Privacy – Preserving Data Dissemination in Untrusted Cloud

Denis Ulybyshev, Bharat Bhargava, Donald Steiner, Leon Li, Jason Kobes, Harry Halpin, Miguel Villarreal – Vasquez, Aala Alsalem, Rohit Ranchal

1 Computer Science, CERIAS, Purdue University
2 Northrop Grumman
3 MIT
4 IBM
Outline

• Problem Statement
• Related Work
• Core Design
• Thesis contributions
• Demonstrations and Experiments
• Future Work

ACKNOWLEDGMENT
This work was funded by the Northrop Grumman Cybersecurity Research Consortium. Paper approved for public release by Northrop Grumman, Case #17-0995. The prototype was implemented in collaboration with Northrop Grumman and W3C / MIT and presented internally to Northrop Grumman in April, 2016. We are thankful to Prof. Leszek Lilien and Prof. Weichao Wang for their collaboration and valuable feedback.
Problem Statement

Privacy – preserving role – based and attribute – based data dissemination

• Authorized service can only access data items for which it is authorized
• Role – based data dissemination
• Attribute- and context – based data dissemination
• Periodic computation of trust level of services
Problem Statement

Scenario of EHR Dissemination in Cloud (suggested by Dr. Leon Li, NGC)
Problem Statement

Data dissemination in SOA

D, P

CLOUD PROVIDER 1
D, P

TRUSTED DOMAIN

SERVICE 1
\(d_1\)

SERVICE 2
\(d_2\)

UNKNOWN DOMAIN

CLOUD PROVIDER 2
D, P

SERVICE 3
\(d_3\)

• Data \((D) = \{d_1, \ldots, d_n\}\)
• Access Control Policies \((P) = \{p_1, \ldots, p_k\}\)

Service 2 forwards \(D\) and \(P\) to other cloud

Data Leaks ?
Privacy – Preserving Data Dissemination based on:

- Active Bundles with policies and policy enforcement engine
- Central Monitor constantly re-computing trust level of services
- Secure Browser with detection of its cryptographic capabilities

Features:

- Is independent from TTP
- Data owner’s availability is not required
- Dissemination considers client's attributes
  - Crypto capabilities of a browser
  - Trust level (which is constantly recomputed)
  - Authentication method
  - Type of client’s device
- On-the-fly data updates are supported
- Secure key generation scheme
Related Work

**Policy-based Data Dissemination**

- Policy enforcement at browser's side [8]  
  (Prof. Matteo Maffei, Saarland University, Germany)
  - Micro-policies specified in terms of tags, used to label URLs, network connections, cookies, etc and a transfer function
  - Transfer function defines permitted operations by the browser based on tags.
  - Trust level of clients is not constantly monitored and recalculated in the data dissemination model
  - Requires browser’s code modification
  - Implemented as a Chrome plugin (MiChrome [9])
Policy-based Data Dissemination

- “Encore” (sticky policies) system [7]
  - Policies and data are made inseparable
  - Policies are enforced by TTP
  - Policies are prone to tamper attacks from malicious recipients
  - Prone to Trusted Third Party (TTP)-related issues

- Privacy – preserving information brokering (PPIB) [6]
  - Divides processing among multiple brokers, no single component has enough control to make a meaningful inference from data disclosed to it
  - Prone to centralized TTP (manages keys, metadata) issues
Attribute and Role–based Data Dissemination

AUTHENTICATED CLIENT

Browser’s Crypto Level: High
Authentication Method: Fingerprint
Client’s device: Desktop
Source network: Corporate Intranet
Role: Doctor

Email
ACCESSIBLE TEXT

AUTHENTICATED CLIENT

Browser’s Crypto Level: Low
Authentication Method: Password
Client’s device: Mobile
Source network: Unknown
Role: Insurance Agent

Email
ACCESSIBLE TEXT
INACCESSIBLE TEXT 1
INACCESSIBLE TEXT 2
Active Bundle (AB) parts [10], [11]

