SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 2014; **7:**1988–2002
Published online 15 November 2013 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.913

RESEARCH ARTICLE

Fully secure ciphertext policy attribute-based encryption with constant length ciphertext and faster decryption

Nishant Doshi* and Devesh C. Jinwala

Sardar Vallabhbhai National Institute of Technology, Surat, India

ABSTRACT

In PKC 2010, Herranz *et al.* proposed the first fully threshold ciphertext policy attribute-based encryption (CP-ABE) scheme with constant length ciphertext. However, their scheme is selectively secure with respect to the chosen plaintext attack. They have left three open problems for CP-ABE with constant ciphertext length, that is, Security against the Chosen Ciphertext Attacks, Security Reduction to a better mathematical problem and to make the scheme Fully Secure. Indeed, in ACISP 2012, Ge *et al.* proposed the solutions to the first two problems but left their proposed scheme selective secure. This makes their scheme weaker because it is secure only for a particular policy. With an aim to propose a fully secure constant ciphertext length CP-ABE scheme, in this paper, we discuss our attempts at extending the approach of Lewko *et al.* (in EUROCRYPT 2010). The scheme that we propose here allows any subset of attributes of the secret key as a part of the ciphertext policy. Copyright © 2013 John Wiley & Sons, Ltd.

KEYWORDS

attribute-based encryption; constant length ciphertext; chosen plaintext security; fully secure

***Correspondence**

Nishant Doshi, Sardar Vallabhbhai National Institute of Technology, Surat, India.
E-mail: doshinikki2004@gmail.com

1. INTRODUCTION

In distributed file systems, complex access-control mechanisms are commonly required to protect various resources. There have been various mechanisms devised that use either cryptographic or non-cryptographic approaches for access-control. Among the cryptographic approaches, one of the popular approaches exploits the association of the identity of a user with the resource to be protected, enabling mere identity checks to allow/reject further access to the resource. However, a user's identity can be described only by defining appropriate associated attributes. Hence, the notion of using a user's identity as the basis of a key, that is, identity-based encryption [1], led eventually to the notion of the attribute-based encryption (ABE) [2]. The use of ABE also enabled its use in a multicast setup that its predecessors viz. the public key cryptosystems [3] and the identity-based encryption systems lacked.

In [4], the authors proposed the first ABE scheme in two different variants viz. *key policy* attribute-based encryption (KP-ABE) and *ciphertext policy* attribute-based encryption (CP-ABE). In KP-ABE, a ciphertext is associated with a defined set of attributes, and user's secret key is associated with a defined policy containing those attributes. Hence, the secret key could be used successfully, only if the attributes in an access policy (aka. Access structure) defined in the key match the attributes in the ciphertext. In [5], the authors proposed a fully functional CP-ABE in which a user's secret key is associated with a defined set of attributes and the ciphertext is associated with a defined policy.

Subsequently, there have been various efforts made to improve the basic CP-ABE scheme too [6–17]. However, all these approaches essentially use *variable* length ciphertext, that is, the size of ciphertext in all these approaches increases linearly w.r.t. the number of attributes in the policy. This not only increases the *communication overhead* but also increases the *computational overhead* during *Decrypt* operation, on the receiver side because of the expensive pairing operations to be carried out there. In order to achieve *faster decryption*, (in terms of execution time), constant length of the ciphertext is to be ensured, giving fixed number of pairing operations. This reduces the execution time of *Decrypt* and makes it faster as compared to *variable length* approaches.

Upon our search for the approaches that offer *constant length* ciphertext, we observe that indeed, an extensive

set of attempts can be found in the literature that support constant length ciphertext-based systems [17–27]. However, we also observe that all of these are only selectively secure [17–23] [27,28]. On the other hand, numerous approaches that offer full security suffer from variable length ciphertext [29–32].

Because of the advantages associated, our primary observation is that it is desirable to ensure not only *constant length ciphertext* in a CP-ABE scheme but also full security of the system. We indeed found one such scheme due to Ren *et al.* in [33], which is the first *fully secure* CP-ABE scheme with *constant length* ciphertext. This scheme is (s, s) threshold scheme. This means that the number of attributes in a receiver's (decryptor) secret key and that in the ciphertext policy must contain *the same s* attributes for successful decryption.

However, we observe that this constraint restricts the application of the scheme to some specific setups. For example, consider a multicast scenario wherein each user in a group has $< p_i, x_1, x_2, x_3, \ldots x_n >$ attributes with the subset $< x_1, x_2, x_3, \ldots x_n >$ being common across all the users, whereas the attribute p_i serves as a distinguisher among the elements in the group. Then, because of the restriction of being a (s, s) threshold scheme, the scheme [33] would fail (having no scope for supporting the distinguisher element).

In real life, a user by default is expected to play different roles, for example, being a student and a bank account holder. Consider a case where Alice, say, is a student of university ABC and the account holder for Bank XYZ. Messages (ciphertext) from ABC entitled to students do not contain attributes for XYZ (as student may have account in any bank) and vice versa. However, according to Ren *et al.* [33], Alice would able to decrypt only if the policy of received ciphertext contains the attributes ABC *and* XYZ. Therefore, even though Alice is a valid user, she would not be able to decrypt individual messages from ABC and/or XYZ. Thus, the scheme in [33] would not be applicable in *all* scenarios in real life. This can be ameliorated if a *subset of the attributes* in the user's secret key *is sufficient* (to be used as a threshold) for establishing the match with the ciphertext policy.

In this backdrop, we propose a fully secure ABE scheme with *a constant length ciphertext* that can be used in a multicast setup also. Our scheme also supports *collusion resistance* using a *single authority* approach. In addition, the proposed approach uses a fixed number of pairing operations during the decryption, irrespective of the number of attributes in the policy. We consider the attributes that contain many values (aka *multivalued*), and during encryption, we multiply them (ANDing) to make the constant ciphertext. Therefore, our proposed scheme is based on AND gate with multivalued attributes.

To the best of our knowledge, ours is the first scheme that supports *fully secure* ABE scheme with *constant* ciphertext length for multicast setup.

The rest of the paper is organized as follows: in Section 2, we present a detailed discussion of various approaches in CP-ABE. In Section 3, we discuss the notations, composite bilinear groups, the basic *three* assumptions for the security and the proposed construction of the scheme. In Section 4, we discuss the proposed approach for constant length ciphertext. In Section 5, we give the full security proof based on the *three* assumptions (as in Section 3). The performance analysis is given in Section 6, whereas the conclusion with the scope for the future work is given in Section 7.

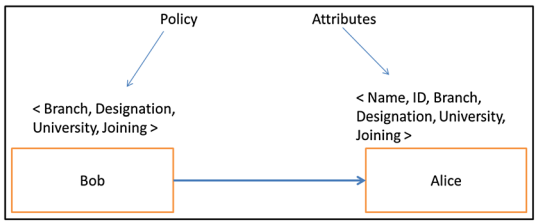

Figure 1. Schematic for the example.

2. RELATED WORK

To exemplify the need for constant length ciphertext as well as full security, we take the help of an illustration in our further discussion that goes as follows:

Say a user *Bob* wants to encrypt some message (like invitation) to all fresh PhD students in the Computer Department of University *X*. Assume one of the users in this system is Alice with the attributes viz. *{"Name=Alice", "ID=D12CO951", "Branch=Computer", "Designation=PhDStudent", "University=X", "Joining=2012"}*. If Bob wishes to address all the fresh PhD students as mentioned before, he has to use the policy viz. *{"Branch=Computer" AND "Designation=PhDStudent" AND "University=X" AND "Joining=2012"}* as shown in Figure 1.

2.1. Constant length ciphertext

The motivation for the constant length ciphertext can best be understood by looking at the limitation of the scheme [5] in which the size of the ciphertext increases linearly with the number of attributes in the policy. We attempt to do so by using an example and the CP-ABE toolkit [34] that provides a set of programs implementing a ciphertext policy attribute-based encryption scheme. We input a sample message text of the size 369 KB by using the CP-ABE scheme as in [5] and obtain the various operational parameter values as shown in Table I using the following notations:

P denotes the number of Pairing operations, E denotes the number of Exponents used, M denotes the number of multiplication operations required at decryptor side for a (t, n) threshold scheme (t or more attributes are required out of total n attributes to decrypt),and S denotes the size of the ciphertext in KB.

