6 Ciphertext-Policy Attribute-Based Encryption with Hidden Access Policy and User Revocation support

6.1 Introduction

The enhancement of BK-CP-ABE with hidden access policy and direct revocation of users has been discussed in the previous chapters. In this Chapter we combine both enhancements in a single scheme. To the best of our knowledge, our scheme is the first CP-ABE to provide hidden access policy and direct revocation in a single scheme. To achieve this, we restrict the access policy to possess only the AND operator, each attribute can take multiple values and the revoked set $S = \{Id_1, ..., Id_r\}$ is of r identities. If the user’s private key satisfies the access policy $P$ and the $ID \notin S$ then the algorithm will decrypt the ciphertext and return the original message.

6.2 Definition and Security Model for CP-ABE-HAPUR

Definition 14 (Access structure)

Let $U = \{a_1, a_2, ..., a_n\}$ be a set of attributes. For $a_i \in U$, $S_i = \{v_{i,1}, v_{i,2}, ..., v_{i,n_i}\}$ is a set of possible values, where $n_i$ is the number of possible values for $a_i$. Let $L = [l_1, l_2, ..., l_n]$ $l_i \in S_i$ be an attribute list for a user, and $P = [\omega_1, \omega_2, ..., \omega_n]$ $P_i \in S_i$ be an access structure. The notation $L \models P$ expresses that an attribute list $L$ satisfies an access structure $P$, namely $L_i = \omega_i (i = 1, 2, n)$. The notation $L \not\models P$ implies $L$ not satisfying the access structure $P$. 
CP-ABE-HAPUR consists of four fundamental algorithms: Setup, Key Generation, Encryption and Decryption.

**Setup:**

The setup algorithm takes no input other than the implicit security parameter. It outputs the public parameters PK and a master key MK.

**KeyGen (MK,PK, L, ID):**

The key generation algorithm takes as input the master key MK, public key PK, an identity ID and the attribute list L. It outputs a private key $SK_L$ for the attribute list L.

**Encrypt (S, PK, $\mathcal{P}$, m):**

The encryption algorithm takes as input a revocation set S of identities, public parameters PK, the message m, and an access policy $\mathcal{P}$ over the universe of attributes. The algorithm will encrypt m and produce a ciphertext CT such that any user with a key for an identity $ID \notin S$ and the attribute list L satisfies the access policy can decrypt.

**Decrypt(CT,SK_L, ID,S):**

The decryption algorithm takes as input the ciphertext CT that was generated for the revoked set S, as well as an identity and a private key $SK_L$ for the attribute list L. If the list L of attributes satisfies the access policy $\mathcal{P}$ and the $ID \notin S$ then
the algorithm will decrypt the ciphertext and return a message $M$.

6.2.1 Security Model for CP-ABE-HAPUR

The selective security notion for CP-ABE-HAPUR is defined in the following game.

**Init**

The adversary chooses the target set $S^*$ of Identities, challenge ciphertext policies $P_0, P_1$ to the challenger.

**Setup**

The challenger runs the Setup algorithm and gives the public parameters, $PK$ to the adversary.

**Phase1**

The adversary makes a secret key request to the KeyGen oracle for any attribute list $L$, user index $ID$, with the restriction that $ID \in S^*$ or $L$ does not satisfy the access policies $P_0, P_1$. The Challenger returns the output of $KeyGen(L, MK, ID, PK)$.

**Challenge**

The adversary submits two equal length messages $M_0$ and $M_1$. The Challenger flips a random coin $d$, and encrypts $M_d$ under $(P_d, S^*)$. The ciphertext $CT^*$ is given to the adversary.
Phase 2

The adversary can continue querying KeyGen with the same restriction as during Phase 1.

Guess

The adversary outputs a guess $d'$ of $d$.

**Definition 15** A ciphertext-policy attribute based encryption scheme is said to be secure against an adaptive chosen-plaintext attack (CPA) if any polynomial time adversaries has only a negligible advantage in the IND-CPA game, where the advantage is defined to be $\epsilon = |Pr[d' = d] - \frac{1}{2}|$.

6.3 CP-ABE-HAPUR Scheme

**Setup ($1^k$)**

The setup algorithm chooses a group $G_0$ of prime order $p$ and a generator $g$.

Let $U = \{a_1, a_2, ..., a_n\}$ be the set of attribute names.

For each attribute $a_i$ where $1 \leq i \leq n$ it chooses $n_q$ random values $t_{i,j} \in \mathbb{Z}_p$, and computes $T_{i,j} = g^{t_{i,j}} \{1 \leq j \leq n_q ; 1 \leq i \leq n\}$

Let $y = e(g, g)^{\alpha}$ where $\alpha \in \mathbb{Z}_p$.

Let $b$ be a random element $\in \mathbb{Z}_p$ and $h \in G$.

The Public Key is $PK = (g, g^b, h, y, T_{i,j}\{1 \leq j \leq n_q; 1 \leq i \leq n\})$ and the Master Secret Key is $MK = (\alpha, b, t_{i,j}\{1 \leq j \leq n_q; 1 \leq i \leq n\})$. 

