Attribute-Based Cryptography: Survey and (Inefficient?) Generic Constructions

Javier Herranz

Universitat Politècnica de Catalunya
Barcelona, Spain

DoE CRYPTODOC, Darmstadt (Germany), November 21st, 2011

Outline

Outline

Traditional Public Key Cryptography

- Each user has a pair of keys $(\mathsf{sk}, \mathsf{pk})$.
- Each ciphertext / signature is **linked** to a particular public key pk.

Traditional Public Key Cryptography

- Each user has a pair of keys $(\mathsf{sk}, \mathsf{pk})$.
- Each ciphertext / signature is **linked** to a particular public key pk.
- Only the user holding the matching sk can decrypt / sign material linked to pk.

Traditional Public Key Cryptography

- Each user has a pair of keys $(\mathsf{sk}, \mathsf{pk})$.
- Each ciphertext / signature is **linked** to a particular public key pk.
- Only the user holding the matching sk can decrypt / sign material linked to pk.
- How to know that public key pk really belongs to the intended receiver ?
  Digital certificates, revocation... **inefficiency !!**

IBC → fuzzy IBC → ABC

Identity Based Cryptography (Shamir, 1984)

Only the owner of the identity which **exactly matches** the chosen identity can decrypt / sign the message.

IBC → fuzzy IBC → ABC

fuzzy Identity Based Cryptography (Sahai-Waters, 2005)

Identities are now **vectors of attributes**.

Only the owners of identities which **match** the chosen identity in **at least** t **positions** can decrypt / sign the message.

[The threshold t is fixed in Setup.]

IBC → fuzzy IBC → ABC

(Threshold) **Attribute Based Cryptography** (Goyal et al., 2006)

Only the owners of identities which **match** the identity chosen by the sender in **at least** t **positions** can decrypt the message.

[The threshold t is **chosen ad-hoc** by the sender / signer.]

Threshold ABC $\rightarrow$ General ABC

More generally, the sender / signer chooses a set of attributes:

$$S = \{\mathsf{at}_1, \mathsf{at}_2, \ldots, \mathsf{at}_n\}$$

and a **(monotone increasing) family** $\Gamma \subset 2^S$ of subsets of S.

Threshold ABC → General ABC

More generally, the sender / signer chooses a set of attributes:

$$S = \{\mathrm{at}_1, \mathrm{at}_2, \ldots, \mathrm{at}_n\}$$

and a **(monotone increasing) family** $\Gamma \subset 2^S$ of subsets of S.

Only users holding a subset of attributes $A \in \Gamma$ can decrypt / sign.

Threshold ABC $\rightarrow$ General ABC

More generally, the sender / signer chooses a set of attributes:

$$S = \{\mathsf{at}_1, \mathsf{at}_2, \ldots, \mathsf{at}_n\}$$

and a **(monotone increasing) family** $\Gamma \subset 2^S$ of subsets of S.

Only users holding a subset of attributes $A \in \Gamma$ can decrypt / sign.

Example: $S = \{\mathsf{at}_1, \mathsf{at}_2, \mathsf{at}_3\}$

$$\Gamma_0 = \{\{\mathsf{at}_1\}, \{\mathsf{at}_2, \mathsf{at}_3\}\}, \quad \Gamma = \mathrm{cl}(\Gamma_0)$$

Threshold ABC → General ABC

More generally, the sender / signer chooses a set of attributes:

$$S = \{\mathrm{at}_1, \mathrm{at}_2, \ldots, \mathrm{at}_n\}$$

and a **(monotone increasing) family** $\Gamma \subset 2^S$ of subsets of S.

Only users holding a subset of attributes $A \in \Gamma$ can decrypt / sign.

Example: $S = \{\mathrm{at}_1, \mathrm{at}_2, \mathrm{at}_3\}$

$$\Gamma_0 = \{\{\mathrm{at}_1\}, \{\mathrm{at}_2, \mathrm{at}_3\}\}, \quad \Gamma = \mathrm{cl}(\Gamma_0)$$

- User with $\{\mathrm{at}_1, \mathrm{at}_2\}$ **can** decrypt / sign.

Threshold ABC → General ABC

More generally, the sender / signer chooses a set of attributes:

$$S = \{\mathsf{at}_1, \mathsf{at}_2, \ldots, \mathsf{at}_n\}$$

and a **(monotone increasing) family** $\Gamma \subset 2^S$ of subsets of S.

Only users holding a subset of attributes $A \in \Gamma$ can decrypt / sign.

Example: $S = \{\mathsf{at}_1, \mathsf{at}_2, \mathsf{at}_3\}$

$$\Gamma_0 = \{\{\mathsf{at}_1\}, \{\mathsf{at}_2, \mathsf{at}_3\}\}, \quad \Gamma = \mathrm{cl}(\Gamma_0)$$

- User with $\{\mathsf{at}_1, \mathsf{at}_2\}$ **can** decrypt / sign.
- User with $\{\mathsf{at}_3\}$ cannot decrypt / sign.

Threshold ABC → General ABC

More generally, the sender / signer chooses a set of attributes:

$$S = \{\mathrm{at}_1, \mathrm{at}_2, \ldots, \mathrm{at}_n\}$$

and a **(monotone increasing) family** $\Gamma \subset 2^S$ of subsets of S.

Only users holding a subset of attributes $A \in \Gamma$ can decrypt / sign.