Table I. Analyzing the CP-ABE scheme in [5] for different values of $< t, n >$.

n	$t = 1$	2	3	4	5
4	P=27 E=13 M=149 S=382.6	P=29 E=14 M=157 S=382.2	P=31 E=15 M=169 S=381.9	**P=33 E=16 M=307 S=381.5**	
5	P=27 E=13 M=149 S=385	P=29 E=14 M=157 S=384.7	P=31 E=15 M=169 S=384.4	P=33 E=16 M=185 S=384	P=35 E=17 M=355 S=383.7

Table II. A comparison based on number of policies supported.

Scheme	Policies that can be decrypted by Alice	Possible number of policies
[33]	{"Name = Alice", "ID = D12CO951", "Branch = Computer", "Designation = PhDStudent", "University = X", "Joining = 2012"}	1
This work	{"Name = Alice"},{"ID = D12CO951"}, ...,{"Name = Alice" , "ID = D12CO951", "Branch = Computer", "Designation = PhDStudent", "University = X", "Joining = 2012"$	$2^6 - 1$

Thus, for the example illustrated earlier, with four attributes in a policy, Alice requires 33 pairing operations (t=4, n=4 in Table I). In addition, the number of pairing operations increases during the decryption that increases the computational overhead on the receiver [21].

To achieve faster decryption using fixed number of pairing operations, in [19], the authors propose a constant length ciphertext by using the (t, t) threshold system, that is, the number of attributes in user's secret key is the same as the number of attributes in the policy. In [20–28], the authors propose different schemes to deal with the constant length ciphertext.

However, none of these approaches achieve fully secure scheme – the motivation for which is discussed in the following section.

2.2. Fully secure CP-ABE schemes

All of the approaches discussed in previous sections provide the selective security. In the selective security model, attacker announces the target access structure (policy) before the Setup phase, which makes the selective security model a weaker model as compared to fully secure [30,33]. More precisely, in selective secure schemes, simulator knows the target access structure before he generates the public parameters. However, the simulator can allow to generate public and to be able to generate private parameters except attributes in target access structure. Therefore, when the simulator generates public parameters, the underlying hard problem is embedded into the secret parameters for target access structure. This makes the selective secure scheme to work for specific access structure and lead to a weaker model. In fully secure model, attacker announces the target access structure after seeing the public parameters that make the scheme secure against any access structure.

The notion of the fully secure model started with [35]. In [35], the authors propose a dual system encryption method in which keys and ciphertext can be of two forms , that is, normal and semi-functional. Semi-functional keys can decrypt only normal ciphertext. However, normal keys can be used to decrypt both types of ciphertext. Semi-functional key and semi-functional ciphertext is used only for proof of the security.

In [29], the authors propose a fully secure CP-ABE scheme based on the dual encryption method [35,36]. In [30–32], the authors propose different schemes to deal with the fully secure notion.

However, in these schemes, if system contains the N attributes, then decryption procedure requires N pairing operations, which makes the system inefficient for larger attribute sets.

2.3. Fully secure CP-ABE with the constant length ciphertext scheme

The first fully secure CP-ABE scheme with constant length ciphertext was proposed in [33]. This scheme is based on the approach proposed in [19] discussed earlier. However, as mentioned earlier, this scheme requires identical and an equal number of attributes in the ciphertext policy as well as in the secret key for successful decryption. In other words, out of all subsets of attribute set s in (s, s) scheme, only one subset, that is, set itself, must be matched with the attribute set of user's secret key (that also has the same set of attributes).

We further illustrate the advantage derived in our scheme in Table II by comparing our approach with the one in [33] in terms of the number of ciphertext policies. It can be perceived from Table II that our approach allows a larger number of possibilities for selecting the ciphertext policy for the sender (encryptor). Therefore, different/same sender can use different policies that suit their requirements.

Specifically, with respect to the illustration that we have discussed earlier, the approach in [33] fails because Alice would not be able to decrypt the ciphertext because of Name and ID attributes in her secret key (the *(s, s)* threshold not being satisfied). However, in our scheme, Alice would be able to do so, as our approach allows the attributes in the ciphertext policy to be the subset of the secret key.

In [37], the authors proposed an approach using *inner-product encryption* (IPE) to achieve the constant length ciphertext with fully secure notion. However, the size of the secret key in [37] is more than double the size of the same, in our scheme. In addition, if we consider a system with n attributes in the universe, then the *KeyGen*, *Encrypt*

DOI: 10.1002/sec

and *Decrypt* of [37] require n attributes to calculate the vector of size *greater* than n, even though fewer attributes are used in the ciphertext or the secret key.

The entire evolution in the research from the basic CP-ABE scheme to that with the support for a CP-ABE scheme with the support for multi-authority, constant length ciphertext, selective secure or fully secure albeit individually has been summarized for a quick reference in Table III. In Table III, fully secure model refers to the dual encryption technique by [36], and IPE model refers to the underlying assumption of the IPE scheme as mentioned in [37].

3. PRELIMINARIES

3.1. Composite order bilinear groups

Definition 1. *(Bilinear map)* [38]. *Assume G and G1 are two multiplicative cyclic group of some order $N = pqr$ where p, q and r are distinct large prime numbers. $Z_N$is set of all positive numbers. G_p, G_q and G_q are subgroup of order p, q and r, respectively. G_{pq} is the subgroup of order pq and so on. Here, G_{pq} can be defined as the $G_{pq} = \{ ab \mid a \in G_p,\ b \in G_q\}$. Here, $G = G_pG_qG_r$. Assume g_1 is a generator of G. A bilinear map $e : G \times G \rightarrow G1$ is a deterministic function, which takes as input two elements from G and output an element in group G1, which satisfies the following criteria:*

- *Bi-linearity : For all $a, b \in Z_N$, $e\left(g_1^a, g_1^b\right) = e(g_1, g_1)^{ab}$.*
- *Non-degeneracy: $e(g_1, g_1) \neq 1$.*
- *e must be computed efficiently.*
- *For $g_p \in G_p, g_r \in G_r$ and $g_q \in G_q$, $e\left(g_p, g_r\right) = e\left(g_p, g_q\right) = e\left(g_q, g_r\right) = 1$. To see this, assume that $h_1 \in G_p$ and $h_2 \in G_q$. Here, g_1^{pq} generates G_r, g_1^{pr} generates G_q and g_1^{rq} generates G_p. Hence, for some a_1, a_2, $h_1 = \left(g_1^{qr}\right)^{a_1}$ and $h_2 = \left(g_1^{pr}\right)^{a_2}$. Then $e(h_1, h_2) = e\left(g_1^{qra_1}, g_1^{pra_2}\right) = e\left(g_1^{ra_1}, g_1^{a_2}\right)^{pqr} = 1$. We will use this orthogonal property to implement semi-functionality of the proposed system.*

Definition 2. *(Access Structure/Policy).*[5] *Let (A1, A2,…,An) be a set of attributes. The collection $A \subseteq 2^{\{A1,A2,\ldots,An\}}$ is monotone if $\forall B,C$: if $B \in A$ and $B \subseteq C$, then $C \in A$. An (monotone) access structure is a (monotone) collection A of non-empty subsets of (A1,A2,…,An), that is, $A \subseteq 2^{\{A1,A2,\ldots,An\}} \backslash \{\phi\}$. The sets in A are called authorized, and the sets that are not in A are called unauthorized sets. We have used monotonic access structure in the proposed scheme that support AND gate.*

3.2. Complexity assumptions [35 – 36]

In the following assumptions, we assume that G_{pq} denotes subgroup of order pq in G, G_{pr} denotes subgroup of order pr in G and so on. All these assumptions are of constant length and secure as given in [35].

Assumption 1. *(Subgroup decision problem for three primes). Given a group generator X, we define the distribution as in Equation (1).*

$$\begin{aligned} &(N = pqr, G, G_1, e) \longleftarrow X, \\ &g \in_R G_p, X_3 \in_R G_r, D = (N, G, G_1, e, g, X_3), \\ &T_1 \in_R G_{pq}, T_2 \in_R G_p \end{aligned} \tag{1}$$

We define the advantage of an algorithm A in breaking Assumption 1 to be

$$Adv1_{G,A}(\lambda) := Pr[A(D, T_1) = 1] - Pr[A(D, T_2) = 1].$$

We note that T_1 can be written (uniquely) as the product of an element of G_p and an element of G_q. We refer to these elements as the "$G_p$partofT_1" and the "$G_q$partofT_1", respectively. We use this terminology in our proofs.

