Enabling Attribute Based Encryption as an Internet Service

Runhua Xu and James Joshi
University of Pittsburgh
runhua.xu@pitt.edu
Cloud Storage Service

• It has been gaining significant success
  • potential “infinite” storage size
  • convenience of synchronization
  • ease of access (at anytime, from anywhere)

• Users/Organizations
  • increasingly utilize/rely on the cloud storage services
Security & Privacy Concerns

Recent advances have enabled applications that generate/collect huge amounts of personal data.

Cloud Storage Providers

Honest-but-Curious
-- run the programs and algorithms correctly
  but gather information related to the stored data.

Insider threat
-- secretly analyzing or leaking customers’ sensitive data

*How users are able to fully trust the CSP regards to their sensitive data*
Initial Solution

- Confidentiality for data
- Fine-grained access control for data

Initial Requirement

- Combination of encryption and access control
- Friendly for access scenario in cloud storage

Ciphertext Policy Attribute Based Encryption

Data : self-protection

Multiple-device Scenarios

• Increasing popularity and adoption of mobile devices
  • pads
  • cell phones
  • IoT sensors
• Traditional application
  → Multiple-device application

When ABE schemes meet Multiple-device application, what’s the situation?

Desktop, laptop, workstation... → fine
Cell phone, pad, IoT sensor.... → not good as expected
Challenges of ABE Adoption

- Global authority center
  - hard to deploy a global authority center trusted by all Internet users

- Multi-device scenarios are pervasive
  - Put ABE adoption into Multi-device scenarios \(\rightarrow\) limitations
    - Computational resources for ABE
    - Battery power for ABE
States

• The lack of an effective deployment approach
  • *to make ABE available broadly as a service*
  • *to support a broad set of mobile cloud applications*

• An attribute based encryption as a service
  • *mechanism to deploy ABE widely over various cloud platforms*
Scenario

Authority Center B

ABEaaS Cloud
Attribute based Encryption as a Service

Access Policy
Encryption Service

Authority Center A

ABEaaS Cloud
Attribute based Encryption as a Service

Decryption Service
Public Key Delivery

Private/Key Delivery
Overview

Two settings:
- hybrid setting
- general setting
From ABE to ABEaaS

• Overview of ABE
  • *Four Algorithms*
    • Setup
    • Key Generation
    • Encryption
    • Decryption
  • *CP-ABE/KP-ABE*
  • Access Structure
    • And-gate, Tree, LSSS
  • *Technique to Outsource Computation*
    • Outsource partial computation to a powerful server without impact on the functionality and security of the ABE scheme

\[
A_1 = (1\land 2\land 3\land 4)
\]

\[
A_3 = (1\land 2\land 3)\lor(1\land 4)
\]
Preliminaries: What’s CP-ABE

CP-ABE in detail

$PK_{CS}, PK_{EE}, ...$

$PK_{PhD}, PK_{ALU}, ...$

$PK_{M}, PK_{F}, ...$

$PK_{1980}, PK_{1981}, ...$

... $C = Enc(PK, P, M)$

$P = CS \text{ AND } (PhD \text{ OR } ALU)$

Storage Server (Untrusted)

$S_A$ satisfies $P$

$S_A = \{CS, PhD\}$

$S_B$ does not satisfy $P$

$S_B = \{EE, PhD\}$

Dept.: CS, EE, ...

Type: PhD Stud., Alumni, ...

Gender: Male, Female

Birth Year: 1980, 1981, ...

...
Architecture of ABE service platform
Manager Node

Request Dispatcher (RD)

*receive the request*

*dispatch the request to an available work node*

Work Node Management (WNM)

*manage a number of work nodes*
Encryption Service Work Node

Authority Management (AM)

Secret/Attribute Intermediate Ciphertext Generator

Pools Management (PM)
Secret Intermediate Ciphertext Pool (SICP)
Attribute Intermediate Ciphertext Pool (AICP)
Decryption Service Work Node

The frequency of using decryption service is several times than the frequency of using encryption service.

Job Sequence

Computation center will calculate the job in parallel
ABEaaS Implementation

• Prototype model of ABE used in ABEaaS
  • *Extend from [2] & [6]*
  • *ABE Instance*

  - $\text{Setup}_{\text{authority}}(\lambda, U) \rightarrow (PK, MSK)$
  - $\text{KeyGen}_{\text{authority}}(MSK, S) \rightarrow (TK, SK)$
  - $\text{Encrypt}_{\text{service}}(PK) \rightarrow (IT)$
  - $\text{Encrypt}_{\text{user}}(PK, IT, AC, data) \rightarrow (CT)$
  - $\text{Decrypt}_{\text{service}}(TK, CT) \rightarrow (CT)$
  - $\text{Decrypt}_{\text{user}}(CT, SK) \rightarrow (data)$


Initialization

- Check the authority list
  - **preload the authority information**
- Initialization of pool
  - **precompute the intermediate components**
  - **store the intermediate components into the pool**

---

Algorithm 1 Service Initialization with General Setting.

**Input:**
- \( \text{type}_{\text{op}} \), the service type (encrypt/decrypt),
- \( \text{type}_{\text{ABE}} \), the ABE type (KP-ABE/CP-ABE),
- \( \text{list} \), the default authority setting list,
- \( \text{size}_{\text{pool}} \), the default size of pools.

