revocation is important for real-time broadcasting applications such as Pay TV.

Existing ABBE Constructions. Several ABBE schemes [17–19] have been proposed in the literature. In [19], Lubicz and Sirvent proposed a CP-ABBE scheme which allows access policies to be expressed in disjunctive normal form, with the OR function provided by ciphertext concatenation. Attrapadung and Imai [18] proposed two KP-ABBE and two CP-ABBE schemes, which are constructed by algebraically combining some existing BE schemes (namely, the Boneh-Gentry-Waters BE scheme [5] and the Sahai-Waters BE scheme [20]) with some existing ABE schemes (namely, the KP-ABE scheme by Goyal et al. [11] and the CP-ABE scheme by Waters [14]). Junod and Karlov [17] also proposed a CP-ABBE scheme that supports boolean access policies with AND, OR and NOT gates. Junod and Karlov's scheme achieved direct revocation by simply treating each user's identity as a unique attribute in the attribute universe.

This Work. In order to use ABBE in real-time applications such as Pay TV, the bandwidth requirement and the decryption cost are the most important factors to be considered. Unfortunately, the ciphertext size of the existing ABBE schemes reviewed above is quite high (See Table 1). The motivation of this work is to construct efficient ABBE schemes in terms of ciphertext and key size, as well as decryption cost.

The contribution of this paper are two efficient ABBE schemes allowing access policies to be expressed using AND-gate with positive ($+$), negative ($-$), and wildcard ($*$) symbols. To give a high-level picture of our constructions, we use the *positions* of different symbols (i.e., positive, negative, and wildcard) to do the matching between the access structure (containing wildcards) and the attribute list (containing no wildcard) in the ABE underlying ABBE schemes. We put the indices of all the positive, negative and wildcard attributes defined in an access structure into three sets. By using the Viète's formulas [21], based on the wildcard set, the decryptor can remove all the wildcard positions, and obtain the correct message if and only if the remaining positive and negative attributes have a perfect position match. We then incorporate the technique of Boneh-Gentry-Waters broadcast encryption scheme [5] into our ABE scheme to enable direct revocation.

Our first ABBE scheme is key policy based, and achieves constant key size and short ciphertext size. The second scheme is ciphertext policy based, achieving constant ciphertext size[1] and short key size. Both schemes require only constant number of pairing operations in decryption. A comparison between our ABBE schemes and the previous ones is given in Table 1.

Table 1. Performance comparison among different ABBE schemes

CP-ABBE	Ciphertext	Private Key	Dec. (Pairing)	Access Structure	Assumption
[19]	$O(r)\|\mathbb{G}\| + 1\|\mathbb{G}_T\|$	$O(t)\|\mathbb{G}\|$	$O(1)$	DNF	GDHE
[18]	$O(n)\|\mathbb{G}\| + 1\|\mathbb{G}_T\|$	$O(t)\|\mathbb{G}\|$	$O(t)$	LSSS	n-BDHE, MEBDH
[17]	$O(n)\|\mathbb{G}\| + 1\|\mathbb{G}_T\|$	$O(m+t)\|\mathbb{G}\|$	$O(1)$	DNF, CNF	GDHE
Ours	$O(1)\|\mathbb{G}\| + 1\|\mathbb{G}_T\|$	$O(N)\|\mathbb{G}\|$	$O(1)$	AND Gates + wildcard	n-BDHE
KP-ABBE	Ciphertext Size	Private Key	Dec. (Pairing)	Access Structure	Assumption
[18]	$O(t)\|\mathbb{G}\| + 1\|\mathbb{G}_T\|$	$O(n)\|\mathbb{G}\|$	$O(t)$	LSSS	n-BDHE, MEBDH
Ours	$O(N)\|\mathbb{G}\| + 1\|\mathbb{G}_T\|$	$O(1)\|\mathbb{G}\|$	$O(1)$	AND Gates + wildcard	n-BDHE

In the table, we compare our ABBE schemes with the previous ones in terms of ciphertext and private key size, decryption cost, access structure, and security assumption. We use "p" to denote the pairing operation, "n" the number of

[1] We should note that in our CP-ABBE scheme the wildcard positions should be attached with the ciphertext. A naive way to do this is to include an n-bit string where a bit "1" indicates wildcard at that position. Similar to the previous works on BE [5] and ABBE [18], this information together with the target receiver set S are not counted when measuring the ciphertext size in Table 1.

attributes in an access structure, "t" the number of attributes in an attribute list, "m" and total number of attributes in the system, "r" the number of revoked users in the system, and "N" the maximum number of wildcard in an access structure in our proposed ABBE schemes.

Paper Organisation. In the next section, we review some primitives that will be used in our constructions, and the formal definition and security model of KP- and CP-ABBE. We then present our KP- and CP-ABBE schemes in Sects. 3 and 4, respectively. We give the formal security proofs for our proposed schemes in Sect. 5, and conclude the paper in Sect. 6.

2 Preliminaries

2.1 Bilinear Map on Prime Order Groups

Let $\mathbb{G}$ and $\mathbb{G}_{\mathbb{T}}$ be two multiplicative cyclic groups of same prime order p, and g a generator of $\mathbb{G}$. Let $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be a bilinear map with the following properties:

1. Bilinearity: $e(u^a, v^b) = e(u^b, v^a) = e(u, v)^{ab}$ for all $u,v \in \mathbb{G}$ and $a,b \in \mathbb{Z}_p$.
2. Non-degeneracy: $e(g, g) \neq 1$.

Notice that the map e is symmetric since $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

Decision n-BDHE Assumption. The Decision n-BDHE problem in $\mathbb{G}$ is defined as follows: Let $\mathbb{G}$ be a bilinear group of prime order p, and g, h two independent generators of $\mathbb{G}$. Denote $\overrightarrow{y}_{g,\alpha,n} = (g_1, g_2, \ldots, g_n, g_{n+2}, \ldots, g_{2n}) \in \mathbb{G}^{2n-1}$ where $g_i = g^{\alpha^i}$ for some unknown $\alpha \in \mathbb{Z}_p^*$. We say that the n-BDHE assumption holds in $\mathbb{G}$ if for any probabilistic polynomial-time algorithm A

$$|\Pr[A(g, h, \overrightarrow{y}_{g,\alpha,n}, e(g_{n+1}, h)) = 1] - \Pr[A(g, h, \overrightarrow{y}_{g,\alpha,n}, T) = 1]| \leq \epsilon(k)$$

where the probability is over the random choive of g, h in $\mathbb{G}$, the random choice $\alpha \in \mathbb{Z}_p^*$, the random choice $T \in \mathbb{G}_T$, and $\epsilon(k)$ is negligible in the security parameter k.

