A Provable and User Revocable Ciphertext-Policy Attribute-Based Encryption with Updatable Ciphertext

Humera Aqeel and Syed Taqi Ali

Abstract Attribute-based encryption (ABE) is one of most favorable tools that enable fine-grained data access control and ensure confidentiality. In our work, we mainly concentrate on Ciphertext-Policy ABE which is viewed to be one of the nearly efficient technologies for data access control as the encryptor is much intelligent to determine who may or may not have access to data encrypted by him as the access control policy lies within the ciphertext. We also illustrate the concept of revocation which corresponds to one of the most sensitive issues in ABE. We give a concrete construction of CP-ABE and prove that our scheme is Chosen-Plaintext Attack (CPA) secured under the Decisional Bilinear Diffie–Hellman assumption (DBDH). We also give the ciphertext updation for the revocation of user along with its security proof.

Keywords Linear secret sharing scheme · Attribute-based encryption
Revocation · Ciphertext-Policy ABE

1 Introduction

Public-key encryption is a strong tool that keeps the transmitted and stored data confidential. A single known user chooses encrypted data for decryption in identity-based encryption (IBE) or traditional public-key encryption systems. In a more advanced sharing of data, this functionality limits the expressiveness required. So in order to tackle these emerging needs, came up an emerging notion of attribute-based encryption (ABE). In recent years, there has been remarkable progress in ABE in terms of effectiveness, security, and diversed assumptions for security. In comparison

H. Aqeel (✉) · S. T. Ali
Department of Computer Engineering, National Institute of Technology,
Kurukshetra 136119, Haryana, India
e-mail: humeraaqeel26@gmail.com

S. T. Ali
e-mail: taqiali110@gmail.com

H. S. Saini et al. (eds.), *Innovations in Computer Science and Engineering*,
Lecture Notes in Networks and Systems 32,
https://doi.org/10.1007/978-981-10-8201-6_44

with IBE, ABE has meaningful benefit as instead of one to one, it gains adaptability for one-to-many encryption, and it is visualized as a favorable paradigm in address to the problems of decentralized access control and fine-grained data sharing. ABE is comprised of two categories mainly CP-ABE and KP-ABE. In KP-ABE, every ciphertext is labeled with set of detailed attributes by encryptor, and access policy is determined within user's secret keys over these attributes. CP-ABE is alike to KP-ABE, besides that the ciphertext determines the access policy and a private key is related with user's attributes. Though ABE is considered as a flexible, efficient, and effective tool, user revocation is still a difficult issue in context to ABE. A revoked user must be restricted from accessing the data, even though his attributes are satisfying the access policy. Since various users can have the similar usable private key related with indistinguishable set of attributes, user revocation is hard to attain in settings of ABE. Revocation is broadly divided as direct revocation and indirect revocation. Our paper mainly focuses on the direct revocation.

In further paper, we discuss as Sect. 2 gives related work, Sect. 3 gives proposed work, Sect. 4 preliminaries, Sect. 5 gives mathematical construction along with correctness, Sect. 6 security proof, and Sect. 7 gives conclusion with future work.

2 Related Work

The first effective IBE system was given by Boneh and Franklin [1]. Further, Sahai and Waters [2] presented the idea of ABE through substituting the identity in IBE with set of attributes. Due to lack of expressiveness of ABE in larger system, Goyal et al. [3] develop a much versatile ABE cryptosystem called KP-ABE. Bethencourt et al. [4] initiated first construction of Ciphertext-Policy ABE proven secured under generic model. Cheung and Newport [5] proposed another construction of CP-ABE which is proved to be secured within the standard model. Various other constructions are given for CP-ABE [6, 7]. Boldyreva et al. [8] proposed a revocable ABE scheme known as indirect revocation. Attrapadung and Imai [9] proposed Broadcast ABE for both KP-ABE and CP-ABE with direct revocation mechanism. Another work is given by Attrapadung and Imai [10] illustrated hybrid direct–indirect revocation scheme and were proved secured under DBDH assumption. Another work given by Shi et al. [11] proposed a novel ABE variant, drvuKPABE, that supports direct revocation as well as verifiable ciphertext delegation. Another work given by Zhang et al. [12] proposed first access control scheme that supports attribute update and user revocability, proven secured under the DBDH assumption.