Definition 3. *We say that G satisfies Assumption 1 if $Adv1_{G,A}(\lambda)$ is a negligible function of λ for any polynomial time algorithm A.*

Assumption 2. *Given a group generator X, we define the distribution as in Equation (2).*

$$\begin{aligned} &(N = pqr, G, G_1, e) \longleftarrow X, \\ &g, X_1 \in_R G_p, X_2, Y_2 \in_R G_q, \\ &X_3, Y_3 \in_R G_r, D = (N, G, G_1, e, g, X_1X_2, X_3, Y_2Y_3), \\ &T_1 \in_R G, T_2 \in_R G_{pr} \end{aligned} \tag{2}$$

We define the advantage of an algorithm A in breaking Assumption 2 to be

$$Adv2_{G,A}(\lambda) := Pr[A(D, T_1) = 1] - Pr[A(D, T_2) = 1].$$

We use G_{pr} to denote the subgroup of order pr in G. We note that T_1 can be (uniquely) written as the product of an element of G_p, an element of G_q and an element of G_r. We refer to these as the "$G_p$partofT_1", the "$G_q$partofT_1" and the "$G_r$partofT_1", respectively. T_2 can similarly be written as the product of an element of G_p and an element of G_r. Note : It is emphasized that Assumption 2 is violated if X_2 is known. However, in our case, the parameter D contains the value of X_2, and there is no way by which an adversary can known X_2, given D.

Definition 4. *We say that G satisfies Assumption 2 if $Adv2_{G,A}(\lambda)$ is a negligible function of λ for any polynomial time algorithm A.*

Assumption 3. *Given a group generator X, we define the distribution as in Equation (3).*

$$\begin{aligned} &(N = pqr, G, G_1, e) \longleftarrow X, \\ &y, s \in_R Z_N, \\ &g \in_R G_p, X_2, Y_2, Z_2 \in_R G_q, \\ &X_3 \in_R G_r, D = (N, G, G_1, e, g, g^yX_2, X_3, g^sY_2, Z_2), \\ &T_1 = e(g, g)^{ys}, T_2 \in_R G_1 \end{aligned} \tag{3}$$

Table III. Comparison of different schemes.

Scheme	Expressiveness of policy	Complexity assumption	Adding of attributes after setup	Constant length ciphertext	Fully secure
[5]	Any Boolean formula	Generic group	Yes	No	No
[7]	Any Boolean Formula	BDH	Yes	No	No
[8]	Any Boolean Formula	Fully secure model	Yes	No	Yes
[9]	Any Boolean Formula	DBDH	Yes	No	No
[10, 11]	Disjunctive Normal Form	Generic Group	Yes	No	No
[12]	Any Boolean Formula	DBDH	No	No	No
[16]	Any Boolean Formula	PBDHE	Yes	No	No
[19]	AND gate multivalued	DBDH	No	Yes	No
[20]	Any Boolean Formula	aMSE-DDH	No	Yes	No
[21]	AND gate multivalued	K-BDHE	Yes	Yes	No
[22]	Any Boolean Formula	q-DBDHE and aMSE-DDH	Yes	Yes	No
[23]	AND gate multivalued	K-BDHE	Yes	Yes	No
[29]	Any Boolean Formula	Fully secure model	No	No	Yes
[30]	AND gate multivalued	Fully secure model	No	No	Yes
[31]	Any Boolean Formula	Fully secure model	Yes	No	Yes
[33]	AND gate multivalued	ABDHE	Yes	Yes	Yes
[37]	Any boolean formula	IPE	Yes	Yes	Yes
[32]	AND gate multivalued	Fully secure model	Yes	No	Yes
[25]	AND gate multivalued	DBDH	Yes	Yes	No
[26]	AND gate with positive, negative and wildcard attribute	DBDH	Yes	Yes	No
[27]	Any Boolean Formula	q-BDHE	Yes	Yes	No
[28]	Any Boolean Formula	aMSE-DDH	Yes	Yes	No
Our scheme	AND gate with multivalued	Fully Secure Model	Yes	Yes	Yes

Security Comm. Networks 2014; **7**:1988–2002 © 2013 John Wiley & Sons, Ltd.
DOI: 10.1002/sec

We define the advantage of an algorithm A in breaking Assumption 3 to be

$$Adv3_{G,A}(\lambda) := Pr[A(D, T_1) = 1] - Pr[A(D, T_2) = 1]$$

Definition 5. *We say that G satisfies Assumption 3 if* $Adv3_{G,A}(\lambda)$ *is a negligible function of* λ *for any polynomial time algorithm A.*

3.3. Proposed construction

The proposed scheme consists of four polynomial algorithms as follows.

(1) Setup(1^γ): Central Authority (CA) runs this algorithm. It will take implicit security parameter γ, that is, Bit security level as input and output public parameter MPK and master key MSK.
(2) Keygen (MSK, L): The key generation algorithm run by CA takes as input the master key of the CA and the set of attributes list L for a user and then generates the secret key SK.
(3) Encrypt (MPK, M, W): The encryption algorithm takes as input the message M, public parameter MPK and access policy W over the universe of attributes. It generates the output CT such that only those users who have a valid set of attributes that satisfy the access policy can decrypt. Assume that the CT implicitly contains access policy W.
(4) Decrypt (CT, SK) : The decryption algorithm run by the user takes as input the ciphertext CT containing access policy W and the secret key SK containing attribute set L. If L satisfies the access policy, then algorithm decrypts the CT and gives M, otherwise decryption fails.

3.4. Security game

(1) Setup: The challenger runs Setup and gives public parameters to the attacker *A*.
(2) Phase 1: The attacker *A* sends a query for attribute sets $S_1, S_2, ..., S_{k'}$. The challenger answers with a secret key. Note that these queries can be repeated adaptively.
(3) Challenge: The attacker *A* sends two equal-length messages M_0 and M_1 and challenges access structure W^* (such that $\forall i \in [1, k'], S_i \not\models W^*$) to the challenger. The challenger selects $\mu \in_R \{0, 1\}$ and runs $CT^* = Encrypt(MPK, M_\mu, W^*)$. The challenger gives the ciphertext CT* to A.
(4) Phase 2: Phase 1 is repeated with a query for attribute sets $S_{k'+1}, S_{k'+2}, ..., S_{q'}$ (such that $\forall i \in [k' + 1, q'], S_i \not\models W^*$). The challenger answers with a secret key.
(5) Guess: A outputs a guess $\mu' \in \{0, 1\}$.

The advantage of A in breaking the Indistinguishability under selective chosen plaintext attacks (IND-sCPA) security of the scheme is defined as

$$Adv(A) := |Pr(\mu' = \mu) - 1/2|$$

The ABE scheme is said to be IND-sCPA secure if Adv(A) is negligible (say ϵ) with respect to security parameter for any polynomial time adversary.

As we can see, A announces the target access structure after seeing the public parameters, which formally proves the fully secure notion of the proposed scheme.

4. THE PROPOSED SCHEME

The proposed scheme consists of four algorithms.

(1) Setup (γ): This algorithm is run by the CA. Based on the security parameter γ, the CA takes the bilinear group G and G_1 of order N, where $G = G_p G_q G_r$ and $N = pqr$. The bilinear map function is $e : G \times G \to G_1$. Then, the CA chooses the set of attributes in universe $U = \{attr_1, attr_2, \ldots, attr_n\}$. Each attribute contains n_i possible values. $v_{i,j}$ represents the j^{th} value of attribute i. We have used the double index, that is, one for attribute and the other for its possible value for better readability with real life. In the example of Section 2, attribute *Branch* has different values like Computer, Electrical and Mechanical. The MPK and MSK are given in Equation (4).

$$g \in G_p, X_3 \in_R G_r, \{t_{i,j} \in_R Z_N\}_1 \le i \le n, 1 \le j \le n_i, y, \beta \in_R Z_N$$

$$\begin{aligned} MPK = \Big\{ & G_r, N, e, g, g^\beta, Y = e(g,g)^y, \\ & T_{i,j} = g^{t_{i,j}} \, (i \in [1,n], j \in [1, n_i]) \Big\} \\ MSK = \{ & y, X_3, \beta \} \end{aligned} \tag{4}$$

(2) KeyGen *(MSK,L,MPK) :* Based on MSK, MPK and attribute list $L = \left\{ v_{1,j_1}, v_{2,j_2}, \ldots, v_{n',j_{n'}} \right\}$ where $n' \le n$, CA generates $t \in_R Z_N, R_0, R'_0, R_{i,j} \in_R G_r$ and calculates the SK of user u as in Equation (5).

$$\begin{aligned} SK_L = \Big\{ & \forall v_{i,j_i} \in L \;\; D_{i,j} = \left(T_{i,j} R_{i,j}\right)^t, \\ & D = (g R_0)^{y+\beta t}, D' = (g R'_0)^t, L \Big\} \end{aligned} \tag{5}$$

(3) Encrypt (MPK,M,W) : Based on the MPK, message M and access policy W, the sender selects $s \in_R Z_N$ and calculates ciphertext CT as in Equation (6).

