Private Ciphertext-Policy Attribute-based Encryption Schemes With Constant-Size Ciphertext Supporting CNF Access Policy

Sébastien Canard[1] and Viet Cuong Trinh[1]

[1]Orange Labs, 42 rue des Coutures, 14000, Caen, France

Abstract. Attribute-based encryption (ABE) is an extension of traditional public key encryption in which the encryption and decryption phases are based on user's attributes. More precisely, we focus on *ciphertext-policy* ABE (CP-ABE) where the secret-key is associated to a set of attributes and the ciphertext is generated with an access policy. It then becomes feasible to decrypt a ciphertext only if one's attributes satisfy the used access policy.
In this paper, we give the first private CP-ABE constructions with a constant-size ciphertext, supporting CNF (Conjunctive Normal Form) access policy, with the simple restriction that each attribute can only appear k_{max} times in the access formula. Our two constructions are based on the BGW scheme at Crypto'05. The first scheme is basic selective secure (in the standard model) while our second one reaches the selective CCA security (in the random oracle model).

Keywords: Attribute-based encryption, ciphertext-policy, CNF

1 Introduction

This is commonly believed that we are currently starting a second period of development of cryptography. This "era of modern cryptography" sees the creation and the improvement of many advanced cryptographic schemes, permitting new and sometimes very complex properties. As an example, in many modern applications, one needs to have stronger and flexible capabilities to encrypt data, such that encrypting a message according to a specific policy. In this case, only receivers with attributes satisfying this specific policy can decrypt the encrypted message.

Attribute-Base Encryption. Addressing this problem, Sahai and Waters [26] introduced the concept of *attribute-based encryption* (ABE) in which the encryption and decryption can be based on the user's attributes. It exists two variants of ABE: *ciphertext-policy* attribute-based encryption (CP-ABE) and *key-policy* attribute-based encryption (KP-ABE). In CP-ABE scheme, the secret key is associated with a set of attributes and the ciphertext is associated with an access policy (structure) over a universe of attributes: a user can then decrypt a given ciphertext if the set of attributes related to his/her secret key satisfies the access policy underlying the ciphertext. In contrast, in KP-ABE scheme, the access policy is for the secret key and the set of attributes is for the ciphertext. In this paper, we focus on CP-ABE which can for example be used in Pay-TV systems, as shown in [17], and for which the size of the ciphertext is essential. We more precisely focus on *private* CP-ABE where the encryption phase is private, meaning that it necessitates the use of some secret keys (in contrast to *public* CP-ABE where anybody can encrypt a message). Again, this case is for example very suitable in the Pay-TV context where only the content broadcaster needs to encrypt something.

1.1 Related Work

Attribute-Base Encryption. Since their introduction in 2005, one can find a lot of papers proposing ABE schemes [26, 14, 23, 18, 17, 15, 5, 22, 25, 8, 29]. The authors in [5, 29] introduced

KP-ABE schemes with constant-size ciphertext. The works in [14] extended the Sahai and Waters' work [26] to propose the first schemes supporting finer-grained access control, specified by a Boolean formula. Non-monotonic access structures permitting to handle the negation of attributes has been considered in subsequent works [23, 5, 29]. Thanks to multilinear maps and cryptographic obfuscations, ABE scheme supporting general access structure has been constructed [12], but as shown recently [11, 16], their real feasibility is now questionable. Adaptive security for ABE scheme was considered in [19, 8, 3, 28] using composite order group, and then in [22, 10] using prime order groups. Similarly, dynamic ABE scheme (unbounded attributes) was first investigated in [20] using composite order groups and then in [25] using prime order groups.

Among those constructions, five of them proposed CP-ABE schemes with a constant size ciphertext, but each time with limited access structure. In fact, the most general case could be to manage Disjunctive Normal Form (DNF, i.e., with disjunctions (OR) of conjunctions (AND)) or Conjunctive Normal Form (CNF, i.e., with conjunctions (AND) of disjunctions (OR)). However, in [18, 9], the access structure is constructed by AND-gates on multi-valued attributes. In [15, 13, 8], the access policy is *threshold*, meaning that there is no distinction among attributes in the access policy: anyone who possesses enough attributes (equal or bigger than a threshold chosen by the sender) will be able to decrypt. To the best of our knowledge, it thus remains an open problem to propose a CP-ABE with constant size ciphertext, supporting general CNF or DNF access policy. In this paper, we fill this gap by proposing a new solution for CNF access policy.

Concurrent and IndependentWork. Concurrently and independently to our work, in [2] and [4] the authors proposed two CP-ABE schemes with constant-size ciphertexts. Compared to our CP-ABE scheme, their schemes are in public key setting which can be applicable to broader context (as we mentioned above we focus on the Pay-TV context) and achieve better security than ours. However, the ciphertext-size and user's key-size in our scheme are shorter, moreover our scheme supports revocation.

1.2 Our Contribution

In this work, we propose the first private CP-ABE supporting CNF access policy and having a constant size ciphertext. For that purpose, we make use of the techniques given in the Junod-Karlov ABBE scheme [17] to achieve CNF access policy and to fight against attribute collusion and the ones from the Multi-Channel Broadcast Encryption (MCBE) scheme given in [24] in order to achieve the constant size of the ciphertext.

More precisely, we present two private CP-ABE schemes with the following properties.

- Both schemes achieve the constant size ciphertext. The key size is linear in the maximal number of attributes in the system. Regarding the access policy, both scheme support restricted CNF access policy in the sense that they introduce a parameter k_{max} in which each attribute can only appear k_{max} times in the access formula used during the encryption phase. The key size is larger than a factor of k_{max} in exchange.
- Both of our schemes are naturally based on the use of an asymmetric bilinear pairing, contrary to previous work based on the symmetric case (even if a generic construction [1] can permit to transform them into the asymmetric case).
- Our first scheme achieves basic selective security under a GDDHE assumption [6], in the standard model.
- Our second scheme improves the first one regarding the security since it achieves selective CCA security under again a similar GDDHE assumption. However, we need to use the random oracle in the security proof.

We give in Table 1 a comparison among our schemes and some others existing CP-ABE schemes.

	Access Policy	Ciphertext	Dec key	Enc key	Assumption
[18]	AND-gates	$O(1)$	$O(1)$	$O(n^2)$	DBDH
[15]	Threshold	$O(1)$	$O(n)$	$O(n)$	GDDHE
[17]	CNF	$O(m)$	$O(n)$	$O(n)$	GDDHE
Our 1st	Restricted CNF	$O(1)$	$O(n.k_{max})$	$O(n.k_{max})$	GDDHE
Our 2nd	Restricted CNF	$O(1)$	$O(n.k_{max})$	$O(1)$	GDDHE+ROM

Table 1. Comparison among our schemes and some previous schemes. n denotes the maximal number of attributes in the system, m denotes the number of clauses in the CNF access policy. Restricted CNF means that each attribute only can appear k_{max} times in an access formula. Our second scheme reachs selective-CCA security, all others schemes in the table achieve selective security.

1.3 Organization of the Paper

The paper is now organized as follows. The next section introduces security definitions and the used assumptions. In Section 3, we introduce our first scheme with basic selective security, while Section 4 describes our second scheme with selective CCA security. We finally conclude in Section 5.

2 Preliminaries

We give in this section several preliminaries regarding security model of private CP-ABE schemes and security assumptions we will need for our construction.

2.1 Private Ciphertext-Policy Attribute-Based Encryption

Formally, we define a *private* CP-ABE scheme consists of three probabilistic algorithms as follows.

Setup$(1^\lambda, \vartheta, \mathcal{B}(u_i)_{1\leq i\leq\vartheta})$: it takes as input the security parameter λ, the total number of users in the system ϑ, and the attribute repartition $\mathcal{B}(u_i)_{1\leq i\leq\vartheta}$ for each user u_i, generates the global parameters param of the system, an encryption key EK, and ϑ decryption keys d_{u_i}. The encryption key EK is kept private from users. The set $\mathcal{K}$ corresponds to the key space for session keys.

Encrypt$(\mathbb{A}, \mathsf{EK}, \mathsf{param})$: it takes as input an access policy $\mathbb{A}$ and the encryption key EK. It outputs the session keys $K \in \mathcal{K}$ and the header Hdr which includes the access policy $\mathbb{A}$.