Example: $S = \{\mathrm{at}_1, \mathrm{at}_2, \mathrm{at}_3\}$

$$\Gamma_0 = \{\{\mathrm{at}_1\}, \{\mathrm{at}_2, \mathrm{at}_3\}\}, \quad \Gamma = \mathrm{cl}(\Gamma_0)$$

- User with $\{\mathrm{at}_1, \mathrm{at}_2\}$ **can** decrypt / sign.
- User with $\{\mathrm{at}_3\}$ cannot decrypt / sign.

Considering $\Gamma = \{A \subset S \ : \ |A| \geq t\}$, we recover the **threshold** case.

CP-ABC: Setup and Key Extraction

- SETUP: master entity runs $(\mathsf{params}, \mathsf{msk}) \leftarrow \mathsf{ABE.Setup}(1^\lambda, \mathcal{P})$, where $\mathcal{P}$ is the total **universe of attributes**.

CP-ABC: Setup and Key Extraction

- SETUP: master entity runs $(\mathsf{params}, \mathsf{msk}) \leftarrow \mathsf{ABE.Setup}(1^\lambda, \mathcal{P})$, where $\mathcal{P}$ is the total **universe of attributes**.
- KEY EXTRACTION: user U **proves** to master entity possession of his attributes $A = \{\mathsf{at}_{i_1}, \ldots, \mathsf{at}_{i_\ell}\} \subset \mathcal{P}$.

CP-ABC: Setup and Key Extraction

- SETUP: master entity runs $(\mathsf{params}, \mathsf{msk}) \leftarrow \mathsf{ABE.Setup}(1^\lambda, \mathcal{P})$, where $\mathcal{P}$ is the total **universe of attributes**.
- KEY EXTRACTION: user U **proves** to master entity possession of his attributes $A = \{\mathsf{at}_{i_1}, \ldots, \mathsf{at}_{i_\ell}\} \subset \mathcal{P}$.
- Master entity gives to U the **secret key** $\mathsf{sk}_A \leftarrow \mathsf{ABE.Ext}(\mathsf{params}, A, \mathsf{msk})$.

ABE: Encryption and Decryption

- ENCRYPTION: to encrypt a message M, sender chooses a set of attributes $S \subset \mathcal{P}$ and a monotone increasing **decryption policy** $\Gamma \subset 2^S$, and runs $C \leftarrow \text{ABE.Enc}(\text{params}, S, \Gamma, M)$.

ABE: Encryption and Decryption

- ENCRYPTION: to encrypt a message M, sender chooses a set of attributes $S \subset \mathcal{P}$ and a monotone increasing **decryption policy** $\Gamma \subset 2^S$, and runs $C \leftarrow \mathrm{ABE.Enc}(\mathrm{params}, S, \Gamma, M)$.
- DECRYPTION: a user holding attributes $A \subset S$ tries to decrypt by running $\tilde{M} \leftarrow \mathrm{ABE.Dec}(\mathrm{params}, C, \Gamma, \mathrm{sk}_A)$.

ABE: Encryption and Decryption

- ENCRYPTION: to encrypt a message M, sender chooses a set of attributes $S \subset \mathcal{P}$ and a monotone increasing **decryption policy** $\Gamma \subset 2^S$, and runs $C \leftarrow \text{ABE.Enc}(\text{params}, S, \Gamma, M)$.
- DECRYPTION: a user holding attributes $A \subset S$ tries to decrypt by running $\tilde{M} \leftarrow \text{ABE.Dec}(\text{params}, C, \Gamma, \text{sk}_A)$.

  [For correctness: $\tilde{M} = M \iff A \in \Gamma$.]

ABS: Signature and Verification

- SIGNATURE: to sign a message M for a **signing policy** (S, Γ), where $\Gamma \subset 2^S$, a signer holding attributes $A \subset S$ runs $\sigma \leftarrow \mathrm{ABS.Sign}(\mathrm{params}, S, \Gamma, M, \mathrm{sk}_A)$.

ABS: Signature and Verification

- SIGNATURE: to sign a message M for a **signing policy** (S, Γ), where $\Gamma \subset 2^S$, a signer holding attributes $A \subset S$ runs $\sigma \leftarrow \mathsf{ABS.Sign}(\mathsf{params}, S, \Gamma, M, \mathsf{sk}_A)$.
- VERIFICATION: the receiver of the signed message runs $1 \text{ or } 0 \leftarrow \mathsf{ABS.Vfy}(\mathsf{params}, S, \Gamma, M, \sigma)$.

ABS: Signature and Verification

- SIGNATURE: to sign a message M for a **signing policy** (S, Γ), where $\Gamma \subset 2^S$, a signer holding attributes $A \subset S$ runs $\sigma \leftarrow \mathsf{ABS.Sign}(\mathsf{params}, S, \Gamma, M, \mathsf{sk}_A)$.
- VERIFICATION: the receiver of the signed message runs $1 \text{ or } 0 \leftarrow \mathsf{ABS.Vfy}(\mathsf{params}, S, \Gamma, M, \sigma)$.

  [For correctness:
  $1 = \mathsf{ABS.Vfy}(\mathsf{params}, S, \Gamma, M, \mathsf{ABS.Sign}(\mathsf{params}, S, \Gamma, M, \mathsf{sk}_A)) \iff A \in \Gamma$.]

ABE Security: IND-CPA

A security game between a challenger and a possible attacker $\mathcal{A}$.

ABE Security: IND-CPA

A security game between a challenger and a possible attacker $\mathcal{A}$.