$$\begin{aligned} C_1 &= M Y^s \\ C_2 &= g^{\beta s} \left(\prod_{v_{i,j} \in W} T_{i,j}{}^s \right) = \left(g^s\right)^\beta \left(g^s\right)^{\sum_{v_{i,j} \in W} t_{i,j}} \\ C_3 &= g^s \\ CT &= \{ C_1, C_2, C_3, W \} \end{aligned} \tag{6}$$

(4) Decrypt (CT, SK_L): User (Receiver) has to identify the AS (such that $(AS \subseteq L) \bigwedge (AS \models W)$) and then has to multiply all the related values, which are given in the secret key, that is, $\prod_{v_{i,j} \in AS} D_{i,j}$. Finally, compute as the following.

$$\frac{C_1\, e(\, C_2, D')}{e\left(C_3, D\, \prod_{v_{i,j} \in AS} D_{i,j}\right)}$$

$$= \frac{M\, e(g,g)^{y\, s}\, e(g^{\beta s}\left(\prod_{v_{i,j} \in W} T_{i,j}{}^{s}\right), g^{t} R_0')}{e\left(g^{s}, g^{y+\beta t} \prod_{v_{i,j} \in AS} \left(g^{t}\right)^{t_{i,j}} R_{i,j}\right)}$$

$$= \frac{M\, e(g,g)^{y\, s} e(g,g)^{ts\beta}\, e(g,g)^{t\, s\, p_1}}{e(g,g)^{y\, s} e(g,g)^{ts\beta}\, e(g,g)^{t\, s\, p_2}}$$

$$= M$$

Here, $p_1 = \sum_{v_{i,j} \in W} t_{i,j}$ and $p_2 = \sum_{v_{i,j} \in AS} t_{i,j}$

As we can see that the size of MPK is $O(n)$, the size of secret key is $O(n')$ and the size of ciphertext is $O(1)$.

Note : $\not\models$ refers to not satisfy symbol, $\models$ refers to satisfy symbol.

4.1. Construction of the secret keys $t_{i,j}$

Here, we assume that $\sum v_{i,j \in L_1} t_{i,j} \neq \sum v_{i,j \in L_1'} t_{i,j}$. If there exists $L_1 \subseteq L$ and $L_1' \subseteq L'$ such that $\sum v_{i,j \in L_1} t_{i,j} = \sum v_{i,j \in L_1'} t_{i,j}$, then L' can decrypt W, where $L' \not\models W$ and $L \models W$. This assumption holds with probability $\frac{N(N-1)\ldots(N-(N-1))}{N^{\delta}} > \frac{(N-(N-1))^{\delta}}{N^{\delta}} = \left(1 - \frac{\delta - 1}{N}\right)^{\delta} > \left(1 - \frac{\delta(\delta-1)}{N}\right) > \left(1 - \frac{\delta^2}{N}\right)$, where $\delta = \prod_{i=1}^{n} n_i$ and N is the group order of G. Therefore, if we select each $t_{i,j}$ as random form Z_N, then our assumption is natural.

5. SECURITY ANALYSIS

In this section, we discuss the full security model based on Assumptions 1, 2 and 3. First, we look at the semi-functional keys and semi-functional ciphertext that we will subsequently use in the proof. Thereafter, we discuss the sequence of hybrid games that we will use to prove the full security of the system. The normal operation of our scheme essentially occurs in G_p. The secret keys are additionally randomized in the subgroup of G_r, while the subgroup of G_q is used for semi-functional space (which is not used in the real system). When a normal key is paired with semi-functional ciphertext or a semi-functional key is paired with normal ciphertext, then the element from G_q will not contribute in pairing as it is orthogonal to G_p and G_r. When we decrypt semi-functional ciphertext with semi-functional key, the extra term from G_q will arise, which makes the decryption fail. When the decryption works, we call it *nominally* semi-functional secret key (say type 1). In that, the G_q term of *nominally* semi-functional key cancels the G_q term of semi-functional ciphertext when paired. If decryption fails, then that key is called the semi-functional secret key of type 2.

5.1. Construction of the semi-functional ciphertext

Let g_q denote the generator of G_q. A semi-functional ciphertext is created as given in the following.

- First, the normal ciphertext is generated based on *Encrypt* algorithm

$$CT' = \left\{C_1', C_2', C_3', W\right\}.$$

- It selects random exponent $c, Z_{i,j} \in_R Z_N$.
- The semi-functional ciphertext is as in Equation (7).

$$CT = \left\{ C_1 = C_1', C_3 = \left(g_q^{c}\right)^{\beta} C_3', \quad C_2 = C_2' \left(\prod_{v_{i,j} \in W} \left(g_q^{c}\right)^{Z_{i,j}} \right), W \right\} \tag{7}$$

5.2. Construction of the semi-functional secret key

A semi-functional secret key can take any of two types as follows:

Type 1: It generates exponents t, $b \in_R Z_N$, R_0, R_0', $R_{i,j} \in_R G_r$

$$SK : \left\{\forall v_{i,j_i} \in L\, D_{i,j} = \left(T_{i,j} R_{i,j}\right)^{t} g_q^{Z_{i,j}}, D = (gR_0)^{y+\beta t} g_q^{b}, \quad D' = (gR_0')^{t} g_q^{\beta}, L\right\}.$$

Type 2: It can generate from type-1 by removing g_q^{β} and $g_q^{Z_{i,j}}$

$$\left\{SK : D = (gR_0)^{y+\beta t} g_q^{\beta}, \forall v_{i,j_i} \in L\ D_{i,j} = \left(T_{i,j} R_{i,j}\right)^{t}, \quad D' = \left(gR_0'\right)^{t}, L\right\}.$$

Here, when we decrypt the semi-functional ciphertext with type 1, then it is successfully decrypted. As compared with type 2, the decryption fails because of the term $e\left(g_q, g_q\right)^{cb\beta}$ being left in the denominator.

5.3. Security Proof

We have proven the security based on the sequence of hybrid games. We will use the same structure of hybrid games and lemmas as in [29]. The $\mathbf{Game_{Real}}(\mathbf{Game_0})$ is the real security game, that is, all the ciphertext and secret keys are in normal form. Let q' denote the total number

of secret key queries (during *phase 1* and *phase 2* in the security game of Section 3.4).

Then, for k from 1 to q'.

In $\mathbf{Game_k}, \mathbf{1}$, the first $k-1$ keys are semi-functional of type 2, the k^{th} key is semi-functional of *type 1* and the remaining keys are normal. The challenge ciphertext is semi-functional.

In $\mathbf{Game_k}, \mathbf{2}$, the first k keys are semi-functional of *type 2* and the remaining keys are normal. The challenge ciphertext is semi-functional.

In the $\mathbf{Game_q'}, \mathbf{2}$, all the keys are semi-functional of *type 2*. In $\mathbf{Game_{Final}}$, all the keys are semi-functional of *type 2*, while challenge ciphertext is semi-functional with encryption of a random message, that is, independent of two messages provided by attacker. We show that in $Game_{Final}$, the advantage of the attacker is negligible based on the games that are indistinguishable as we show later in four lemmas. For notational purposes, we have used $Game_{0,2}$ as another way of denoting $Game_0$. In lemmas, a simulator generates the parameters based on Assumptions 1,2 and 3, and gives it to the *Algorithm B*, which will use the output of *Algorithm A* (attacker) to break the underlying assumptions with negligible advantage ϵ.

One can see that we start with real game that contains a normal key and normal ciphertext, and then in each subsequent game, we change one key from normal to semi-functional with negligible advantage for A. In the final game, all the keys are semi-functional with semi-functional ciphertext.

Lemma 1. *If there exists a polynomial time algorithm A with $Game_{Real}Adv_A - Game_0Adv_A = \epsilon$, then we can construct algorithm B in polynomial time with advantage ϵ in breaking Assumption 1.*

Proof. Simulator gives g, X_3, T to B. B simulates $Game_{Real}$ or $Game_0$ with A. B generates exponents $\{t_{i,j} \in_R Z_N\}_{1 \le i \le n, 1 \le j \le n_i}$, y, $\beta \in_R Z_N$. B calculates MPK as in Equation (8).

$$MPK = \Big\{ G_r, N, e, g, g^{\beta}, Y = e(g,g)^y, \quad T_{i,j} = g^{t_{i,j}} \; (i \in [1,n], j \in [1,n_i]) \Big\} \tag{8}$$

B sends the MPK to A. B can generate normal keys to the response of A's secret key query, using *KeyGen* as it knows MSK= $y, X_3, \beta\}$.