Decrypt$(\mathsf{Hdr}, d_{u_i}, \mathcal{B}(u_i), \mathsf{param})$: it takes as input the header Hdr, the decryption key d_{u_i} of a user u_i with attributes $\mathcal{B}(u_i)$, together with the parameters param. It outputs the session keys K if and only if $\mathcal{B}(u_i)$ satisfies $\mathbb{A}$. Otherwise, it outputs $\perp$.

Security Model: In this paper, we will consider the same security model as in [17] which is called *semantic security with full static collusions*. In fact, a private CP-ABE scheme is said to be secure in this model if given a challenge header and all the decryption keys of revoked users to an adversary, moreover the adversary also can ask the encryption query or decryption query. It is impossible for the adversary to infer any information about the session key. Formally, we now define the security model for a private CP-ABE scheme by the following probabilistic game between an attacker $\mathcal{A}$ and a challenger $\mathcal{C}$:

Both $\mathcal{A}$ and $\mathcal{C}$ are given a system consisting of n attributes $A_1, \ldots, A_n$.

$\mathcal{A}$ outputs target access policy $\mathbb{A}^*$ as well as a repartition $\mathcal{B}(u_i)_{1\leq i\leq\vartheta}$ which he intends to attack.

Setup$(1^\lambda, \vartheta, \mathcal{B}(u_i)_{1\leq i\leq\vartheta})$ The challenger runs the **Setup**$(1^\lambda, \vartheta, \mathcal{B}(u_i)_{1\leq i\leq\vartheta})$ algorithm, he gives the decryption keys d_{u_i} where $\mathcal{B}(u_i)$ does not satisfy the target access policy $\mathbb{A}^*$ and param to $\mathcal{A}$. Decryption lists Λ_D is set to empty list.

Query phase 1. The adversary $\mathcal{A}$ adaptively asks queries.

1. Decryption query on the header Hdr with u_i. The challenger answers with **Decrypt**$(\mathsf{Hdr}, d_{u_i}, \mathcal{B}(u_i), \mathsf{param})$. The full header Hdr is appended to the decryption list Λ_D;
2. Encryption query for the access policy $\mathbb{A}$. The challenger answers with **Encrypt**$(\mathbb{A}, \mathsf{EK}, \mathsf{param})$. Remark that he/she can ask encryption query on target access policy $\mathbb{A}^*$ since the encryption algorithm uses a fresh random coin for each time of the encryption.

Challenge. The challenger runs **Encrypt**$(\mathbb{A}^*, \mathsf{EK}, \mathsf{param})$ and gets (K^*, Hdr^*). Next, the challenger picks a random $b \overset{\$}{\leftarrow} \{0,1\}$. If $b=0$, the challenger sets $K = K^*$. Else, it picks a random $K \overset{\$}{\leftarrow} \mathcal{K}$. It outputs (K, Hdr^*) to $\mathcal{A}$. Note that if $b=0$, K is the real key, encapsulated in Hdr^*, and if $b=1$, K is random, independent of the header.

Query phase 2. The adversary $\mathcal{A}$ continues to adaptively ask queries as in the first phase.

Guess. The adversary $\mathcal{A}$ eventually outputs its guess $b' \in \{0,1\}$ for b.

We say the adversary wins the game if $b' = b$, but only if $\mathsf{Hdr}^* \notin \Lambda_D$. We then denote the advantage of the adversary to win the game by

$$\mathbf{Adv}^{\mathsf{ind}}(1^\lambda, \vartheta, \mathcal{B}(u_i)_{1\leq i\leq\vartheta}, \mathcal{A}) = |2\Pr[b=b'] - 1|.$$

Definition 1 (Basic Selective Security). A private CP-ABE scheme is said to be basic selective security if the advantage of the adversary in the above security game is negligible where the adversary cannot ask the encryption query and the decryption query.

Definition 2 (Selective−CCA Security). A private CP-ABE scheme is said to be selective−CCA security if the advantage of the adversary in the above security game is negligible where the adversary can ask any types of queries.

Note that this definition is much stronger than basic selective security because it allows the adversary to make any types of queries.

2.2 Bilinear Maps, CDH and (P, Q, f) − GDDHE Assumptions

Let $\mathbb{G}$, $\widetilde{\mathbb{G}}$ and $\mathbb{G}_T$ denote three finite multiplicative abelian groups of large prime order $p > 2^\lambda$ where λ is the security parameter. Let g be a generator of $\mathbb{G}$ and $\tilde{g}$ be a generator of $\widetilde{\mathbb{G}}$. We assume that there exists an admissible asymmetric bilinear map $e : \mathbb{G} \times \widetilde{\mathbb{G}} \to \mathbb{G}_T$, meaning that for all $a, b \in \mathbb{Z}_p$

1. $e(g^a, \tilde{g}^b) = e(g, \tilde{g})^{ab}$,
2. $e(g^a, \tilde{g}^b) = 1$ iff $a = 0$ or $b = 0$,
3. $e(g^a, \tilde{g}^b)$ is efficiently computable.

In the sequel, the set $(p, \mathbb{G}, \widetilde{\mathbb{G}}, \mathbb{G}_T, e)$ is called a bilinear map group system.

Definition 3 (CDH Assumption). The (t, ε) − CDH assumption says that for any t-time adversary $\mathcal{A}$ that is given $(g, g^t, h) \in \mathbb{G}$, its probability to output h^t is bounded by ε:

$$\mathbf{Succ}^{\mathsf{cdh}}(\mathcal{A}) = \Pr[\mathcal{A}(g, g^t, h) = h^t] \leq \varepsilon.$$

Let $(p, \mathbb{G}, \widetilde{\mathbb{G}}, \mathbb{G}_T, e)$ be a bilinear map group system and $g \in \mathbb{G}$ (resp. $\tilde{g} \in \widetilde{\mathbb{G}}$) be a generator of $\mathbb{G}$ (resp. $\widetilde{\mathbb{G}}$). We set $g_T = e(g, \tilde{g}) \in \mathbb{G}_T$. Let s, n be positive integers and $P, Q, R \in \mathbb{F}_p[X_1, \ldots, X_n]^s$ be three s-tuples of n-variate polynomials over $\mathbb{F}_p$. Thus, P, Q and R are just three lists containing s multivariate polynomials each. We write $P = (p_1, p_2, \ldots, p_s)$, $Q = (q_1, q_2, \ldots, q_s)$ $R = (r_1, r_2, \ldots, r_s)$ and impose that $p_1 = q_1 = r_1 = 1$. For any function $h : \mathbb{F}_p \to \Omega$ and vector $(x_1, \ldots, x_n) \in \mathbb{F}_p^n$, $h(P(x_1, \ldots, x_n))$ stands for $(h(p_1(x_1, \ldots, x_n)), \ldots, h(p_s(x_1, \ldots, x_n))) \in \Omega^s$. We use a similar notation for the s-tuples Q and R. Let $f \in \mathbb{F}_p[X_1, \ldots, X_n]$. It is said that f depends on (P, Q, R), which denotes $f \in \langle P, Q, R \rangle$, when there exists a linear decomposition (with an efficient isomorphism between $\mathbb{G}$ and $\widetilde{\mathbb{G}}$)

$$f = \sum_{1 \le i,j \le s} a_{i,j} \cdot p_i \cdot q_j + \sum_{1 \le i,j \le s} b_{i,j} \cdot p_i \cdot p_j + \sum_{1 \le i \le s} c_i \cdot r_i, \qquad a_{i,j}, b_{i,j}, c_i \in \mathbb{Z}_p.$$

We moreover have $b_{i,j} = 0$ when there is no efficiently computable homomorphism between $\mathbb{G}$ and $\widetilde{\mathbb{G}}$.

Let P, Q, R be as above and $f \in \mathbb{F}_p[X_1, \ldots, X_n]$. The $(P, Q, R, f) - \mathsf{GDDHE}$ problem is defined as follows.

Definition 4. $((P, Q, R, f) - \mathsf{GDDHE})$ [6].
Given $H(x_1, \ldots, x_n) = (g^{P(x_1, \ldots, x_n)}, \tilde{g}^{Q(x_1, \ldots, x_n)}, g_T^{R(x_1, \ldots, x_n)}) \in \mathbb{G}^s \times \widetilde{\mathbb{G}}^s \times \mathbb{G}_T^s$ as above and $T \in \mathbb{G}_T$ decide whether $T = g_T^{f(x_1, \ldots, x_n)}$.