2.2 The Viète's formulas

Both of our schemes introduced in this paper are based on the Viète's formulas [21] which is reviewed below. Consider two vectors $\overrightarrow{v} = (v_1, v_2, \ldots, v_L)$ and $\overrightarrow{z} = (z_1, z_2, \ldots, z_L)$. Vector v contains both alphabets and wildcards, and vector z only contains alphabets. Let $J = \{j_1, \ldots, j_n\} \subset \{1, \ldots, L\}$ denote the positions of the wildcards in vector $\overrightarrow{v}$. Then the following two statements are equal:

$$\begin{gathered} v_i = z_i \vee v_i = * \text{ for } i = 1 \ldots L \\ \sum_{i=1, i \notin J}^{L} v_i \prod_{j \in J} (i - j) = \sum_{i=1}^{L} z_i \prod_{j \in J} (i - j). \end{gathered} \tag{1}$$

Expand $\prod_{j\in J}(i-j) = \sum_{k=0}^{n} a_k i^k$, where a_k are the coefficients dependent on J, then (1) becomes:

$$\sum_{i=1,i\notin J}^{L} v_i \prod_{j\in J}(i-j) = \sum_{k=0}^{n} a_k \sum_{i=1}^{L} z_i i^k \tag{2}$$

To hide the computations, we choose random group elemen H_i and put v_i, z_i as the exponents of group elements: $H_i^{v_i}, H_i^{z_i}$. Then (2) becomes:

$$\prod_{i=1,i\notin J}^{L} H_i^{v_i \prod_{j\in J}(i-j)} = \prod_{k=0}^{n} (\prod_{i=1}^{L} H_i^{z_i i^k})^{a_k} \tag{3}$$

Using Viète's formulas we can construct the coefficient a_k in (2) by:

$$a_{n-k} = (-1)^k \sum_{1\le i_1<i_2<\ldots<i_k\le n} j_{i_1} j_{i_2} \ldots j_{i_k},\ 0 \le k \le n. \tag{4}$$

where $n = |J|$. If we have $J = \{j_1, j_2, j_3\}$, the polynomial is $(x-j_1)(x-j_2)(x-j_3)$, then:

$$\begin{aligned} a_3 &= 1 \\ a_2 &= -(j_1 + j_2 + j_3) \\ a_1 &= (j_1 j_2 + j_1 j_3 + j_2 j_3) \\ a_0 &= -j_1 j_2 j_3. \end{aligned}$$

2.3 Access Structure

Let $U = \{Att_1, Att_2, ..., Att_L\}$ be the universe of attributes in the system. Each attribute Att_i has two possible values: positive and negative. Let $W = \{Att_1, Att_2, ..., Att_L\}$ be an AND-gates access policy with wildcards. A wildcard '*' means "don't care" (i.e., both positive and negative attributes are accepted). We use the notation $S \models W$ to denote that the attribute list S of a user satisfies W.

For example, suppose $U = \{Att_1 = \text{CS}, Att_2 = \text{EE}, Att_3 = \text{Faculty}, Att_4 = \text{Student}\}$. Alice is a student in the CS department; Bob is a faculty in the EE department; Carol is a faculty holding a joint position in the EE and CS department. Their attribute lists are illustrated in Table 2. The access structure W_1 can be satisfied by all the CS students, while W_2 can be satisfied by all CS people.

2.4 KP-ABBE Definition

Let U denote the set of all user indices, and N the set of all user attributes. A key-policy attribute based broadcast encryption scheme consists of four algorithms:

- **Setup**(1^λ): The setup algorithm takes the security parameter 1^λ as input and outputs the public parameters PK and a master key MSK.

Table 2. List of attributes and policies

Attributes	Att_1	Att_2	Att_3	Att_4
Description	CS	EE	Faculty	Student
Alice	+	−	−	+
Bob	−	+	+	−
Carol	+	+	+	−
W_1	+	−	−	+
W_2	+	−	*	*

- **Encrypt**(S, L, M, PK): The encryption algorithm takes as input the public parameters PK, a message M, a set of user index $S \subseteq U$ and a set of attributes $L \subseteq N$, and outputs a ciphertext CT.
- **Key Generation**(ID, W, MSK, PK)**:** The key generation algorithm takes as input the master key MSK, public parameters PK, a user index $ID \in U$, and an access structure W, and outputs a private key SK.
- **Decrypt**$(PK,\ CT,\ SK)$: The decryption algorithm takes as input the public parameters PK, a ciphertext CT, and a private key SK, and outputs a message M or a special symbol '$\perp$'.

Security Definition for KP-ABBE. We define the Selective IND-CPA security for KP-ABBE via the following game.

- **Init:** The adversary commits to the challenge user indices S^* and target attribute set L^*.
- **Setup:** The challenger runs the Setup algorithm and gives PK to the adversary.
- **Phase 1:** The adversary queries for private keys with pairs of user index and access structure (ID, W) such that $L^* \not\models W$ or $ID \notin S^*$.
- **Challenge:** The adversary submits messages M_0, M_1 to the challenger. The challenger flips a random coin β and passes the ciphertext $ct^* = Encrypt(PK, M_\beta, L^*, S^*)$ to the adversary.
- **Phase 2:** Phase 1 is repeated.
- **Guess:** The adversary outputs a guess β' of β.

Definition 1. *We say a KP-ABBE scheme is selective IND-CPA secure if for any probabilistic polynomial time adversary*

$$\boldsymbol{Adv}_{kp}^{s\text{-}ind\text{-}cpa}(\lambda) = |\Pr[\beta' = \beta] - 1/2|$$

is a negligible function of λ.

2.5 CP-ABBE Definition

A ciphertext-policy attribute based broadcast encryption scheme consists of four algorithms:

- **Setup**(1^λ)**:** The setup algorithm takes the security parameter 1^λ as input and outputs the public parameters PK and a master key MSK.
- **Encrypt**(S, W, M, PK)**:** The encryption algorithm takes as input the public parameters PK, a message M, an access structure W, a set of user index $S \subseteq U$, and outputs a ciphertext CT.
- **Key Generation**(ID, L, MSK, PK)**:** The key generation algorithm takes as input the master key MSK, public parameters PK, a user index $ID \in U$, and a set of attributes $L \subseteq N$, and outputs a private key SK.
- **Decrypt**(PK, CT, SK)**:** The decryption algorithm takes as input the public parameters PK, a ciphertext CT, and a private key SK, and outputs a message M or a special symbol '$\perp$'.

Security Definition for CP-ABBE. We define the Selective IND-CPA security for CP-ABBE via the following game.

- **Init:** The adversary commits to the challenge user indices S^* and target access structure W^*.
- **Setup:** The challenger runs the Setup algorithm and gives PK to the adversary.
- **Phase 1:** The adversary queries for private keys with pairs of user index and a user attribute list (ID, L) such that $L^* \not\models W$ or $ID \notin S^*$.
- **Challenge:** The adversary submits messages M_0, M_1 to the challenger. The challenger flips a random coin β and passes the ciphertext $ct^* = Encrypt(PK, M_\beta, W^*, S^*)$ to the adversary.
- **Phase 2:** Phase 1 is repeated.
- **Guess:** The adversary outputs a guess β' of β.

Definition 2. *We say a CP-ABBE scheme is selective IND-CPA secure if for any probabilistic polynomial time adversary*

$$\boldsymbol{Adv}_{cp}^{s\text{-}ind\text{-}cpa}(\lambda) = |\Pr[\beta' = \beta] - 1/2|$$

is a negligible function of λ.

