Arbitrary-State Attribute-Based Encryption with Dynamic Membership

Speaker: Chun-I Fan

Chun-I Fan, National Sun Yat-sen University, Kaohsiung, Taiwan
Vincent Shi-Ming Huang, Industry Technology Research Institute, Hsinchu, Taiwan
He-Ming Ruan, National Taiwan University, Taipei, Taiwan
IEEE Transactions on Computers, 2014

Outline

Public Key Infrastructure (PKI)

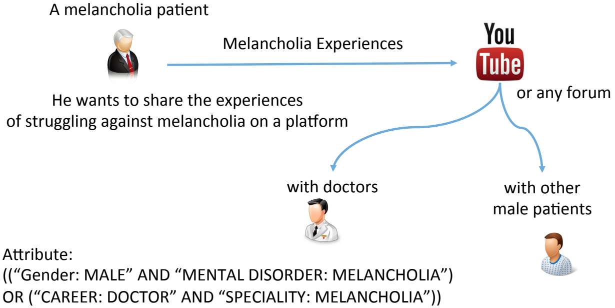
Traditional PKI has some features:

- A sender should get the public key (or the identity) of the receiver in advance.
- The more receivers the system has, the more bandwidth it consumes.
- Broadcast encryption may be able to solve the problem of performance, but the sender needs to have the receiver list.

Attribute-Based Encryption (ABE)

- In 1993, Fiat, and Naor proposed a broadcast encryption system.
- In 2001, Boneh and Franklin proposed an identity-based encryption (IBE) scheme based on Weil pairing, and enhanced it by the technique from Fujisaki and Okamoto.
- In 2005, Sahai and Waters proposed a new type of IBE, fuzzy IBE, which was the prototype of ABE.
- In 2007, Baek, Susilo, and Zhou presented another fuzzy IBE with new construction.

Our Contribution



- Our scheme is the first one which supports dynamic membership and arbitrary-state attributes.
- It is CCA secure under a standard model (without using random oracles).

Dynamic Membership

- **Expandability.** A new user is able to enroll in the system.
- **Renewability.** Each user's private key, attribute set, and attribute values can be renewed and the old private keys should be useless to those ciphertexts which are encrypted after these parameters associated with the user are renewed.
- **Revocability.** A user's private key can be revoked and the revoked private keys should be useless to those ciphertexts which are encrypted after these private keys are revoked.
- **Independence.** When a user's leaving or attribute updating occurs, the other users are not required to interact with KGC (Key Generation Center) to renew their private keys.

Backgrounds

Lagrange Interpolation

Lagrange interpolating polynomial is a polynomial p *of degree not greater than* $(n-1)$ *that passes through* n *points* $(x_i, y_1), \ldots, (x_n, y_n)$, *and is given by*

$$p(x) = \sum_{j=1}^{n} p_j(x), \textit{ where } p_j(x) = y_j \prod_{k=i,\ldots,n,k\neq j} \frac{x - x_k}{x_j - x_k}.$$

For $i \in \mathbb{Z}$ *and* $S \subseteq \mathbb{Z}$, *the Lagrange coefficient* $\Delta_{i,S}(x)$ *is defined as*

$$\Delta_{i,S}(x) = \prod_{\forall j \in S, j \neq i} \frac{x - j}{i - j}$$

Bilinear Mapping

Let G_0, G_1, *and* G_T *be three cyclic groups of prime order* q. *A bilinear mapping* $e : G_0 \times G_1 \rightarrow G_T$ *satisfies the following properties:*

- **Bilinearity:** $e(aP, bQ) = e(P, Q)^{ab}$, $\forall P \in G_0$, $Q \in G_1$ *and* $a, b \in \mathbb{Z}_q$.
- **Non-Degeneracy:** *The mapping does not map all pairs in* $G_0 \times G_1$ *to the identity in* G_T.
- **Computability:** *There is an efficient algorithm to compute* $e(P, Q), \forall P \in G_0, Q \in G_1$.

Decisional Bilinear Diffie-Hellman Problem

Let G_0, G_1, and G_T be three cyclic groups of prime order q, P and Q be arbitrarily-chosen generators of G_0 and G_1, respectively, and $e : G_0 \times G_1 \rightarrow G_T$ be a bilinear mapping. Given $(P, Q, aP, bP, cP, aQ, bQ, cQ, Z)$ for some $a, b, c \in \mathbb{Z}_q^*$ and $Z \in_R \{e(P, Q)^{abc}, Y \in_R G_T\}$, decide if $Z = e(P, Q)^{abc}$.