The $(P, Q, R, f) - \mathsf{GDDHE}$ assumption says that it is hard to solve the $(P, Q, R, f) - \mathsf{GDDHE}$ problem if f is independent of (P, Q, R). In this paper, we will prove our schemes secure under this assumption.

3 Our First Scheme

In this section, we introduce our first scheme that is secure in the standard model, and achieves the basic selective security.

3.1 Intuition

Our construction is based on the BGW broadcast encryption scheme [7]. Then, we first make use of the techniques given in the Junod-Karlov ABBE scheme [17] to fight against attribute collusion. We finally integrate the techniques from the MCBE scheme in [24] to obtain a ciphertext with a constant size.

More precisely, in [7], each element of the header has the form

$$\Big(g^r, (v \cdot \prod_{j \in \beta_k} g_{n+1-j})^r\Big).$$

In the Junod-Karlov scheme [17], the authors manage to transform many instances of the BGW scheme [7] to an attribute-based encryption scheme, such that one instance of the BGW scheme corresponds to one clause in the CNF access policy. The resulting attribute-based encryption scheme then contains m BGW instances where m is the maximal number of clauses in the CNF access policy. However, this necessitates to manage a ciphertext with $m + 1$ parts. More precisely, for a CNF access policy $\mathbb{A} = a_1 \wedge \cdots \wedge a_m$, each component β_k is related to a BGW header as

$$\Big(g^{rt_k}, (v^r \prod_{j \in \beta_k} g_{n+1-j}^r)^{t_k}\Big).$$

In the MCBE scheme given in [24], the authors introduce a technique to multiply many BGW instances in one single value in order to support the new property of multi-channel for a broadcast encryption. For this purpose, they introduce new integers x_j and provide a unique header given by

$$\Big(g^r, \prod_{k=1}^{m}(v \cdot \prod_{j\in\beta_k} g_{n+1-j})^{r+\sum_{j\in\beta_k} x_j}\Big).$$

Inspired by the technique given in [24], we multiply the m instances of the BGW schemes to achieve an ABE scheme with constant-size ciphertext. Our scheme therefore inherits the properties of the MCBE scheme, as the private property and the basic selective security.

3.2 Construction

We now give the details of our construction by describing each procedure.

Setup$(1^\lambda, \vartheta, \mathcal{B}(u_i)_{1\le i\le\vartheta})$: the algorithm takes as input the security parameter λ, the total number of users in the system ϑ, and the attribute repartition $\mathcal{B}(u_i)_{1\le i\le\vartheta}$ for each user u_i, generates the global parameters param of the system, the encryption key EK, and ϑ decryption keys $d_{u_i}, 1 \le i \le \vartheta$ as follows:
Let $(p, \mathbb{G}, \widetilde{\mathbb{G}}, \mathbb{G}_T, e)$ be a bilinear map group system and let n be the maximal number of attributes in the system. The set of all possible attributes is $\{A_1, \dots, A_n\}$. All these elements are considered to be known to each participant.
The algorithm first picks random generators $g \in \mathbb{G}$ and $\tilde{g} \in \widetilde{\mathbb{G}}$. It then chooses a random scalar $\alpha \in \mathbb{Z}_p$ and computes for all $i \in [1, 2n] \setminus \{n+1\}$, the values $g_i = g^{\alpha^i}$ and $\tilde{g}_i = \tilde{g}^{\alpha^i}$. It also chooses at random $r \in \mathbb{Z}_p$ and computes $R = g^r$ and then, for all $i \in [1, 2n] \setminus \{n+1\}$, $h_i = g_i^r \in \mathbb{G}$. Next, it picks random scalars $\beta, \gamma \in \mathbb{Z}_p$ and sets $B = g_n^\beta$, $v = g^\gamma$ and $V = v^r$. It also picks additional random scalars $x_1, x_2, \dots, x_n \in \mathbb{Z}_p$ and sets $X_i = R^{x_i}$ for all $i \in [1, n]$.
The public parameters are then

$$\mathsf{param} = (g, \tilde{g}, B, R, V, g_n, \tilde{g}_1^r, h_1, \dots, h_n, h_{n+2}, \dots, h_{2n}, X_1, \dots, X_n)$$

The encryption key is $\mathsf{EK} = \mathsf{param} \cup \{x_1, \dots, x_n\}$.
To generate a decryption key d_u, let $\mathcal{B}(u) = (A_{i_1}, \dots, A_{i_N})$ be the set of attributes of user u (among the set of all possible attributes). The algorithm first picks a random scalar $s_u \in \mathbb{Z}_p$, and computes $\tilde{d}_{u_0} = \tilde{g}_1^{r(\beta+s_u)}$, then $\tilde{d}_{u_i} = \tilde{g}_i^{s_u}$ for all $i \in [1, 2n]\setminus\{n+1\}$, and finally $\tilde{d}_j = \tilde{g}_j^{\gamma\cdot s_u}$ for all $j \in \{i_1, \cdots, i_N\}$. The private decryption key for u is

$$d_u = (\tilde{d}_{u_0}, \tilde{d}_{u_1}, \dots, \tilde{d}_{u_n}, \tilde{d}_{u_{n+2}}, \dots, \tilde{d}_{u_{2n}}, \tilde{d}_{i_1}, \dots, \tilde{d}_{i_N}).$$

Encrypt$(\mathbb{A}, \mathsf{EK}, \mathsf{param})$: assuming that the access policy is expressed in CNF $\mathbb{A} = \beta_1 \wedge \cdots \wedge \beta_m$. The encryption phase works as follows: It first picks a random scalar $t \in \mathbb{Z}_p$ and sets the session key as

$$K = e(B, \tilde{g}_1^r)^{m.t+\sum_{k=1}^m \sum_{j\in\beta_k} x_j} = e(g_{n+1}, \tilde{g})^{r.\beta(m.t+\sum_{k=1}^m \sum_{j\in\beta_k} x_j)}.$$

It then computes the following values:

$$C_1 = R^t,\ C_2 = \prod_{k=1}^{m}(V \cdot \prod_{j\in\beta_k} h_{n+1-j})^{t+\sum_{j\in\beta_k} x_j},\ C_3 = g_n^{m.t+\sum_{k=1}^m \sum_{j\in\beta_k} x_j}.$$

The header is finally set to $\mathsf{Hdr} = (C_1, C_2, C_3)$ and the couple (Hdr, K) is output.

Decrypt$(\mathsf{Hdr}, \mathcal{B}(u), d_u, \mathsf{param})$: this algorithm first parses $\mathsf{Hdr} = (C_1, C_2, C_3)$. For each clause $\beta_k, k \in [1, m]$, the user u chooses an attribute $A_i \in \big(\beta_k \cup \mathcal{B}(u)\big)$ and computes K_k as

$$\frac{e(C_2, \tilde{d}_{u_i})}{e(C_1 \cdot \prod_{j\in\beta_k} X_j, \tilde{d}_i \cdot \prod_{\substack{j\in\beta_k\\ j\neq i}} \tilde{d}_{u_{n+1-j+i}}) \cdot \prod_{\substack{\ell=1\\ \ell\neq k}}^{\ell=m} e(C_1 \cdot \prod_{j\in\beta_\ell} X_j, \tilde{d}_i \cdot \prod_{j\in\beta_\ell} \tilde{d}_{u_{n+1-j+i}})}$$
$$= e(g_{n+1}, \tilde{g})^{(t+\sum_{j\in\beta_k} x_j)r.s_u}.$$

We remark that $\prod_{k=1}^m K_k = e(g_{n+1}, \tilde{g})^{(m.t+\sum_{k=1}^m \sum_{j\in\beta_k} x_j)r.s_u}$. The session key then is computed as

$$\frac{e(\tilde{d}_{u_0}, C_3)}{\prod_{k=1}^m K_k} = \frac{e(\tilde{g}_1^{r(\beta+s_u)}, g_n^{m.t+\sum_{k=1}^m \sum_{j\in\beta_k} x_j})}{e(g_{n+1}, \tilde{g})^{(m.t+\sum_{k=1}^m \sum_{j\in\beta_k} x_j)r.s_u}}$$
$$= e(g_{n+1}, \tilde{g})^{r.\beta(m.t+\sum_{k=1}^m \sum_{j\in\beta_k} x_j)} = K.$$