- *Sensitive data:*
  - Encrypted data items

- *Metadata:* describe AB and its access control policies
  - Policies [14], [15] manage AB interaction with services and hosts

- *Policy Engine* [18]: enforces policies specified in AB
  - Provides tamper-resistance of AB [1]
Key-value pair stored in the Active Bundle:

\{ \text{“ab.patientID”} : \text{“Enc(0123456789)”} \}
\{ \text{“ab.name”} : \text{“Enc(‘Monica Latte’)”} \}

**Policy Examples:**

<table>
<thead>
<tr>
<th>Resource</th>
<th>patientID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject's Role</td>
<td>Doctor, Insurance, Researcher</td>
</tr>
<tr>
<td>Action</td>
<td>Read</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject's Role</td>
<td>Doctor, Insurance</td>
</tr>
<tr>
<td>Action</td>
<td>Read</td>
</tr>
</tbody>
</table>
Key Generation

1. AB Template [1] used to generate new ABs with data and policies (specified by data owner).
2. AB Template includes implementation of invariant parts (monitor) and placeholders for customized parts (data and policies).
3. AB Template is executed to simulate interaction between AB and service requesting access to each data item of AB.

\[ \text{Aggregation}\{d_i\} \]

- Generated AB modules execution info;
- \(\text{Digest(AB Modules)}\);
- Resources: authentication code + CA certificate, authorization code, applicable policies + evaluation code

Key Derivation Module

\[ \text{(javax.crypto SecetKeyFactory)} \]

\[ \text{ENC}_{k_i}(d_i) \]

- \(K_i\)
Key Generation (Cont.)

- Info generated during the execution and digest (modules) and AB resources are collected into a single value.

- Value for each data item is input into a Key Derivation module (such as `SecretKeyFactory`, `PBEKeySpec`, `SecretKeySpec` from `javax.crypto` library).

- Key Derivation module outputs the specific key relevant to the data item.

- This key is used to encrypt the related data item [1].
Key Derivation

Aggregation\{d_i\} \ (\ - \textit{Generated AB modules execution info};
\ - \textit{Digest(AB Modules)},
\ - \textit{Resources: authentication code + CA certificate, authorization code, applicable policies + evaluation code})

- AB receives data item request from a service
- AB authenticates the service and authorizes its request (evaluates access control policies) \[1\]

Key Derivation (Cont.)

• Info generated during the AB modules execution in interaction with service, and digest (AB modules) and AB resources are aggregated into a single value for each data item [1]

• Value for each data item is input into the Key Derivation module

• Key Derivation module outputs specific key relevant to data item

• This key is used decrypt the requested data item

• If any module fails (i.e. service is not authentic or the request is not authorized) or is tampered, the derived key is incorrect and the data is not decrypted
Other Key Distribution Methods

• Centralized Key Management Service
  • TTP used for key storage and distribution
  • TTP is a single point of failure

• Key included inside AB
  • Prone to attacks!
Tamper Resistance of AB

- Key is not stored inside AB [2]
- Separate symmetric key is used for each separate data value
- Ensure protection against tampering attacks

Aggregation \{d_i\} (\text{Execution info; Digest(AB Modules); Resources})

Aggregation \{d_i\} (\text{Tampered (Execution info; Digest(AB Modules); Resources)})
AB Use Cases

- **Hospital Information System (collection of EHRs)**
  - Doctor, Researcher and Insurance are authorized for different parts of patient's EHR [5]
  - Database of EHRs is hosted by untrusted cloud provider

- **Secure Email**
  - Email is AB
  - Entire email can be sent to the whole mailing list
  - Recipients are authorized for different fragments of email
  - It is guaranteed for the sender that each recipient will only see those email fragments it is authorized for
  - No need for multiple mailing lists for different authorization levels

- **Online shopping**
  - Decentralized data accesses: data can travel across the services

**References**

AB Use Cases: Online Shopping

- Name
- Email
- Payment type
- Credit card
- Shipping preference
- Mailing address