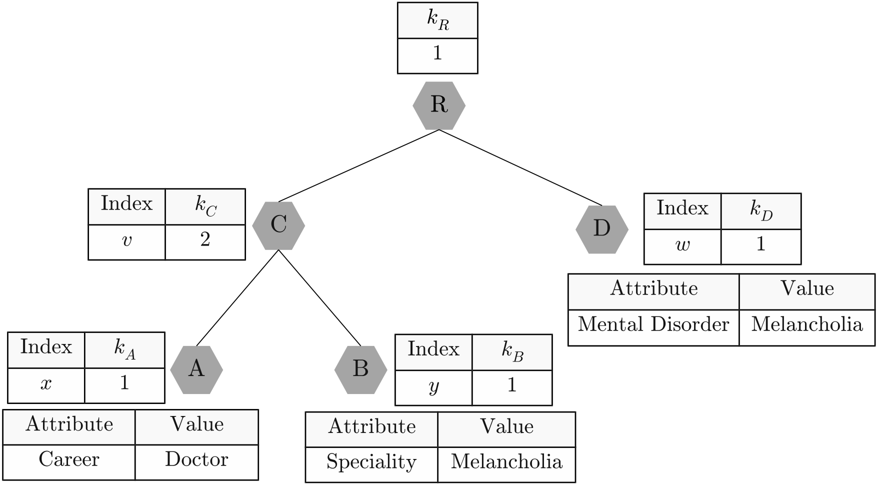
Definition 1 (The DBDH Assumption)

We define that an adversary $\mathcal{C}$ with an output $b' \in \{0,1\}$ has advantage ϵ' in solving the DBDH problem if

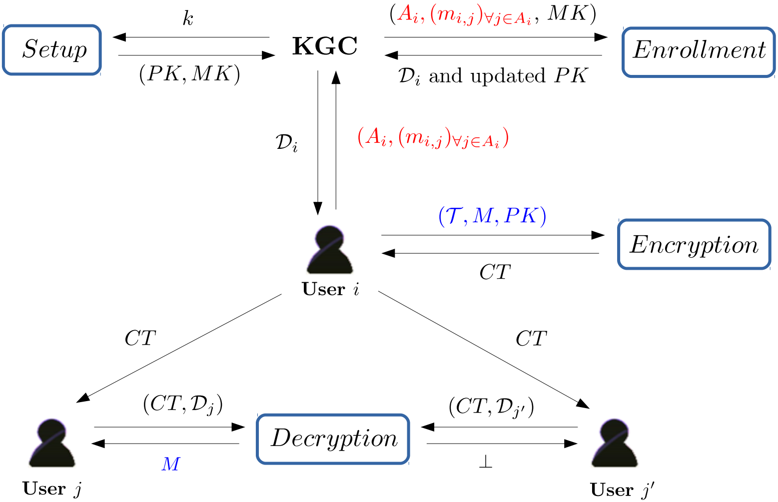
$$|Pr[\mathcal{C}(P,Q,aP,bP,cP,aQ,bQ,cQ,e(P,Q)^{abc}) = 1] - Pr[\mathcal{C}(P,Q,aP,bP,cP,aQ,bQ,cQ,Z) = 1]| \geq \epsilon'$$

where the probability is over the random choice of $a,b,c \in \mathbb{Z}_q^*$ and the random choice $Z \in \{e(P,Q)^{abc}, Y \in_R G_T\}$.