For the correctness: we used the relations $\tilde{d}_i = \tilde{g}^{\gamma s_u.\alpha^i}, \tilde{d}_{u_i} = \tilde{g}_i^{s_u}, \tilde{d}_{u_{n+1-j+i}} = \tilde{g}_{n+1-j+i}^{s_u}$, and $g_{n+1-j+i} = g_{n+1-j}^{\alpha^i}, \tilde{g}_{n+1-j+i} = \tilde{g}_{n+1-j}^{\alpha^i}, g_{n+1-i}^{\alpha^i} = g_{n+1}, \tilde{g}_{n+1-i}^{\alpha^i} = \tilde{g}_{n+1}$, and $V = v^r, h_i = g_i^r$. It follows that

$$K_k = \frac{e(\prod_{\ell=1}^{\ell=m}(v^r \cdot \prod_{j\in\beta_\ell} g_{n+1-j}^r)^{t+\sum_{j\in\beta_\ell} x_j}, \tilde{g}^{s_u.\alpha^i})}{e(g^{r(t+\sum_{j\in\beta_k} x_j)}, \tilde{g}^{\gamma s_u.\alpha^i} \cdot (\prod_{\substack{j\in\beta_k\\ j\neq i}} \tilde{g}_{n+1-j}^{s_u})^{\alpha^i})} \cdot$$
$$\cdot \frac{1}{\prod_{\substack{\ell=1\\ \ell\neq k}}^{\ell=m} e(g^{r(t+\sum_{j\in\beta_\ell} x_j)}, \tilde{g}^{\gamma s_u.\alpha^i} \cdot (\prod_{j\in\beta_\ell} \tilde{g}_{n+1-j}^{s_u})^{\alpha^i})}$$

$$= \frac{e((g^\gamma \cdot \prod_{j\in\beta_k} g_{n+1-j})^{\alpha^i}, \tilde{g}^{t+\sum_{j\in\beta_k} x_j})^{r.s_u}}{e(g^{t+\sum_{j\in\beta_k} x_j}, (\tilde{g}^\gamma \cdot \prod_{\substack{j\in\beta_k\\ j\neq i}} \tilde{g}_{n+1-j})^{\alpha^i})^{r.s_u}} \cdot$$
$$\cdot \prod_{\substack{\ell=1\\ \ell\neq k}}^{\ell=m} \frac{e((g^\gamma \cdot \prod_{j\in\beta_\ell} g_{n+1-j})^{\alpha^i}, \tilde{g})^{r.s_u.(t+\sum_{j\in\beta_\ell} x_j)}}{e(g, (\tilde{g}^\gamma \cdot \prod_{j\in\beta_\ell} \tilde{g}_{n+1-j})^{\alpha^i})^{r.s_u.(t+\sum_{j\in\beta_\ell} x_j)}}.$$
$$= e(g_{n+1-i}^{\alpha^i}, \tilde{g}^{t+\sum_{j\in\beta_k} x_j})^{r.s_u} = e(g_{n+1}, \tilde{g}^{t+\sum_{j\in\beta_k} x_j})^{r.s_u}$$
$$= e(g_{n+1}, \tilde{g})^{(t+\sum_{j\in\beta_k} x_j)r.s_u}$$

Remark 5. In the first scheme, the encryption key EK contains $\mathsf{EK} = \mathsf{param} \cup \{x_1, \ldots, x_n\}$ and thus cannot be public since with the knowledge of $x_i's$ adversary can break the semantic security of the first scheme. However, from the encryption key one cannot generate decryption keys for users. Like the first scheme in [24], we thus can separate the role of group manager (who generates the decryption keys) and broadcaster (who encrypts and broadcasts the content).

Remark 6. In the above construction, the attributes cannot be reused in the access policy since each β_k is a disjoint subset (following the technique in [24]). To deal with this drawback, as in [27], we allow each attribute to have many copies of itself. If we assume that k_{max} is the maximal number of times in which each attribute can appear in the access formula, then each attribute will have k_{max} copies of itself. For example, the attribute professor can be represented by k_{max} different attributes $\mathsf{professor}_1, \ldots, \mathsf{professor}_{k_{max}}$ corresponding to k_{max} different secret keys $d_{i_1}, \ldots, d_{i_{k_{max}}}$. A user possessing the attribute $\mathsf{professor}$ will receive k_{max} corresponding secret keys $d_{i_1}, \ldots, d_{i_{k_{max}}}$. Therefore, the construction above can support CNF access policy with the cost that the key size is a factor of k_{max} larger.

Remark 7. The notion of attribute-based broadcast encryption (ABBE) has then been introduced in [21] to address the problem of user revocation in an attribute-based encryption scheme. More precisely, in such system, the broadcaster is capable of revoking any receiver he wants, despite that these receivers can possess sufficient attributes to satisfy the access policy.

In fact, following the work in [17], the construction above can easily be extended to support revocation. For that purpose, we consider the identity of each user as an additional attribute (without the need to have copies of this special attribute). Then, to do the revocation, the encryption procedure needs to add one more set β_{m+1} containing the identities of privileged (unrevoked) users. The users outside the set β_{m+1} (revoked users) cannot decrypt because it lacks the partial session key related to the set β_{m+1}. It follows that the key size in our scheme will be similar to the one in Junod-Karlov scheme [17], that is linear in the maximal number of users in the system.

This way, we obtain the first ABBE scheme with a constant size ciphertext.

3.3 Security

In this section, we first give a theorem to prove that our first scheme *basic selective* secure under a $(P, Q, R, f) - \mathsf{GDDHE}$ assumption. We then show that this assumption holds in the generic group model.

More precisely, following the security model we define above the adversary will output the target access policy $\mathbb{A}^*$ as well as a repartition $\mathcal{B}(u_i)_{1\leq i\leq \vartheta}$ which he intends to attack. The challenger then runs the setup algorithm and returns the param, decryption keys of all user u_i where $\mathcal{B}(u_i)$ does not satisfy the target access policy $\mathbb{A}^*$ to the adversary, he also computes and returns the challenge header to the adversary. The adversary finally needs to make his guess on bit b. According to the framework of GDDHE assumption, we can describe this fact as a $(P, Q, R, f) - \mathsf{GDDHE}$ assumption as follows. Let P, Q, R be the list of polynomials consisting of all elements corresponding to the public global parameters, the private decryption keys of corrupted users, and the challenge header.

$$\begin{aligned}
P = \{&1, r, \alpha^n\beta, r\gamma, \alpha^n, r\alpha, \ldots, r\alpha^n, r\alpha^{n+2}, \ldots, r\alpha^{2n}, x_1 r, \ldots, x_n r, \\
&rt, \alpha^n(mt + \sum_{k=1}^{m}\sum_{j\in\beta_k} x_j), \sum_{k=1}^{m}(r\gamma + \sum_{j\in\beta_k}\alpha^{n+1-j}r)(t + \sum_{j\in\beta_k} x_j)\} \\
Q = \{&1, \alpha r, r(\beta + s_u)\alpha, \alpha s_u, \ldots, \alpha^n s_u, \alpha^{n+2}s_u, \ldots, \alpha^{2n}s_u, \alpha^{i_1}\gamma s_u, \ldots, \alpha^{i_N}\gamma s_u\} \\
R = \{&1\}, \qquad\qquad f = \alpha^{n+1}r\beta(mt + \sum_{k=1}^{m}\sum_{j\in\beta_k} x_j)
\end{aligned}$$

For all corrupted user u, $1 \leq N = |\mathcal{B}(u)| \leq n$.

Theorem 8. *If there exists an adversary $\mathcal{A}$ that solves the basic selective security of our first scheme with advantage ε, then we can construct a simulator to solve the $(P, Q, R, f) - \mathsf{GDDHE}$ assumption above with the same advantage ε in polynomial time.*

Proof. Assume that $\mathcal{B}$ is a simulator that solves the $(P, Q, R, f) - \mathsf{GDDHE}$ assumption above. At the beginning, $\mathcal{B}$ is given an instance of the $(P, Q, R, f) - \mathsf{GDDHE}$ assumption, i.e., all elements corresponding to the public global parameters, the private decryption keys of corrupted users, and the challenge header (denoted $g^{P(\dots)}, \tilde{g}^{Q(\dots)}, g_T^{R(\dots)}$), as well as an element K such that $K = e(g, \tilde{g})^f$ if bit $b = 0$, and K is a random element in $\mathbb{G}_T$ if $b = 1$. $\mathcal{B}$ will use this instance to simulate $\mathcal{A}$ and use the output of $\mathcal{A}$ to guess bit b. To do that, in the setup phase $\mathcal{B}$ gives $\mathcal{A}$ the public global parameters, the private decryption keys of corrupted users. Finally in the challenge phase, $\mathcal{B}$ gives $\mathcal{A}$ the challenge header as well as K. We note that all of these information are in $g^{P(\dots)}, \tilde{g}^{Q(\dots)}$. When $\mathcal{A}$ outputs its guess for b, $\mathcal{B}$ uses this guess to break the security of the $(P, Q, R, f) - \mathsf{GDDHE}$ assumption. Since the simulation is perfect and $\mathcal{A}$ has advantage ε, therefore $\mathcal{B}$ also has the same advantage ε in solving the $(P, Q, R, f) - \mathsf{GDDHE}$ assumption. □

We are now going to prove that (P, Q, R) and f are independent, so that the $(P, Q, R, f) - \mathsf{GDDHE}$ assumption holds in our case.