3 Proposed Work

In our paper, we have given a CP-ABE scheme considering revocation and a detailed security proof of the scheme against Chosen-Plaintext Attack (CPA) proven to be

secured under Decisional Bilinear Diffie–Hellman (DBDH) assumption. Also, we give the concept of update ciphertext additionally with the scheme and its security proof. With update ciphertext, we determine that the diffusion of ciphertexts produced by *Encryption* algorithm is indistinguishable to that of the ciphertexts produced by *Update* algorithm where both corresponds to same number of terms and revocation lists $R_{list} \subset R'_{list}$, respectively.

4 Preliminaries

1. **Linear Secret Sharing Scheme**- A LSSS represents the access control policy P defined as (T, ρ) where T determines $l \times k$ matrix where entries belongs to Z_q and $\rho : (1, ..., l) \rightarrow U_{att}$ is an one-to-one function which outlines mapping of a row to the attribute. The given attribute set $S \subset U_{att}$, denoted by $F(S, P) = 1$ if policy P is satisfied by S. A LSSS determines two algorithms: share and reconstruction. Along with the reconstruction algorithm, we give following Lemma:

   Lemma 1 [6]: Suppose (T, ρ) be the LSSS that represents policy P. So in every attribute in S that is not being satisfied by P, there is a algorithm in polynomial time which returns a vector $N = (n_1, ..., n_k) \in Z_q$ such that $n_1 = -1$ and $T_i.N = 0$ for every $i \in [1, ..., l]$ where $\rho(i) \in S$.
2. **Subset Cover Technique**- As revocation is a major issue in our scheme, subset cover technique is an efficient way to encode revoked users in the revocation list [11].
3. **Ciphertext-Policy ABE**- It consists of algorithms as follows:-
   - $Setup(1^l) \rightarrow (MK, PK)$: It inputs a security parameter 1^l and outputs a MK and PK as master key and public key, respectively. The PK corresponds to encryption, whereas the MK produces user private keys which is confined by centralized authority.
   - $KeyGen(MK, S, \texttt{uid}) \rightarrow sk$: It takes master key MK and a attributes set, $S \subseteq U_{att}$ as inputs. It returns secret key sk for users satisfied by `uid` associated with S.
   - $Enc(m, (T, \rho)) \rightarrow cph$: It takes a m and (T, ρ) as message and access structure respectively as inputs. It further encrypts m using (T, ρ) returns a ciphertext cph.
   - $Dec(cph, sk) \rightarrow (m, \perp)$: The decryption is successfully done only when the attribute set S associated to decryption key is been satisfied by access control policy P which is specified by ciphertext and identity of user described by decryption key is not likely revoked with respect to revocation list which is given by cph. Otherwise, it would give an error message $\perp$.
4. **Correctness**- The correctness of given scheme is illustrated as:
   $Setup(1^l) \rightarrow (PK, MK)$, given any message m, set of attribute S, and revocation list R_{list}, let $cph \rightarrow Enc(m, P, R_{list})$, the scheme would be correct only when

following always holds: Given $sk \leftarrow KeyGen(MK, S, \texttt{uid})$ where $F(S, P) = 1$ and $\texttt{uid} \neq R, Dec(cph, sk) = m$.

