Constant-Size Ciphertext Attribute-Based Encryption from Multi-channel Broadcast Encryption

Sébastien Canard[1(✉)] and Viet Cuong Trinh[1,2]

[1] Orange Labs - Applied Crypto Group, Caen, France
sebastien.canard@orange.com
[2] Hong Duc University, Thanh Hoa, Viet Nam

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. CP-ABE scheme with constant-size ciphertext supporting fine-grained access control has been investigated at AsiaCrypt'15 and then at TCC'16. The former makes use of the conversion technique between ABE and spatial encryption, and the later studies the pair encodings framework.

In this paper, we give a new approach to construct such kind of CP-ABE scheme. More precisely, we propose private CP-ABE schemes with 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

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 [28] introduced the concept of *attribute-based encryption* (ABE) in which

I. Ray et al. (Eds.): ICISS 2016, LNCS 10063, pp. 193–211, 2016.
DOI: 10.1007/978-3-319-49806-5_10

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 [19], 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 [5,8,12,15,17,19,24,25,27,28,31]. The authors in [5,31] introduced KP-ABE schemes with constant-size ciphertext. The works in [15] extended the Sahai and Waters' work [28] 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 [5,25,31]. Thanks to multilinear maps and cryptographic obfuscations, ABE scheme supporting general access structure has been constructed [13], but as shown recently [11,18,23], their real feasibility is questionable. Adaptive security for ABE schemes was considered in [3,8,20,30] using composite order group, and then in [10,24] using prime order groups. Similarly, dynamic ABE scheme (unbounded attributes) was first investigated in [21] using composite order groups and then in [27] using prime order groups.

Among those constructions, five of them propose CP-ABE schemes with constant size ciphertext supporting limited access structure. In [9,12], the access structure is constructed by AND-gates on multi-valued attributes. In [8,14,17], 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, there exists only two interesting approaches to construct CP-ABE schemes with constant size ciphertext supporting fine-grained access control. The first one [4] makes use of the conversion technique between ABE and spatial encryption [16]. More precisely, starting from a KP-ABE scheme with constant-size ciphertext, such that [5,31], one first converts it to a spatial encryption scheme with constant-size ciphertext. Then, from this spatial encryption scheme, one continues to convert it to a CP-ABE schemes with constant size ciphertext. The second approach [2] comes from the pair encodings technique [3,30], in which it is proposed a new relaxed but still information theoretic security property that is sufficient to achieve a CP-ABE schemes with constant size ciphertext. The weakness of these both approaches is that the key-size is relatively large.

1.2 Our Contribution

In this work, we propose a new approach to construct CP-ABE schemes with constant size ciphertext supporting CNF access policy. For that purpose, we make use of the techniques given in the Junod-Karlov ABBE scheme [19] to achieve CNF access policy and to fight against attribute collusion and the ones from the Multi-Channel Broadcast Encryption (MCBE) scheme given in [26] 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 schemes 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.

When comparing to the two interesting existing approaches [2,4], ours leads to a scheme with better key size. However, the schemes in [2,4] are in public setting and are large universe CP-ABE schemes. We give in Table 1 a comparison among our schemes and some other existing CP-ABE schemes. We moreover argue that our approach to construct constant-size ciphertext ABE is new and can lead to better schemes in the future. We also notice that using the technique given in [19], we are able to turn our scheme into the first attribute-based broadcast encryption [22] (ABBE) with a constant size ciphertext.

1.3 Organization of the Paper

The next section introduces security definitions and the used assumptions. In Sect. 3, we introduce our first scheme with basic selective security, while Sect. 4 describes our second scheme with selective CCA security. Finally, in Sect. 5 we give the conclusion.

Table 1. Comparison among our schemes and some previous schemes. n denotes the number of attributes in the system, m denotes the number of clauses in the CNF access policy, k denotes the maximal size of an attribute set associated with a secret key, ℓ denotes the maximal number of rows of a span program matrix associated with a ciphertext (fixed at the setup, thus should be n). Restricted CNF means that each attribute only can appear k_{max} times in an access formula. We note that [4] supports large universe and obtains adaptive security, [2] supports large universe and obtains selective security.