Lemma 9. *In the $(P, Q, R, f) - \mathsf{GDDHE}$ assumption above, (P, Q, R) and f are independent.*

Proof. We refer the proof of this lemma to Appendix A.

4 Our Second Scheme

We now give the details of a second scheme, which aims at improving the first one regarding the security. More precisely, it achieves selective CCA security under again a similar GDDHE assumption, in the random oracle model.

4.1 Construction

In this construction, instead of generating the terms X_i, we use a random oracle to generate them at the time of encryption. In addition, to help the security proof, we add a dummy clause containing only one attribute A_n to any access formula, and allow all users in the system to possess this attribute. This way, we are able to reach the selective CCA security.

Setup(1^λ) :
similar to the one in the first construction, except that the algorithm here uses an additional random oracle $\mathcal{H}$ on to $\mathbb{G}$. The public parameters[1] are then

$$\mathsf{param} = (g, \tilde{g}, R, V, g_n, h_1, \dots, h_n, h_{n+2}, \dots, h_{2n}, \mathcal{H})$$

The encryption key is $\mathsf{EK} = (r, \beta, \gamma, \alpha) \cup \mathsf{param}$.
To generate the decryption key for user u, similar to the one in the first construction, let $\mathcal{B}(u) = (A_{i_1}, \dots, A_{i_N}, A_n)$ be the set of attributes of user u. The private decryption key for u is

$$d_u = (\tilde{d}_{u_0}, \tilde{d}_{u_1}, \dots, \tilde{d}_{u_n}, \tilde{d}_{u_{n+2}}, \dots, \tilde{d}_{u_{2n}}, \tilde{d}_{i_1}, \dots, \tilde{d}_{i_N}, \tilde{d}_n).$$

Encrypt$(\mathbb{A}, \mathsf{EK}, \mathsf{param})$: assume that the access policy is expressed in CNF $\mathbb{A} = \beta_1 \wedge \beta_2 \wedge \dots \wedge \beta_m$, where β_m is a dummy clause that only contains the attribute A_n. The encryption phase works as follows: it first picks a random scalar $t \xleftarrow{\$} \mathbb{Z}_p$, and then computes $Y_i = \mathcal{H}(i, R^t) = R^{y_i}$ for $i = 1, \dots, m$ with unknown scalars y_i. The session key is then computed as:

$$K = e(g, \tilde{g}_{n+1})^{r.\beta.m.t} \prod_{k=1}^{m} e(Y_k, \tilde{g}_{n+1}^{\beta}) = e(g_{n+1}, \tilde{g})^{r.\beta(m.t + \sum_{k=1}^{m} y_k)}.$$

[1] We make the choice of putting all these values into param, so that the encryptor doesn't need to re-compute these values when encrypting. Another possibility is to set $\mathsf{param} = \{g, \tilde{g}, \mathcal{H}\}$ and re-compute all others values when encrypting.

Next, sets $\mathsf{Hdr} = (C_1, C_2, C_3, C_4)$ where:

$$C_1 = R^t, C_2 = \prod_{k=1}^{k=m} Y_k^{\gamma} V^t \prod_{j\in\beta_k} Y_k^{\alpha^{n+1-j}} h_{n+1-j}^t = \prod_{k=1}^{k=m} (V \cdot \prod_{j\in\beta_k} h_{n+1-j})^{t+y_k},$$

$$C_3 = g_n^{m.t} \cdot \prod_{k=1}^{m} ((Y_k)^{r^{-1}})^{\alpha^n} = g_n^{m.t+\sum_{k=1}^{m} y_k}, C_4 = \mathcal{H}(C_1, C_2, C_3)^t$$

The broadcaster can easily compute K and Hdr because it knows $r, \beta, \alpha, \gamma, g, \tilde{g}$ from EK. The header is finally set to $\mathsf{Hdr} = (C_1, C_2, C_3, C_4)$ and the couple (Hdr, K) is output.

Decrypt$(\mathsf{Hdr}, \mathcal{B}(u), d_u, \mathsf{param})$: the user u first checks whether the equation $e(C_1, \mathcal{H}(C_1, C_2, C_3)) = e(R, C_4)$ holds, then computes $Y_i = \mathcal{H}(i, C_1)$ for $i = 1, \dots, m$. For each clause β_k, the user u chooses an attribute $A_i \in \big(\beta_k \cup \mathcal{B}(u)\big)$ and computes, as in the previous scheme, for each $k \in [1, m]$:

$$\begin{aligned} K_k &= \frac{e(C_2, \tilde{d}_{u_i})}{e(C_1 \cdot Y_k, \tilde{d}_i \cdot \prod_{\substack{j\in\beta_k \\ j\neq i}} \tilde{d}_{u_{n+1-j+i}}) \cdot \prod_{\substack{\ell=1 \\ \ell\neq k}}^{\ell=m} e(C_1 \cdot Y_\ell, \tilde{d}_i \cdot \prod_{j\in\beta_\ell} \tilde{d}_{u_{n+1-j+i}})} \\ &= e(g_{n+1}, \tilde{g})^{(t+y_k)r.s_u}. \end{aligned}$$

We remark that $\prod_{k=1}^{m} K_k = e(g_{n+1}, \tilde{g})^{(m.t+\sum_{k=1}^{m} y_k)r.s_u}$. The session key is then computed as:

$$K = \frac{e(C_3, \tilde{d}_{u_0})}{\prod_{k=1}^{m} K_k} = \frac{e(g_n^{m.t+\sum_{k=1}^{m} y_k}, \tilde{g}_1^{r(\beta+s_u)})}{e(g_{n+1}, \tilde{g})^{(m.t+\sum_{k=1}^{m} y_k)r.s_u}} = e(g_{n+1}, \tilde{g})^{r.\beta(m.t+\sum_{k=1}^{m} y_k)}.$$

Remark 10. In the second scheme, from the encryption key EK one can generate the decryption keys of users, thus we do not separate the role of group manager (who generates the decryption keys) and broadcaster (who encrypts and broadcasts the content). With the similar arguments as in the first scheme, our second scheme can also support CNF access policy and revocation.

4.2 Security

In this section, we first give a theorem to prove that our second scheme is selective CCA secure under a $(P, Q, R, f) - \mathsf{GDDHE}$ assumption. We then show that this assumption holds in the generic group model.

The $(P, Q, R, f) - \mathsf{GDDHE}$ assumption that we need is, in fact, similar to the one given in Section 3.3, except that the terms $rx_1, \dots, rx_n$ are now replaced by the terms $ry_1, \dots, ry_m, z, zt$. More precisely, let P, Q, R be the list of polynomials consisting of all elements corresponding to the public global parameters, the private decryption keys of corrupted users, and the challenge header.

$$\begin{aligned} P = \{&1, r, r\gamma, \alpha^n, r\alpha, \dots, r\alpha^n, r\alpha^{n+2}, \dots, r\alpha^{2n}, ry_1, \dots, ry_m, z, zt, \\ &rt, \alpha^n(mt + \sum_{k=1}^{m} y_k), \sum_{k=1}^{m} (r\gamma + \sum_{j\in\beta_k} \alpha^{n+1-j} r)(t + y_k)\} \\ Q = \{&1, r(\beta + s_u)\alpha, \alpha s_u, \dots, \alpha^n s_u, \alpha^{n+2} s_u, \dots, \alpha^{2n} s_u, \alpha^{i_1}\gamma s_u, \dots, \alpha^{i_N}\gamma s_u, \alpha^n\gamma s_u\}, \\ R = \{&1\}, \text{ and } f = \alpha^{n+1} r\beta(mt + \sum_{k=1}^{m} y_k). \end{aligned}$$

For each user u belonging to the set of corrupted users, we have $1 \leq N = |\mathcal{B}(u)| \leq n$. This assumption can now be written as follows. Given

$$g, \tilde{g}, \tilde{g}_1^{r(\beta+s_u)}, \tilde{g}_1^{s_u}, \ldots, \tilde{g}_n^{s_u}, \tilde{g}_{n+2}^{s_u}, \ldots, \tilde{g}_{2n}^{s_u}, \tilde{g}_{i_1}^{\gamma s_u}, \ldots, \tilde{g}_{i_N}^{\gamma s_u}, \tilde{g}_n^{\gamma s_u}$$
$$R, V, g_n, h_1, \ldots, h_n, h_{n+2}, \ldots, h_{2n}, R^{y_1}, \ldots, R^{y_m}, g^z, g^{zt}$$
$$R^t, \prod_{k=1}^{k=m} (V \cdot \prod_{j \in \beta_k} h_{n+1-j})^{t+y_k}, g_n^{m.t+\sum_{k=1}^m y_k} \quad .$$

for all corrupted user u, distinguish between the value $e(g_{n+1}, \tilde{g})^{r.\beta(m.t+\sum_{k=1}^m y_k)}$ and a random $T \in \mathbb{G}_T$.