5. **Security Model**- In this model, we give the definition of Chosen-Plaintext Attack (CPA) security with user revocation where a PPT *Adv* at outset of security game shall commit to a challenge access structure (T^*, ρ^*).
   - **Initialization**: A PPT *Adv* returns (T^*, ρ^*) to $\mathcal{C}$.
   - **Setup**: It chooses a revocation list R_{list} and sends that to $\mathcal{C}$ who when executes the setup algorithm produces PK, MK, gives the PK to *Adv*, and keeps MK secret.
   - **Phase 1**: *Adv* may flexibly submit any attributes set S to $\mathcal{C}$, and queries to $\mathcal{C}$ for the secret key with respect to any attributes set S with constraint that S should not be satisfied by access structure (T^*, ρ^*). For the set of attributes S, $\mathcal{C}$ runs the *KeyGen* algorithm as $sk \leftarrow KeyGen(MK, PK, S, \texttt{uid})$ and sends the corresponding private key sk to *Adv*. In case $F(S, P) = 1$ and $\texttt{uid} \notin R_{list}$ simultaneously, then abort.
   - **Challenge**: The challenger $\mathcal{C}$ receives two equal length messages M_0, M_1 submitted by the *Adv* after which $\mathcal{C}$ arbitrarily selects one bit $\sigma \in \{0, 1\}$ and runs $cph^* \leftarrow Enc(M_\sigma, (T^*, \rho^*), R_{list})$. Finally, *Adv* gets the challenge ciphertext cph^* from $\mathcal{C}$.
   - **Phase 2**: *Adv* queries more about secret key sk in the similar manner like in Phase 1 with same limitations.
   - **Guess**: *Adv* outputs a σ' as a guess. It will win the game if $\sigma = \sigma'$.

5 Proposed Construction

This section gives the detailed construction of our scheme that comprises of four algorithms as follows:

1. $Setup(1^l, U_{att})$- This algorithm selects a security parameter l and an universal attribute system $U_{att} = (1, 2, ..., m)$ of size m as input. Consider a bilinear map [3] $e : G_0 \times G_0 \rightarrow G_T$ having G_0 as bilinear group with prime order q along with g as a generator. Further, it picks $\alpha, \beta \in Z_q$ randomly as two exponents. For every $i \in U_{att}$, algorithm chooses randomly $t_i \in Z_q$. The public key is published and master key is generated as:
   $PK = [G_0, g, g^\alpha e(g, g)^\beta, \{pk_i = g^{t_i} | i \in U_{att}\}]$, $MK = [\alpha, \beta, t_1, t_2, ..., t_m]$
2. $KeyGen(MK, PK, S, \texttt{uid})$- Here, we take a master key, public key, attributes set S as input parameters. Also, we define a universal set W be the users universe in system and 2^d be number of users, so, $|Y| = 2^d$, depth of all leaves in full binary tree is d, set user identity $\texttt{uid} \in W$. Select a random $p, h \in Z_q$, $D(1) = g^\beta . g^{ah}$, $D_{At}(2) = g^{h \cdot \frac{1}{At}}$ $\forall At \in S$
   Given the user identity $\texttt{uid} \in Y$, suppose path(uid) in the full binary tree Tr such that $v_{i_0} = root$ and $v_{i_{\texttt{depth(uid)}}} = \texttt{uid}$ is path(uid) $= (v_{i_0}, v_{i_1}, ..., v_{i_{\texttt{depth(uid)}}})$. Let,

$D_v(3) = (v)^p$; $\forall v \in \text{path}(\texttt{uid}), D(4) = g^{\alpha.p}$ The decryption key is $sk = [\texttt{uid}, \{D_{At}(1)\}_{At \in S}, D(2), \{D_v(3)\}_{v \in \text{path}(\texttt{uid})}, D(4)]$