76
KeyGen (ID,MK,PK,L)

This algorithm takes as input the user index $ID$, master secret key, public key and the attribute list of the user and performs the following:

Let $L = [l_1, l_2, .., l_n] = \{v_{1,t_1}, v_{2,t_2}, .., v_{n,t_n}\}$ be the attribute list of the user for which he obtains the corresponding secret key.

a) Select random values $a,r,\omega \in \mathbb{Z}_p$ and compute the following

\[
d_0 = g^a g^{ar} g^{b\omega}; d_2 = g^{\omega}; d_3 = (g^{bID_1})^{\omega}\]

b) For each attribute $v_{i,t_i} \in L$, compute $d^*_i = g^{art_i} \{1 \leq i \leq n\}$

The secret key is $SK_L = \{d_0, d_2, d_3, \forall l_i \in L : d^*_i\}$

Encrypt(S,PK, $\mathcal{P}$, m)

The encryption algorithm takes as input a user index set $S = \{ID_1, .., ID_r\}$, public key PK, message $m \in G_1$ to encrypt and the access policy $\mathcal{P} = [\omega_1, .., \omega_n]$.

**Step 1:** Select a random element $s \in [-2^\ell, 2^\ell]$ and compute $C_0 = g^s$.

M is the distribution matrix constructed for the access policy $\mathcal{P}$.

Choose $\rho = (s, \rho_2, ..., \rho_e)^T$, where $\rho_i$'s are uniformly random chosen integers in $[-2^\ell_0+k, 2^\ell_0+k]$.

**Step 2:**

a) Computes $M \cdot \rho = (s_1, ..., s_d)^T$
b) $C_1 = m \cdot g^s = m \cdot e(g, g)^{as}$; $C'_{k} = g^{sk}$; $C^+_{k} = (g^{bID_k h})^{sk}$; $k = 1$ to $r$

c) For each $v_i,t_i \in w_i$, calculate $C_{i,j} = T_{i,j}^{s_i}$ \{1 $\leq j \leq n_q$ ; $1 \leq i \leq n$\} using the corresponding shares of the attribute.

The ciphertext is published as $CT = (C_0, C_1, C^*_i, C'_k, C^+_k)$.

Decrypt($CT, SK_L, ID, S$)

The decryption algorithm takes as input the ciphertext $CT$, revoked set $S$ as well as the identity and a private key $SK_L$ for the attribute list $L$. By using Lemma1, it is possible to construct a reconstruction vector $\lambda_A \in Z^d$ such that $M_A^T \lambda_A = \xi$. With this, it is possible to reconstruct the secret using $\sum_{i \in A} \lambda_i s_i = s$.

If $ID \not\in S$ then the decryption algorithm computes

$$E = \prod_{i \in L} \frac{e(C_0, d_0)}{e(C^*_i, (d^*_i)^{s_i})} \prod_{k = 1}^q \left[ \frac{e(d_2, C^+_i)}{e(d_3, C'_k)} \right]^{\lambda_k_{ID_k}}$$

$$= e(g, g)^{as}$$

Note that this computation is defined if $ID \neq ID_k$ for $k = 1,..,r$. It then obtains

$$m = C_1 / E.$$
6.4 Security Analysis

Theorem 6.1 Suppose the DBDH assumption holds, then no polynomial adversary can selectively break the CP-ABE-HAPUR system.

Proof: We prove that our scheme is selectively secure under the DBDH assumption. The adversary commits to the challenge ciphertext policies $\mathcal{P}_0, \mathcal{P}_1$ in advance. We use a sequence of hybrid games to prove that the adversary cannot win the original security game denoted by $G$ with non-negligible probability. We begin by slightly modifying the game $G$ into a game $G_0$. Games $G$ and $G_0$ are the same except for how the challenge ciphertext is generated. In $G_0$, if $M_0 \neq M_1$, then the challenge ciphertext component $C_1$ is a random element of $G_T$ regardless of the random coin $d$, other parts of the ciphertext is generated in normal way. If $M_0 = M_1$, then the challenge ciphertext in $G_0$ is generated correctly, in this case, we have $G = G_0$.

In this theorem first we will prove that the difference of advantage of $A$ in game $G$ and game $G_0$ is negligible for any polynomial adversary $A$, then the game $G_0$ is modified by changing the ciphertext parts $C_{i,j}$. We define a sequence of games as follows. For $v_{i,t}$ such that $\{v_{i,t} \in \omega_{0,i} \land v_{i,t} \in \omega_{1,i}\}$ or $\{v_{i,t} \notin \omega_{0,i} \land v_{i,t} \notin \omega_{1,i}\}$, the components $C_{i,j}$ are generated as in the real scheme through the sequence of all the games.

