SecLoc: Securing Location-Sensitive Storage in the Cloud

Jingwei Li$^1$, Anna Squicciarini$^2$, Dan Lin$^3$, Shuang Liang$^1$, Chunfu Jia$^1$

$^1$Nankai University

$^2$Pennsylvania State University

$^3$Missouri University of Science and Technology

May 28, 2015
Outline

1. Introduction
2. SecLoc (Secure Location-Sensitive Storage)
3. tABE (Transformable ABE)
4. Efficiency Analysis
5. Conclusion
Outline

1. Introduction

2. SecLoc (Secure Location-Sensitive Storage)

3. tABE (Transformable ABE)

4. Efficiency Analysis

5. Conclusion
Introduction

Background

- In the cloud, the providers take care of data management, storage and recovery on behalf of data owners.

Problem

- The outsourcing paradigm makes it hard to prevent the cloud users’ data from being copied to risky locations.
- Location sensitivity is important in data storage, e.g., Canadian law has clearly specified that data related to personal identifiable information must be stored within Canada.
## Goals & Contribution

### Goals
- Addressing the cloud users’ security and privacy concerns incurred by the storage location constraints.
- Allowing cloud users to take advantage of computing services offered by the cloud.

### Contributions
- Propose the *SecLoc* framework for data confidentiality, location sensitivity and computing efficiency.
- Propose the *tABE* primitive, which integrates with SecLoc to ensure the user data become unavailable if they are copied to unintelligent locations.

Anna Squicciarini (PSU)  SACMAT 2015  May 28, 2015  5 / 22
State-of-the-Art

Many works denote into confidentiality and integrity

- Confidentiality, e.g., SE, ABE.
- Integrity, e.g., PoR, PDP.

Few works consider the storage location

- PoL [Watson 12] does not consider any protections on the user data being copied to other ineligible locations.
- DLAS [Noman 12] just focuses on preserving the location information, not data.
- Some other works need fully trusted modules.
Outline

1. Introduction
2. SecLoc (Secure Location-Sensitive Storage)
3. tABE (Transformable ABE)
4. Efficiency Analysis
5. Conclusion
Architecture

- Cloud service providers and Cloud users.
- Region servers. independent of service providers and responsible for storage nodes authentication.

Assumptions

- Region server is with minimal trusted. reliably authenticate the storage nodes so long as they are not compromised.
- Cloud providers and their storage nodes are not trusted.
- The storage nodes and region server in the same region are not be compromised simultaneously.
Threat Model & Design Goals

Threat Model

- Cloud provider provides a tampered list of candidate nodes to the users, misleading to store data in a wrong place.
- Storage nodes colludes with each other, or even with some compromised region servers, to extract (encrypted) data.

Design Goals

- Location-sensitive access control.
- Content-based access control.
- Keyless.
SecLoc Overview

(a) Registration
(b) Storage
(c) Retrieval

Note
Any tamper of the candidate node list can be caught through verifying the location of the storage nodes with the aid of region servers.
Outline

1. Introduction
2. SecLoc (Secure Location-Sensitive Storage)
3. tABE (Transformable ABE)
4. Efficiency Analysis
5. Conclusion
tABE Integration

We propose the tABE primitive to be integrated into SecLoc.

tABE Integration

Setup($\lambda$) = ($pk, msk$);
KeyGen($msk$) = ($dk_1, sd$);
Encrypt($m, \{u_i\}, t_i$) = ($ct_{t_i}, sig$);
Delegate($ct_{t_i}, sig$) = ($st_{t_i \rightarrow t_j}, dk_2$);
tABE Overview

Basic Idea

tABE is based on KP-ABE [Goyal 06] but with two tricks:

- We use a *share* of the master key $msk$ for generating the ABE secret key $dk_1$, while assigning another share $sd$ to region server, which makes the decryption depend on both storage node’s secret key $dk_1$ and a temp key $dk_2$ generated from the region server.

- We introduce a *coupling* between the ciphertext and the corresponding “ticket” $sig$, so that the storage node can fetch the corresponding $dk_2$ from the region server only for some specified ciphertext.
Brief Construction

- **Setup**($\lambda$) = ($pk$, $msk$) where $pk = (\{g^{li}\}_{i=1}^n, e(g,g)^\alpha)$ and $msk = \alpha$.

- **KeyGen**($msk, P, t_i$) = ($dk_1$, $sd$) where $dk_1 = (\{g^{qx(0)/li}\}$ and $sd = (t_i, \alpha_2)$.