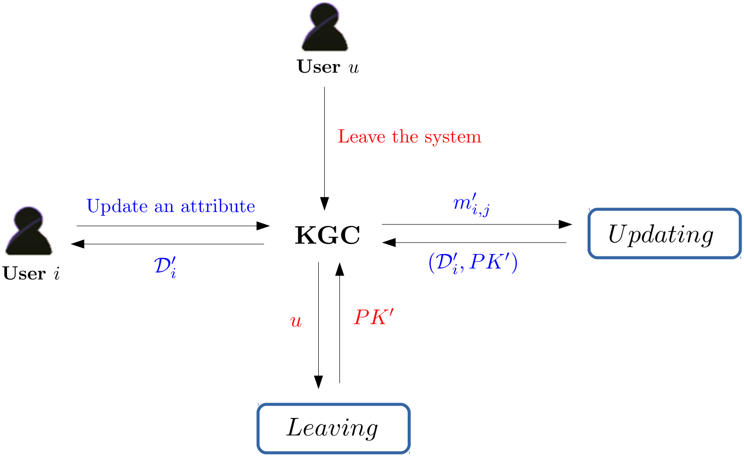
An Access Tree



Overview: *Enrollment*, *Encryption*, *Decryption*



The Proposed Scheme
Overview: *Leaving*, *Updating*



- **Step 1:** G_0, G_1, G_T with prime order q, $e : G_0 \times G_1 \rightarrow G_T$, generator P of G_0, generator Q of G_1
- **Step 2:** $\alpha, \beta \in_R \mathbb{Z}_q^*$ and $H : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$.
- **Step 3:** Generate two default users:
  1. $\mathcal{U} = \{0,1\}$, $\{v_{i,j}\}_{\forall i \in \mathcal{U}, j \in A} \in_R \mathbb{Z}_q^*$, $\{t_i\}_{\forall i \in \mathcal{U}} \in_R \mathbb{Z}_q^*$.
  2. $$\begin{cases} \{V_j = (\prod_{\forall i \in \mathcal{U}} v_{i,j})Q\}_{\forall j \in A} \\ \{\overline{v_{i,j}} = t_i \prod_{\forall k \neq i, k \in \mathcal{U}} v_{k,j}^{-1} + v_{i,j} \bmod q\}_{\forall i \in \mathcal{U}, \forall j \in A}. \end{cases}$$
- **Step 4:** Public parameter $PK = (G_0, G_1, G_T, e, H, P, Q, U = e(P,Q)^{\alpha(\beta-1)}, e(P,Q)^{\alpha\beta}, \{V_j, \{\overline{v_{i,j}}\}_{\forall i \in \mathcal{U}}\}_{\forall j \in A}, \mathbb{V})$
  The master key of KGC is $MK = \alpha Q$.

The asymmetric setting of e can be implemented by BN-Curve which is with faster software implementations.

- **Step 1:** $r_i, t_i, \{v_{i,j}\}_{\forall j \in A}, \{r_{i,j}\}_{\forall j \in A_i} \in_R \mathbb{Z}_q^*$
- **Step 2:** $\mathcal{U} = \mathcal{U} \cup \{i\}$
- **Step 3:**

$$\begin{cases} \{h_{i,j} = H(m_{i,j})\}_{\forall j \in A_i} \\ \{V_j = v_{i,j} V_j\}_{\forall j \in A} \\ \{\overline{v_{i,j}} = t_i \prod_{\forall k \neq i, k \in \mathcal{U}} v_{k,j}^{-1} + v_{i,j} \bmod q\}_{\forall j \in A} \\ \{\overline{v_{k,j}} = (\overline{v_{k,j}} - v_{k,j}) v_{i,j}^{-1} + v_{k,j} \bmod q\}_{\forall k \neq i, k \in \mathcal{U}, \forall j \in A} \end{cases}$$

- **Step 4:** Generate user i's private key $\mathcal{D}_i =$

$$\begin{cases} D_i = \alpha Q + t_i r_i Q \\ \{D_{i,j} = v_{i,j}^{-1}(r_i P + r_{i,j} h_{i,j} P)\}_{\forall j \in A_i} \\ \{D'_{i,j} = t_i r_{i,j} P\}_{\forall j \in A_i} \\ \{D''_{i,j} = r_i P + r_{i,j} h_{i,j} P\}_{\forall j \in A_i} \end{cases}$$

- **Step 5:** Increase $\mathbb{V}$ and update $(\{V_j, \{\overline{v_{i,j}}\}_{\forall i \in \mathcal{U}}\}_{\forall j \in A}, \mathbb{V})$ in PK.

KGC increases $\mathbb{V}$ and updates PK as follows:

$$\begin{cases} \{V_j = v_{u,j}^{-1} V_j\}_{\forall j \in A} \\ \{\overline{v_{k,j}} = (\overline{v_{k,j}} - v_{k,j}) v_{u,j} + v_{k,j} \bmod q\}_{\forall k \neq u, k \in \mathcal{U}, \forall j \in A} \end{cases}$$

Finally, it sets $\mathcal{U} = \mathcal{U} \backslash \{u\}$ and deletes $\{\overline{v_{u,j}}\}_{\forall j \in A}$ in PK.

- **Step 1:** $v'_{i,j}, r'_i, r'_{i,j} \in_R \mathbb{Z}_q^*$
- **Step 2:** $h'_{i,j} = H(m'_{i,j})$
- **Step 3:** Give

$$\begin{cases} D_i = \alpha Q + t_i r'_i Q \\ D_{i,j} = v'^{-1}_{i,j}(r'_i P + r'_{i,j} h'_{i,j} P) \\ \{D_{i,k} = v^{-1}_{i,k}(r'_i P + r_{i,k} h_{i,k} P)\}_{\forall k \in A_i \setminus \{j\}} \\ D'_{i,j} = t_i r'_{i,j} P \\ D''_{i,j} = r'_i P + r'_{i,j} h'_{i,j} P \\ \{D''_{i,k} = r'_i P + r_{i,k} h_{i,k} P)\}_{\forall k \in A_i \setminus \{j\}} \end{cases}$$

to user i.