3 KP-ABBE Scheme

In our KP-ABBE scheme, we assume that $|U| \leq n$ and $|N| \leq n$ where n is a system parameter. Let N_1, N_2, N_3 be three upper bounds for the user attributes:

- N_1: the maximum number of wildcard in an access structure.
- N_2: the maximum number of positive attribute in an attribute list L.
- N_3: the maximum number of negative attribute in an attribute list L.

▶ **Setup**(1^λ): The setup algorithm first generates bilinear groups $\mathbb{G}, \mathbb{G}_T$ with order p, and selects random generators $g, h_1, \ldots, h_N \in_R \mathbb{G}$, and $\alpha \in_R \mathbb{Z}_p$. Then compute $g_i = g^{\alpha_i} \in \mathbb{G}$ for $i = 1, 2, \ldots, n, n+2, \ldots, 2n$, randomly choose $\gamma, \delta, \theta, x_1, \ldots, x_{N_1} \in_R \mathbb{Z}_p$, and set:

$$\nu = g^{\gamma}, V_0 = g^{\delta}, V_1 = g^{\theta},$$
$$V_{01} = (g^{\delta})^{x_1}, \ldots, V_{0N_1} = (g^{\delta})^{x_{N_1}},$$
$$V_{11} = (g^{\theta})^{x_1}, \ldots, V_{1N_1} = (g^{\theta})^{x_{N_1}},$$

The public key and master secret key are defined as:

$$\begin{aligned} PK &= (g, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}, h_1, \ldots, h_N, \nu, V_0, V_1, V_{01}, \ldots, V_{0N_1}, \\ &\quad V_{11}, \ldots, V_{1N_1}) \\ MSK &= (\alpha, \gamma, \delta, \theta, x_1, \ldots, x_{N_1}). \end{aligned}$$

▶ **Encrypt**(S, L, M, PK): Given a user index set $S \subseteq U$, an attribute list L which contains:
– $n_2 \leq N_2$ positive attributes at positions $V = \{v_1, \ldots, v_{n_2}\}$;
– $n_3 \leq N_3$ negative attributes at positions $Z = \{z_1, \ldots, z_{n_3}\}$;
the algorithm randomly chooses $r \in \mathbb{Z}_p$ and computes:

$$C_0 = M \cdot e(g_n, g_1)^r, C_1 = g^r, C_2 = (\nu \prod_{j \in S} g_{n+1-j})^r,$$

$$\begin{pmatrix} C_{3,0} = (V_0 \prod_{i \in V} h_i)^r \\ C_{3,1} = (V_{01} \prod_{i \in V} h_i^i)^r \\ \ldots \\ C_{3,N_1} = (V_{0N_1} \prod_{i \in V} h_i^{i^{N_1}})^r \end{pmatrix}, \begin{pmatrix} C_{4,0} = (V_1 \prod_{i \in Z} h_i)^r \\ C_{4,1} = (V_{11} \prod_{i \in Z} h_i^i)^r \\ \ldots \\ C_{4,N_1} = (V_{1N_1} \prod_{i \in Z} h_i^{N_1})^r \end{pmatrix}.$$

The ciphertext is $CT = (C_0, C_1, C_2, C_{3,0}, \ldots, C_{3,N_1}, C_{4,0}, \ldots, C_{4,N_1})$.

▶ **Key Generation**(ID, W, MSK, PK): Suppose that the access structure W contains:
– $n_1 \leq N_1$ wildcards at positions $J = \{w_1, \ldots, w_{n_1}\}$.
– $n_2 \leq N_2$ positive attributes at positions $V' = \{v'_1, \ldots, v'_{n_2}\}$.
– $n_3 \leq N_3$ negative attributes at positions $Z' = \{z'_1, \ldots, z'_{n_3}\}$.
Randomly choose $s_1, s_2 \in \mathbb{Z}_p$, and apply the Viete formulas on J to compute $a_k (0 \leq k \leq n_1)$ and set $t = \sum_{k=0}^{n_1} x_k a_k$ where $x_0 = 1$. Then compute

$$D_1 = g^{\alpha^{ID}\gamma + \delta s_1 + \theta s_2}, D_2 = g^{\frac{s_1}{t}}, D_3 = g^{\frac{s_2}{t}},$$

$$D_4 = (\prod_{i \in V'} h_i^{\prod_{j=0}^{n_1}(i-w_j)})^{\frac{s_1}{t}}, D_5 = (\prod_{i \in Z'} h_i^{\prod_{j=0}^{n_1}(i-w_j)})^{\frac{s_2}{t}}.$$

and set the secret key $SK = (D_1, D_2, D_3, D_4, D_5)$.

► **Decrypt**(PK, CT, SK): The decryption algorithm first applies the Viete formulas on J included in the secret key to compute a_k for $0 \le k \le n_1$, and

$$\begin{aligned} e(D_1, C_1) &= e(g^{\alpha^{ID}\gamma+\delta s_1+\theta s_2}, g^r) \\ &= e(g^{\alpha^{ID}\gamma}, g^r)e(g,g)^{\delta s_1 r}e(g,g)^{\theta s_2 r} \end{aligned}$$

$$e(D_4, C_1) = e((\prod_{i\in V'} h_i^{\prod_{j=0}^{n_1}(i-w_j)})^{s_1/t}, g^r)$$

$$e(D_5, C_1) = e((\prod_{i\in Z'} h_i^{\prod_{j=0}^{n_1}(i-w_j)})^{s_2/t}, g^r)$$

$$\begin{aligned} e(g_{ID}, C_2) &= e(g^{\alpha^{ID}}, (\nu \prod_{j\in S} g_{n+1-j})^r) \\ &= e(g^{\alpha^{ID}}, \nu)^r e(g^{\alpha^{ID}}, \prod_{j\in S} g_{n+1-j})^r \end{aligned}$$

$$e(\prod_{j\in S, j\neq ID} g_{n+1-j+ID}, C_1) = e(\prod_{j\in S, j\neq ID} g_{n+1-j+ID}, g^r)$$

$$\Rightarrow e(g_{ID}, C_2)/e(\prod_{j\in S, j\neq ID} g_{n+1-j+ID}, C_1) = e(g^{\alpha^{ID}}, \nu)^r \cdot e(g_n, g_1)^r$$

$$\begin{aligned} e(D_2, \prod_{k=0}^{n_1} C_{3,k}^{a_k}) &= e(g^{s_1/t}, V_0^{r\sum_{k=0}^{n_1} x_k a_k} \prod_{i\in V} h_i^{\sum_{k=0}^{n_1} i^k a_k r}) \\ &= e(g, V_0)^{s_1 r} e(\prod_{i\in V} h_i^{\prod_{j=0}^{n_1}(i-w_j)r}, g^{s_1/t}) \end{aligned}$$

$$\begin{aligned} e(D_3, \prod_{k=0}^{n_1} C_{4,k}^{a_k}) &= e(g^{s_2/t}, V_1^{r\sum_{k=0}^{n_1} x_k a_k} \prod_{i\in Z} h_i^{\sum_{k=0}^{n_1} i^k a_k r}) \\ &= e(g, V_1)^{s_2 r} e(\prod_{i\in Z} h_i^{\prod_{j=0}^{n_1}(i-w_j)r}, g^{s_2/t}) \end{aligned}$$