3. $Enc(m, (T, \rho), R_{list})$- Select $s \rightarrow Z_q$ and set $C = m.e(g,g)^{\beta.s}$, $C(1) = g^{\alpha.s}$. Let $d = (s, y_2, ..., y_k)$ where $y_2, ..., y_k \leftarrow Z_q$, for $i = (1, ..., l)$, T is a $l \times k$ matrix having entries belonging to Z_q and $\rho : (1, ..., l) \rightarrow U_{att}$ is an one-to-one function which does mapping of a row of T to an attribute. Compute $\lambda_{\rho(i)} = T_i.d$ where T_i is corresponding *ith* row of T and $\rho(i)$ is the attribute from U_{att}. On computing, we get a secret share value $\lambda_{\rho(i)}$, then set, $C_i(2) = g^{a.\rho(i).\lambda_{\rho(i)}}$, $C_i(3) = g^{\lambda_{\rho(i)}}$
   As we are using direct revocation mechanism, we have revocation list R_{list} as input. Then we run $cover(R_{list})$ which describes the cover set with respect to revocation list R_{list} to find a set of minimal nodes that cover $Y \backslash R_{list}$. Let $\text{path}(u) = (u_{i_0}, ..., u_{i_{\text{depth(u)}}})$ such that $u_{i_0} = root$ and $u_{i_{\text{depth(u)}}} = u$ for every $u \in cover(R_{list})$ and computes $P_u = \prod(u_{i_j})$ where $j = 1, ..., u$ where $u_{i_j} \in \text{Tr}$ and set, $C_u(4) = P_u^s \ \forall u \in cover(R_{list})$. The ciphertext is:
   $cph = [(T, \rho), C, C(1), \{C_i(2), C_i(3)\}_{i \in [1,l]}, C_u(4)_{u \in cover(R_{list})}]$
4. $Dec(cph, sk)$- Given cph and sk, decryption is as follows:

   - When either identity of user $\texttt{uid} \in R_{list}$ or set of attributes is not been satisfied by access control policy identified by (T, ρ), then output null.
   - Since $\texttt{uid} \neq R_{list}$, there would always be a node v such that $v \in \text{path}((\texttt{uid}) \cap cover(R_{list}))$. Suppose $\text{path}(\texttt{uid}) = (v_{i_0}, ..., v_{i_{\text{depth(v)}}}, ..., v_{i_{\text{depth(uid)}}})$ where $v_{i_{\text{depth}(v)}} = v$. Let $P'_{\texttt{uid}} = C_u(4)$ where $u \in cover(R_{list})$ and $u = v$.
   - Since the access control policy determined by (T, ρ) and satisfied by attribute set S, there exists $c_i's$ such that $\Sigma_{\rho(i) \in S} c_i.T_i = s$ and,

$$K = \prod_{\rho(i) \in S} \left(\frac{e(C_i(3), D(1))}{e(C_i(2), D_{\rho(i)}(2))}\right)^{c_i} . \frac{e(C(1), \prod_{v' \in \text{path}(v)} D_{v'}(3))}{e(P'_{\texttt{uid}}, D(4))}$$

   The message is obtained as: $m = \frac{C}{K}$

5.1 *Update*(cph, R'_{list})

Given a new revocation list (R'_{list}), ciphertext can be updated as: Let $cover(R_{list})$ and $cover(R'_{list})$ are the cover sets of R_{list} and R'_{list}, respectively. Given $v' \in cover(R'_{list})$,

- Suppose there exists $v \in cover(R_{list})$ such as $v = v'$, then set $C_v(4) = C'_{v'}(4)$.
- Else, there exists $v \in cover(R_{list})$ such as v is the predecessor of v'. Let $path(v') = path(v) \cup (v_{i_{depth(v)+1}} ... v_{i_{depth(v')}})$ where $v_{i_{depth(v)}} = v$ and $v_{i_{depth(v')}} = v'$, and set $P'_{v_{i_{depth(v)}}} = C_v(4)$. For $j = v, ..., v'$, compute $P'_v = \prod(v_{i_j})$ where $v_{i_j} \in \text{Tr}$. Set, $C'_{v'}(4) = P'_{v'}$
- Suppose $C'(1) = C(1), C'_i(2) = C_i(2), C'_i(3) = C_i(3)$.

The updated ciphertext is: $cph^{'} = [(T, \rho), R'_{list}, C', C'(1), [C'_i(2), C'_i(3)]_{i\in[1,l]}, C'_{v'}(4)_{v'\in cover(R'_{list})}]$

5.2 Correctness

The verification of correctness of the decryption is as illustrated: As we know,

$$K = \prod_{\rho(i)\in S} \left(\frac{e(C_i(3), D(1))}{e(C_i(2), D_{\rho(i)}(2))}\right)^{c_i} . \frac{e(C(1), \prod_{v'\in \mathtt{path}(v)} D_{v'}(3))}{e(P'_{\mathtt{uid}}, D(4))}$$