- **Step 4:** Increase $\mathbb{V}$ and update PK.

$$\begin{cases} V_j = v^{-1}_{i,j} v'_{i,j} V_j \\ \overline{v_{i,j}} = (\overline{v_{i,j}} - v_{i,j}) + v'_{i,j} \mod q \\ \overline{v_{k,j}} = (\overline{v_{k,j}} - v_{k,j}) v_{i,j} v'^{-1}_{i,j} + v_{k,j} \mod q, \forall k \in \mathcal{U} \setminus \{i\} \end{cases}$$

Access tree structure construction.

- **For the root node R:**
  1. $s \in_R \mathbb{Z}_q^*$ and k_R is the threshold value of R.
  2. Randomly choose a polynomial q_R of degree $d_R = k_R - 1$ with $q_R(0) = s$.
  3. Assign a unique index number x for each child of R.
- **For each internal node N other than R:**
  1. k_N is the threshold value of N.
  2. Randomly choose a polynomial q_N of degree $d_N = k_N - 1$ with $q_N(0) = q_{parent(N)}(index(N))$.
  3. Assign a unique index number x for each child of node N.
- **For each leaf node N_L:** Randomly choose a polynomial q_{N_L} of degree 0 with $q_{N_L}(0) = q_{parent(N_L)}(index(N_L))$.

Ciphertext generation: $K \in_R G_T$ and $\mathcal{N}_L = \{\text{the leaves of } \mathcal{T}\}$.

The ciphertext is

$$CT = (\mathcal{T}, \tilde{C} = e(P,Q)^{\alpha\beta s}K, C = sP, C' = U^s, \overline{M} = E_K(M), C_r = H(K||\overline{M})P, \{C_N = q_N(0)V_{att(N)}, C'_N = q_N(0)H(val(N))Q, \{\overline{v_{i,att(N)}}\}_{\forall i \in \mathcal{U}}\}_{\forall N \in \mathcal{N}_L}, \mathbb{V}).$$

$\{\{\overline{v_{i,att(N)}}\}_{\forall i \in \mathcal{U}}\}_{\forall N \in \mathcal{N}_L}$ can be excluded from the ciphertext.

$DecryptNode(CT, \mathcal{D}_i, N)$: Let $V_j = v_j P$.

- **If N is a leaf node:** Let $j = att(N)$. If j is in A_i and $m_{i,j} = val(N)$, then

$$
\begin{aligned}
& DecryptNode(CT, \mathcal{D}_i, N) \\
&= \frac{e(D_{i,j}, \overline{v_{i,j}} C_N)}{e(D'_{i,j}, C'_N) e(D''_{i,j}, C_N)} \\
&= \frac{e(v_{i,j}^{-1}(r_i P + r_{i,j} h_{i,j} P), (t_i v_j^{-1} v_{i,j} + v_{i,j}) q_N(0) v_j Q)}{e(t_i r_{i,j} P, q_N(0) h_{i,j} Q) e(r_i P + r_{i,j} h_{i,j} P, v_j q_N(0) Q)} \\
&= \frac{e(v_{i,j}^{-1}(r_i P + r_{i,j} h_{i,j} P), t_i v_{i,j} q_N(0) Q + v_{i,j} v_j q_N(0) Q)}{e(t_i r_{i,j} P, q_N(0) h_{i,j} Q) e(r_i P + r_{i,j} h_{i,j} P, v_j q_N(0) Q)} \\
&= \frac{e(r_i P + r_{i,j} h_{i,j} P, t_i q_N(0) Q) e(r_i P + r_{i,j} h_{i,j} P, v_j q_N(0) Q)}{e(t_i r_{i,j} P, q_N(0) h_{i,j} Q) e(r_i P + r_{i,j} h_{i,j} P, v_j q_N(0) Q)} \\
&= \frac{e(r_i P, t_i q_N(0) Q) e(r_{i,j} h_{i,j} P, t_i q_N(0) Q)}{e(t_i r_{i,j} P, q_N(0) h_{i,j} Q)} \\
&= e(P, Q)^{t_i r_i q_N(0)}.
\end{aligned}
$$

Otherwise, $DecryptNode(CT, \mathcal{D}_i, N) = \perp$.

- **If N is an internal node**:

1. For each child N_c of N, $F_{N_c} = DecryptNode(CT, \mathcal{D}_i, N_c)$.
2. Let $\mathcal{I}_c$ be a k_N-sized set containing the indexes of the child nodes N_c's such that $F_{N_c} \neq \perp$ for each N_c. Return

$$
\begin{aligned}
F_N &= \prod_{\forall index(N_c)=z\in\mathcal{I}_c} F_{N_c}^{\Delta_{z,\mathcal{I}_c}(0)} \\
&= \prod_{\forall index(N_c)=z\in\mathcal{I}_c} (e(P,Q)^{t_i r_i q_{N_c}(0)})^{\Delta_{z,\mathcal{I}_c}(0)} \\
&= \prod_{\forall index(N_c)=z\in\mathcal{I}_c} (e(P,Q)^{t_i r_i q_{parent(N_c)}(z)})^{\Delta_{z,\mathcal{I}_c}(0)} \\
&= \prod_{\forall index(N_c)=z\in\mathcal{I}_c} (e(P,Q)^{t_i r_i q_N(z)})^{\Delta_{z,\mathcal{I}_c}(0)} \\
&= e(P,Q)^{t_i r_i q_N(0)}
\end{aligned}
$$

.