If $L \models W$ and $ID \in S$, then we have:

$$M = \frac{C_0 \cdot e(g^{\alpha^{ID}\gamma}, g^r)e(g,g)^{\delta s_1 r}e(g,g)^{\theta s_2 r}e((\prod_{i\in V'} h_i^{\prod_{j=0}^{n_1}(i-w_j)})^{s_1/t}, g^r)e((\prod_{i\in Z'} h_i^{\prod_{j=0}^{n_1}(i-w_j)})^{s_2/t}, g^r)}{e(g^{\alpha^{ID}}, \nu)^r \cdot e(g_n, g_1)^r e(g, V_0)^{s_1 r} e(\prod_{i\in V} h_i^{\prod_{j=0}^{n_1}(i-w_j)r}, g^{s_1/t})e(g, V_1)^{s_2 r} e(\prod_{i\in Z} h_i^{\prod_{j=0}^{n_1}(i-w_j)r}, g^{s_2/t})}.$$

4 CP-ABBE Scheme

Our CP-ABBE scheme is the dual-form of our KP-ABBE scheme.

► **Setup**(1^λ): The setup algorithm first generates bilinear groups $\mathbb{G}, \mathbb{G}_T$ with order p, and selects random generators $g, h_1, \ldots, h_N \in_R \mathbb{G}$, and $\alpha \in_R \mathbb{Z}_p$.

Then compute $g_i = g^{\alpha_i} \in \mathbb{G}$ for $i = 1, 2, \ldots, n, n+2, \ldots, 2n$, randomly choose $\gamma, \delta, \theta \in_R \mathbb{Z}_p$, and set:

$$\nu = g^{\gamma}, V_0 = g^{\delta}, V_1 = g^{\theta}.$$

The public key and master secret key are defined as:

$$\begin{aligned} PK &= (g, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}, h_1, \ldots, h_N, \nu, V_0, V_1) \\ MSK &= (\alpha, \gamma, \delta, \theta). \end{aligned}$$

► **Encrypt**(S, W, M, PK): Given a user index set $S \subseteq U$, and an access structure W containing:
– $n_1 \leq N_1$ wildcards at positions $J = \{w_1, \ldots, w_{n_1}\}$;
– $n_2 \leq N_2$ positive attributes at positions $V = \{v_1, \ldots, v_{n_2}\}$;
– $n_3 \leq N_3$ negative attributes at positions $Z = \{z_1, \ldots, z_{n_3}\}$;
the algorithm randomly chooses $r \in \mathbb{Z}_p$ and computes:

$$C_0 = M \cdot e(g_n, g_1)^r, C_1 = g^r, C_2 = (\nu \prod_{j \in S} g_{n+1-j})^r,$$

$$C_3 = (V_0 \prod_{i \in V} h_i^{\prod_{j=0}^{n_1}(i-w_j)})^r, C_4 = (V_1 \prod_{i \in Z} h_i^{\prod_{j=0}^{n_1}(i-w_j)})^r.$$

The ciphertext is $CT = (J, C_0, C_1, C_2, C_3, C_4)$.

► **Key Generation**(ID, L, MSK, PK): Given a user identity ID and an attribute list L which contains:
– $n_2 \leq N_2$ positive attributes at positions $V' = \{v'_1, \ldots, v'_{n_2}\}$;
– $n_3 \leq N_3$ negative attributes at positions $Z' = \{z'_1, \ldots, z'_{n_3}\}$;
randomly choose $s_1, s_2 \in \mathbb{Z}_p$ and compute:

$$D_1 = g^{\alpha^{ID}\gamma + \delta s_1 + \theta s_2}, D_2 = g^{s_1}, D_3 = g^{s_2}$$

$$\begin{pmatrix} D_{4,0} = (\prod_{i \in V'} h_i)^{s_1} \\ D_{4,1} = (\prod_{i \in V'} h_i^i)^{s_1} \\ \cdots \\ D_{4,N_1} = (\prod_{i \in V'} h_i^{i^{N_1}})^{s_1} \end{pmatrix}, \begin{pmatrix} D_{5,0} = (\prod_{i \in Z'} h_i)^{s_2} \\ D_{5,1} = (\prod_{i \in Z'} h_i^i)^{s_2} \\ \cdots \\ D_{5,N_1} = (\prod_{i \in Z'} h_i^{i^{N_1}})^{s_2} \end{pmatrix},$$

and set the secret key $SK = (D_1, D_2, D_3, D_{4,0}, \ldots, D_{4,N_1}, D_{5,0}, \ldots, D_{5,N_1})$.

► **Decrypt**(PK, CT, SK): The decryption algorithm first applies the Viete formulas on J included in the ciphertext to compute a_k for $0 \leq k \leq n_1$:

$$
\begin{aligned}
e(D_1, C_1) &= e(g^{\alpha^{ID}\gamma+\delta s_1+\theta s_2}, g^r) \\
&= e(g^{\alpha^{ID}\gamma}, g^r) e(g,g)^{\delta s_1 r} e(g,g)^{\theta s_2 r} \\
e((\prod_{k=0}^{n_1} D_{4,k}^{a_k}), C_1) &= e(\prod_{i\in V'} h_i^{\sum_{k=0}^{n_1} i^k a_k s_1}, g^r) \\
&= e(\prod_{i\in V'} h_i^{\prod_{j=0}^{n_1}(i-w_j)s_1}, g^r)
\end{aligned}
$$

$$
\begin{aligned}
e((\prod_{k=0}^{n_1} D_{5,k}^{a_k}), C_1) &= e(\prod_{i\in Z'} h_i^{\sum_{k=0}^{n_1} i^k a_k s_2}, g^r) \\
&= e(\prod_{i\in Z'} h_i^{\prod_{j=0}^{n_1}(i-w_j)s_2}, g^r)
\end{aligned}
$$

$$
\begin{aligned}
e(g_{ID}, C_2) &= e(g^{\alpha^{ID}}, (\nu \prod_{j\in S} g_{n+1-j})^r) \\
&= e(g^{\alpha^{ID}}, \nu)^r e(g^{\alpha^{ID}}, \prod_{j\in S} g_{n+1-j})^r
\end{aligned}
$$

$$
e(\prod_{j\in S,\, j\neq ID} g_{n+1-j+ID}, C_1) = e(\prod_{j\in S,\, j\neq ID} g_{n+1-j+ID}, g^r)
$$

$$
\Rightarrow e(g_{ID}, C_2)/e(\prod_{j\in S,\, j\neq ID} g_{n+1-j+ID}, C_1) = e(g^{\alpha^{ID}}, \nu)^r \cdot e(g_n, g_1)^r
$$

$$
\begin{aligned}
e(D_2, C_3) &= e(g^{s_1}, (V_0 \prod_{i\in V} h_i^{\prod_{j=0}^{n_1}(i-w_j)})^r) \\
&= e(g^{s_1}, V_0^r) e(g^{s_1}, \prod_{i\in V} h_i^{\prod_{j=0}^{n_1}(i-w_j)})^r
\end{aligned}
$$

$$
\begin{aligned}
e(D_3, C_4) &= e(g^{s_2}, (V_1 \prod_{i\in Z} h_i^{\prod_{j=0}^{n_1}(i-w_j)})^r) \\
&= e(g^{s_2}, V_1^r) e(g^{s_2}, \prod_{i\in Z} h_i^{\prod_{j=0}^{n_1}(i-w_j)})^r
\end{aligned}
$$