**Output:**
- \( m_{\text{authority}} \), a map for the authorities information,
- \( m_{\text{AICP}} \), a map for the AICP.

```plaintext
1: initialize the map, \( m_{\text{authority}} \)
2: for \( \text{id} \) in \( \text{list} \) do
3: \( \text{pk}_{\text{id}} \leftarrow \text{request the public key from authority} \)
4: push \((\text{id}, \text{pk}_{\text{id}})\) \( \rightarrow m_{\text{authority}} \)
5: end for
6: if \( \text{type}_{\text{op}} \) == “Encrypt” then
7: initialize the maps \( m_{\text{AICP}} \).
8: for \( \text{id} \) in \( \text{list} \) do
9: \( \text{pk}_{\text{id}} \leftarrow m_{\text{authority}}[	ext{id}] \)
10: \( s \leftarrow \text{random}(\mathbb{Z}_{p_{\text{ps}}}^*) \)
11: for \( i = 0 \) to \( \text{size} \) do
12: if \( \text{type}_{\text{ABE}} \) == “CP-ABE” then
13: \( \lambda, x, t \leftarrow \text{random}(\mathbb{Z}_{p_{\text{ps}}}) \)
14: \( C_1 = g_{\text{id}}^\lambda, C_2 = (u_{\text{id}}^{\lambda t})^x, C_3 = g^t \)
15: add tuple \((\lambda, x, t, C_1, C_2, C_3)\) \( \rightarrow \text{list}_{\text{AICP}} \)
16: else
17: \( r, s \leftarrow \text{random}(\mathbb{Z}_{p_{\text{ps}}}) \)
18: \( C_1 = u_{\text{id}}^r, C_2 = (u_{\text{id}}^{\lambda t})^s \)
19: add tuple \((r, s, C_1, C_2)\) \( \rightarrow \text{list}_{\text{AICP}} \)
20: end if
21: end for
22: push \((\text{id}, \text{list}_{\text{AICP}})\) \( \rightarrow m_{\text{AICP}} \)
23: end for
24: return \( m_{\text{authority}}, m_{\text{AICP}} \)
25: else
26: return \( m_{\text{authority}} \)
27: end if
```

*Note:* the function \( \text{random}(\mathbb{A}) \) generates random elements between 0 and \(|\mathbb{A}|\).
Encryption Service

- Find required authority information from DB
- “calculate” the intermediate components (IC)
  - check the current pool
  - if no enough IC
    - return signal to change to another node
  - if having enough IC
    - randomly select from the pool
    - remove the selected IC from the pool

```plaintext
Algorithm 2 Encryption Service.
Input: id, the authority id of the user,
       size_attribute, the number of attributes size,
       mAICP, a map represented the AICP,
       m_authority, the authorities information.
Output: it_attribute, the tuple of attribute intermediate ciphertext.
1: if id in m_authority then
2:   pull pubid ← m_authority
3: else
4:   execute the initialization with the id
5:   end if
6: list_AICP:id ← mAICP[id]
7: if |list_AICP:id| > size_attribute then
8:   for i = 0 to sizeattribute do
9:     index_random ← random(|list_AICP:id|)
10:    ituple ← pop list_AICP:id[index_random]
11:   add ituple → it_attribute
12: end for
13: return it_attribute
14: else
15: return signal empty
16: end if
Note: that size_att < size_pool, which indicates the size of requested attributes set is much smaller than the size of pool. |A| denotes the size of list A.
```
Decryption Service

• Find required authority information from DB
  • if no, query from the authority and store it

• Computation job
  • add delegation computing job to job sequence
  • (multiple processing in parallel)
  • return the intermediate computing result

---

Algorithm 3 Decryption Service.

Input: \( id \), the authority id of the user,
\( S \), the job sequences,
\( CT \), the ciphertext,
\( TK \), the temporary key of CP-ABE.

Output: \( CT \), the intermediate ciphertext.

1: if \( id \) in \( m_{authority} \) then
2:   pull \( pk_{id} \leftarrow m_{authority} \)
3: else
4:   execute the initialization with the \( id \)
5: end if
6: push tuple \( (job_{id}, < pk_{id}, CT, TK >) \) \( \rightarrow S \)
7: for true do
8:   if status of \( job_{id} == signal_{done} \) then
9:     \( CT \leftarrow S[job_{id}] \)
10:    return \( CT \)
11: end if
12: if time out then
13:   return signal_{time.out}
14: end if
15: end for
Security Discussion

• Security of Encryption Service Node
  • sensitive modules
    • general setting: AICG, AICP
    • hybrid setting: additional SICG, SICP
  • The AIC does not include any secret
  • The SIC includes secret information
    • only used in the hybrid setting
  • The AIC/SIC is disposable
    • when the intermediate component is used, it will be destroyed immediately
  • The intermediate component is randomly selected

• Security of Decryption Service Node
  • we does not change the structure of delegation computation algorithm
Performance Analysis

- Scalability and Availability
  - *dual-master multi-slave architecture*
    - a backup manager node with real-time synchronization
    - multiple work nodes
    - computing of each work node
- Efficiency of using ABEaaS
  - *Efficiency estimates*
    - theoretical analysis
  - *Experiment Result*
### TABLE I

**USER’S COMPUTATION ESTIMATES**

<table>
<thead>
<tr>
<th>Schemes</th>
<th>$\text{ABE [5]}$</th>
<th>$\text{ABEaaS}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encryption</td>
<td>$B + (5</td>
<td>P</td>
</tr>
<tr>
<td>Decryption</td>
<td>$(</td>
<td>P'</td>
</tr>
</tbody>
</table>

$^1$ Let $B$, $E$ and $M_P$ be the bilinear map, exponentiation, and multiplication operations, respectively.

$^2$ Let $|P|$ and $|P'|$ be the complexity of the access policy and the size of the minimal set of attributes, respectively.
Users’ operation time

- Users’ operation time
  - Original ABE scheme v.s. ABEaaS scheme in General Setting
- More attributes, more time saving
Thanks

Q & A