1. The challenger sends a universe of attributes $\mathcal{P}$ to $\mathcal{A}$.

ABE Security: IND-CPA

A security game between a challenger and a possible attacker $\mathcal{A}$.

1. The challenger sends a universe of attributes $\mathcal{P}$ to $\mathcal{A}$.
2. The challenger runs $(\mathsf{params}, \mathsf{msk}) \leftarrow \mathsf{ABE.Setup}(1^{\lambda}, \mathcal{P})$ and gives params to $\mathcal{A}$.

ABE Security: IND-CPA

A security game between a challenger and a possible attacker $\mathcal{A}$.

1. The challenger sends a universe of attributes $\mathcal{P}$ to $\mathcal{A}$.
2. The challenger runs $(\mathsf{params}, \mathsf{msk}) \leftarrow \mathsf{ABE.Setup}(1^\lambda, \mathcal{P})$ and gives params to $\mathcal{A}$.
3. **Secret key queries:** $\mathcal{A}$ adaptively chooses subsets $B \subset \mathcal{P}$ and must receive $\mathsf{sk}_B \leftarrow \mathsf{ABE.Ext}(\mathsf{params}, B, \mathsf{msk})$.

ABE Security: IND-CPA

A security game between a challenger and a possible attacker $\mathcal{A}$.

1. The challenger sends a universe of attributes $\mathcal{P}$ to $\mathcal{A}$.
2. The challenger runs $(\mathsf{params}, \mathsf{msk}) \leftarrow \mathsf{ABE.Setup}(1^\lambda, \mathcal{P})$ and gives params to $\mathcal{A}$.
3. **Secret key queries:** $\mathcal{A}$ adaptively chooses subsets $B \subset \mathcal{P}$ and must receive $\mathsf{sk}_B \leftarrow \mathsf{ABE.Ext}(\mathsf{params}, B, \mathsf{msk})$.
4. $\mathcal{A}$ outputs two messages M_0, M_1 of the same length, a set of attributes $S \subset \mathcal{P}$ and a decryption policy $\Gamma \subset 2^S$.

ABE Security: IND-CPA

A security game between a challenger and a possible attacker $\mathcal{A}$.

1. The challenger sends a universe of attributes $\mathcal{P}$ to $\mathcal{A}$.
2. The challenger runs $(\mathsf{params}, \mathsf{msk}) \leftarrow \mathsf{ABE.Setup}(1^\lambda, \mathcal{P})$ and gives params to $\mathcal{A}$.
3. **Secret key queries:** $\mathcal{A}$ adaptively chooses subsets $B \subset \mathcal{P}$ and must receive $\mathsf{sk}_B \leftarrow \mathsf{ABE.Ext}(\mathsf{params}, B, \mathsf{msk})$.
4. $\mathcal{A}$ outputs two messages M_0, M_1 of the same length, a set of attributes $S \subset \mathcal{P}$ and a decryption policy $\Gamma \subset 2^S$.
5. **Challenge:** the challenger chooses $b^\star \in_R \{0, 1\}$, computes $C^\star \leftarrow \mathsf{ABE.Enc}(\mathsf{params}, S, \Gamma, M_{b^\star})$ and gives $C^\star$ to $\mathcal{A}$.

ABE Security: IND-CPA

A security game between a challenger and a possible attacker $\mathcal{A}$.

1. The challenger sends a universe of attributes $\mathcal{P}$ to $\mathcal{A}$.
2. The challenger runs $(\text{params}, \text{msk}) \leftarrow \text{ABE.Setup}(1^\lambda, \mathcal{P})$ and gives params to $\mathcal{A}$.
3. **Secret key queries:** $\mathcal{A}$ adaptively chooses subsets $B \subset \mathcal{P}$ and must receive $\text{sk}_B \leftarrow \text{ABE.Ext}(\text{params}, B, \text{msk})$.
4. $\mathcal{A}$ outputs two messages M_0, M_1 of the same length, a set of attributes $S \subset \mathcal{P}$ and a decryption policy $\Gamma \subset 2^S$.
5. **Challenge:** the challenger chooses $b^\star \in_R \{0, 1\}$, computes $C^\star \leftarrow \text{ABE.Enc}(\text{params}, S, \Gamma, M_{b^\star})$ and gives $C^\star$ to $\mathcal{A}$.
6. Step 4 is repeated.

ABE Security: IND-CPA

A security game between a challenger and a possible attacker $\mathcal{A}$.

1. The challenger sends a universe of attributes $\mathcal{P}$ to $\mathcal{A}$.
2. The challenger runs $(\text{params}, \text{msk}) \leftarrow \text{ABE.Setup}(1^\lambda, \mathcal{P})$ and gives params to $\mathcal{A}$.
3. **Secret key queries:** $\mathcal{A}$ adaptively chooses subsets $B \subset \mathcal{P}$ and must receive $\text{sk}_B \leftarrow \text{ABE.Ext}(\text{params}, B, \text{msk})$.
4. $\mathcal{A}$ outputs two messages M_0, M_1 of the same length, a set of attributes $S \subset \mathcal{P}$ and a decryption policy $\Gamma \subset 2^S$.
5. **Challenge:** the challenger chooses $b^\star \in_R \{0, 1\}$, computes $C^\star \leftarrow \text{ABE.Enc}(\text{params}, S, \Gamma, M_{b^\star})$ and gives $C^\star$ to $\mathcal{A}$.
6. Step 4 is repeated.
7. $\mathcal{A}$ outputs a bit b, and wins if $b = b^\star$.