A sends M_0, M_1 and policy W to B. To make challenge ciphertext, B will implicitly set g^s as G_p part of T (as T is the product of $g^s \in G_p$ and possibly an element of G_q). It selects $\mu \in_R \{0,1\}, Z_{i,j} \in_R Z_N$ and sets

$$C1 = M_{\mu} e\left(g^y, T\right), C2 = T^{\beta + \sum_{v_{i,j} \in W} Z_{i,j}}, C3 = T$$

If $T \in G_p$, then this is normal ciphertext.

If $T \in G_{pr}$, then we let g_q^c denote G_q part of T (i.e., $T = g^s g_q^c$). Then we have semi-functional ciphertext that implicitly sets $Z_{i,j} = t_{i,j}$. By re-using the values of G_p part here, we prevent unwanted correlations. The values of $c, t_{i,j} \; mod \; q, \forall v_{i,j} \in W$ are uncorrelated to their *mod p* values because of the *Chinese Reminder* theorem. Hence, this is a properly distributed semi-functional ciphertext. Therefore, B will use the output of A to have ϵ advantage in breaking Assumption 1. □

Lemma 2. *If there exists a polynomial time algorithm A with $Game_{k-1,2}Adv_A - Game_{k,1}Adv_A = \epsilon$, then we can construct algorithm B in polynomial time with advantage ϵ in breaking Assumption 2.*

Proof. Simulator gives $g, X_1X_2, X_3, Y_2Y_3, T$ to B. B simulates $Game_{k-1,2}$ or $Game_{k,1}$ with A. B generates exponents $\{t_{i,j} \in_R Z_N\}_{1 \le i \le n, 1 \le j \le n_i}$, y, $\beta \in_R Z_N$. B calculates MPK as in Equation (9).

$$MPK = \Big\{ G_r, N, e, g, g^{\beta}, Y = e(g,g)^y, \quad T_{i,j} = g^{t_{i,j}} \; (i \in [1,n], j \in [1,n_i]) \Big\} \tag{9}$$

B sends the MPK to A. To make first $k-1$ semi-functional secret key of type 2, B answers each query (with attribute list say L that is different for each query) by selecting exponents $t \in_R Z_N, R_0, R_0' \in_R G_r$ and sets

$$D = (gR_0)^{y+\beta t} g_q^{\beta}, \forall v_{i,j_i} \in L \;\; D_{i,j} = \left(T_{i,j} R_{i,j}\right)^t,$$
$$D' = (gR_0')^t, L$$

Here, D is properly distributed as its value *mod q* and r are uncorrelated to its value *mod p*. To make a normal key for $> k$ queries, B can run the *KeyGen* algorithm, as it knows MSK.

To make key k, B implicitly sets g^t as G_p part of T. B selects exponents $R_0, R_0', R_{i,j} \in_R G_r$ and sets

$$D = g^y T^{\beta} R_0{}^{y+\beta t} g_q^{\beta}, \forall v_{i,j_i} \in L D_{i,j} = (T)^{t_{i,j}} R_{i,j}{}^t,$$
$$D' = T(R_0')^t, L$$

If $T \in G_{pr}$, then this properly distributes normal secret key. If $T \in G$, then this is a semi-functional secret key of type 1 that implicitly sets $Z_{i,j} = t_{i,j}$. If we denote g_q^b to be G_q part of T, then we are left with $b\beta \; mod \; q$. Here, G_q part of D is g_q^b, G_q part of D' is g_q^{β} and G_q part of $D_{i,j}$ is $g_q^{Z_{i,j}}$. The values of $Z_{i,j} \; mod \; q$ are uncorrelated from the $t_{i,j} \; mod \; p$.

A sends M_0, M_1 and policy W to B. To make challenge semi-functional ciphertext, B will implicitly set $g^s = X_1$ and $g_q^c = X_2$. It selects $\mu \in_R \{0,1\}, Z_{i,j} \in_R Z_N$ and sets

$$C1 = M_{\mu} e\left(g^y, X_1X_2\right), C2 = (X_1X_2)^{\beta + \sum_{v_{i,j} \in W} Z_{i,j}},$$
$$C3 = X_1X_2$$

Here, s is shared in the G_p subgroup and cy is shared in the G_q subgroup. If k^{th} key could decrypt the challenge ciphertext, then we will have $cb\beta = 0 \ mod \ q$. Therefore, the k^{th} key is either normal or *nominally* semi-functional secret key. One can see that as long as values of $c, b, \beta \ mod \ q$ are non-zero, the value shared in G_q subgroup is information theoretically hidden from A's viewpoint. Thus, for the random values of c, b and $\beta \ mod \ q$, this leads to a negligible advantage from A's viewpoint. Hence, the ciphertext and k^{th} key are properly distributed from A's viewpoint with negligible advantage. Thus, if $T \in G_{pr}$, then B has properly simulated $Game_{k-1,2}$, or if (with c, b and β values are non-zero), B has properly simulated $Game_{k,1}$. Therefore, B will use the output of A to have ϵ advantage in breaking Assumption 2. □

Lemma 3. *If there exists a polynomial time algorithm A with $Game_{k,1}Adv_A - Game_{k,2}Adv_A = \epsilon$, then we can construct an algorithm B in polynomial time with an advantage ϵ in breaking Assumption 2.*

Proof. The proof of this lemma is similar to previous lemma but without information theoretically hidden argument.

Simulator gives $g, X_1X_2, X_3, Y_2Y_3, T$ to B. B simulates $Game_{k,1}$ or $Game_{k,2}$ with A. B generates exponents $\{t_{i,j} \in_R Z_N\}_{1 \le i \le n, 1 \le j \le n_i}$, y, $\beta \in_R Z_N$. B calculates MPK as in Equation (10).

$$MPK = \Big\{ G_r, N, e, g, g^{\beta}, Y = e(g,g)^{y}, \\ T_{i,j} = g^{t_{i,j}} \ (i \in [1,n], j \in [1, n_i]) \Big\} \tag{10}$$

B sends the MPK to A. To make first $k-1$ semi-functional secret key of type 2, this works the same as of Lemma 2. To make a normal key for $> k$ queries, B can run the *KeyGen* algorithm, as it knows MSK.

For the k^{th} query, it will add one additional term $f \in Z_N$ and sets

$$D = g^{y} T^{\beta} R_0{}^{y+\beta t} (Y_2Y_3)^{f} g_q^{\beta}, D' = T(R_0')^{t}, \\ \forall v_{i,j_i} \in L \ D_{i,j} = (T)^{t_{i,j}} R_{i,j}{}^{t}, L$$

The additional term $(Y_2Y_3)^f$ will randomize the G_q part of D; so now, key is no longer semi-functional of type 1. Thus, decryption would fail for semi-functional ciphertext (we no longer have $cb\beta \equiv 0 \ mod \ q$).

If $T \in G_{pr}$, then this is normal semi-functional secret key of type 2. If $T \in G$, then this is semi-functional secret key of type 1. Therefore, B will use the output of A to have ϵ advantage in breaking Assumption 2. □

Lemma 4. *If there exists a polynomial time algorithm A with $Game_{q',2}Adv_A - Game_{Final}Adv_A = \epsilon$, then we can construct algorithm B in polynomial time with advantage ϵ in breaking Assumption 3.*

Proof. Simulator gives $g, X_3, T, g^{y}X_2, g^{s}Y_2, Z_2$ to B. B simulates $Game_{q',2}$ or $Game_{Final}$ with A. B generates exponents $\{t_{i,j} \in_R Z_N\}_{1 \le i \le n, 1 \le j \le n_i}$, $\beta \in_R Z_N$. B calculates MPK as in Equation (11).

$$MPK \Big\{ = G_r, N, e, g, g^{\beta}, Y = e\left(g, g^{y}X_2\right), \\ T_{i,j} = g^{t_{i,j}} \ (i \in [1,n], j \in [1, n_i]) \Big\} \tag{11}$$

B sends the MPK to A. In order to make semi-functional secret key of type 2, B answers each query (with attribute list say L that is different for each query) by selecting exponents $t \in_R Z_N, R_0, R_0', R_{i,j} \in_R G_r$ and sets

$$\forall v_{i,j_i} \in L \ D_{i,j} = \left(T_{i,j}R_{i,j}\right)^{t}, D = (gR_0)^{y+\beta t} Z_2{}^{t} g_q^{\beta}, \\ D' = (gR_0')^{t}, L$$

As in the previous lemmas, A sends M_0, M_1 and policy W to B. To make challenge semi-functional ciphertext, B selects $\mu \in_R \{0,1\}, Z_{i,j} \in_R Z_N$ and sets

$$C1 = M_{\mu}T, C2 = (g^{s}Y_2)^{\beta + \sum_{v_{i,j} \in W} Z_{i,j}}, C3 = g^{s}Y_2$$

If $T = e(g,g)^{ys}$, then this properly distributes semi-functional encryption of message M_{μ}. If $T \in G_1$, then this properly distributes semi-functional encryption of some random message from G_1. Therefore, B will use the output of A to have ϵ advantage in breaking Assumption 3. □

Theorem 1. *If Assumptions 1, 2 and 3 hold, then the proposed constant length ciphertext CP-ABE scheme is fully secure.*

Proof. We have shown that if Assumptions 1, 2 and 3 hold, then $Game_{Real}$ is indistinguishable from $Game_{Final}$ based on the aforementioned lemmas, and the value of μ is information theoretically hidden from the attacker's viewpoint. Hence, the attacker has negligible advantages in breaking the proposed CP-ABE system. □

6. PERFORMANCE EVALUATION

In this section, we discuss the implementation and the evaluation of the scheme in [33] and compare it against our scheme. For the practical performance, we have used the single threshold gate to represent the access structure of the proposed as well as that of scheme in [33].