Theorem 11. *Our second scheme is selective*$-\mathsf{CCA}$ *secure under* CDH *assumption and the* $(P, Q, R, f) - \mathsf{GDDHE}$ *assumption above.*

Proof. Let $\mathsf{Hdr} = (C_1, C_2, C_3, C_4)$ be the challenge header. We will prove the security of our second scheme in two steps. First, we prove that the adversary cannot produce any decryption query of the form $\mathsf{Hdr}' = (C_1, C_2', C_3', C_4')$ under the CDH assumption. In the second step, we prove that our second scheme is selective$-\mathsf{CCA}$ secure under $(P, Q, R, f) - \mathsf{GDDHE}$ assumption with the requirement that the adversary doesn't ask any query $\mathsf{Hdr}' = (C_1, C_2', C_3', C_4')$.

First step. The simulator is first given a CDH instance g, g^t, h. It needs to compute h^t. To do that, it simulates the adversary $\mathcal{A}$ who can produce a decryption query of the form $\mathsf{Hdr}' = (C_1, C_2', C_3', C_4')$ and uses the output of $\mathcal{A}$ to compute h^t. To this aim, the simulator first receives the target access policy $\mathbb{A} = \beta_1 \wedge \beta_2 \wedge \cdots \wedge \beta_m$ from the adversary $\mathcal{A}$ as well as the repartition of attributes for each user, he randomly chooses the encryption key $\mathsf{EK} = (r, \beta, \alpha, \gamma)$ and $\mathsf{param} = (g, \tilde{g})$, he easily can compute the decryption keys for $\mathcal{A}$ and answers all types of queries from $\mathcal{A}$. He then manages to produce the challenge header with the unknown scalar t as follows.

The simulator randomly chooses $y_i \in \mathbb{Z}_p, i = 1, \ldots, m$, sets $\mathcal{H}(i, R^t) = R^{y_i}$, and appends the tuple (q_i, R^{y_i}, y_i) to the hash list related to the random oracle. He then computes:

$$C_1 = (g^t)^r, C_2 = \prod_{k=1}^{k=m} ((g^t)^{\gamma r} \cdot \prod_{j \in \beta_k} (g^t)^{\alpha^{n+1-j} r})(V \cdot \prod_{j \in \beta_k} h_{n+1-j})^{y_k}$$
$$= \prod_{k=1}^{k=m} (V \cdot \prod_{j \in \beta_k} h_{n+1-j})^{t+y_k}, C_3 = (g^t)^{m\alpha^n} \cdot g_n^{\sum_{k=1}^m y_k} = g_n^{m.t+\sum_{k=1}^m y_k}$$

It next randomly chooses $z \in \mathbb{Z}_p$, set $\mathcal{H}(C_1, C_2, C_3) = g^z$ and computes: $C_4 = (g^t)^z = \mathcal{H}(C_1, C_2, C_3)^t$. It is easy to see that the header $\mathsf{Hdr} = (C_1, C_2, C_3, C_4)$ is valid. The simulator can also easily compute the session key.

During the simulation, the simulator can answer all the hash queries $\mathcal{H}(C_1, X, Y) = h^u, u \in \mathbb{Z}_p$ for any X, Y, and appends the tuple (q_u, h^u, u) to the hash list. It can also answer all the other queries by using EK.

Next, $\mathcal{A}$ produces a valid decryption query $\mathsf{Hdr} = (C_1, C_2', C_3', C_4')$ and sends it to the simulator. Since $\mathcal{H}(C_1, C_2', C_3') = h^u$ for some known u, therefore $C_4' = h^{ut}$, from that the simulator can compute $h^t = (C_4')^{\frac{1}{u}}$ and then break the CDH assumption.

Second step. First, the simulator is given the instance of $(P, Q, R, f) - \mathsf{GDDHE}$ assumption above. Let $\mathcal{A}$ be an adversary against the security of our second scheme. The simulator will use the guess of $\mathcal{A}$ to break the instance of $(P, Q, R, f) - \mathsf{GDDHE}$ assumption above. For that purpose, the simulator first receives the target access policy $\mathbb{A} = \beta_1 \wedge \beta_2 \wedge \cdots \wedge \beta_m$ from the adversary $\mathcal{A}$ as well as the repartition of attributes for each user, from the instance of $(P, Q, R, f) - \mathsf{GDDHE}$ assumption above the simulator gives $\mathcal{A}$ the public parameters, and the decryption keys of all corrupted users. The simulator also needs to answer the following types of queries.

1. *Hash query*: There are two types of hash queries, $(j, h^*) \in \mathbb{Z}_p \times \mathbb{G}$ or $(h_1^*, h_2^*, h_3^*) \in \mathbb{G}^3$. For any query q, if it has been asked before, the same answer is sent back. Otherwise, for the (j, h^*) queries the simulator randomly chooses $y \in \mathbb{Z}_p$ and sets $\mathcal{H}(q) = R^y$, and appends the tuple (q, R^y, y) to the hash list. If the value y is unknown, it is replaced by $\perp$. For the (h_1^*, h_2^*, h_3^*) query, the simulator randomly chooses $z^* \in \mathbb{Z}_p$ and set $\mathcal{H}(q) = g^{z^*}$, and appends the tuple (q, g^{z^*}, z^*) to the hash list. If the value z^* is unknown, it is replaced by $\perp$.
2. *Encryption query*: $\mathcal{A}$ sends an access policy $\mathbb{A} = \beta_1' \wedge \beta_2' \wedge \cdots \wedge \beta_\ell'$ to simulator. The simulator first randomly chooses $t', z', y_1', \ldots, y_\ell' \in \mathbb{Z}_p$ and appends to the hash list the tuple $(q_{z'}', g^{z'}, z')$ and for all $i = 1, \ldots, \ell$, the tuples $(q_i', R^{y_i'}, y_i')$. It takes the private decryption key of a user u and then computes:

$$K = \left(\frac{e(\tilde{g}_1^{r(\beta+s_u)}, g_n)}{e(\tilde{g}_n^{s_u}, g_1^r)}\right)^{t'\ell+\sum_{k=1}^{\ell} y_k'} = e(g_{n+1}, \tilde{g})^{r.\beta(t'\ell+\sum_{k=1}^{\ell} y_k')}$$

$$C_1 = R^{t'}, C_2 = \prod_{k=1}^{k=\ell}(V \cdot \prod_{j\in\beta_k} h_{n+1-j})^{t'+y_k'}, C_3 = g_n^{t'\ell+\sum_{k=1}^{\ell} y_k'}, C_4 = g^{z't'}.$$

3. *Decryption query*: we assume that $\mathcal{A}$ sends the following ciphertext to the simulator (note that $t' \neq t$):

$$C_1 = R^{t'}, C_2 = \prod_{k=1}^{k=m'}(V \cdot \prod_{j\in\beta_k} h_{n+1-j})^{t'+y_k'}, C_3 = g_n^{m'.t'+\sum_{k=1}^{m'} y_k'}, C_4 = \mathcal{H}(C_1, C_2, C_3)^{t'}$$