ABE Security: IND-CPA

A security game between a challenger and a possible attacker $\mathcal{A}$.

1. The challenger sends a universe of attributes $\mathcal{P}$ to $\mathcal{A}$.
2. The challenger runs $(\text{params}, \text{msk}) \leftarrow \text{ABE.Setup}(1^\lambda, \mathcal{P})$ and gives params to $\mathcal{A}$.
3. **Secret key queries:** $\mathcal{A}$ adaptively chooses subsets $B \subset \mathcal{P}$ and must receive $\text{sk}_B \leftarrow \text{ABE.Ext}(\text{params}, B, \text{msk})$.
4. $\mathcal{A}$ outputs two messages M_0, M_1 of the same length, a set of attributes $S \subset \mathcal{P}$ and a decryption policy $\Gamma \subset 2^S$.
5. **Challenge:** the challenger chooses $b^\star \in_R \{0, 1\}$, computes $C^\star \leftarrow \text{ABE.Enc}(\text{params}, S, \Gamma, M_{b^\star})$ and gives $C^\star$ to $\mathcal{A}$.
6. Step 4 is repeated.
7. $\mathcal{A}$ outputs a bit b, and wins if $b = b^\star$.

If $\Pr[\mathcal{A} \text{ wins}] \approx 1/2$, then the ABE scheme is IND-CPA secure.

ABE Security: sIND-CPA

A security game between a challenger and a possible attacker $\mathcal{A}$.

1. The challenger sends a universe of attributes $\mathcal{P}$ to $\mathcal{A}$.
2. $\mathcal{A}$ selects $S \subset \mathcal{P}$ and a decryption policy $\Gamma \subset 2^S$.
3. The challenger runs $(\text{params}, \text{msk}) \leftarrow \text{ABE.Setup}(1^\lambda, \mathcal{P})$ and gives params to $\mathcal{A}$.
4. **Secret key queries:** $\mathcal{A}$ adaptively chooses subsets $B \subset \mathcal{P}$ s.t. $B \cap S \notin \Gamma$, and must receive $\text{sk}_B \leftarrow \text{ABE.Ext}(\text{params}, B, \text{msk})$.
5. $\mathcal{A}$ outputs two messages M_0, M_1 of the same length.
6. **Challenge:** the challenger chooses $b^\star \in_R \{0, 1\}$, computes $C^\star \leftarrow \text{ABE.Enc}(\text{params}, S, \Gamma, M_{b^\star})$ and gives $C^\star$ to $\mathcal{A}$.
7. Step 4 is repeated.
8. $\mathcal{A}$ outputs a bit b, and wins if $b = b^\star$.

If $\Pr[\mathcal{A} \text{ wins}] \approx 1/2$, then the ABE scheme is sIND-CPA secure.

ABS Security: (s)EUF-CMA

A security game between a challenger and a possible attacker $\mathcal{F}$.

1. The challenger sends a universe of attributes $\mathcal{P}$ to $\mathcal{F}$.
2. (In the selective case), $\mathcal{F}$ selects $S \subset \mathcal{P}$ and a decryption policy $\Gamma \subset 2^S$.
3. The challenger runs $(\text{params}, \text{msk}) \leftarrow \text{ABS.Setup}(1^\lambda, \mathcal{P})$ and gives params to $\mathcal{F}$.
4. **Secret key queries:** $\mathcal{F}$ adaptively chooses subsets $B \subset \mathcal{P}$ s.t. $B \cap S \notin \Gamma$ (selective), and must receive $\text{sk}_B \leftarrow \text{ABS.Ext}(\text{params}, B, \text{msk})$.
5. **Signature queries:** $\mathcal{F}$ adaptively chooses tuples (S', Γ', M') and must receive $\sigma' \leftarrow \text{ABS.Sign}(\text{params}, S', \Gamma', M', \text{sk}_A)$, where $sk_A \leftarrow \text{ABS.Ext}(\text{params}, A, \text{msk})$ and $A \in \Gamma'$.
6. $\mathcal{F}$ outputs a tuple (S, Γ, M, σ).
7. $\mathcal{F}$ wins if (S, Γ, M, σ) has not been obtained in Step 5 and $1 = \text{ABS.Vfy}(\text{params}, S, \Gamma, M, \sigma)$.

If $\Pr[\mathcal{F} \text{ wins}] \approx 0$, then the ABS scheme is sEUF-CMA secure.

ABS Security: Privacy

It can be formalized through an indistinguishability game...

Intuitively: a signature $\sigma \leftarrow \mathsf{ABS.Sign}(\mathsf{params}, S, \Gamma, M, \mathsf{sk}_A)$ must reveal no information about the set of attributes A.

ABS Security: Privacy

It can be formalized through an indistinguishability game...

Intuitively: a signature $\sigma \leftarrow \mathsf{ABS.Sign}(\mathsf{params}, S, \Gamma, M, \mathsf{sk}_A)$ must reveal no information about the set of attributes A.

This property can be achieved **computationally** (relation to a hard problem) or **perfectly**.