**Shopping Service**

1. order request
   - Active Bundle

2. verify request
   - Active Bundle

**Seller Service**

- E(Name)
- E(Email)
- E(Payment type)
- E(Credit card)
- Shipping preference
- E(Mailing address)

**Payment Service**

3. shipping request
   - Active Bundle

**Shipping Service**

- Name
- E(Email)
- E(Payment type)
- E(Credit card)
- E(Shipping preference)
- Mailing address
NGC TechFest’16 Demo: Electronic Health Record Dissemination in Cloud

**Active Bundle: Contact, Medical and Billing Information**

- **Researcher**
  - E(Contact Info)
  - Medical Info
  - Billing Info

- **Insurance**
  - Contact Info
  - E(Medical Info)
  - Billing Info

- **Doctor**
  - Contact Info
  - Medical Info
  - Billing Info

- **Hospital (NodeJS Server)**

- **Client (Browser)**

1. HTTP GET Request
2. Hospital’s Web Page
3. HTTP POST with Data Request and Role
4. HTTP 302 with AB Request and Role
5. HTTP Get Request
6. AS Web Page
7. HTTP POST with Credentials
8. HTTP 302 with Ticket
9. HTTP Get Request with Ticket
10. Data provided by AB
Data dissemination features

**Data Dissemination based on:**

- Access control policies
- Trust level of a subject (service, user)
- Context (e.g. emergency vs. normal)
- Security level of client’s browser (crypto capabilities) [16], [17]
- Authentication method (password-based, fingerprint, etc)
- Source network (secure intranet vs. unknown network)
- Type of client’s device: desktop vs. mobile (detected by Authentication Server)
Lightweight encryption

- Can be used in Active Bundle instead of regular AES [1]

<table>
<thead>
<tr>
<th>_cipher</th>
<th>key size [bits]</th>
<th>block size [bits]</th>
<th>throughput at 4 MHz [kbit/sec]</th>
<th>relative throughput (% of AES)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hardware-oriented block ciphers</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DES</td>
<td>56</td>
<td>64</td>
<td>29.6</td>
<td>38.4</td>
</tr>
<tr>
<td>DESXL</td>
<td>184</td>
<td>64</td>
<td>30.4</td>
<td>39.3</td>
</tr>
<tr>
<td>Hight</td>
<td>128</td>
<td>64</td>
<td>80.3</td>
<td>104.2</td>
</tr>
<tr>
<td><strong>Software-oriented block ciphers</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AES</td>
<td>128</td>
<td>128</td>
<td>77.1</td>
<td>100.0</td>
</tr>
<tr>
<td>IDEA</td>
<td>128</td>
<td>64</td>
<td>94.8</td>
<td>123</td>
</tr>
</tbody>
</table>
Notes

1. Assumption: hardware and OS are trusted

2. Data is extracted from Active Bundle at a server side and send to client via https
   - Data can't be tampered
Contributions

Contributes to Data Privacy, Integrity and Confidentiality

- Dissemination does not require data owner’s availability
- TTP-independent for recipient’s key generation
- Trust level of subjects is constantly recalculated
- On-the-fly key generation
- Supports data updates for multiple subjects
- Agnostic to policy language and evaluation engine
- Tamper-resistance: data and policies integrity is provided
- Compatible with industry-standard SOA / cloud frameworks
Performance overhead of Active Bundle with detection of browser's crypto capabilities on / off
Performance overhead of Active Bundle for data request from insecure / secure browser
Evaluation

Performance overhead of Active Bundle, hosted by Google Cloud
Deliverables

- **Prototype implementation:**
  - Privacy – Preserving Data Dissemination Prototype
  - Active Bundle Module
    - AB implementation as an executable JAR file
    - AB API implementation using Apache Thrift RPC framework
    - Policy specification in JSON and evaluation using WSO2 Balana


- **Documentation:**
  - Deployment and user manual
Future Work

- Lightweight encryption schemes in Active Bundle instead of AES
- Isolated AB Execution (Linux Docker Containers)
- Data Leakage Detection
- Encrypted Search over Database of Active Bundles
References