- **Encrypt**($\omega, m, t_i$) = ($ct_{t_i}$, $sig$) where $ct_{t_i} = (m \cdot e(g,g)^{\alpha s}, g^s, \{g^{li_s}\}_{i \in \omega}, g^{rs})$ and $sig = g^r$.

- **Delegate**($sd, sig, t_i$) = ($st_{t_i \rightarrow t_k}$, $dk_2$) where $st_{t_i \rightarrow t_k} = t_k / t_i$ and $dk_2 = (g^{\alpha_{21}/t_k}, g^{\alpha_{22}/r})$.

- **Transform**($st_{t_i \rightarrow t_k}$, $ct_{t_i}$) = $ct_{t_k}$ where $ct_{t_k} = (m \cdot e(g,g)^{\alpha s}, \{g^{li}\}_{i \in \omega}, g^{tks}, g^{rs})$.

- **Decrypt**($dk_1$, $dk_2$, $ct_{t_k}$) = $m$. 

Anna Squicciarini (PSU)  SACMAT 2015  May 28, 2015  14 / 22
Security Analysis

- $A_I$ models a set of “curious” storage nodes (not within the target region) colluding with some compromised region servers.
- $A_{II}$ models the (qualified) storage node colluding with storage nodes and (compromised) region servers in other regions.

Security Results

- We introduce sDBDH assumption and show it is not stronger than existing wDBDHI assumption (Theorem 1).
- tABE scheme is secure against $A_I$ and $A_{II}$ under sDBDH assumption (Theorem 2, 3).
Outline

1. Introduction
2. SecLoc (Secure Location-Sensitive Storage)
3. tABE (Transformable ABE)
4. Efficiency Analysis
5. Conclusion
## Complexity Analysis on SecLoc

<table>
<thead>
<tr>
<th>Stage</th>
<th>Computation Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>Registration Stage</td>
<td>$(2n + 1)E_G + 1P$</td>
</tr>
<tr>
<td>Storage Stage</td>
<td>$(</td>
</tr>
<tr>
<td>Retrieval Stage</td>
<td>$2E_G$ (for region server)</td>
</tr>
<tr>
<td></td>
<td>$1E_G + 3P + \xi_2</td>
</tr>
</tbody>
</table>

$\xi_1, \xi_2$ are the coefficients of computation complexity in symmetric encryption and decryption.
Simulation

We adopted a trivial method as a baseline, where user encrypts file expected to be stored in one region per key. It is impractical for suffering from a large number of encryption keys.

Registration Stage:

<table>
<thead>
<tr>
<th>Number of Regions</th>
<th>Registration Time (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>2.553</td>
</tr>
<tr>
<td>20</td>
<td>3.816</td>
</tr>
<tr>
<td>30</td>
<td>4.189</td>
</tr>
<tr>
<td>40</td>
<td>6.114</td>
</tr>
<tr>
<td>50</td>
<td>6.505</td>
</tr>
</tbody>
</table>

- The time of SecLoc grows linearly with the number of regions.
- The baseline approach does not need the parameter assignment.
SecLoc takes (slightly) longer than the baseline approach. SecLoc additionally utilizes tABE to encapsulate (storage stage) and decapsulate (retrieval stage) the symmetric data key.
Summary

- SecLoc is slower than the baseline approach due to two main reasons:
  - SecLoc requires a “heavy” registration for assignments;
  - SecLoc takes longer for encapsulating and decapsulating key.

- We argue the registration overhead is required only once for initialization, while tABE could be restricted on millisecond level, which is negligible when the file size is large enough.

- Compared with the baseline approach, SecLoc introduces small overhead, and achieves securing location-sensitive storage, while simplifying the key management.
1. Introduction

2. SecLoc (Secure Location-Sensitive Storage)

3. tABE (Transformable ABE)

4. Efficiency Analysis

5. Conclusion
Conclusion

- We propose the SecLoc framework guaranteeing
  - cloud user’s data is stored and can be processed only at expected locations;
  - user data becomes unaccessible at ineligible locations.

- SecLoc construction is based on the tABE primitive, the security of which is proved under a new introduced computational assumption sDBDH.

- We also show sDBDH assumption is at least not stronger than the non-standard wDBDHI assumption and the standard DLP assumption.

- Both complexity and experimental analysis demonstrate the effectiveness of SecLoc.