3. If no such set exists, N is not satisfied and return $F_N = \perp$.

Call $DecryptNode(CT, \mathcal{D}_i, R)$:

$$\begin{aligned} A &= DecryptNode(CT, \mathcal{D}_i, R) \\ &= e(P, Q)^{t_i r_i q_R(0)} \\ &= e(P, Q)^{t_i r_i s} \end{aligned}$$

Compute the session key K:

$$\begin{aligned} \frac{A \cdot \tilde{C}}{e(C, D_i) \cdot C'} &= \frac{e(P,Q)^{t_i r_i s} \cdot e(P,Q)^{\alpha\beta s} K}{e(sP, (\alpha + t_i r_i)Q) \cdot e(P,Q)^{\alpha(\beta-1)s}} \\ &= \frac{e(P,Q)^{\alpha\beta s} \cdot e(P,Q)^{t_i r_i s} K}{e(P,Q)^{(\alpha s + \alpha(\beta-1)s + t_i r_i s)}} \\ &= \frac{e(P,Q)^{\alpha\beta s + t_i r_i s} K}{e(P,Q)^{\alpha\beta s + t_i r_i s}} \\ &= K \end{aligned}$$

The decryption procedure will return $\perp$ if $C_r \neq H(K||\overline{M})P$.
Otherwise, it returns M by computing $M = D_K(\overline{M})$.

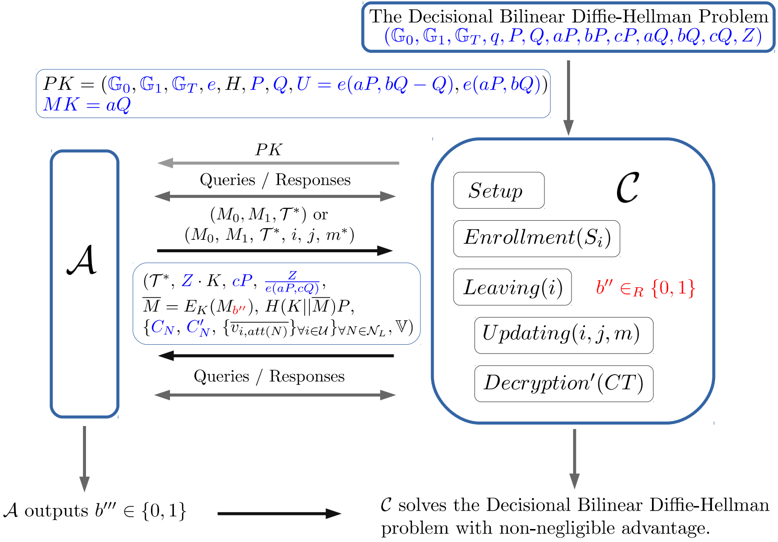
Dynamic Membership

- The *Enrollment* algorithm:
  1. Expandability
- The *Leaving* and *Updating* algorithms:
  1. Revocability
  2. Renewability
  3. Independence

Security Proof

Theorem

The proposed CP-ABE-DM scheme is CCA_{DM} *secure under the DBDH assumption in a standard model.*



If $b''' = b''$, $\mathcal{C}$ outputs $b' = 1$, otherwise $b' = 0$

$Cipher(\mathcal{T}, N, \mathbb{C}_0)$

Locate the node N in $\mathcal{T}$;
If (N is a leaf node) {
    Set $j = att(N)$ and $m = val(N)$;
    Compute $u_m = H(m)$;
    Compute $C_N = v_j \mathbb{C}_0$; // $v_j = \prod_{\forall i \in \mathcal{U}} v_{i,j}$
    Compute $C'_N = u_m \mathbb{C}_0$;
    Store (C_N, C'_N) in $\mathcal{L}_C$; }
Else {
    Randomly select $k_N - 1$ elements $c_i \in \mathbb{Z}_q^*$;
    For each child N_C of the node N {
        Set $x = index(N_C)$;
        If ($k_N - 1 > 0$) {
            Compute $\overline{\mathbb{C}_0} = \mathbb{C}_0 + \sum_{i=1}^{k_N - 1} c_i x^i Q$; }
        Else {
            Set $\overline{\mathbb{C}_0} = \mathbb{C}_0$; }
        Call $Cipher(\mathcal{T}, N_C, \overline{\mathbb{C}_0})$; } }

Initially, $\mathcal{C}$ calls $Cipher(\mathcal{T}^*, R, cQ)$.