	Access Policy	Ciphertext	Dec key	Enc key	Assumption
[12]	AND-gates	$O(1)$	$O(1)$	$O(n^2)$	DBDH
[17]	Threshold	$O(1)$	$O(n)$	$O(n)$	GDDHE
[19]	CNF	$O(m)$	$O(n)$	$O(n)$	GDDHE
[4]	LSSS	$O(1)$	$O(k^4.\ell^4)$	$O(k^2.\ell^2)$	Parametrized
[2]	LSSS	$O(1)$	$O(n.\ell^2)$	$O(n.\ell)$	Parametrized
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

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 which 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 ($\mathcal{B}(u_i)$ is the attribute set of 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, a decryption key d_{u_i} and the attribute set $\mathcal{B}(u_i)$ of user 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 consider the same security model as in [19] 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 to an adversary (i) a challenge header, (ii) all the decryption keys of revoked users and (iii) a access to both encryption and decryption oracles, it is impossible for the adversary to infer any information about the session key. Formally, we now define the security model for a private $\mathsf{CP\text{-}ABE}$ 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 $\mathcal{B}(u_i)_{1 \le i \le \vartheta}$ which he intends to attack.

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

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

Definition 1 (Basic Selective Security). *A private* $\mathsf{CP\text{-}ABE}$ *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* $\mathsf{CP\text{-}ABE}$ *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.*

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 $e : \mathbb{G} \times \widetilde{\mathbb{G}} \rightarrow \mathbb{G}_T$, 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 $(t, \varepsilon) - \mathsf{CDH}$ 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 $h : \mathbb{F}_p \rightarrow \Omega$ 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 $\widetilde{\mathbb{G}}$)

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

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 the security of our schemes 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 two main techniques. First, we make use of the techniques given in the Junod-Karlov ABBE scheme [19] to fight against attribute collusion. We finally integrate the techniques from the MCBE scheme in [26] to obtain a ciphertext with a constant size. Note that both ABBE scheme [19] and MCBE scheme in [26] are constructed from BGW scheme [7].

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 [19], 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 leads to a ciphertext with $m+1$ parts. More precisely, for a CNF access policy $\mathbb{A} = \beta_1 \wedge \cdots \wedge \beta_m$, each component $\beta_k, k \in [m]$, is related to a BGW header as

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

In the MCBE scheme given in [26], 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 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 [26], we manage to 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

$\{A_1, \ldots, A_n\}$. 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 $i \in [1, 2n]\backslash\{n+1\}$, 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 $i \in [1, 2n]\backslash\{n+1\}$, $h_i = g_i^r \in \mathbb{G}$. 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 $x_1, x_2, \ldots, x_n \in \mathbb{Z}_p$ and sets $X_i = R^{x_i}$ for all $i \in [1, n]$. The public parameters are then

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

The encryption key is $\mathsf{EK} = \mathsf{param} \cup \{x_1, \ldots, x_n\}$.
To generate a decryption key d_u, let $\mathcal{B}(u) = (A_{i_1}, \ldots, A_{i_N})$ 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 $i \in [1, 2n]\backslash\{n+1\}$, 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

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

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 $\mathsf{Hdr} = (\mathbb{A}, C_1, C_2, C_3)$, and the pair (Hdr, K) is the output.

Decrypt$(\mathsf{Hdr}, d_u, \mathcal{B}(u), \mathsf{param})$: This algorithm first parses $\mathsf{Hdr} = (\mathbb{A}, C_1, C_2, C_3)$. Then, it computes a partial session key K_k for each clause β_k in $\mathbb{A}$, $k \in [1, m]$. For that purpose, the user u chooses an attribute $A_i \in \big(\beta_k \cap \mathcal{B}(u)\big)$, retrieves the corresponding private decryption key $\tilde{d}_i$ and first computes

$$T_i = e(C_1 \cdot \prod_{j \in \beta_k} X_j, \tilde{d}_i \cdot \prod_{\substack{j \in \beta_k \\ j \neq i}} \tilde{d}_{u_{n+1-j+i}}).$$

The partial session key K_k is then computed as

$$K_k = \frac{e(C_2, \tilde{d}_{u_i})}{T_i \cdot \prod_{\substack{\ell=1 \\ \ell \neq k}}^{\ell=m} e(C_1 \cdot \prod_{j \in \beta_\ell} X_j, \tilde{d}_i \cdot \prod_{j \in \beta_\ell} \tilde{d}_{u_{n+1-j+i}})}.$$

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

$$K = \frac{e(\tilde{d}_{u_0}, C_3)}{\prod_{k=1}^{m} K_k}.$$

For the correctness: We first focus on the partial session key K_k. We use 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

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

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

Now focusing on the session key K, we have

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

which exactly corresponds to the key K generated at the encryption step.