If $L \models W$ and $ID \in S$, then we have

$$
M = \frac{C_0 \cdot e(g^{\alpha^{ID}\gamma}, g^r) e(g,g)^{\delta s_1 r} e(g,g)^{\theta s_2 r} \cdot e(\prod_{i\in V'} h_i^{\prod_{j=0}^{n_1}(i-w_j)s_1}, g^r) e(\prod_{i\in Z'} h_i^{\prod_{j=0}^{n_1}(i-w_j)s_2}, g^r)}{e(g^{\alpha^{ID}}, \nu)^r \cdot e(g_n, g_1)^r e(g^{s_1}, V_0^r) e(g^{s_1}, \prod_{i\in V} h_i^{\prod_{j=0}^{n_1}(i-w_j)})^r e(g^{s_2}, V_1^r) e(g^{s_2}, \prod_{i\in Z} h_i^{\prod_{j=0}^{n_1}(i-w_j)})^r}.
$$

5 Security Analysis

We prove that the proposed KP-ABBE and CP-ABBE schemes are selectively secure under the Decision n-BDHE assumption.

Theorem 1. *Assume that the Decision n-BDHE assumption holds, then no polynomial-time adversary against our KP-ABBE scheme can have a non-negligible advantage over random guess in the Selective IND-CPA security game.*

Proof: Suppose that there exists an adversary $\mathcal{A}$ which can attack our scheme with non-negligible advantage ϵ, we construct another algorithm $\mathcal{B}$ which uses $\mathcal{A}$ to solve the Decision n-BDHE problem. On input $(g, h, \overrightarrow{y}_{g,\alpha,n} = (g_1, g_2, \ldots, g_n, g_{n+2}, \ldots, g_{2n}), T)$, where $g_i = g^{\alpha^i}$ and for some unknown $\alpha \in \mathbb{Z}_p^*$, the goal of $\mathcal{B}$ is to determine whether $T = e(g_{n+1}, h)$ or a random element of $\mathbb{G}_T$.

Init: $\mathcal{A}$ gives $\mathcal{B}$ the challenge user indices S^* and the target attribute set L^* with $n_2 \leq N_2$ positive attributes which occur at positions $V^* = \{v_1^*, \ldots, v_{n_2}^*\}$, and $n_3 \leq N_3$ negative attributes which occur at positions $Z^* = \{z_1^*, \ldots, z_{n_3}^*\}$ at the beginning of the game.

Setup: $\mathcal{B}$ chooses $d, v_0, v_1, u_1, \ldots, u_n, x_1, \ldots, x_{N_1} \in \mathbb{Z}_p$ and generates:

$$\begin{aligned}
\nu &= g^d(\prod_{j \in S^*} g_{n+1-j}^{-1}) = g^{d-\sum_{j\in S^*} \alpha^{n+1-j}} = g^{\gamma}, \\
V_{0j} &= (g^{v_0})^{x_j} \prod_{i \in V^*} g^{\alpha^{n+1-i} i^j} = (g^{v_0})^{x_j} g^{\sum_{i\in V^*} \alpha^{n+1-i} i^j}, \text{ for } j = 0, \ldots, N_1 \\
V_{1j} &= (g^{v_1})^{x_j} \prod_{i \in Z^*} g^{\alpha^{n+1-i} i^j} = (g^{v_1})^{x_j} g^{\sum_{i\in Z^*} \alpha^{n+1-i} i^j}, \text{ for } j = 0, \ldots, N_1
\end{aligned}$$

where $x_0 = 1$, and $h_i = g^{u_i - \alpha^{n+1-i}}$, then $\mathcal{B}$ sets public key as:

$$PK = (g, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}, h_1, \ldots, h_N, \nu, V_0, V_1, V_{01}, \ldots, V_{0N_1}, V_{11}, \ldots, V_{1N_1}).$$

Phase 1: $\mathcal{A}$ submits a pair of user index and access structure (ID, W) in a secret key query, which satisfies $L^* \not\models W$ or $ID \notin S^*$. Assume W consists of $n_1 \leq N_1$ wildcards which occur at positions $J = \{w_1, \ldots, w_{n_1}\}$, $n_2 \leq N_2$ positive attributes which occur at positions $V = \{v_1, \ldots, v_{n_2}\}$, and $n_3 \leq N_3$ negative attributes which occur at positions $Z = \{z_1, \ldots, z_{n_3}\}$. $\mathcal{B}$ applies the Viete formulas on $J = \{j_1, \ldots, j_{n_1}\}$ to get a_k and set $t = \sum_{k=0}^{n_1} x_k a_k$. Consider the following two cases in Phase 1:

– **Case 1:** $ID \notin S^*$. $\mathcal{B}$ first selects a random number $s_1, s_2 \in \mathbb{Z}_p$, then computes:

$$\begin{aligned}
D_1 &= g_{ID}^d \prod_{j\in S^*} (g_{n+1-j+ID})^{-1} g^{v_0 s_1} \prod_{i\in V^*} (g_{n+1-i})^{s_1} g^{v_1 s_2} \prod_{i\in Z^*} (g_{n+1-i})^{s_2} \\
&= g^{\alpha^{ID}(d-\sum_{j\in S^*} \alpha^{n+1-j})} (g^{v_0 + \sum_{i\in V^*} \alpha^{n+1-i}})^{s_1} (g^{v_1 + \sum_{i\in Z^*} \alpha^{n+1-i}})^{s_2} \\
&= g^{\alpha^{ID}\gamma + \delta s_1 + \theta s_2}.
\end{aligned}$$

$$D_2 = g^{\frac{s_1}{t}},$$
$$D_3 = g^{\frac{s_2}{t}},$$
$$D_4 = (\prod_{i\in V}(g^{u_i-\alpha^{n+1-i}})^{\prod_{j\in J}(i-w_j)})^{\frac{s_1}{t}} = (\prod_{i\in V} h_i^{\prod_{j\in J}(i-w_j)})^{\frac{s_1}{t}},$$
$$D_5 = (\prod_{i\in Z}(g^{u_i-\alpha^{n+1-i}})^{\prod_{j\in J}(i-w_j)})^{\frac{s_2}{t}} = (\prod_{i\in Z} h_i^{\prod_{j\in J}(i-w_j)})^{\frac{s_2}{t}}.$$

– **Case 2:** $ID \in S^*$. In this case, due to the constraint $L^* \not\models W$, W has at least one position i^* which has a different attribute value from L^*, which means $\{V \cup Z^*\} \neq \emptyset$ or $\{Z \cup V^*\} \neq \emptyset$.