Comparisons

- **Dynamic Membership**: It allows an ABE system to manage member enrollment, attribute updating, and member revocation efficiently.
- **Sender Updating**: A sender must grab the newest public information before she/he encrypts a message.
- **Receiver Updating**: A receiver must interact with the system to refresh her/his private key or retrieve the newest public information before she/he decrypts a ciphertext.
- **No Private Key Refreshment**: Members do not have to interact with the system to refresh their private keys when the membership or any of the members' attributes has been changed.
- **Arbitrary-State Attribute**: The domain of each attribute is a variable-length string, not a binary bit only.

Feature Comparisons

Scheme	Dynamic Membership		ASA	Special Feature
	Updating	Leaving		
Ours	Yes	Yes	Yes	No Private Key Refreshment
[4]	No	No	Yes	Direct and Indirect Revocation
[5]	No	Yes	No	Multi-Authority
[12]	No	No	No	Dual Policy
[13]	No	No	No	Multi-Authority
[15]	No	No	No	Multi-Authority
[7]	No	No	No	Full Logic Expression
[8]	No	No	No	Key Delegation
[9]	No	No	No	Multi-Authority
[10]	No	No	No	-

Feature Comparisons

Scheme	Dynamic Membership		ASA	Special Feature
	Updating	Leaving		
Ours	Yes	Yes	Yes	No Private Key Refreshment
[11]	No	No	No	-
[14]	No	No	No	-
[25]	No	No	No	Full Logic Expression
[26]	No	No	No	
[27]	No	No	No	Multi-Authority
[28]	No	No	No	Multi-Authority
[29]	No	No	No	Attribute Hierarchy
[30]	No	No	Yes	Unbounded Attribute
[31]	No	No	No	Constant Ciphertext Size
[32]	No	No	No	Multi-Authority
[33]	No	Yes	No	-
ASA: Arbitrary-State Attribute				

Notations for Performance Comparisons

Notation	Meaning
n	the number of the members in an ABE system
m	the number of the attributes provided in an ABE system
m_c	the number of attributes associated to a ciphertext
m_d	the maximum of $\|Cover(R)\|$ where R is the set of revoked users and $m_d < m$
m_u	the number of a user's attributes
m_a	the number of authorities
m_{or}	the number of OR gates of the access rule associated to a ciphertext
m_{and}	the number of the AND conjunction attributes of the access rule associated to a ciphertext
m_{not}	the number of the NOT conjunction attributes of the access rule associated to a ciphertext
DR, IR	Direct Mode and Indirect Mode
SA, OA	Subject Attribute and Objected Attribute