Remark 1. 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 $\{x_1, \ldots, x_n\}$ 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 [26], we thus can separate the role of group manager (who generates the decryption keys) and broadcaster (who encrypts and broadcasts the content).

Remark 2. 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 [26]). To deal with this drawback, as in [29], 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 $\mathsf{professor}$ 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 3. The notion of attribute-based broadcast encryption (ABBE) has then been introduced in [22] 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 [19], 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 (non revoked) users. The users outside the set β_{m+1} (revoked users) cannot decrypt because it lacks the partial session key corresponding 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 [19], that is linear in the maximal number of users in the system.

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

3.3 Security

In this section, we first give a theorem to prove that our first scheme achieves *basic selective* security 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 in Sect. 2.1 the adversary first outputs 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) – 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 1. *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 $g^{P(\cdots)}, \tilde{g}^{Q(\cdots)}, g_T^{R(\cdots)}$), 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 $g^{P(\cdots)}, \tilde{g}^{Q(\cdots)}$. 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, $\mathcal{B}$ 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 1. *In the $(P, Q, R, f) - \mathsf{GDDHE}$ assumption above, (P, Q, R) and f are linearly independent.*

Proof. 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 Λ_C is the list of corrupted users. We will use β to analyze f, set $q_u = \alpha r(\beta + s_u), u \in \Lambda_C$, $P' = P\backslash\{q_u\}_{u\in\Lambda_C}$. We rewrite f as follows:

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

Consider f_1, we rewrite it as follows:

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

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 $\{u, v\} \subset \Lambda_C$, or we have $f_1 = 0$.

Consider $f_2 = \sum_{u \in \Lambda_C, p_i \in P'} a_{u,i} p_i q_u$, 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:

$$\begin{aligned} f_2 &= \sum_{u \in \Lambda_C, p_i \in P'_1} a_{u,i} p_i q_u + \sum_{u \in \Lambda_C, p_i \in P'_2} a_{u,i} p_i q_u \\ &= \sum_{u \in \Lambda_C, p_i \in P'_1} a_{u,i} p_i \alpha r(\beta + s_u) + \sum_{u \in \Lambda_C, p_i \in P'_2} a_{u,i} p_i q_u. \end{aligned}$$

We therefore obtain the equation

$$\begin{aligned} f &= \alpha^{n+1} r\beta(mt + \sum_{k=1}^{m} \sum_{j \in \beta_k} x_j) \\ &= \sum_{u \in \Lambda_C, p_i \in P'_1} a_{u,i} p_i \alpha r(\beta + s_u) + \sum_{u \in \Lambda_C, p_i \in P'_2} a_{u,i} p_i q_u + f_3 \end{aligned} \tag{1}$$

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

On the other hand, to make the term

$$f = \alpha^{n+1} r\beta(mt + \sum_{k=1}^{m} \sum_{j \in \beta_k} x_j)$$

appear, 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

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

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

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

Since each user $u \in \varLambda_C$ 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 Eq. (1) cannot hold or f is independent to (P, Q, R). □

4 Our Second Scheme

We now give the details of our second scheme, which aims at improving the first scheme 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, 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$(1^\lambda, \vartheta, \mathcal{B}(u_i)_{1 \le i \le \vartheta})$: Similar to the one in the first construction, except that the algorithm here uses an additional random oracle $\mathcal{H}$ on to $\mathbb{G}$ and $\tilde{h} = \tilde{g}^r$. The public parameters[1] are then

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

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 $\mathcal{B}(u) = (A_{i_1}, \ldots, A_{i_N}, A_n)$ be the set of attributes of user u. The private decryption key for u is

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

Encrypt$(\mathbb{A}, \mathsf{EK}, \mathsf{param})$: Assume that the access policy is expressed in CNF $\mathbb{A} = \beta_1 \wedge \beta_2 \wedge \cdots \wedge \beta_m$, 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 $t \overset{\$}{\leftarrow} \mathbb{Z}_p$, and then computes $Y_i = \mathcal{H}(i, h^t) = h^{y_i}$ for $i = 1, \ldots, m$ with unknown scalars y_i. The session key is then computed as:

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

[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 $\mathsf{param} = \{g, \tilde{g}, h, \tilde{h}, \mathcal{H}\}$ and re-compute all others values when encrypting.