Suppose $K = K'.K''$, then

$$K' = \prod_{\rho(i)\in S} \left(\frac{e(C_i(3), D(1)),}{e(C_i(2), D_{\rho(i)}(2))}\right)^{c_i}$$

$$K' = \prod_{\rho(i)\in S} \left(\frac{e(g^{\lambda_{\rho(i)}}, g^{\beta}.g^{a.h}}{e(g^{a.\rho(i).\lambda_{\rho(i)}}, g^{h.\frac{1}{\rho(i)}})}\right)^{c_i} \tag{1}$$

$$= \prod_{\rho(i)\in S} e(g^{\lambda_{\rho(i)}}, g^{\beta})^{c_i}$$

$$= e(g, g)^{\Sigma_{\rho(i)\in S}\beta.\lambda_{\rho(i)}.c_i} \tag{2}$$

$$= e(g, g)^{\beta.s} \tag{3}$$

In the above-defined equations, (1) illustrates the construction, (2) illustrates property of bilinear map, (3) illustrates characterstics of linear secret sharing scheme, i.e., linear reconstruction. Similarly, we have

$$K'' = \frac{e(C(1), \prod_{v'\in \mathtt{path}(v)} D_{v'}(3))}{e(P'_{\mathtt{uid}}, D(4))} = 1$$

Further, combining K' and K'', we have $K = K'K'' = e(g, g)^{\beta s}$ Therefore, we compute the message as: $\frac{C}{K}$,

$$= \frac{m.e(g, g)^{\beta s}}{e(g, g)^{\beta s}} = m$$

6 Security Proof

Theorem *If PPT Adv with non-negligible advantage wins CP-ABE security game, then considering a PPT algorithm $\mathcal{B}$ such that a DBDH tuple can be differentiated from an arbitrary tuple with non-negligible advantage.*

Proof Consider a bilinear map $e : G_0 \times G_0 \rightarrow G_T$ having G_0 as bilinear group with prime order q along with g as a generator. Firstly, DBDH challenger C selects randomly: $a, b, c \in Z_q, \sigma \in \{0, 1\}$ and $R \in G_T$ as some random element. We let Z as $e(g, g)^{abc}$ if $\sigma = 0$, otherwise R. Then C sends $< g, A, B, C, Z >=< g, g^a, g^b, g^c, Z >$ to $\mathcal{B}$. Now $\mathcal{B}$ would play part of C in further game.