Performance Comparisons: Computation Cost

	Encryption Cost of the Sender	Decryption Cost of the Receiver	The Necessary Computation Cost of the Center		
			Enrollment	Updating	Leaving
Ours	$\mathcal{O}(m_c)$	$\mathcal{O}(m_c)$	$\mathcal{O}(m+m_u)$	$\mathcal{O}(m+m_u)$	$\mathcal{O}(m)$
[4]	$\mathcal{O}(m \times m_c + m_d)$	$\mathcal{O}(m_c)$	$\mathcal{O}(m \times m_p \times m_u)$	$\mathcal{O}(n \times m_p \times m_u \times m)$	DR: $\mathcal{O}(1)$, IR: $\mathcal{O}(n \times m_d^2)$
[5]	$\mathcal{O}(m)$	$\mathcal{O}(m_c)$	$\mathcal{O}(m \times m_u)$	$\mathcal{O}(n \times m \times m_u)$	$\mathcal{O}(1)$
[12]	$\mathcal{O}(m \times m_c)$	$\mathcal{O}(m_c)$	$\mathcal{O}(m \times m_u)$	SA: $\mathcal{O}(n \times m)$, DA: $\mathcal{O}(n \times m \times m_u)$	$\mathcal{O}(n \times m \times m_u)$
[13]	$\mathcal{O}(m_{and} \times m_{or})$	$\mathcal{O}(m_c)$	$\mathcal{O}(m_u)$	$\mathcal{O}(n)$	$\mathcal{O}(n \times m_u)$
[15]	$\mathcal{O}(m_c)$	$\mathcal{O}(m_c)$	$\mathcal{O}(m^2)$	$\mathcal{O}(n \times m^2)$	$\mathcal{O}(n \times m^2)$
[7]	$\mathcal{O}(m_c)$	$\mathcal{O}(m_c + m_u)$	$\mathcal{O}(m_u)$	$\mathcal{O}(n \times m_u)$	$\mathcal{O}(n \times m_u)$
[8]	$\mathcal{O}(m_c)$	$\mathcal{O}(m_c)$	$\mathcal{O}(m)$	$\mathcal{O}(n \times m)$	$\mathcal{O}(n \times m)$
[9]	$\mathcal{O}(m_c)$	$\mathcal{O}(m_c \times m_L)$	$\mathcal{O}(m)$	$\mathcal{O}(n)$	$\mathcal{O}(n \times m)$
[10]	$\mathcal{O}(m_{and} + m_{not})$	$\mathcal{O}(m_u + m_{not})$	$\mathcal{O}(m_u)$	$\mathcal{O}(n \times m_u)$	$\mathcal{O}(n \times m_u)$
[11]	$\mathcal{O}(m_c)$	$\mathcal{O}(m_c)$	$\mathcal{O}(m_u)$	$\mathcal{O}(n)$	$\mathcal{O}(n \times m_u)$
[14]	$\mathcal{O}(m \times m_c + m_c^2)$	$\mathcal{O}(m \times m_c m_c^2)$	$\mathcal{O}(m_u^2)$	$\mathcal{O}(n)$	$\mathcal{O}(n \times m_u^2)$
[25]	$\mathcal{O}(m)$	$\mathcal{O}(m_u)$	$\mathcal{O}(m_u)$	$\mathcal{O}(n \times m_u)$	$\mathcal{O}(n \times m_u)$
[26]	$\mathcal{O}(m)$	$\mathcal{O}(m^2)$	$\mathcal{O}(m)$	$\mathcal{O}(n \times m)$	$\mathcal{O}(n \times m)$
[27]	$\mathcal{O}(m)$	$\mathcal{O}(m_c \times m_u)$	$\mathcal{O}(m_a \times m_u)$	$\mathcal{O}(n \times m_a \times m_u)$	$\mathcal{O}(n \times m_a \times m_u)$
[28]	$\mathcal{O}(m_a + m_c)$	$\mathcal{O}(m_a \times m_u)$	$\mathcal{O}(m_a \times m_u)$	$\mathcal{O}(n \times m_a \times m_u)$	$\mathcal{O}(n \times m_a \times m_u)$
[29]	$\mathcal{O}(m_c^2)$	$\mathcal{O}(m_c^2)$	$\mathcal{O}(m_u)$	$\mathcal{O}(n \times m_u)$	$\mathcal{O}(n \times m_u)$
[30]	$\mathcal{O}(m_c)$	$\mathcal{O}(m_c)$	$\mathcal{O}(m_u)$	$\mathcal{O}(n \times m_u)$	$\mathcal{O}(n \times m_u)$
[31]	$\mathcal{O}(m)$	$\mathcal{O}(m_c)$	$\mathcal{O}(m \times m_u)$	$\mathcal{O}(n \times m \times m_u)$	$\mathcal{O}(n \times m \times m_u)$
[32]	$\mathcal{O}(m_c)$	$\mathcal{O}(m_c^2)$	$\mathcal{O}(m_u)$	$\mathcal{O}(n \times m_u)$	$\mathcal{O}(n \times m_u)$
[33]	$\mathcal{O}(m_c)$	$\mathcal{O}(m_c)$	$\mathcal{O}(m_u)$	$\mathcal{O}(n \times m_u)$	$\mathcal{O}(n \times m_u)$