Outline

Properties of AB Systems

- **Expressiveness:** (n, n)-threshold << (t, n)-threshold << LSSS monotone policies << LSSS (non-)monotone policies

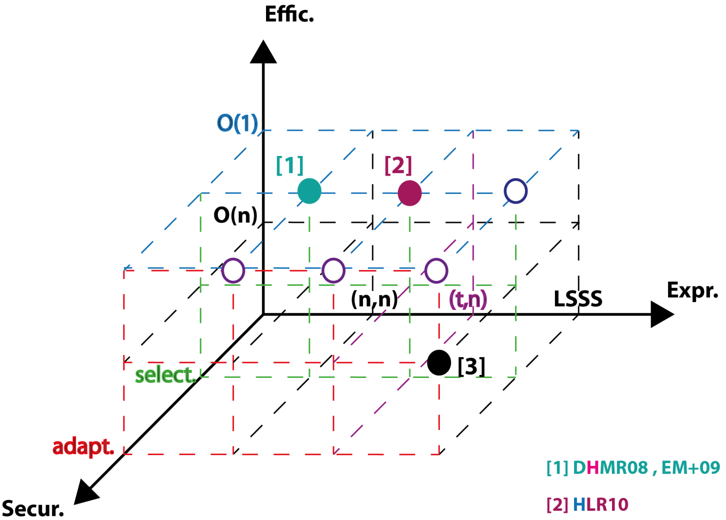
Properties of AB Systems

- **Expressiveness:** (n, n)-threshold $<<$ (t, n)-threshold $<<$ LSSS monotone policies $<<$ LSSS (non-)monotone policies
- **Efficiency:** $\left(\ |C| = |\sigma| = \mathcal{O}(n)\ \right) << \left(\ |C| = |\sigma| = \mathcal{O}(1)\ \right)$

Properties of AB Systems

- **Expressiveness:** (n,n)-threshold $<<$ (t,n)-threshold $<<$ LSSS monotone policies $<<$ LSSS (non-)monotone policies
- **Efficiency:** $\left(\ |C| = |\sigma| = \mathcal{O}(n)\ \right) << \left(\ |C| = |\sigma| = \mathcal{O}(1)\ \right)$
- **Security:** $\begin{array}{rcl} \text{selective} & << & \text{adaptive} \\ \text{ROM} & << & \text{standard model} \end{array}$

CP-ABE Panorama

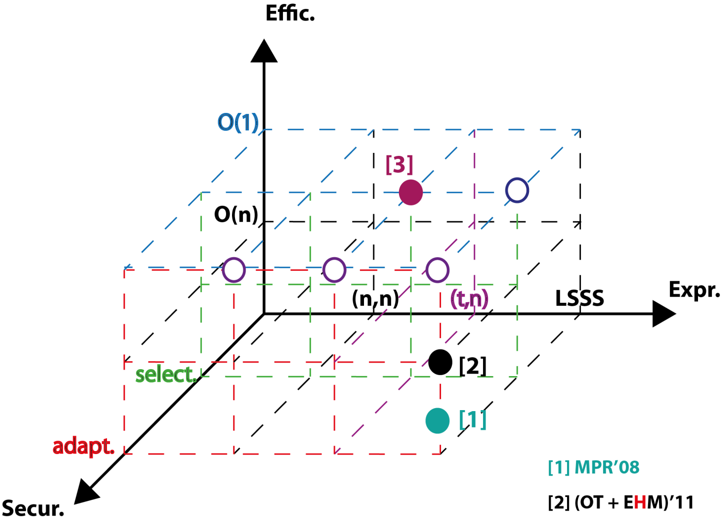


[1] DHMR08 , EM+09

[2] HLR10

[3] (OT + LSW)10

ABS Panorama



[1] MPR'08

[2] (OT + EHM)'11

[3] HLLR'12

Furthermore...

- **All** existing ABE schemes employ bilinear pairings.
- **Agrawal et al.** (ePrint, 2011) have proposed a fuzzy-IBE scheme from **lattices**.

Furthermore...

- **All** existing ABE schemes employ bilinear pairings.
- **Agrawal et al.** (ePrint, 2011) have proposed a fuzzy-IBE scheme from **lattices**.
- More or less the same for ABS schemes (except **generic** construction of **Maji et al., CT-RSA'11**).

Furthermore...

- **All** existing ABE schemes employ bilinear pairings.
- **Agrawal et al.** (ePrint, 2011) have proposed a fuzzy-IBE scheme from **lattices**.
- More or less the same for ABS schemes (except **generic** construction of **Maji et al., CT-RSA'11**).
- What about generic constructions of ABE schemes ?

Outline

Identity Based Dynamic Distributed Encryption (IB-DDE)

Identity Based Dynamic Distributed Encryption (IB-DDE)

- KEY EXTRACTION: a user with **identity** id_i obtains from a master entity the secret key

$$\mathsf{sk}_{\mathsf{id}_i} \leftarrow \mathsf{IBDDE.Ext}(\mathsf{msk}, \mathsf{id}_i)$$

Identity Based Dynamic Distributed Encryption (IB-DDE)

- KEY EXTRACTION: a user with **identity** id_i obtains from a master entity the secret key

$$\mathsf{sk}_{\mathsf{id}_i} \leftarrow \mathsf{IBDDE.Ext}(\mathsf{msk}, \mathsf{id}_i)$$

- ENCRYPTION: the sender chooses a set of people, $S = \{\mathsf{id}_1, \ldots, \mathsf{id}_s\}$ and a **decryption policy** $\Gamma \subset 2^S$, monotone increasing:

$$C \leftarrow \mathsf{IBDDE.Enc}(M, S, \Gamma)$$

Identity Based Dynamic Distributed Encryption (IB-DDE)