◇ If there exists an $i^* \in \{V \cup Z^*\} \neq \emptyset$:
$\mathcal{B}$ selects two random numbers $s_1', s_2' \in \mathbb{Z}_p$ and implicitly sets s_1, s_2 as: $\begin{cases} s_1 = s_1' \\ s_2 = s_2' + \alpha^{i^*} \end{cases}$ by setting $D_2 = g^{s_1'} = g^{s_1}, D_3 = g^{s_2'+\alpha^{i^*}} = g^{s_2}$. Then $\mathcal{B}$ can compute D_1, D_4, D_5 as follows:

$$\begin{aligned}
D_1 &= g^{\alpha^{ID}\gamma+\delta s_1+\theta s_2}. \\
&= g^{\alpha^{ID}(d-\sum_{j\in S^*}\alpha^{n+1-j})} g^{v_0 s_1} \prod_{i\in V^*}(g_{n+1-i})^{s_1} g^{v_1 s_2} \prod_{i\in Z^*}(g_{n+1-i})^{s_2} \\
&= g_{ID}^d \prod_{j\in S^*}(g_{n+1-j+ID})^{-1} \\
&\quad (g^{v_0})^{s_1'}(g^{\sum_{i\in V^*}\alpha^{n+1-i}})^{s_1'}(g^{v_1})^{s_2'+\alpha^{i^*}}(g^{\sum_{i\in Z^*}\alpha^{n+1-i}})^{s_2'+\alpha^{i^*}} \\
&= g_{ID}^d \prod_{j\in S^*, j\neq ID}(g_{n+1-j+ID})^{-1} \cdot g^{-\alpha^{n+1}} \\
&\quad (g^{v_0})^{s_1'}(g^{\sum_{i\in V^*}\alpha^{n+1-i}})^{s_1'} \\
&\quad (g^{v_1})^{s_2'+\alpha^{i^*}}(g^{\sum_{i\in Z^*}\alpha^{n+1-i}})^{s_2'}(g^{\sum_{i\in Z^*, i\neq i^*}\alpha^{n+1-i+i^*}})g^{\alpha^{n+1}} \\
&= g_{ID}^d \prod_{j\in S^*, j\neq ID}(g_{n+1-j+ID})^{-1}(g^{v_0})^{s_1'}(g^{\sum_{i\in V^*}\alpha^{n+1-i}})^{s_1'} \\
&\quad (g^{v_1})^{s_2'+\alpha^{i^*}}(g^{\sum_{i\in Z^*}\alpha^{n+1-i}})^{s_2'}(g^{\sum_{i\in Z^*, i\neq i^*}\alpha^{n+1-i+i^*}}), \\
D_4 &= (\prod_{i\in V}(g^{u_i-\alpha^{n+1-i}})^{\prod_{j\in J}(i-w_j)})^{s_1'/t} = (\prod_{i\in V} h_i^{\prod_{j\in J}(i-w_j)})^{s_1/t}, \\
D_5 &= (\prod_{i\in Z}(g^{u_i-\alpha^{n+1-i}})^{\prod_{j\in J}(i-w_j)})^{(s_2'+\alpha^{i^*})/t} = (\prod_{i\in Z} h_i^{\prod_{j\in J}(i-w_j)})^{s_2/t}.
\end{aligned}$$

We should note that since $i^* \notin Z$, the item $g^{\alpha^{n+1}}$ will not occur in the calculation of D_5.

◇ If there exists an $i^* \in \{Z \cup V^*\} \neq \emptyset$:
the simulation can be performed in a similar way by choosing two random numbers $s_1', s_2' \in \mathbb{Z}_p$ and implicitly setting s_1, s_2 as: $\begin{cases} s_1 = s_1' + \alpha^{i^*} \\ s_2 = s_2' \end{cases}$. We omit the details here.

$\mathcal{B}$ returns to $\mathcal{A}$ the secret key $SK = (D_1, D_2, D_3, D_4, D_5)$.

Challenge: The adversary gives two messages M_0 and M_1 to $\mathcal{B}$. Then $\mathcal{B}$ flips a coin b and generate the challenge ciphertext by setting $C_1 = g^\tau = h$ for some unknown τ and

$$\begin{aligned} C_2 &= h^d = (g^d)^\tau \\ &= (g^d \prod_{j\in S^*}(g_{n+1-j})^{-1} \prod_{j\in S^*}(g_{n+1-j}))^\tau = (\nu \prod_{j\in S^*}(g_{n+1-j}))^\tau \\ C_{3,k} &= h^{v_0 x_k + \sum_{i\in V^*} u_i i^k} = (g^{v_0 x_k + \sum_{i\in V^*} u_i i^k})^\tau, \\ C_{4,k} &= h^{v_1 x_k + \sum_{i\in Z^*} u_i i^k} = (g^{v_1 x_k + \sum_{i\in Z^*} u_i i^k})^\tau. \end{aligned}$$

$\mathcal{B}$ then sends the following challenge ciphertext to $\mathcal{A}$

$$CT^* = (M_b T, C_1, C_2, \{C_{3,k}\}, \{C_{4,k}\}).$$

Phase II: Same as Phase I.

Guess: $\mathcal{A}$ output $b' \in \{0,1\}$. If $b' = b$ then $\mathcal{B}$ outputs 1, otherwise outputs 0.

Analysis: If $T = e(g_{n+1}, h)$, then the simulation is the same as in the real game. Hence, $\mathcal{A}$ will have the probability $\frac{1}{2} + \epsilon$ to guess b correctly. If T is a random element of $\mathbb{G}_T$, then $\mathcal{A}$ will have probability $\frac{1}{2}$ to guess b correctly. Therefore, $\mathcal{B}$ can solve the Decision n-BDHE assumption also with advantage ϵ. □

Theorem 2. *Assume that the Decision n-BDHE assumption holds, then no polynomial-time adversary against our CP-ABBE scheme can have a non-negligible advantage over random guess in the Selective IND-CPA security game.*

Proof: Suppose that there exists an adversary $\mathcal{A}$ which can attack our scheme with non-negligible advantage ϵ, we construct another algorithm $\mathcal{B}$ which uses $\mathcal{A}$ to solve the Decision n-BDHE problem. On input $(g, h, \overrightarrow{y}_{g,\alpha,n} = (g_1, g_2, \ldots, g_n, g_{n+2}, \ldots, g_{2n}), T)$, where $g_i = g^{\alpha^i}$ and for some unknown $\alpha \in \mathbb{Z}_p^*$, the goal of $\mathcal{B}$ is to determine whether $T = e(g_{n+1}, h)$ or a random element of $\mathbb{G}_T$.

Init: $\mathcal{A}$ gives $\mathcal{B}$ the challenge user indexes S^* and the challenge access structure W^* with $n_1 \le N_1$ wildcards which occur at positions $J^* = \{w_1^*, \ldots, w_{n_1}^*\}$, $n_2 \le N_2$ positive attributes which occur at positions $V^* = \{v_1^*, \ldots, v_{n_2}^*\}$, $n_3 \le N_3$ negative attributes which occur at positions $Z^* = \{z_1^*, \ldots, z_{n_3}^*\}$ at the beginning of the game.