In order to compare our performance and that of [33] in detail, we have implemented both the schemes and compared them empirically based on their running times. The results are shown in Tables IV–VI. The scheme of [33] requires the same set of attributes in the policy and ciphertext, that is, $m = n'$. Therefore, in Tables IV–VI, we have written "NO" for $m < n'$ as the decryption fails for the scheme of [33], where n' represents the total number of

Table IV. Time in milliseconds for *KeyGen*, at various security levels, for selected values of m and n'.

	n'	$m = 5$		10		15		20		25		30		35		40		45		50	
80 bit	10	NO	36	27	38																
	15	NO	52	NO	48	28	51														
	20	NO	60	NO	63	NO	62	26	61												
	25	NO	78	NO	74	NO	75	NO	76	27	78										
	30	NO	85	NO	83	NO	87	NO	82	NO	79	25	87								
	35	NO	101	NO	107	NO	106	NO	110	NO	96	NO	97	29	102						
	40	NO	198	NO	218	NO	212	NO	217	NO	222	NO	222	NO	214	27	227				
	45	NO	244	NO	240	NO	235	NO	237	NO	235	NO	240	NO	238	NO	236	25	240		
	50	NO	261	NO	266	NO	269	NO	259	NO	256	NO	255	NO	259	NO	254	NO	263	27	262
112 bit	10	NO	102	71	103																
	15	NO	144	NO	129	73	131														
	20	NO	167	NO	160	NO	160	74	163												
	25	NO	195	NO	192	NO	197	NO	190	72	197										
	30	NO	228	NO	226	NO	219	NO	220	NO	221	74	222								
	35	NO	258	NO	257	NO	253	NO	258	NO	260	NO	257	74	252						
	40	NO	280	NO	309	NO	289	NO	294	NO	288	NO	288	NO	302	72	294				
	45	NO	320	NO	318	NO	320	NO	354	NO	316	NO	323	NO	307	NO	322	75	321		
	50	NO	371	NO	363	NO	342	NO	337	NO	348	NO	352	NO	339	NO	357	NO	328	74	362
128 bit	10	NO	147	52	145																
	15	NO	190	NO	191	53	190														
	20	NO	238	NO	233	NO	235	54	236												
	25	NO	276	NO	287	NO	287	NO	284	58	279										
	30	NO	332	NO	332	NO	326	NO	336	NO	332	53	335								
	35	NO	376	NO	385	NO	380	NO	379	NO	383	NO	383	55	379						
	40	NO	420	NO	421	NO	418	NO	411	NO	420	NO	422	NO	427	49	429				
	45	NO	476	NO	474	NO	460	NO	463	NO	471	NO	469	NO	463	NO	457	57	469		
	50	NO	506	NO	521	NO	535	NO	522	NO	509	NO	510	NO	524	NO	518	NO	521	54	523
256 bit	10	NO	473	342	473																
	15	NO	643	NO	630	351	626														
	20	NO	784	NO	784	NO	798	357	779												
	25	NO	903	NO	934	NO	935	NO	915	358	919										
	30	NO	1081	NO	1071	NO	1078	NO	1079	NO	1096	353	1070								
	35	NO	1218	NO	1230	NO	1259	NO	1241	NO	1243	NO	1230	355	1230						
	40	NO	1382	NO	1412	NO	1388	NO	1355	NO	1367	NO	1395	NO	1362	361	1334				
	45	NO	1527	NO	1521	NO	1502	NO	1508	NO	1478	NO	1522	NO	1489	NO	1507	352	1518		
	50	NO	1675	NO	1679	NO	1666	NO	1707	NO	1745	NO	1691	NO	1686	NO	1654	NO	1652	358	1691

First number in each column is for [33], second number is for our scheme. "NO" means decryption is not possible with given m and n' values.

Table V. Time in milliseconds for *Encrypt*, at various security levels, for selected values of m and n'.

	n'	$m = 5$		10		15		20		25		30		35		40		45		50	
80 bit	10	NO	7	29	7																
	15	NO	8	NO	7	30	8														
	20	NO	9	NO	8	NO	9	28	9												
	25	NO	10	NO	10	NO	12	NO	11	33	11										
	30	NO	12	NO	13	NO	12	NO	12	NO	13	30	14								
	35	NO	12	NO	12	NO	13	NO	13	NO	12	NO	13	29	14						
	40	NO	14	NO	11	NO	13	NO	15	NO	13	NO	14	NO	12	30	15				
	45	NO	14	NO	16	NO	11	NO	12	NO	14	NO	19	NO	14	NO	14	33	14		
	50	NO	16	NO	13	NO	12	NO	11	NO	15	NO	15	NO	15	NO	15	NO	10	33	15
112 bit	10	NO	19	83	19																
	15	NO	19	NO	18	84	19														
	20	NO	21	NO	21	NO	18	84	19												
	25	NO	19	NO	18	NO	19	NO	19	76	20										
	30	NO	19	NO	19	NO	19	NO	19	NO	20	78	19								
	35	NO	19	NO	19	NO	19	NO	19	NO	19	NO	20	83	19						
	40	NO	18	NO	19	NO	19	NO	20	NO	19	NO	19	NO	21	84	20				
	45	NO	19	NO	19	NO	19	NO	19	NO	19	NO	19	NO	19	NO	20	85	20		
	50	NO	30	NO	20	NO	19	NO	18	NO	19	NO	19	NO	19	NO	19	NO	19	91	20
128 bit	10	NO	28	62	29																
	15	NO	27	NO	29	60	28														
	20	NO	26	NO	30	NO	30	61	30												
	25	NO	29	NO	30	NO	26	NO	29	58	26										
	30	NO	30	NO	30	NO	29	NO	29	NO	29	60	29								
	35	NO	31	NO	30	NO	30	NO	28	NO	30	NO	30	61	27						
	40	NO	28	NO	29	NO	26	NO	28	NO	29	NO	36	NO	29	63	31				
	45	NO	32	NO	26	NO	30	NO	31	NO	34	NO	26	NO	32	NO	30	64	30		
	50	NO	31	NO	27	NO	27	NO	28	NO	30	NO	30	NO	28	NO	25	NO	24	60	24
256 bit	10	NO	91	404	94																
	15	NO	94	NO	95	415	106														
	20	NO	91	NO	93	NO	99	429	95												
	25	NO	91	NO	93	NO	92	NO	92	432	93										
	30	NO	93	NO	92	NO	91	NO	94	NO	94	424	92								
	35	NO	93	NO	91	NO	96	NO	94	NO	102	NO	124	432	96						
	40	NO	94	NO	91	NO	93	NO	92	NO	94	NO	95	NO	96	426	90				
	45	NO	93	NO	94	NO	90	NO	92	NO	91	NO	93	NO	93	NO	93	425	96		
	50	NO	95	NO	92	NO	93	NO	94	NO	93	NO	103	NO	88	NO	91	NO	94	430	91

First number in each column is for [33], second number is for our scheme. "NO" means decryption is not possible with given m and n' value.