- KEY EXTRACTION: a user with **identity** id_i obtains from a master entity the secret key

$$\mathrm{sk}_{\mathrm{id}_i} \leftarrow \mathrm{IBDDE.Ext}(\mathrm{msk}, \mathrm{id}_i)$$

- ENCRYPTION: the sender chooses a set of people, $S = \{\mathrm{id}_1, \ldots, \mathrm{id}_s\}$ and a **decryption policy** $\Gamma \subset 2^S$, monotone increasing:

$$C \leftarrow \mathrm{IBDDE.Enc}(M, S, \Gamma)$$

- DECRYPTION: if a subset of people $A \in \Gamma$ cooperate, they can **jointly** decrypt by using their secret keys:

$$\tilde{M} \leftarrow \mathrm{IBDDE.Dec}(C, \{\mathrm{sk}_i\}_{\mathrm{id}_i \in A})$$

Identity Based Dynamic Distributed Encryption (IB-DDE)

- KEY EXTRACTION: a user with **identity** id_i obtains from a master entity the secret key

$$\mathsf{sk}_{\mathsf{id}_i} \leftarrow \mathsf{IBDDE.Ext}(\mathsf{msk}, \mathsf{id}_i)$$

- ENCRYPTION: the sender chooses a set of people, $S = \{\mathsf{id}_1, \ldots, \mathsf{id}_s\}$ and a **decryption policy** $\Gamma \subset 2^S$, monotone increasing:

$$C \leftarrow \mathsf{IBDDE.Enc}(M, S, \Gamma)$$

- DECRYPTION: if a subset of people $A \in \Gamma$ cooperate, they can **jointly** decrypt by using their secret keys:

$$\tilde{M} \leftarrow \mathsf{IBDDE.Dec}(C, \{\mathsf{sk}_i\}_{\mathsf{id}_i \in A})$$

[Again, $\tilde{M} = M \iff A \in \Gamma$.]

From IB-DDE to ABE...

Take an IB-DDE scheme, and **replace identities with attributes**.

From IB-DDE to ABE...

Take an IB-DDE scheme, and **replace identities with attributes**.

- ABE.Setup: same as IBDDE.Setup.

From IB-DDE to ABE...

Take an IB-DDE scheme, and **replace identities with attributes**.

- ABE.Setup: same as IBDDE.Setup.
- $\mathsf{ABE.Ext}(A, \mathsf{msk})$: run $\mathsf{sk}_{\mathsf{at}_i} \leftarrow \mathsf{IBDDE.Ext}(\mathsf{msk}, \mathsf{at}_i)$ for each $\mathsf{at}_i \in A$, and define

$$\mathsf{sk}_A = \{\mathsf{sk}_{\mathsf{at}_i}\}_{\mathsf{at}_i \in A}$$

From IB-DDE to ABE...

Take an IB-DDE scheme, and **replace identities with attributes**.

- ABE.Setup: same as IBDDE.Setup.
- ABE.Ext(A, msk): run $\mathsf{sk}_{\mathsf{at}_i} \leftarrow \mathsf{IBDDE.Ext}(\mathsf{msk}, \mathsf{at}_i)$ for each $\mathsf{at}_i \in A$, and define
$$\mathsf{sk}_A = \{\mathsf{sk}_{\mathsf{at}_i}\}_{\mathsf{at}_i \in A}$$
- ABE.Enc(M, S, Γ): works exactly as IBDDE.Enc(M, S, Γ).

From IB-DDE to ABE...

Take an IB-DDE scheme, and **replace identities with attributes**.

- ABE.Setup: same as IBDDE.Setup.
- ABE.Ext(A, msk): run $\mathsf{sk}_{\mathsf{at}_i} \leftarrow \mathsf{IBDDE.Ext}(\mathsf{msk}, \mathsf{at}_i)$ for each $\mathsf{at}_i \in A$, and define
$$\mathsf{sk}_A \;=\; \{\mathsf{sk}_{\mathsf{at}_i}\}_{\mathsf{at}_i \in A}$$
- ABE.Enc(M, S, Γ): works exactly as $\mathsf{IBDDE.Enc}(M, S, \Gamma)$.
- ABE.Dec$(C, \Gamma, \mathsf{sk}_A)$: works exactly as $\mathsf{IBDDE.Dec}(C, \{\mathsf{sk}_i\}_{\mathsf{at}_i \in A})$.

... Without Coalition-Resistance !!

- Suppose a message is encrypted for $S = \{\mathsf{at}_1, \ldots, \mathsf{at}_4\}$ with a **threshold** decryption policy, $t = 3$.

... Without Coalition-Resistance !!

- Suppose a message is encrypted for $S = \{\mathrm{at}_1, \ldots, \mathrm{at}_4\}$ with a **threshold** decryption policy, $t = 3$.
- With the construction based on IB-DDE, a **coalition** of a user holding $\{\mathrm{at}_1, \mathrm{at}_2\}$ and a user holding $\{\mathrm{at}_3\}$ will be **able to decrypt**.

  [This contradicts the security requirements for ABE.]

Basic Mathematics...?

ABE − 'coalition-resistance' = IB-DDE

Basic Mathematics...?

ABE − 'coalition-resistance' = IB-DDE

IBDDE = IBE + 'secret sharing'

Basic Mathematics...?