Setup: $\mathcal{B}$ chooses $d, v_0, v_1, u_1, \ldots, u_n \in \mathbb{Z}_p$ and generates:

$$\begin{aligned} \nu &= g^d(\prod_{j\in S^*} g_{n+1-j}^{-1}) = g^{d-\sum_{j\in S^*}\alpha^{n+1-j}} = g^\gamma, \\ V_0 &= g^{v_0} \prod_{i\in V^*} g^{\alpha^{n+1-i}\prod_{j\in J^*}(i-w_j^*)} = g^{v_0+\sum_{i\in V^*}\alpha^{n+1-i}\prod_{j\in J^*}(i-w_j^*)} = g^\delta, \\ V_1 &= g^{v_1} \prod_{i\in Z^*} g^{\alpha^{n+1-i}\prod_{j\in J^*}(i-w_j^*)} = g^{v_1+\sum_{i\in Z^*}\alpha^{n+1-i}\prod_{j\in J^*}(i-w_j^*)} = g^\theta, \end{aligned}$$

and $h_i = g^{u_i - \alpha^{n+1-i}}$, then $\mathcal{B}$ sets public key as:

$$PK = (g, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}, h_1, \ldots, h_N, \nu, V_0, V_1).$$

Phase 1: $\mathcal{A}$ submits (ID, L) in a secret key query, where $L^* \not\models W$ "or" $ID \notin S^*$. Suppose the attribute set L contains $n_2 \leq N_2$ positive attributes which occur at positions $V = \{v_1, \ldots, v_{n_2}\}$, and $n_3 \leq N_3$ negative attributes which occur at positions $Z = \{z_1, \ldots, z_{n_3}\}$. We consider two cases in Phase 1:

– **Case 1:** $ID \notin S^*$. $\mathcal{B}$ first selects random numbers $s_1, s_2 \in \mathbb{Z}_p$ and computes:

$$\begin{aligned}
D_1 &= g_{ID}^d \prod_{j \in S^*} (g_{n+1-j+ID})^{-1} g^{v_0 s_1} \prod_{i \in V^*} (g_{n+1-i}^{\prod_{j \in J^*} (i - w_j^*)})^{s_1} g^{v_1 s_2} \prod_{i \in Z^*} (g_{n+1-i}^{\prod_{j \in J^*} (i - w_j^*)})^{s_2} \\
&= g^{\alpha^{ID}(d - \sum_{j \in S^*} \alpha^{n+1-j})} \\
&\quad (g^{v_0 + \sum_{i \in V^*} \alpha^{n+1-i} \prod_{j \in J^*} (i - w_j^*)})^{s_1} (g^{v_1 + \sum_{i \in Z^*} \alpha^{n+1-i} \prod_{j \in J^*} (i - w_j^*)})^{s_2} \\
&= g^{\alpha^{ID} \gamma + \delta s_1 + \theta s_2}, \\
D_2 &= g^{s_1}, \\
D_3 &= g^{s_2}, \\
D_{4,k} &= \prod_{i \in V} (g^{u_i - \alpha^{n+1-i}})^{i^k s_1} = \prod_{i \in V} h_i^{i^k s_1}, \\
D_{5,k} &= \prod_{i \in Z} (g^{u_i - \alpha^{n+1-i}})^{i^k s_2} = \prod_{i \in Z} h_i^{i^k s_2}.
\end{aligned}$$

– **Case 2:** $ID \in S^*$. In this case, due to the constraint $L^* \not\models W$, L has at least one position i^* which has a different attribute value from W^*, which means $\{V \cup Z^*\} \neq \emptyset$ or $\{Z \cup V^*\} \neq \emptyset$.

◇ If there exists $i^* \in \{V \cup Z^*\} \neq \emptyset$:

$\mathcal{B}$ selects two random numbers $s_1', s_2' \in \mathbb{Z}_p$ and implicitly sets s_1, s_2 as: $\begin{cases} s_1 = s_1' \\ s_2 = s_2' + \dfrac{\alpha^{i^*}}{\prod_{j \in J^*} (i^* - w_j^*)} \end{cases}$ by setting $D_2 = g^{s_1'} = g^{s_1}, D_3 = g^{s_2' + \frac{\alpha^{i^*}}{\prod_{j \in J^*} (i^* - w_j^*)}} = g^{s_2}$. Then $\mathcal{B}$ can compute $D_1, D_{4,k}, D_{5,k}$ as follows:

$$\begin{aligned}
D_1 &= g^{\alpha^{ID} \gamma + \delta s_1 + \theta s_2}. \\
&= g^{\alpha^{ID}(d - \sum_{j \in S^*} \alpha^{n+1-j})} g^{v_0 s_1} \prod_{i \in V^*} (g_{n+1-i}^{\prod_{j \in J^*} (i - w_j^*)})^{s_1} g^{v_1 s_2} \prod_{i \in Z^*} (g_{n+1-i}^{\prod_{j \in J^*} (i - w_j^*)})^{s_2} \\
&= g_{ID}^d \prod_{j \in S^*} (g_{n+1-j+ID})^{-1} (g^{v_0})^{s_1'} (g^{\sum_{i \in V^*} \alpha^{n+1-i} \prod_{j \in J^*} (i - w_j^*)})^{s_1'} \\
&\quad (g^{v_1})^{s_2' + \frac{\alpha^{i^*}}{\prod_{j \in J^*} (i^* - w_j^*)}} (g^{\sum_{i \in Z^*} \alpha^{n+1-i} \prod_{j \in J^*} (i - w_j^*)})^{s_2' + \frac{\alpha^{i^*}}{\prod_{j \in J^*} (i^* - w_j^*)}} \\
&= g_{ID}^d \prod_{j \in S^*, j \neq ID} (g_{n+1-j+ID})^{-1} g^{-\alpha^{n+1}} \\
&\quad (g^{v_0})^{s_1'} (g^{\sum_{i \in V^*} \alpha^{n+1-i} \prod_{j \in J^*} (i - w_j^*)})^{s_1'} \\
&\quad (g^{v_1})^{s_2' + \frac{\alpha^{i^*}}{\prod_{j \in J^*} (i^* - w_j^*)}} (g^{\sum_{i \in Z^*} \alpha^{n+1-i} \prod_{j \in J^*} (i - w_j^*)})^{s_2'}
\end{aligned}$$

$$
\begin{aligned}
&(g^{\frac{\sum_{i\in Z^*, i\neq i^*}\alpha^{n+1-i+i^*}\prod_{j\in J^*}(i-w_j^*)}{\prod_{j\in J^*}(i^*-w_j^*)}})g^{\alpha^{n+1}}\\
=\ & g_{ID}^d \prod_{j\in S^*, j\neq ID}(g_{n+1-j+ID})^{-1}\\
&(g^{v_0})^{s_1'}(g^{\sum_{i\in V^*}\alpha^{n+1-i}\prod_{j\in J^*}(i-w_j^*)})^{s_1'}\\
&(g^{v_1})^{s_2'+\frac{\alpha^{i^*}}{\prod_{j\in J^*}(i^*-w_j^*)}}(g^{\sum_{i\in Z^*}\alpha^{n+1-i}\prod_{j\in J^*}(i-w_j^*)})^{s_2'}\\
&(g^{\frac{\sum_{i\in Z^*, i\neq i^*}\alpha^{n+1-i+i^*}\prod_{j\in J^*}(i-w_j^*)}{\prod_{j\in J^*}(i^*-w_j^*)}})\\
D_{4,k} =\ & \prod_{i\in V}(g^{u_i-\alpha^{n+1-i}})^{i^k s_1'} = \prod_{i\in V} h_i^{i^k s_1}\\
D_{5,k} =\ & \prod_{i\in Z}(g^{u_i-\alpha^{n+1-i}})^{i^k(s_2'+\frac{\alpha^{i^*}}{\prod_{j\in J^*}(i^*-w_j^*)})} = \prod_{i\in Z} h_i^{i^k s_2}
\end{aligned}
$$

◇ If there exists an $i^* \in \{Z \cup V^*\} \neq \emptyset$:
the simulation can be performed in a similar way by choosing two random numbers $s_1', s_2' \in \mathbb{Z}_p$ and implicitly setting s_1, s_2 as: $\begin{cases} s_1 = s_1' + \frac{\alpha^{i^*}}{\prod_{j\in J^*}(i^*-w_j^*)} \\ s_2 = s_2' \end{cases}$. We omit the details here.