If there is $v_{i,t}$ such that $\{v_{i,t} \in \omega_{0,i} \land v_{i,t} \notin \omega_{1,i}\}$ or $\{v_{i,t} \notin \omega_{0,i} \land v_{i,t} \in \omega_{1,i}\}$, the components $C_{i,j}$ generated properly in game $G_{i-1}$ are replaced with the random
values in the new modified game $G_i$ regardless of the random coin $b$. This process is repeated until there is no component satisfying $\{v_{i,t} \notin \omega_{0,i} \land v_{i,t} \in \omega_{1,i}\}$ or $\{v_{i,t} \notin \omega_{0,i} \land v_{i,t} \in \omega_{1,i}\}$. At the end of the sequence game, the advantage of the adversary is zero because the adversary is given a ciphertext chosen from the same distribution regardless of the random coin $d$.

Suppose we have an adversary $A$ with non-negligible advantage $\epsilon$ in the selective security game against our construction. We show how to use the adversary $A$ to build a simulator $B$ that is able to solve the DBDH assumption. The Challenger gives the simulator $B$ the DBDH challenge: $(g,A,B,C,D) = (g,g^a,g^b,g^s,D)$.

**Init.** The Adversary $A$ gives two challenge ciphertext policies $\mathcal{P}_0 = [\omega_{0,1},..,\omega_{0,n}]$, $\mathcal{P}_1 = [\omega_{1,1},..,\omega_{1,n}]$ and a revocation set $S^* = \{Id_1,Id_2,..,Id_r\}$ to the simulator. The simulator flips a fair binary coin $d \in \{0,1\}$.

**Setup** The simulator selects at random $a' \in Z_p$ and implicitly sets $\alpha = ab + a'$ by letting $e(g,g)^\alpha = e(g^a,g^b) e(g,g)^{a'}$. For all $v_{i,t} \in \omega_{d,i}$ chooses a random $q_i \in Z_p$ and set $T_i = g^{\left\lfloor \frac{1}{\sigma_i} \right\rfloor}$ if $v_{i,t} \notin \omega_{d,i}$, otherwise $T_i = g^{q_i}$. It selects $g^h = \prod_{1 \leq i \leq r^*} g^{q_i}$, $h = \prod_{1 \leq i \leq r^*} \{g^{q_i}\}^{-1}$, $g^y$. The simulator $B$ sends the above public parameters to $A$.

**Phase 1** In this phase the simulator answers private key queries. Suppose the simulator is given a private key for a list $L$ where $L$ does not satisfy $\mathcal{P}_d$ OR $ID \notin S^*$. On each request $B$ chooses a random variable $v, \delta \in Z_p$, and finds a vector $k =$
\((k_1, k_2, \ldots, k_e)^T \in \mathbb{Z}^e\) such that \(M' \cdot k = 0\) with \(k_1 = 1\). By the definition of Sweeping vector, such a vector must exist. The simulator sets \(r = v - k_j b\) and chooses \(k_j\) as \(k_1\) to compute, \(d_0 = g^{a} g^{ar} g^{b\omega} = \prod_{1 \leq j \leq n} g^{ab+a'} g^{a(v-k_j b)} g^{a_\delta} = \prod_{1 \leq j \leq n} g^{a'} A^v g^{a_\delta}\)

In calculating \(d_i^*\) we have the term \(M' a \cdot k_j b\) get cancelled because of \(M' \cdot k = 0\)  
\(d_i^* = g^{(v-k_j b)a_j} M'_i = A^{vM'_j} q_j\)

\(d_2 = g^{b}\)

\(d_3 = \prod_{1 \leq i,j \leq n} g^{(a_i-a_j)(I^D - I D_j)} g^{a_\delta}\)

**Challenge**  
A submits two messages \(m_0, m_1 \in G_1\), turns the encryption of \(m_d\). The encryption of \(m_d\) can be done as follows:

\(C_0 = g^s, C_1 = m_d De(g^s, g^{a'})\)

The simulator will choose uniformly random integers \(z_2, \ldots, z_h \in [-2^{\ell_0+k}, 2^{\ell_0+k}]\) and share the secret using the vector \(\Phi = (x, z_2, \ldots, z_h)\).

Create the distribution matrix \(M\), for the access policy \(P_d\). Compute \(M \cdot \Phi\) and use the shares to encrypt the access policy with corresponding \(q_j\) for the attributes present in the access policy i.e., \(v_{i,t_i} \in \omega_{d,i}\), \(C_{i,j} = T_{i,j}^{x_j}\).

**Phase 2**  
Same as Phase 1.

**Guess**  
A outputs a guess \(d'\) of \(d\). The simulator then outputs 1 to the guess if \(d' = d\), otherwise 0. By our assumption, the probability that \(A\) guesses \(d\) correctly in the game \(G\) has a non-negligible difference from that of guessing \(d\) correctly in \(G_0\).
When $D = e(g, g)^{xyz}$, $A$ is in game $G$ and when $D$ is random, $A$ is in game $G_0$. Therefore the simulator $B$ has the same advantage as the adversary’s in the DBDH game.

6.5 Summary

We constructed the CP-ABE-HAPUR scheme that can hide the access policy partially as well as revoke the users in a single scheme. To achieve this, we combine the Privacy Aware CP-ABE scheme and CP-ABE-UR scheme with the restricted access structure containing AND operators alone. Security analysis of this scheme is provided under the DBDH assumption.