ABE $-$ 'coalition-resistance' $=$ IB-DDE

IBDDE $=$ IBE $+$ 'secret sharing'

ABE $-$ 'coalition-resistance' $=$ IBE $+$ 'secret sharing'

Basic Mathematics...?

ABE − 'coalition-resistance' = IB-DDE

IBDDE = IBE + 'secret sharing'

ABE − 'coalition-resistance' = IBE + 'secret sharing'

ABE = IBE + 'secret sharing' + 'coalition-resistance'

From IB-DDE to ABE

The approach ABE = IB-DDE + ‘coalition-resistance’ has been followed for specific schemes (with pairings).

From IB-DDE to ABE

The approach ABE = IB-DDE + 'coalition-resistance' has been followed for specific schemes (with pairings).

To achieve **coalition-resistance**, one can *try* to modify ABE.Ext: linking the values $\{\mathsf{sk}_{\mathsf{at}_i}\}_{\mathsf{at}_i \in A}$ with some additional value, **different** for each user.

From IB-DDE to ABE: Precedents

1. IB-DDE scheme in [DHMR,ProvSec'07] $\longrightarrow$ ABE scheme in [DHMR,AAECC'10] (available at ePrint 2008/502 since 2008).

   Schemes work for LSSS monotone policies, have selective security, and $|C| = 2(n - t) + \mathcal{O}(1)$.

2. IB-DDE scheme in [DelPoi,Crypto'08] $\longrightarrow$ ABE scheme in [HLR,PKC'10].

   Schemes work for threshold policies, have selective security, and $|C| = \mathcal{O}(1)$.

From IB-DDE to ABE: Precedents

1. IB-DDE scheme in [DHMR,ProvSec'07] $\longrightarrow$ ABE scheme in [DHMR,AAECC'10] (available at ePrint 2008/502 since 2008).

   Schemes work for LSSS monotone policies, have selective security, and $|C| = 2(n-t) + \mathcal{O}(1)$.

2. IB-DDE scheme in [DelPoi,Crypto'08] $\longrightarrow$ ABE scheme in [HLR,PKC'10].

   Schemes work for threshold policies, have selective security, and $|C| = \mathcal{O}(1)$.

But ... is there a **generic** way to achieve **'coalition-resistance'** ?

Outline

1. Attribute-Based Cryptography
2. ABC: State of the Art
3. Relation between ABE and IB-DDE
4. (Inefficient) Generic Constructions of ABE Schemes
5. Conclusions

The Simple Idea

ABE = IBE + ‘secret sharing’ + ‘coalition-resistance’

The Simple Idea

ABE = IBE + ~~'secret sharing' + 'coalition-resistance'~~

The Simple Idea

ABE $=$ IBE $+$ 'brute force approach'

The Simple Idea

ABE = IBE + 'brute force approach'

Consider **all** the subsets of A for sk_A, and **all** the subsets in Γ_0 for C.

The Scheme: Setup and Key Extraction

Let $\mathsf{IBE} = (\mathsf{IBE.Setup}, \mathsf{IBE.Ext}, \mathsf{IBE.Enc}, \mathsf{IBE.Dec})$ be an IBE scheme.

The Scheme: Setup and Key Extraction

Let IBE $=$ (IBE.Setup, IBE.Ext, IBE.Enc, IBE.Dec) be an IBE scheme.

ABE.Setup$(1^\lambda, \mathcal{P})$:

1. Run $(\mathsf{params}_{\mathsf{IBE}}, \mathsf{msk}_{\mathsf{IBE}}) \leftarrow \mathsf{IBE.Setup}(1^\lambda)$.
2. Let ID be the identity space of IBE, included in $\mathsf{params}_{\mathsf{IBE}}$. Choose a hash function $H : \{0,1\}^* \rightarrow \mathsf{ID}$.
3. Define $\mathsf{params} = (\mathsf{params}_{\mathsf{IBE}}, H)$ and $\mathsf{msk} = \mathsf{msk}_{\mathsf{IBE}}$.

The Scheme: Setup and Key Extraction

Let $\mathsf{IBE} = (\mathsf{IBE.Setup}, \mathsf{IBE.Ext}, \mathsf{IBE.Enc}, \mathsf{IBE.Dec})$ be an IBE scheme.

$\mathsf{ABE.Setup}(1^\lambda, \mathcal{P})$:

1. Run $(\mathsf{params}_{\mathsf{IBE}}, \mathsf{msk}_{\mathsf{IBE}}) \leftarrow \mathsf{IBE.Setup}(1^\lambda)$.
2. Let ID be the identity space of IBE, included in $\mathsf{params}_{\mathsf{IBE}}$. Choose a hash function $H : \{0,1\}^* \rightarrow \mathsf{ID}$.
3. Define $\mathsf{params} = (\mathsf{params}_{\mathsf{IBE}}, H)$ and $\mathsf{msk} = \mathsf{msk}_{\mathsf{IBE}}$.

$\mathsf{ABE.Ext}(\mathsf{params}, A, \mathsf{msk})$:

1. For every subset $A' \subseteq A$, $A' \neq \emptyset$, run $\mathsf{sk}_{A'} \leftarrow \mathsf{IBE.Ext}(\mathsf{params}_{\mathsf{IBE}}, H(A'), \mathsf{msk})$.
2. Define $\mathsf{sk}_A = \{\mathsf{sk}_{A'}\}_{A' \subseteq A}$.