Performance Comparisons: Storage and Communication Cost

	Size of Private Key	Size of Ciphertext	Size of Public Parameters	The Communication Cost (between the center and a user)		
				Enrollment	Updating	Leaving
Ours	$\mathcal{O}(m_u)$	$\mathcal{O}(m_c)$	$\mathcal{O}(n \times m)$	$\mathcal{O}(m_u)$	$\mathcal{O}(m_u)$	$\mathcal{O}(1)$
[4]	$\mathcal{O}(m_u)$	$\mathcal{O}(m_c)$	$\mathcal{O}(n + m)$	$\mathcal{O}(m_u \times m_p)$	$\mathcal{O}(n \times m_u \times m_p)$	DR: $\mathcal{O}(1)$, IR: $\mathcal{O}(m_d)$
[5]	$\mathcal{O}(m_u)$	$\mathcal{O}(m_c)$	$\mathcal{O}(n + m)$	$\mathcal{O}(m_u)$	$\mathcal{O}(n \times m_u)$	$\mathcal{O}(1)$
[12]	$\mathcal{O}(m_u)$	$\mathcal{O}(m_c)$	$\mathcal{O}(n + m)$	$\mathcal{O}(m_u)$	$\mathcal{O}(n \times m_u)$	$\mathcal{O}(n \times m_u)$
[13]	$\mathcal{O}(m_u)$	$\mathcal{O}(m_c)$	$\mathcal{O}(n + m)$	$\mathcal{O}(m_u)$	$\mathcal{O}(n \times m_u)$	$\mathcal{O}(n \times m_u)$
[15]	$\mathcal{O}(m)$	$\mathcal{O}(m_c)$	$\mathcal{O}(m^2)$	$\mathcal{O}(m^2)$	$\mathcal{O}(n \times m^2)$	$\mathcal{O}(n \times m^2)$
[7]	$\mathcal{O}(m_u)$	$\mathcal{O}(m_c)$	$\mathcal{O}(m)$	$\mathcal{O}(m_u)$	$\mathcal{O}(n \times m_u)$	$\mathcal{O}(n \times m_u)$
[8]	$\mathcal{O}(m)$	$\mathcal{O}(m_c)$	$\mathcal{O}(1)$	$\mathcal{O}(m)$	$\mathcal{O}(n \times m)$	$\mathcal{O}(n \times m)$
[9]	$\mathcal{O}(m)$	$\mathcal{O}(m_c)$	$\mathcal{O}(m)$	$\mathcal{O}(m_u)$	$\mathcal{O}(n \times m_u)$	$\mathcal{O}(n \times m_u)$
[10]	$\mathcal{O}(m_u)$	$\mathcal{O}(m_{and} + m_{or})$	$\mathcal{O}(m)$	$\mathcal{O}(m_u)$	$\mathcal{O}(n \times m_u)$	$\mathcal{O}(n \times m_u)$
[11]	$\mathcal{O}(m_u)$	$\mathcal{O}(m_c)$	$\mathcal{O}(m)$	$\mathcal{O}(m_u)$	$\mathcal{O}(n)$	$\mathcal{O}(n \times m_u)$
[14]	$\mathcal{O}(m_L^2 \times m_u)$	$\mathcal{O}(m_L \times m_c)$	$\mathcal{O}(m \times m_L)$	$\mathcal{O}(m_L^2 \times m_u)$	$\mathcal{O}(n \times m_L)$	$\mathcal{O}(n \times m_L \times m_u)$
[25]	$\mathcal{O}(m_u)$	$\mathcal{O}(m_c)$	$\mathcal{O}(m)$	$\mathcal{O}(m_u)$	$\mathcal{O}(n \times m_u)$	$\mathcal{O}(n \times m_u)$
[26]	$\mathcal{O}(m)$	$\mathcal{O}(m)$	$\mathcal{O}(m)$	$\mathcal{O}(m)$	$\mathcal{O}(n \times m)$	$\mathcal{O}(n \times m)$
[27]	$\mathcal{O}(m_a \times m_u)$	$\mathcal{O}(m_a \times m_c)$	$\mathcal{O}(m \times m_a)$	$\mathcal{O}(m_a \times m_u)$	$\mathcal{O}(n \times m_a \times m_u)$	$\mathcal{O}(n \times m_a \times m_u)$
[28]	$\mathcal{O}(m_a \times m_u)$	$\mathcal{O}(m_a \times m_c)$	$\mathcal{O}(n \times m_a \times m)$	$\mathcal{O}(m_a \times m_u)$	$\mathcal{O}(n \times m_a \times m_u)$	$\mathcal{O}(n \times m_a \times m_u)$
[29]	$\mathcal{O}(m_u)$	$\mathcal{O}(m_c)$	$\mathcal{O}(m)$	$\mathcal{O}(m_u)$	$\mathcal{O}(n \times m_u)$	$\mathcal{O}(n \times m_u)$
[30]	$\mathcal{O}(m_u)$	$\mathcal{O}(m_c)$	$\mathcal{O}(1)$	$\mathcal{O}(m_u)$	$\mathcal{O}(n \times m_u)$	$\mathcal{O}(n \times m_u)$
[31]	$\mathcal{O}(m \times m_u)$	$\mathcal{O}(1)$	$\mathcal{O}(m)$	$\mathcal{O}(m \times m_u)$	$\mathcal{O}(n \times m \times m_u)$	$\mathcal{O}(n \times m \times m_u)$
[32]	$\mathcal{O}(m_u)$	$\mathcal{O}(m_c)$	$\mathcal{O}(m)$	$\mathcal{O}(m_u)$	$\mathcal{O}(n \times m_u)$	$\mathcal{O}(n \times m_u)$
[33]	$\mathcal{O}(m_u)$	$\mathcal{O}(m)$	$\mathcal{O}(m)$	$\mathcal{O}(m_u)$	$\mathcal{O}(n \times m_u)$	$\mathcal{O}(n \times m_u)$

Conclusions

- An attribute-based encryption scheme with dynamic membership has been proposed.
- This is the first ABE scheme which can support arbitrary-state attributes and attribute (and value) updating with Sender Updating only.
- It has been formally proved to be CCA secure under a standard model.


Thanks for Listening