DOI: 10.1002/sec

Table VI. Time in milliseconds for *Decrypt*, at various security levels, for selected values of m and n'.

	n'	$m = 5$		10		15		20		25		30		35		40		45		50	
80 bit	10	NO	5	5	5																
	15	NO	5	NO	5	5	5														
	20	NO	5	NO	5	NO	5	5	5												
	25	NO	5	NO	8	NO	5	NO	5	6	5										
	30	NO	5	NO	5	NO	5	NO	5	NO	5	7	7								
	35	NO	5	NO	6	NO	5	NO	5	NO	5	NO	6	10	10						
	40	NO	5	NO	7	NO	12	NO	10	NO	11	NO	13	NO	12	12	12				
	45	NO	10	NO	9	NO	13	NO	12	NO	9	NO	9	NO	10	NO	12	12	12		
	50	NO	11	NO	11	NO	10	NO	11	NO	10	NO	9	NO	10	NO	12	NO	15	12	13
112 bit	10	NO	16	15	15																
	15	NO	15	NO	14	15	15														
	20	NO	15	NO	15	NO	15	16	15												
	25	NO	16	NO	14	NO	15	NO	15	16	17										
	30	NO	15	NO	15	NO	15	NO	15	NO	17	17	15								
	35	NO	16	NO	16	NO	15	NO	16	NO	17	NO	16	16	16						
	40	NO	14	NO	14	NO	16	NO	16	NO	16	NO	15	NO	18	19	16				
	45	NO	15	NO	15	NO	15	NO	15	NO	15	NO	16	NO	15	NO	16	15	15		
	50	NO	23	NO	16	NO	16	NO	14	NO	16	NO	16	NO	15	NO	15	NO	15	16	16
128 bit	10	NO	23	23	24																
	15	NO	20	NO	22	22	23														
	20	NO	20	NO	23	NO	21	23	21												
	25	NO	22	NO	21	NO	21	NO	20	23	22										
	30	NO	22	NO	22	NO	20	NO	21	NO	22	24	23								
	35	NO	22	NO	20	NO	22	NO	23	NO	22	NO	19	25	24						
	40	NO	21	NO	21	NO	25	NO	21	NO	19	NO	25	NO	24	25	24				
	45	NO	20	NO	23	NO	22	NO	22	NO	21	NO	23	NO	20	NO	20	23	24		
	50	NO	23	NO	20	NO	23	NO	25	NO	22	NO	22	NO	26	NO	26	NO	30	28	30
256 bit	10	NO	86	80	86																
	15	NO	86	NO	89	81	85														
	20	NO	86	NO	84	NO	84	85	90												
	25	NO	84	NO	86	NO	85	NO	85	82	87										
	30	NO	86	NO	86	NO	83	NO	90	NO	88	81	86								
	35	NO	85	NO	86	NO	88	NO	87	NO	96	NO	94	81	94						
	40	NO	89	NO	85	NO	86	NO	85	NO	87	NO	87	NO	91	85	83				
	45	NO	88	NO	88	NO	83	NO	86	NO	84	NO	92	NO	86	NO	85	92	90		
	50	NO	89	NO	86	NO	86	NO	91	NO	89	NO	92	NO	82	NO	85	NO	86	86	85

First number in each column is for [33], second number is for our scheme. "NO" means decryption is not possible with given m and n' value.

attributes in secret key of the user and m represents the total number of attributes in the policy of the ciphertext.

The implementation is based on Ben Lynn's PBC library (version 0.5.12) [39] that is publically available. For the pairing, we have used the Type A1 curve where the order of subgroup (G, G_1) is $N = pqr$. We have evaluated the performance of both schemes at 80, 112, 128 and 256 bits security levels, as per NIST standard [[40], Table 2]. All the tests were run on Intel Core i5 with 1 GB RAM and 32-bit mode Ubuntu 10.4 operating system.

For *KeyGen*, the asymptotic complexity of the proposed scheme is $O(n')(T_1 + T_2 + T_3)$ and $O(n')T_1 + O(1)(T_2 + T_3)$ for [33]. Therefore, we can see that in Table IV, the execution time of the proposed scheme is *lesser* as compared to that in [33]. The T_1, T_2, T_3 and T_4 are the cost of addition, multiplication, exponentiation and pairing operation, respectively.

For *Encrypt*, the asymptotic complexity of the proposed scheme is $O(m)T_2 + O(1)T_3+$ and $O(n')T_2 + O(1)(T_3 + T_4)$ for [33]. This shows that the scheme of [33] uses extra *pairing* operations and *multiplication* operations as compared to the proposed scheme. Therefore, we can see that in Table V, the execution time of the proposed scheme is *lesser* as compared to that in [33].

For *Decrypt*, the asymptotic complexity of the proposed scheme is $O(m)T_2 + O(1)T_4$ and $O(1)(T_2 + T_3 + T_4)$ for [33]. As can be seen, in the proposed scheme, there are *multiplication* operations as compared to *exponentiation* operation in [33]. Therefore, we can see that in Table VI, the execution time of the proposed scheme is *nearly* equal to that in [33].

As can be seen, the number of decryption policies supported by our scheme is greater than that in [33]. The resulting advantage is that the number of policies decrypted by user is higher.

7. CONCLUSION AND FUTURE WORK

As is commonly understood, in order to reduce the associated overhead in CP-ABE schemes, it is desirable to design fully secure CP-ABE schemes that offer a constant length ciphertext. Although one can find numerous approaches in the literature that support constant length ciphertext and are full secure individually, there is only one prevalent attempt that supports a fully secure constant length ciphertext CP-ABE scheme due to *Ren et al*. However, this scheme entails the limitation that it is a *(s,s)* threshold scheme and hence cannot be applied in certain restricted setups that demand multicast communication. We argue that in such setups, it is desirable to have a CP-ABE scheme that allows the number of attributes in the ciphertext to be a subset of the attributes in the receiver key. In this paper, we propose a constant length ciphertext CP-ABE scheme for a single authority system under the constraint that the number of attributes in the ciphertext policy is a subset of attributes in the receiver's secret key. Our approach is based on the AND gate with multivalued attribute. One can extend our scheme for fully threshold constant length ciphertext with fully secure setup.

ACKNOWLEDGEMENTS

The authors are thankful to *Dr. Wei Wang* for the painstaking efforts and to *anonymous reviewers* for their valuable suggestions to improve the quality of this paper. Without their support, this paper would not be in shape, it is, as of now.

REFERENCES

1. Shamir A. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology*, vol. 196, Blakley G, Chaum D (eds), Lecture Notes in Computer Science. Springer Berlin / Heidelberg: Springer, Heidelberg, 1985; 47–53.
2. Sahai A, Waters B. Fuzzy identity-based encryption. In *Advances in Cryptology EUROCRYPT 2005*, vol. 3494, Cramer R (ed), Lecture Notes in Computer Science. Springer Berlin / Heidelberg: Springer, Heidelberg, 2005; 557–557.
3. Rivest RL, Shamir A, Adleman L. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM* Feb.1978; **21**(2): 120–126.
4. Goyal V, Pandey O, Sahai A, Waters B. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*, CCS '06. ACM: ACM New York, NY, USA, 2006; 89–98.
5. Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption, *Security and Privacy, 2007. SP '07. IEEE Symposium on*, The Claremont Resort, Oakland, California, USA, May 2007; 321–334.
6. Goyal V, Jain A, Pandey O, Sahai A. Bounded ciphertext policy attribute based encryption. In *Automata, Languages and Programming*, vol. 5126, Lecture Notes in Computer Science. Springer Berlin / Heidelberg: Springer, Heidelberg, 2008; 579–591.
7. Chase M. Multi-authority attribute based encryption. In *Theory of Cryptography*, vol. 4392, Vadhan S (ed), Lecture Notes in Computer Science. Springer Berlin / Heidelberg: Springer, Heidelberg, 2007; 515–534.
8. Lewko A, Waters B. Decentralizing attribute-based encryption. In *Advances in Cryptology EUROCRYPT 2011*, vol. 6632, Paterson K (ed), Lecture Notes