The Scheme: Encryption and Decryption

$\mathsf{ABE.Enc}(\mathsf{params}, S, \Gamma, M)$:

1. Find the basis Γ_0 of minimal subsets of Γ.
2. For each $B \in \Gamma_0$, compute $c_B \leftarrow \mathsf{IBE.Enc}(\mathsf{params}_{\mathsf{IBE}}, H(B), M)$.
3. Define $C = \{c_B\}_{B \in \Gamma_0}$.

The Scheme: Encryption and Decryption

$\mathsf{ABE.Enc}(\mathsf{params}, S, \Gamma, M)$:

1. Find the basis Γ_0 of minimal subsets of Γ.
2. For each $B \in \Gamma_0$, compute $c_B \leftarrow \mathsf{IBE.Enc}(\mathsf{params}_{\mathsf{IBE}}, H(B), M)$.
3. Define $C = \{c_B\}_{B \in \Gamma_0}$.

$\mathsf{ABE.Dec}(\mathsf{params}, C, \Gamma, \mathsf{sk}_A)$:

1. Find a subset $A' \subseteq A$ such that $A' \in \Gamma_0$.
2. Extract $c_{A'}$ from C, and extract $\mathsf{sk}_{A'}$ from sk_A.
3. Output $M \leftarrow \mathsf{IBE.Dec}(\mathsf{params}_{\mathsf{IBE}}, c_{A'}, H(A'), \mathsf{sk}_{A'})$.

The Scheme: (Bad) Efficiency, and "Improvements"

- Let $n = |\mathcal{P}|$ be the number of attributes.
- Then $|\mathsf{sk}_A| = 2^{|A|} - 1 \leq 2^n$.
- And $|C| = 2^{|\Gamma_0|} \leq 2^n$.

The Scheme: (Bad) Efficiency, and "Improvements"

- Let $n = |\mathcal{P}|$ be the number of attributes.
- Then $|\mathrm{sk}_A| = 2^{|A|} - 1 \leq 2^n$.
- And $|C| = 2^{|\Gamma_0|} \leq 2^n$.

- Using **HIBE** or **IBBE** instead of IBE leads to similar constructions, with shorter sk_A or C.

The Scheme: (Bad) Efficiency, and "Improvements"

- Let $n = |\mathcal{P}|$ be the number of attributes.
- Then $|\mathrm{sk}_A| = 2^{|A|} - 1 \leq 2^n$.
- And $|C| = 2^{|\Gamma_0|} \leq 2^n$.

- Using **HIBE** or **IBBE** instead of IBE leads to similar constructions, with shorter sk_A or C.

So, if $n \leq \log[\mathrm{poly}(\lambda)]$, the protocols of ABE are all poly-time...

The Scheme: (Bad) Efficiency, and "Improvements"

- Let $n = |\mathcal{P}|$ be the number of attributes.
- Then $|\mathsf{sk}_A| = 2^{|A|} - 1 \leq 2^n$.
- And $|C| = 2^{|\Gamma_0|} \leq 2^n$.
- Using **HIBE** or **IBBE** instead of IBE leads to similar constructions, with shorter sk_A or C.

So, if $n \leq \log[\mathrm{poly}(\lambda)]$, the protocols of ABE are all poly-time...

What about AB Signatures ? Same ideas, using **IB ring signatures** instead of IBE.

Can this Simple Construction Be Useful ?

- ABE from any IBE: pairings, lattices, quadratic residuosity (ROM)...
- If the IBE scheme is **adaptively** secure, so the ABE scheme is.

Can this Simple Construction Be Useful ?

- ABE from any IBE: pairings, lattices, quadratic residuosity (ROM)...
- If the IBE scheme is **adaptively** secure, so the ABE scheme is.

- Is AB crypto being used somewhere, in real life ?
  [In theory: access control, cloud computing...]

Can this Simple Construction Be Useful ?

- ABE from any IBE: pairings, lattices, quadratic residuosity (ROM)...
- If the IBE scheme is **adaptively** secure, so the ABE scheme is.

- Is AB crypto being used somewhere, in real life ?
  [In theory: access control, cloud computing...]

- If the answer is **YES**, what are the typical values for $n, |A|, |\Gamma_0|$?

Outline

1. Attribute-Based Cryptography
2. ABC: State of the Art
3. Relation between ABE and IB-DDE
4. (Inefficient) Generic Constructions of ABE Schemes
5. Conclusions

AB Crypto: Theory and Practice

Theory

Practice

AB Crypto: Theory and Practice

Theory

- Designing new AB cryptosystems is challenging (strong security requirements).
- Many open problems $\rightarrow$ possible theoretical crypto **papers** !
- In particular, is there any **efficient and generic** way to achieve **'coalition-resistance'**, when IB-DDE $\rightarrow$ ABE ?

Practice

AB Crypto: Theory and Practice

Theory

- Designing new AB cryptosystems is challenging (strong security requirements).
- Many open problems $\rightarrow$ possible theoretical crypto **papers** !
- In particular, is there any **efficient and generic** way to achieve **'coalition-resistance'**, when IB-DDE $\rightarrow$ ABE ?

Practice

- Theoretical research should be complemented with practical issues.
- Real **needs of the market** in terms of AB crypto ?
- Maybe for a small company which implements access control for its workers, IBE $\rightarrow$ ABE suffices...

Attribute-Based Cryptography: Survey and (Inefficient?) Generic Constructions

Javier Herranz

Universitat Politècnica de Catalunya
Barcelona, Spain

DoE CRYPTODOC, Darmstadt (Germany), November 21st, 2011