Next, one computes:

$$C_1 = h^t, \quad \tilde{C}_1 = \tilde{h}^t,$$

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

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

The broadcaster can easily compute K and Hdr because it knows the values $r, \beta, \alpha, \gamma, g, \tilde{g}$ from EK. The header is set to $\mathsf{Hdr} = (\mathbb{A}, C_1, \tilde{C}_1, C_2, C_3, C_4)$, and the pair (Hdr, K) is the output.

Decrypt$(\mathsf{Hdr}, d_u, \mathcal{B}(u), \mathsf{param})$: The user u first parses the header Hdr as above: $(\mathbb{A}, C_1, \tilde{C}_1, C_2, C_3, C_4)$. It then checks whether the equations

$$e(C_1, \tilde{h}) = e(h, \tilde{C}_1) \text{ and } e(\mathcal{H}(C_1, C_2, C_3), \tilde{C}_1) = e(\tilde{h}, C_4)$$

hold. It then computes $Y_i = \mathcal{H}(i, C_1)$ for $i = 1, \ldots, m$. For each clause β_k in $\mathbb{A}$, the user u chooses an attribute $A_i \in \big(\beta_k \cap \mathcal{B}(u)\big)$ and computes, as in the previous scheme, for each $k \in [1, m]$:

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

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:

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

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 Sect. 3.3, except that the terms $rx_1, \ldots, rx_n$ are now replaced by the terms $ry_1, \ldots, ry_m, z, zt$. 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 revoked users, and the challenge header.

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

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

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

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 2. *Our second scheme is selective*$-$CCA *secure under* CDH *assumption and the* $(P, Q, R, f) -$ GDDHE *assumption above.*

Proof. Let $\mathsf{Hdr} = (\mathbb{A}, C_1, \tilde{C}_1, C_2, C_3, C_4)$ be the challenge header. Similar to the proof of MCBE_2 scheme, we will prove the security of $\mathsf{CP\text{-}ABBE}_2$ scheme in two steps. First, we prove that the adversary cannot produce any decryption query of the form $\mathsf{Hdr}' = (\mathbb{A}, C_1, \tilde{C}'_1, C'_2, C'_3, C'_4)$ under the CDH assumption. In the second step, we prove that our second scheme is selective$-$CCA secure under $(P, Q, R, f)-$GDDHE assumption with the requirement that the adversary doesn't ask any query $\mathsf{Hdr}' = (\mathbb{A}, C_1, \tilde{C}'_1, C'_2, C'_3, C'_4)$.

First step. This step is similar to the first step in the proof of MCBE_2 scheme, we thus refer the reader to the one in the proof of MCBE_2 scheme.

Second step. First, the simulator is given the instance of aforementioned $(P, Q, R, f) -$ GDDHE assumption. 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 $(P, Q, R, f) -$ GDDHE assumption. For that purpose, the simulator first receives the target access policy $\mathbb{A}$ from the adversary $\mathcal{A}$ as well as the repartition of attributes for each user, from the instance of $(P, Q, R, f) -$ GDDHE assumption 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 $\mathcal{H}(q) = h^y$, and appends the tuple (q, h^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 where $\beta'_\ell = A_n$. 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 $(q'_i, h^{y'_i}, y'_i)$. It takes the private decryption key of a user u and then computes:

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

$$C_1 = h^{t'}, \tilde{C}_1 = \tilde{h}^{t'}, C_2 = \prod_{k=1}^{k=\ell}(V \cdot \prod_{j\in\beta_k} h_{n+1-j})^{t'+y'_k},$$

$$C_3 = g_n^{t'\ell+\sum_{k=1}^{\ell} y'_k}, C_4 = g^{z't'}.$$

3. *Decryption query*: we assume that $\mathcal{A}$ sends the following ciphertext to the simulator (note that $t' \neq t$ since one cannot reuse the C_1 in the challenge header):

$$C_1 = h^{t'}, \tilde{C}_1 = \tilde{h}^{t'}, C_2 = \prod_{k=1}^{k=m'}(V \cdot \prod_{j\in\beta_k} h_{n+1-j})^{t'+y'_k},$$

$$C_3 = g_n^{m'.t'+\sum_{k=1}^{m'} y'_k}, C_4 = \mathcal{H}(C_1, C_2, C_3)^{t'}$$