$\mathcal{B}$ returns to $\mathcal{A}$ the secret key $SK = (D_1, D_2, D_3, \{D_{4,k}\}, \{D_{5,k}\})$.

Challenge: The adversary gives two messages M_0 and M_1 to $\mathcal{B}$. Then $\mathcal{B}$ flips a coin b and generates the challenge ciphertext by setting $C_1 = g^\tau = h$ for some unknown τ and

$$
\begin{aligned}
C_2 &= h^d = (g^d)^\tau\\
&= (g^d \prod_{j\in S^*}(g_{n+1-j})^{-1} \prod_{j\in S^*}(g_{n+1-j}))^\tau\\
&= (\nu \prod_{j\in S^*}(g_{n+1-j}))^\tau
\end{aligned}
$$

$$
\begin{aligned}
C_3 &= h^{v_0+\sum_{i\in V^*} u_i \prod_{j\in J^*}(i-w_j^*)} = (g^{v_0+\sum_{i\in V^*} u_i \prod_{j\in J^*}(i-w_j^*)})^\tau\\
C_4 &= h^{v_1+\sum_{i\in Z^*} u_i \prod_{j\in J^*}(i-w_j^*)} = (g^{v_1+\sum_{i\in Z^*} u_i \prod_{j\in J^*}(i-w_j^*)})^\tau
\end{aligned}
$$

$\mathcal{B}$ sends the following challenge ciphertext to $\mathcal{A}$:

$$CT^* = (M_b T, C_1, C_2, C_3, C_4).$$

Phase II: Same as Phase I.

Guess: $\mathcal{A}$ outputs $b' \in \{0, 1\}$. If $b' = b$ then $\mathcal{B}$ outputs 1, otherwise outputs 0.

Analysis: If $T = e(g_{n+1}, h)$, then the simulation is the same as in the real game. Hence, $\mathcal{A}$ will have the probability $\frac{1}{2} + \epsilon$ to guess b correctly. If T is a random element of $\mathbb{G}_T$, then $\mathcal{A}$ will have probability $\frac{1}{2}$ to guess b correctly. Therefore, $\mathcal{B}$ can solve the Decision n-BDHE assumption also with advantage ϵ. □

6 Conclusion

We proposed two efficient Attribute Based Broadcast Encryption (ABBE) schemes allowing access policies to be expressed using AND-gate with positive, negative, and wildcard symbols. Our first key policy ABBE scheme achieves constant secret key size, while the second ciphertext policy ABBE scheme achieves constant ciphertext size, and both schemes require only constant number of pairing operations in decryption. We also proved the security of our schemes under the Decision n-BDHE assumption. One open problem is to construct an ABBE scheme that has constant ciphertext and secret key, and we leave it as our future work.

References

1. Berkovits, S.: How to broadcast a secret. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 535–541. Springer, Heidelberg (1991)
2. Fiat, A., Naor, M.: Broadcast encryption. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 480–491. Springer, Heidelberg (1994)
3. Naor, D., Naor, M., Lotspiech, J.: Revocation and tracing schemes for stateless receivers. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, p. 41. Springer, Heidelberg (2001)
4. Dodis, Y., Fazio, N.: Public key trace and revoke scheme secure against adaptive chosen ciphertext attack. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 100–115. Springer, Heidelberg (2002)
5. Boneh, D., Gentry, C., Waters, B.: Collusion resistant broadcast encryption with short ciphertexts and private keys. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 258–275. Springer, Heidelberg (2005)
6. Boneh, D., Waters, B.: A fully collusion resistant broadcast, trace, and revoke system. In: ACM CCS, pp. 211–220 (2006)
7. Delerablée, C., Paillier, P., Pointcheval, D.: Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys. In: Takagi, T., Okamoto, T., Okamoto, E., Okamoto, T. (eds.) Pairing 2007. LNCS, vol. 4575, pp. 39–59. Springer, Heidelberg (2007)
8. Gentry, C., Waters, B.: Adaptive security in broadcast encryption systems (with short ciphertexts). In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 171–188. Springer, Heidelberg (2009)
9. Phan, D.-H., Pointcheval, D., Shahandashti, S.F., Strefler, M.: Adaptive CCA broadcast encryption with constant-size secret keys and ciphertexts. In: Susilo, W., Mu, Y., Seberry, J. (eds.) ACISP 2012. LNCS, vol. 7372, pp. 308–321. Springer, Heidelberg (2012)
10. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
11. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: ACM CCS, pp. 89–98 (2006)
12. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE S&P, pp. 321–334 (2007)
13. Cheung, L., Newport, C.: Provably secure ciphertext policy ABE. In: ACM CCS, pp. 456–465 (2007)

14. Waters, B.: Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 53–70. Springer, Heidelberg (2011)
15. Attrapadung, N., Libert, B., de Panafieu, E.: Expressive key-policy attribute-based encryption with constant-size ciphertexts. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 90–108. Springer, Heidelberg (2011)
16. Lewko, A., Waters, B.: New proof methods for attribute-based encryption: achieving full security through selective techniques. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 180–198. Springer, Heidelberg (2012)
17. Junod, P., Karlov, A.: An efficient public-key attribute-based broadcast encryption scheme allowing arbitrary access policies. In: ACM Workshop on Digital Rights Management, pp. 13–24 (2010)
18. Attrapadung, N., Imai, H.: Conjunctive broadcast and attribute-based encryption. In: Shacham, H., Waters, B. (eds.) Pairing 2009. LNCS, vol. 5671, pp. 248–265. Springer, Heidelberg (2009)
19. Lubicz, D., Sirvent, T.: Attribute-based broadcast encryption scheme made efficient. In: Vaudenay, S. (ed.) AFRICACRYPT 2008. LNCS, vol. 5023, pp. 325–342. Springer, Heidelberg (2008)
20. Sahai, A., Waters, B.: Revocation systems with very small private keys. IACR Cryptology ePrint Archive 2008/309
21. Sedghi, S., van Liesdonk, P., Nikova, S., Hartel, P., Jonker, W.: Searching keywords with wildcards on encrypted data. In: Garay, J.A., De Prisco, R. (eds.) SCN 2010. LNCS, vol. 6280, pp. 138–153. Springer, Heidelberg (2010)