   The simulator first checks whether the equation $e(C_1, \mathcal{H}(C_1, C_2, C_3)) = e(R, C_4)$ holds, takes the private decryption key of a user u and then uses the secret key $\tilde{d}_n$ corresponding to attribute A_n in the clause $\beta_{m'}$ to compute the value $e(g_{n+1}, \tilde{g})^{(t'+y_{m'}')r.s_u}$. It extracts the value $y_{m'}'$ from the hash list (since $t' \neq t$) and compute $e(\tilde{g}_n^{s_u}, g_1^r)^{y_{m'}'}$. This permits to obtain the value

$$\frac{e(g_{n+1}, \tilde{g})^{(t'+y_{m'}')r.s_u}}{e(\tilde{g}_n^{s_u}, g_1^r)^{y_{m'}'}} = e(g_{n+1}, \tilde{g})^{t'.r.s_u}.$$

   Next, it extracts all the values from y_1' to $y_{m'-1}'$ from the hash list (since $t' \neq t$) and computes the partial session keys related to each clause $\beta_i, i = 1, \ldots, m'-1$

$$K_i = e(g_{n+1}, \tilde{g})^{t'.r.s_u} \cdot e(\tilde{g}_n^{s_u}, g_1^r)^{y_i'} = e(g_{n+1}, \tilde{g})^{(t'+y_i')r.s_u}.$$

   The simulator can finally recover the following session key and forwards the result to $\mathcal{A}$.

$$K = e(g_{n+1}, \tilde{g})^{r.\beta(t'm'+\sum_{k=1}^{m'} y_k')}.$$

Next, during the challenge phase, the simulator first appends to the hash list the values $\mathcal{H}(i, R^t) = (q_i, R^{y_i}, \perp)$, for all $i = 1, \ldots, m$ and the values $\mathcal{H}(C_1, C_2, C_3) = (q_z, g^z, \perp)$. It then sends the following challenge ciphertext to $\mathcal{A}$:

$$C_1 = R^t, C_2 = \prod_{k=1}^{k=m}(V \cdot \prod_{j\in\beta_k} h_{n+1-j})^{t+y_k}, C_3 = g_n^{m.t+\sum_{k=1}^{m} y_k}, C_4 = g^{zt}.$$

If $\mathcal{A}$ make new requests to the different oracles, the simulator can use again the above strategy. Finally, when $\mathcal{A}$ outputs its guess for b, the simulator uses this guess to break the security of the $(P, Q, R, f) - \mathsf{GDDHE}$ assumption. Since $\mathcal{A}$ has advantage ε and the probability of programmation of $\mathcal{H}$ failed is negligible, therefore the simulator has the same advantage ε in solving the $(P, Q, R, f) - \mathsf{GDDHE}$ assumption. $\square$

The following lemma finally shows that in the above $(P, Q, R, f) - \mathsf{GDDHE}$ assumption, (P, Q, R) and f are independent. The proof of this lemma is similar to the one given for Lemma 9 (see Appendix A) and, therefore, we do not repeat it again.

Lemma 12. *In the $(P, Q, R, f) - \mathsf{GDDHE}$ assumption above, (P, Q, R) and f are independent.*

5 Conclusion

In this paper, we proposed two private CP-ABE schemes which possess a nice property: the constant size of the ciphertext. Our schemes support a restricted form of CNF access policy, and can naturally be extended to allow the revocation. We leave the challenging problem of how to improve the efficiency of our schemes for the future work.

References

1. M. Abe, J. Groth, M. Ohkubo, and T. Tango. Converting cryptographic schemes from symmetric to asymmetric bilinear groups. In J. A. Garay and R. Gennaro, editors, *Advances in Cryptology – CRYPTO 2014, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 241–260, Santa Barbara, CA, USA, Aug. 17–21, 2014. Springer, Berlin, Germany.
2. S. Agrawal and M. Chase. A study of pair encodings: predicate encryption in prime order groups. IACR Cryptology ePrint Archive, 2015:413, 2015.
3. N. Attrapadung. Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In P. Q. Nguyen and E. Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 557–577, Copenhagen, Denmark, May 11–15, 2014. Springer, Berlin, Germany.
4. N. Attrapadung, G. Hanaokay, and S. Yamada. Conversions among several classes of predicate encryption and applications to abe with various compactness tradeoffs. IACR Cryptology ePrint Archive, 2015:431, 2015.
5. N. Attrapadung, B. Libert, and E. de Panafieu. Expressive key-policy attribute-based encryption with constant-size ciphertexts. In D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, editors, *PKC 2011: 14th International Workshop on Theory and Practice in Public Key Cryptography*, volume 6571 of *Lecture Notes in Computer Science*, pages 90–108, Taormina, Italy, Mar. 6–9, 2011. Springer, Berlin, Germany.
6. D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In R. Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 440–456, Aarhus, Denmark, May 22–26, 2005. Springer, Berlin, Germany.
7. D. Boneh, C. Gentry, and B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In V. Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 258–275, Santa Barbara, CA, USA, Aug. 14–18, 2005. Springer, Berlin, Germany.
8. C. Chen, J. Chen, H. W. Lim, Z. Zhang, D. Feng, S. Ling, and H. Wang. Fully secure attribute-based systems with short ciphertexts/signatures and threshold access structures. In E. Dawson, editor, *Topics in Cryptology – CT-RSA 2013*, volume 7779 of *Lecture Notes in Computer Science*, pages 50–67, San Francisco, CA, USA, Feb. 25 – Mar. 1, 2013. Springer, Berlin, Germany.
9. C. Chen, Z. Zhang, and D. Feng. Efficient ciphertext policy attribute-based encryption with constant-size ciphertext and constant computation-cost. In X. Boyen and X. Chen, editors, *ProvSec 2011: 5th International Conference on Provable Security*, volume 6980 of *Lecture Notes in Computer Science*, pages 84–101, Xi'an, China, Oct. 16–18, 2011. Springer, Berlin, Germany.
10. J. Chen, R. Gay, and H. Wee. Improved dual system abe in prime-order groups via predicate encodings. In E. Oswald and M. Fischlin, editors, *Proceedings of EUROCRYPT*, LNCS 9057, pages 595–624. Springer, 2015.
11. J. H. Cheon, K. Han, C. Lee, Hansol, and D. Stehle. Cryptanalysis of the multilinear map over the integers. Cryptology ePrint Archive, Report 2014/906, 2014. http://eprint.iacr.org/2014/906.
12. S. Garg, C. Gentry, S. Halevi, A. Sahai, and B. Waters. Attribute-based encryption for circuits from multilinear maps. In R. Canetti and J. A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 479–499, Santa Barbara, CA, USA, Aug. 18–22, 2013. Springer, Berlin, Germany.