The simulator first checks whether the equations $e(C_1, \tilde{h}) = e(h, \tilde{C}_1)$ and $e(\mathcal{H}(C_1, C_2, C_3), \tilde{C}_1) = e(\tilde{h}, C_4)$ hold, takes the private decryption key of a corrupted 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 $\mathcal{H}(i, h^t) = (q_i, h^{y_i}, \perp)$, 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 $\mathcal{A}$:

$$C_1 = h^t, \tilde{C}_1 = \tilde{h}^t, C_2 = \prod_{k=1}^{k=m} (V \cdot \prod_{j \in \beta_k} h_{n+1-j})^{t+y_k}, C_3 = g_n^{m.t+\sum_{k=1}^{m} y_k}, C_4 = g^{zt}.$$

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. □

The following lemma finally shows that in the aforementioned $(P, Q, R, f) - \mathsf{GDDHE}$ assumption, (P, Q, R) and f are linearly independent. The proof of this lemma is similar to the one given for Lemma 1 and, therefore, we do not repeat it again.

Lemma 2. *In the* $(P, Q, R, f) - \mathsf{GDDHE}$ *assumption above,* (P, Q, R) *and* f *are linearly independent.*

5 Conclusion

In this paper, we proposed two private CP-ABE schemes with 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.

Acknowledgement. This work is supported by the European Union SUPERCLOUD Project (H2020 Research and Innovation Program grant 643964 and Swiss Secretariat for Education, Research and Innovation contract 15.0091). It was partially conducted within the context of the Vietnamese Project Pervasive and Secure Information Service Infrastructure for Internet of Things based on Cloud Computing.

References

1. Abe, M., Groth, J., Ohkubo, M., Tango, T.: Converting cryptographic schemes from symmetric to asymmetric bilinear groups. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8616, pp. 241–260. Springer, Heidelberg (2014). doi:10.1007/978-3-662-44371-2_14
2. Agrawal, S., Chase, M.: A study of pair encodings: predicate encryption in prime order groups. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016. LNCS, vol. 9563, pp. 259–288. Springer, Heidelberg (2016). doi:10.1007/978-3-662-49099-0_10
3. Attrapadung, N.: Dual system encryption via doubly selective security: framework, fully secure functional encryption for regular languages, and more. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 557–577. Springer, Heidelberg (2014). doi:10.1007/978-3-642-55220-5_31