1. **Initialization**: $\mathcal{B}$ receives a challenge access structure (T^*, ρ^*) that is selected by *Adv*.
2. **Setup**: As per imparting public key *PK* to Adversary *Adv*, $\mathcal{B}$ selects $\alpha, \beta' \in Z_q$ randomly, set $\beta = \beta' + ab$, compute $e(g, g)^{\beta}$ as $e(g, g)^{\beta'} e(g, g)^{ab}$. For each $i \in U_{att}$, challenger $\mathcal{B}$ selects a random $d_i \in Z_q$ and compute g^{d_i}. Finally, challenger $\mathcal{B}$ sends the $PK = (G_0, e(g, g)^{\beta}, g, g^{\alpha}, g^{d_i} | i \in U_{att})$ parameters to adversary *Adv*. Moreover, given the revocation list R_{list},
   Let $\chi_{R_{list}} = \{v \in \text{path(uid)} | \text{uid} \in R_{list}\}$
3. **Phase 1**: Here $\mathcal{B}$ answers private key queries from adversary *Adv*. *Adv* can submit adaptively any set of attribute $S \subseteq U_{att}$ to challenger $\mathcal{B}$ and performs private key query for S with the limitation that access structure (T^*, ρ^*) should not be satisfied by S. On each request, simulator$\mathcal{B}$ finds a vector $N = (n_1, ..., n_k) \in Z_q^k$ such that $n_1 = -1$ and $\forall \rho(i) \in S, T_i.N = 0$, by Lemma 1, likewise a vector definitely exist. Also, the simulator implicitly defines h value as $1 + n_j.b$. Here we choose n_j as n_1 to compute. Select $p' \in Z_q$, we have
   $D(1) = g^{\beta} g^{a.h} = g^{\beta' + ab} g^{a.(1-b)} = g^{\beta'}.A$, $D_{At}(2) = g^{h.\frac{1}{\rho(i)}}$
   Suppose path(uid) $= (v_0, ..., v_d = \text{uid})$, and there exists some $v_j \in \chi_{R_{list}}$, then it sets and computes,
   $D(3) = g^{v^{p'}}$ $\forall v \in \text{path(uid)}$, $D(4) = g^{\alpha.p'}$
4. **Challenge**: *Adv* outputs M_0, M_1 as two equal length messages and sends them to $\mathcal{B}$. It selects $y_i' \leftarrow Z_q$ for $i = (2, ..., k)$, and sets $d' = (\beta_1, y_2', ...y_k')$. Compute $\lambda_{\rho(i)} = T_i.d = T_i.d'$. Therefore,
   $C = m_{\sigma}.e(g, g)^{\beta.c} = m_{\sigma}.e(g, g)^{(\beta' + ab)c} = m_{\sigma} e(g, g)^{\beta'.c}$
   $C(1) = g^{\alpha.s}$, $C_i(2) = g^{a.\rho(i).T_i.d'}$, $C_i(3) = g^{T_i.d'}$
   Given every $v \in cover(R_{list})$, suppose path$(v) = (v_0, ..., v_{depth(v)}(= v))$, then $v_i \in \chi_{R_{list}}$, $i = 0, ..., v$ and $v \notin \chi_{R_{list}}$, it sets, $C_v(4) = g^{v^c}$.
   Finally, $\mathcal{B}$ gives *Adv* challenge ciphertext as,
   $cph^* = (C(1), (C_i(2), C_i(3))_{(i \in [1,l])}, C_v(4)_{v \in cover(R_{list})})$.
5. **Phase 2**: It is similar to Phase 1 with the same limitation.
6. **Guess**: *Adv* outputs v' of v as a guess. $\mathcal{B}$ returns 0 to specify $Q = e(g, g)^{abc}$ if $v' = v$ otherwise, returns 1 to guess $Q = R$. Thus,
   $X = Pr[\mathcal{B}(g, g^a, g^b, g^c, Q = e(g, g)^{abc}) = 0] = \frac{1}{2} + \epsilon$

If $Q = R$, then cph^* is random from Adv view completely. Therefore, $Y = Pr[\mathcal{B}(g, g^a, g^b, g^c, Q = R) = 0] = \frac{1}{2}$
Lastly, benefit of $\mathcal{B}$ in the game is $\mathcal{B} = \frac{1}{2}(X + Y) - \frac{1}{2} = \frac{1}{2}(\frac{1}{2} + \epsilon + \frac{1}{2}) - \frac{1}{2}$ $= \frac{\epsilon}{2}$

6.1 Security Proof for Update(cph, R'_{list})

Theorem *Given the DBDH assumption, scheme proposed in updated ciphertext is secured in random oracle model.*

Here, the strategy is to understand if distribution of ciphertexts which algorithm *Encryption* produces is identical to that of ciphertexts which *Update* algorithm produces, then scheme over updated ciphertext achieves security. Given message m, access policy (T, ρ), suppose we have a revocation list R'_{list} changed from R_{list} such that $R_{list} \subset R'_{list}$, then, $Encryption(m, (T, \rho), R_{list})$ is:

$$cph = [(T, \rho), R_{list}, C, C(1), \{C_i(2), C_i(3)\}_{i\in[1,l]}, C_v(4)_{v\in cover(R_{list})}]$$

where $s' \in Z_q$ and $C = m.e(g, g)^{\beta.s'}$,
$C_i(2) = g^{a.\rho(i).\lambda_{\rho(i)}}, C_i(3) = g^{\lambda_{\rho(i)}}, C_v(4) = g^{s'}.P_v^{s'}$
Similarly, $Encryption(m, (T, \rho), R'_{list})$ is:

$$cph^* = [(T, \rho), R'_{list}, C^*, C^*(1), \{C_i^*(2), C_i^*(3)\}_{i\in[1,l]}, (C_{v'}^*(4))_{v'\in cover(R_{list})}]$$

where $C^* = m.e(g, g)^{\beta.s}$,
$C_i^*(2) = g^{a.\rho(i).\lambda_{\rho(i)}}, C_i^*(3) = g^{\lambda_{\rho(i)}}, C_{v'}^*(4) = g^s.P_{v'}^s$
Then the updated ciphertext is $Update(cph, R'_{list})$:

$$cph' = [(T, \rho), R'_{list}, C', C'(1), [C'_i(2), C'_i(3)]_{i\in[1,l]}, C'_v(4)_{v\in cover(R'_{list})}]$$

where $C' = m.e(g, g)^{\beta.s'}$,
$C'_i(2) = g^{a.\rho(i).\lambda_{\rho(i)}}, C'_i(3) = g^{\lambda_{\rho(i)}}$.
For all, $v' \in cover(R_{list}) \cap cover(R'_{list}), C'_{v'}(4) = g^{s'}.P_{v'}^{s'}$
and for all, $x' \in cover(R'_{list}) - cover(R_{list}), C'_{v'}(4) = P'_{v'} = P_{v'}^{s'}$.

Here s and s' are random values respectively from Z_q, also both updated ciphertext (cph') and original ciphertext (cph^*) have similar number of terms. *Adv* cannot distinguish about the generation of ciphertext, i.e., from *Encryption* algorithm or *update* algorithm as the distribution of terms is identical in both. Here, if *Adv* can break the security of update ciphertext, then eventually it can break the original ciphertext's security.

7 Conclusion

Our paper focuses on CP-ABE scheme considering revocation along with its security proof-proven secured under DBDH assumption. An algorithm $\mathcal{B}$ is constructed, and it is assumed that if this algorithm would break the DBDH assumption, then Adv can break the security of our scheme. Also, we give ciphertext update parameter in addition with the scheme along with its security proof.

In future, we can give update verifiability to verify the correctness of updated ciphertext.

References

1. Boneh D; Franklin M; (2001), Identity-Based Encryption from the weil pairing, In CRYPTO volume 2139 of LNCS Springer-Verlag pp. 213–229.
2. Sahai A; Waters B; (2005), Fuzzy Identity-Based Encryption, EUROCRYPT, pp. 457–473.
3. Goyal V; Pandey O; Sahai A; Waters B; (2006), Attribute-based encryption for fine-grained access control of encrypted data, ACM Conference on Computer and Communications Security, pp. 89–98.
4. Bethencourt J; Sahai A; Waters B; (2007), Ciphertext-policy attribute-based encryption, IEEE Symposium on Security and Privacy, pp. 321–334.
5. Cheung L; Newport C; (2007), Provably Secure Ciphertext Policy ABE, ACM Conference on Computer and Communications Security, pp. 456–465.
6. Waters B; (2011), Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization, Public Key Cryptography, pp. 53–70.
7. Balu A; Kuppusamy K; (2013), An expressive and provably secure ciphertext-policy attribute-based encryption, Information Sciences 276, pp. 354–362.
8. Boldyreva A; Goyal V; Kumar V; (2008), Identity-based encryption with efficient revocation, ACM Conference on Computer and Communications Security, pp. 417–426.
9. Attrapadung N; Imai H; (2009), Conjunctive broadcast and attribute-based encryption, LNCS volume 5671 Springer-Verlag, pp. 248–265.
10. Attrapadung N; Imai H; (2009), Attribute-based encryption supporting direct/indirect revocation modes, Cryptography and Coding LNCS volume 5921 Springer-Verlag, pp. 278–300.
11. Shi Y; Zheng Q; Liu J; Han Z; (2015), Directly revocable key-policy attribute-based encryption with verifiable ciphertext delegation, Information Sciences 295, pp. 221–231.
12. Zhang P; Chen Z; Liang K; Wang S; Wang T; (2016), A Cloud-Based Access Control Scheme with User Revocation and Attribute Update, LNCS volume 9722 Springer, pp. 525–540.