in Computer Science. Springer Berlin / Heidelberg: Springer, Heidelberg, 2011; 568–588.
9. Bovi V, Socek D, Steinwandt R, Villnyi V I. Multi-authority attribute-based encryption with honest-but-curious central authority. *International Journal of Computer Mathematics* 2012; **89**(3): 268–283.
10. Muller S, Katzenbeisser S, Eckert C. Distributed attribute-based encryption. In *Information Security and Cryptology ICISC 2008*, vol. 5461, Lee P, Cheon J (eds), Lecture Notes in Computer Science. Springer Berlin / Heidelberg: Springer, Heidelberg, 2009; 20–36.
11. Müller S, Katzenbeisser S, Eckert C. On multi-authority ciphertext-policy attribute-based encryption. *Bulletin of the Korean Mathematical Society* 2009; **46**(4): 803–819.
12. Lin H, Cao Z, Liang X, Shao J. Secure threshold multi authority attribute based encryption without a central authority. *Information Sciences* 2010; **180**(13): 2618–2632.
13. Cheung L, Newport C. Provably secure ciphertext policy abe. In *Proceedings of the 14th ACM Conference on Computer and Communications Security*, CCS '07. ACM: ACM New York, NY, USA, 2007; 456–465.
14. Kapadia A, Tsang PP, Smith SW. Attribute-based publishing with hidden credentials and hidden policies, *In The 14th Annual Network and Distributed System Security Symposium (NDSS 07)*, Catamaran Resort Hotel - San Diego, CA, USA, 2007; 179–192.
15. Lubicz D, Sirvent T. Attribute-based broadcast encryption scheme made efficient. In *Progress in Cryptology AFRICACRYPT 2008*, vol. 5023, Vaudenay S (ed), Lecture Notes in Computer Science. Springer Berlin/Heidelberg: Springer, Heidelberg, 2008; 325–342. 10.1007/978-3-540-68164-9_22.
16. Waters B. Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In *Public Key Cryptography PKC 2011*, vol. 6571, Catalano D, Fazio N, Gennaro R, Nicolosi A (eds), Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2011; 53–70. 10.1007/978-3-642-19379-8_4.
17. Hur J, Park C, Hwang S. Fine-grained user access control in ciphertext-policy attribute-based encryption. *Security and Communication Networks* 2012; **5**(3): 253–261.
18. Daza V, Herranz J, Morillo P, Rfols C. Extensions of access structures and their cryptographic applications. *Applicable Algebra in Engineering, Communication and Computing* 2010; **21**: 257–284, 10.1007/s00200-010-0125-1.
19. Emura K, Miyaji A, Nomura A, Omote K, Soshi M. A ciphertext-policy attribute-based encryption scheme with constant ciphertext length. In *Information Security Practice and Experience*, vol. 5451, Bao F, Li H, Wang G (eds), Lecture Notes in Computer Science. Springer Berlin / Heidelberg: Springer, Heidelberg, 2009; 13–23. 10.1007/978-3-642-00843-6_2.
20. Herranz J, Laguillaumie F, Rfols C. Constant size ciphertexts in threshold attribute-based encryption. In *Public Key Cryptography PKC 2010*, vol. 6056, Nguyen P, Pointcheval D (eds), Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2010; 19–34. 10.1007/978-3-642-13013-7_2.
21. Zhou Z, Huang D. On efficient ciphertext-policy attribute based encryption and broadcast encryption: extended abstract. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, CCS '10. ACM: New York, NY, USA, 2010; 753–755.
22. Attrapadung N, Herranz J, Laguillaumie F, Libert B, de Panafieu E, Rfols C. Attribute-based encryption schemes with constant-size ciphertexts. *Theoretical Computer Science* 2012; **422**(0): 15–38.
23. Chen C, Zhang Z, Feng D. Efficient ciphertext policy attribute-based encryption with constant-size ciphertext and constant computation-cost. In *Provable Security*, vol. 6980, Boyen X, Chen X (eds), Lecture Notes in Computer Science. Springer Berlin / Heidelberg: Springer, Heidelberg, 2011; 84–101. 10.1007/978-3-642-24316-5_8.
24. Venugopalan S. Attribute based cryptology. *M.Tech. Dissertation*, Indian Institute of Technology, IIT Madras, 2011. Available from: https://researcher.ibm.com/files/in-subhvenu/thesis.pdf.
25. Doshi N, Jinwala D. Constant ciphertext length in multi-authority ciphertext policy attribute based encryption, *Computer and Communication Technology (ICCCT), 2011 2nd International Conference on*, Motilal Nehru National Institute of Technology, Allahabad (U.P.), INDIA, sept.2011; 451–456.
26. Döshi N, Jinwala D. Hidden access structure ciphertext policy attribute based encryption with constant length ciphertext. In *Advanced Computing, Networking and Security*, vol. 7135, Thilagam P, Pais A, Chandrasekaran K, Balakrishnan N (eds), Lecture Notes in Computer Science. Springer Berlin / Heidelberg: Springer, Heidelberg, 2012; 515–523. 10.1007/978-3-642-29280-4_60.
27. Ge A, Zhang R, Chen C, Ma C, Zhang Z. Threshold ciphertext policy attribute-based encryption with constant size ciphertexts. In *Information Security and Privacy*, vol. 7372, Susilo W, Mu Y, Seberry J (eds),

Lecture Notes in Computer Science. Springer Berlin / Heidelberg: Springer, Heidelberg, 2012; 336–349. 10.1007/978-3-642-31448-3_25.

28. Kitak Kim JHP, Koo WK, Lee DH. Chosen ciphertext secure ciphertext-policy attribute-based encryption with constant ciphertext length and threshold policy. In *International Conference on Information Science and Technology (IST 2012)*, vol. 3, Adrian S, Jeong J K, Sabah Mohammed E G and Yvette, Ronnie DC (eds), Information Science and Technology: SERSC, Sandy Bay, Tasmania, Australia, 2012. Available from: http://onlinepresent.org/proceedings/vol3_2012/88.pdf.
29. Lewko A, Okamoto T, Sahai A, Takashima K, Waters B. Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption. In *Advances in Cryptology EUROCRYPT 2010*, vol. 6110, Gilbert H (ed), Lecture Notes in Computer Science. Springer Berlin / Heidelberg: Springer, Heidelberg, 2010; 62–91. 10.1007/978-3-642-13190-5_4.
30. Lai J, Deng R, Li Y. Fully secure cipertext-policy hiding cp-abe. In *Information Security Practice and Experience*, vol. 6672, Bao F, Weng J (eds), Lecture Notes in Computer Science. Springer Berlin / Heidelberg: Springer, Heidelberg, 2011; 24–39. 10.1007/978-3-642-21031-0_3.
31. Qian Jl, Dong Xl. Fully secure revocable attribute-based encryption. *Journal of Shanghai Jiaotong University (Science)* 2011; **16**: 490–496, 10.1007/s12204-011-1178-4.
32. Liu Z, Cao Z, Huang Q, Wong D, Yuen T. Fully secure multi-authority ciphertext-policy attribute-based encryption without random oracles. In *Computer Security ESORICS 2011*, vol. 6879, Atluri V, Diaz C (eds), Lecture Notes in Computer Science. Springer Berlin / Heidelberg: Springer, Heidelberg, 2011; 278–297. 10.1007/978-3-642-23822-2_16.
33. Ren Y, Wang S, Zhang X, Qian Z. Fully secure ciphertext-policy attribute-based encryption with constant size ciphertext, *Multimedia Information Networking and Security (MINES), 2011 Third International Conference on*, Nanjing Huamao International Hotel, Nanjing, China, nov.2011; 380 –384.
34. Bethencourt J, Sahai A, Waters B. *The cp-abe toolkit, advanced crypto software collection*, December 1, 2006. Available from: http://acsc.cs.utexas.edu/cpabe/.
35. Lewko A, Waters B. New techniques for dual system encryption and fully secure hibe with short ciphertexts. In *Theory of Cryptography*, vol. 5978, Micciancio D (ed), Lecture Notes in Computer Science. Springer Berlin / Heidelberg: Springer, Heidelberg, 2010; 455–479. 10.1007/978-3-642-11799-2_27.
36. Waters B. Dual system encryption: realizing fully secure ibe and hibe under simple assumptions. In *Advances in Cryptology – CRYPTO 2009*, vol. 5677, Halevi S (ed), Lecture Notes in Computer Science. Springer Berlin / Heidelberg: Springer, Heidelberg, 2009; 619–636. 10.1007/978-3-642-03356-8_36.
37. Chen C, Chen J, Lim H, Zhang Z, Feng D, Ling S, Wang H. Fully secure attribute-based systems with short ciphertexts/signatures and threshold access structures. In *Topics in Cryptology CT-RSA 2013*, vol. 7779, Dawson E (ed), Lecture Notes in Computer Science. Springer Berlin Heidelberg: Springer, Heidelberg, 2013; 50–67.
38. Boneh D, Goh E J, Nissim K. Evaluating 2-DNF formulas on ciphertexts. In *Theory of Cryptography*, vol. 3378, Kilian J (ed), Lecture Notes in Computer Science. Springer Berlin / Heidelberg: Springer, Heidelberg, 2005; 325–341. 10.1007/978-3-540-30576-7_18.
39. Lynn B. *PBC library*, 2006. Available from: http://crypto.stanford.edu/pbc.
40. Barker E, Barker W, Burr W, Polk W, Smid M. NIST special publication 800-57. recommendation for key management part 1: General(revised). *NIST Special Publication. National Institute of Standards and Technology (NIST)* 2007; **800**(57): 1–142.