4. Attrapadung, N., Hanaoka, G., Yamada, S.: Conversions among several classes of predicate encryption and applications to ABE with various compactness tradeoffs. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9452, pp. 575–601. Springer, Heidelberg (2015). doi:10.1007/978-3-662-48797-6_24
5. Attrapadung, N., Libert, B., 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). doi:10.1007/978-3-642-19379-8_6
6. Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical identity based encryption with constant size ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005). doi:10.1007/11426639_26
7. 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). doi:10.1007/11535218_16
8. Chen, C., Chen, J., Lim, H.W., Zhang, Z., Feng, D., Ling, S., Wang, H.: Fully secure attribute-based systems with short ciphertexts/signatures and threshold access structures. In: Dawson, E. (ed.) CT-RSA 2013. LNCS, vol. 7779, pp. 50–67. Springer, Heidelberg (2013). doi:10.1007/978-3-642-36095-4_4
9. Chen, C., Zhang, Z., Feng, D.: Efficient ciphertext policy attribute-based encryption with constant-size ciphertext and constant computation-cost. In: Boyen, X., Chen, X. (eds.) ProvSec 2011. LNCS, vol. 6980, pp. 84–101. Springer, Heidelberg (2011). doi:10.1007/978-3-642-24316-5_8
10. Chen, J., Gay, R., Wee, H.: Improved dual system ABE in prime-order groups via predicate encodings. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9057, pp. 595–624. Springer, Heidelberg (2015). doi:10.1007/978-3-662-46803-6_20
11. Cheon, J.H., Han, K., Lee, C., Ryu, H., Stehle, D.: Cryptanalysis of the multilinear map over the integers. Cryptology ePrint Archive, Report 2014/906 (2014). http://eprint.iacr.org/2014/906
12. Emura, K., Miyaji, A., Nomura, A., Omote, K., Soshi, M.: A ciphertext-policy attribute-based encryption scheme with constant ciphertext length. In: Bao, F., Li, H., Wang, G. (eds.) ISPEC 2009. LNCS, vol. 5451, pp. 13–23. Springer, Heidelberg (2009). doi:10.1007/978-3-642-00843-6_2
13. Garg, S., Gentry, C., Halevi, S., Sahai, A., Waters, B.: Attribute-based encryption for circuits from multilinear maps. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8043, pp. 479–499. Springer, Heidelberg (2013). doi:10.1007/978-3-642-40084-1_27
14. Ge, A., Zhang, R., Chen, C., Ma, C., Zhang, Z.: Threshold ciphertext policy attribute-based encryption with constant size ciphertexts. In: Susilo, W., Mu, Y., Seberry, J. (eds.) ACISP 2012. LNCS, vol. 7372, pp. 336–349. Springer, Heidelberg (2012). doi:10.1007/978-3-642-31448-3_25
15. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Juels, A., Wright, R.N., Vimercati, S. (eds.) ACM CCS 2006: 13th Conference on Computer and Communications Security, pp. 89–98, Alexandria, Virginia, USA, 30 Oct - 3 Nov 2006. ACM Press (2011). Available as Cryptology ePrint Archive Report 2006/309
16. Hamburg, M.: Spatial encryption. Cryptology ePrint Archive: Report 2011/389 (2011)

17. Herranz, J., Laguillaumie, F., Ràfols, C.: Constant size ciphertexts in threshold attribute-based encryption. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 19–34. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13013-7_2
18. Hu, Y., Jia, H.: Cryptanalysis of GGH map. Cryptology ePrint Archive: Report 2015/301 (2014). http://eprint.iacr.org/2015/301
19. 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. ACM Press (2010)
20. Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 62–91. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13190-5_4
21. Lewko, A., Waters, B.: Unbounded HIBE and attribute-based encryption. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 547–567. Springer, Heidelberg (2011). doi:10.1007/978-3-642-20465-4_30
22. 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). doi:10.1007/978-3-540-68164-9_22
23. Miles, E., Sahai, A., Zhandry, M.: Annihilation attacks for multilinear maps: cryptanalysis of indistinguishability obfuscation over GGH13. In: Crypto 2016 (2016, to appear). https://eprint.iacr.org/2016/147
24. Okamoto, T., Takashima, K.: Fully secure unbounded inner-product and attribute-based encryption. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 349–366. Springer, Heidelberg (2012). doi:10.1007/978-3-642-34961-4_22
25. Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: Ning, P., di Vimercati, S.D.C., Syverson, P.F. (eds.) 14th Conference on Computer and Communications Security, ACM CCS 2007, pp. 195–203, Alexandria, Virginia, USA, 28–31 October 2007. ACM Press (2011)
26. Phan, D.H., Pointcheval, D., Trinh, V.C.: Multi-channel broadcast encryption. In: Proceedings of the 8th ACM Symposium on InformAtion, Computer and Communications Security (ASIACCS 2013). ACM Press (2013)
27. Rouselakis, Y., Waters, B.: Practical constructions, new proof methods for large universe attribute-based encryption. In: Sadeghi, A.-R., Gligor, V.D., Yung, M. (eds.) 20th Conference on Computer and Communications Security, ACM CCS 2013, pp. 463–474, Berlin, Germany, 4–8 November 2013. ACM Press (2011)
28. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005). doi:10.1007/11426639_27
29. 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). doi:10.1007/978-3-642-19379-8_4
30. Wee, H.: Dual system encryption via predicate encodings. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 616–637. Springer, Heidelberg (2014). doi:10.1007/978-3-642-54242-8_26
31. Yamada, S., Attrapadung, N., Hanaoka, G., Kunihiro, N.: A framework and compact constructions for non-monotonic attribute-based encryption. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 275–292. Springer, Heidelberg (2014). doi:10.1007/978-3-642-54631-0_16