13. A. Ge, R. Zhang, C. Chen, C. Ma, and Z. Zhang. Threshold ciphertext policy attribute-based encryption with constant size ciphertexts. In W. Susilo, Y. Mu, and J. Seberry, editors, *ACISP 12: 17th Australasian Conference on Information Security and Privacy*, volume 7372 of *Lecture Notes in Computer Science*, pages 336–349, Wollongong, NSW, Australia, July 9–11, 2012. Springer, Berlin, Germany.
14. V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In A. Juels, R. N. Wright, and S. Vimercati, editors, *ACM CCS 06: 13th Conference on Computer and Communications Security*, pages 89–98, Alexandria, Virginia, USA, Oct. 30 – Nov. 3, 2006. ACM Press. Available as Cryptology ePrint Archive Report 2006/309.
15. J. Herranz, F. Laguillaumie, and C. Ràfols. Constant size ciphertexts in threshold attribute-based encryption. In P. Q. Nguyen and D. Pointcheval, editors, *PKC 2010: 13th International Conference on Theory and Practice of Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 19–34, Paris, France, May 26–28, 2010. Springer, Berlin, Germany.
16. Y. Hu and H. Jia. Cryptanalysis of ggh map. Cryptology ePrint Archive: Report 2015/301, 2014. http://eprint.iacr.org/2015/301.
17. P. Junod and A. Karlov. An efficient public-key attribute-based broadcast encryption scheme allowing arbitrary access policies. In *ACM Workshop on Digital Rights Management*, pages 13–24. ACM Press, 2010.
18. E. K, M. A, N. A, O. K, and S. M. A ciphertext-policy attribute-based encryption scheme with constant ciphertext length. In B. F, L. H, and W. G, editors, *Proceedings of ISPEC*, LNCS 5451, pages 13–23. Springer, 2009.
19. A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In H. Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 62–91, French Riviera, May 30 – June 3, 2010. Springer, Berlin, Germany.
20. A. B. Lewko and B. Waters. Unbounded HIBE and attribute-based encryption. In K. G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 547–567, Tallinn, Estonia, May 15–19, 2011. Springer, Berlin, Germany.
21. D. Lubicz and T. Sirvent. Attribute-based broadcast encryption scheme made efficient. In S. Vaudenay, editor, *AFRICACRYPT 08: 1st International Conference on Cryptology in Africa*, volume 5023 of *Lecture Notes in Computer Science*, pages 325–342, Casablanca, Morocco, June 11–14, 2008. Springer, Berlin, Germany.
22. T. Okamoto and K. Takashima. Fully secure unbounded inner-product and attribute-based encryption. In X. Wang and K. Sako, editors, *Advances in Cryptology – ASIACRYPT 2012*, volume 7658 of *Lecture Notes in Computer Science*, pages 349–366, Beijing, China, Dec. 2–6, 2012. Springer, Berlin, Germany.
23. R. Ostrovsky, A. Sahai, and B. Waters. Attribute-based encryption with non-monotonic access structures. In P. Ning, S. D. C. di Vimercati, and P. F. Syverson, editors, *ACM CCS 07: 14th Conference on Computer and Communications Security*, pages 195–203, Alexandria, Virginia, USA, Oct. 28–31, 2007. ACM Press.
24. D. H. Phan, D. Pointcheval, and V. C. Trinh. Multi-Channel Broadcast Encryption. In *Proceedings of the 8th ACM Symposium on InformAtion, Computer and Communications Security (ASIACCS '13)*, ACM Press, 2013.
25. Y. Rouselakis and B. Waters. Practical constructions and new proof methods for large universe attribute-based encryption. In A.-R. Sadeghi, V. D. Gligor, and M. Yung, editors, *ACM CCS 13: 20th Conference on Computer and Communications Security*, pages 463–474, Berlin, Germany, Nov. 4–8, 2013. ACM Press.
26. A. Sahai and B. R. Waters. Fuzzy identity-based encryption. In R. Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473, Aarhus, Denmark, May 22–26, 2005. Springer, Berlin, Germany.
27. B. Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, editors, *PKC 2011: 14th International Workshop on Theory and Practice in Public Key Cryptography*, volume 6571 of *Lecture Notes in Computer Science*, pages 53–70, Taormina, Italy, Mar. 6–9, 2011. Springer, Berlin, Germany.
28. H. Wee. Dual system encryption via predicate encodings. In Y. Lindell, editor, *TCC 2014: 11th Theory of Cryptography Conference*, volume 8349 of *Lecture Notes in Computer Science*, pages 616–637, San Diego, CA, USA, Feb. 24–26, 2014. Springer, Berlin, Germany.
29. S. Yamada, N. Attrapadung, G. Hanaoka, and N. Kunihiro. A framework and compact constructions for non-monotonic attribute-based encryption. In H. Krawczyk, editor, *PKC 2014: 17th International Workshop on Theory and Practice in Public Key Cryptography*, volume 8383 of *Lecture Notes in Computer Science*, pages 275–292, Buenos Aires, Argentina, Mar. 26–28, 2014. Springer, Berlin, Germany.

A Proof of Lemma 8

We prove for the general case where we allow all polynomials in P, Q to multiply with each other, which is exactly the symmetric pairing when P, Q are in the same group. For notational simplicity, we denote $P = P \cup Q$.

Suppose that f is not independent to (P, Q, R), *i.e.*, one can find $a_{i,j}, c_i$ such that the following equation holds

$$f = \sum_{\{p_i, p_j\} \subset P} a_{i,j} \cdot p_i \cdot p_j + c_i$$

Assume that $\mathit{\Lambda}_C$ is the list of corrupted users. We will use β to analyze f, set $q_u = \alpha r(\beta + s_u), u \in \mathit{\Lambda}_C$, $P' = P \setminus \{q_u\}_{u\in\mathit{\Lambda}_C}$. We rewrite f as follows:

$$f = \sum_{\{u,v\}\subset\mathit{\Lambda}_C} a_{u,v} q_u q_v + \sum_{u\in\mathit{\Lambda}_C, p_i\in P'} a_{u,i} p_i q_u + \sum_{\{p_i,p_j\}\subset P'} a_{i,j} p_i p_j + c_i = f_1 + f_2 + f_3$$

Consider f_1, we rewrite it as follows:

$$f_1 = \sum_{\{u,v\}\subset\mathit{\Lambda}_C} a_{u,v} q_u q_v = \sum_{\{u,v\}\subset\mathit{\Lambda}_C} a_{u,v} \alpha^2 r^2 (\beta^2 + \beta s_u + \beta s_v + s_u s_v)$$

Since s_u, s_v are random elements thus the value $a_{u,v}\alpha^2 r^2 s_u s_v$ is unique. On the other hand, this value doesn't appear in $f = \alpha^{n+1} r\beta(mt + \sum_{k=1}^{m}\sum_{j\in\beta_k} x_j)$, this leads to the fact that $a_{u,v} = 0$ for any $\{u, v\} \subset \mathit{\Lambda}_C$, or we have $f_1 = 0$.

Consider $f_2 = \sum_{u\in\mathit{\Lambda}_C, p_i\in P'} a_{u,i} p_i q_u$, to let it appear the needed term $\alpha^{n+1} r\beta$ we divide the polynomials $p_i \in P'$ into two subsets, one containing the term α^n denoted P'_1, and one doesn't denoted P'_2. We now rewrite f_2 as follows: $f_2 =$

$$\sum_{u\in\mathit{\Lambda}_C, p_i\in P'_1} a_{u,i} p_i q_u + \sum_{u\in\mathit{\Lambda}_C, p_i\in P'_2} a_{u,i} p_i q_u = \sum_{u\in\mathit{\Lambda}_C, p_i\in P'_1} a_{u,i} p_i \alpha r(\beta + s_u) + \sum_{u\in\mathit{\Lambda}_C, p_i\in P'_2} a_{u,i} p_i q_u$$

We therefore lead to the equation

$$f = \alpha^{n+1} r\beta(mt + \sum_{k=1}^{m}\sum_{j\in\beta_k} x_j) = \sum_{u\in\mathit{\Lambda}_C, p_i\in P'_1} a_{u,i} p_i \alpha r(\beta + s_u) + \sum_{u\in\mathit{\Lambda}_C, p_i\in P'_2} a_{u,i} p_i q_u + f_3 (1)$$

Since the term $\alpha^{n+1} r\beta$ only appear in $\sum_{u\in\mathit{\Lambda}_C, p_i\in P'_1} a_{u,i} p_i \alpha r(\beta + s_u)$, to make the equation (1) hold one needs to remove the term related to s_u in $\sum_{u\in\mathit{\Lambda}_C, p_i\in P'_1} a_{u,i} p_i \alpha r(\beta + s_u)$, and the only way to do that is to produce the term $\sum_{u\in\mathit{\Lambda}_C, p_i\in P'_1} a_{u,i} p_i \alpha r s_u$ for each $u \in \mathit{\Lambda}_C$.

On the other hand, to make appear the term $f = \alpha^{n+1} r\beta(mt + \sum_{k=1}^{m}\sum_{j\in\beta_k} x_j)$, the polynomial $p_i, p_i \in P'_1$, cannot have the form containing $\alpha^n\beta$ or $\alpha^n r$, or $\alpha^n s_u$ (if not, it will make the redundancy when multiplying with $q_u = \alpha r(\beta + s_u)$). The only one such p_i comes from $p_i = \alpha^n(mt + \sum_{k=1}^{m}\sum_{j\in\beta_k} x_j)$. This leads to the fact that one only can produce the term

$$\sum_{u\in\mathit{\Lambda}_C} a_{u,i} \alpha^{n+1} r(\beta + s_u)(mt + \sum_{k=1}^{m}\sum_{j\in\beta_k} x_j)$$

That means one needs to produce the term related to s_u:

$$f' = \sum_{u\in\mathit{\Lambda}_C} a_{u,i} \alpha^{n+1} r s_u (mt + \sum_{k=1}^{m}\sum_{j\in\beta_k} x_j)$$

Since each user $u \in \mathit{\Lambda}_C$ lacks at least one term $\alpha^{n+1} r s_u(t + \sum_{j\in\beta_k} x_j)$ for some β_k and no one can help because of the unique value s_u, therefore one cannot reach to f'. That means the equation (1) cannot hold or f is independent to (P, Q, R). $\square$