Integrating OpenID with proxy re-encryption
to enhance privacy in cloud-based identity services

David Nuñez, Isaac Agudo, and Javier Lopez

Network, Information and Computer Security Laboratory (NICS Lab)
Universidad de Málaga, Spain
Email: dnunez@lcc.uma.es

December 4, 2012
1. Introduction
   Motivation
   Proposal

2. Support technologies
   OpenID
   Proxy Re-Encryption

3. Privacy-preserving IDaaS system
   General overview
   System operation
   Implementation
   Analysis

4. Conclusions
Introduction

- Identity Management is a ubiquitous service
- Costly $\Rightarrow$ specific applications and personnel

**Identity Management as a Service (IDaaS)**
- Cloud computing solution to this problem
- Organizations can outsource their IdM services to the cloud
- Cloud providers specialized in Identity Management
- New business opportunities to cloud providers
Motivation

- Classic problem of cloud computing
  ⇒ The user loses the control of his data

- Now we are talking about identity data...
  ⇒ Data protection laws and regulations

- Current solution: Service Level Agreements (SLAs)
  ⇒ It is just an agreement not a technical safeguard

- Trust problem ⇒ Users are obliged to trust the provider

- **Goal**: To define technical safeguards that allow an IdM service without compromising users’ data
Proposal: Privacy-preserving IDaaS

- Privacy-preserving IDaaS system

- Based in OpenID Attribute Exchange and Proxy Re-Encryption

- Identity attributes are encrypted by the user and decrypted by the requester

- The Identity Provider (IdP) stores encrypted attributes
  \[\Rightarrow\] Still capable of offering an identity service

- First proposal that tackles this problem
OpenID: Overview

- Decentralized model for identity management
- User’s identity is represented by an *OpenID identifier*
- Current version is OpenID 2.0
- Defines an extension for attribute exchange
  ⇒ OpenID Attribute Exchange 1.0
OpenID Authentication protocol

Figure: OpenID Authentication sequence diagram
OpenID: Problems

- Identity information assurance
- Lack of trust framework
- Privacy
Proxy Re-Encryption: Overview

A PRE scheme is a public-key encryption scheme that permits a proxy to transform ciphertexts under Alice’s public key into ciphertexts under Bob’s public key.

The proxy needs a re-encryption key $r_{A\rightarrow B}$ to make this transformation possible.

![Proxy Re-Encryption flow](image)

Figure: Proxy Re-Encryption flow
Proxy Re-Encryption: AFGH scheme

Global parameters:
- $\mathbb{G}_1, \mathbb{G}_2$ are groups of prime order $q$
- $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is a bilinear pairing
- $g \in \mathbb{G}_1, Z = e(g, g) \in \mathbb{G}_2$

Primitives:
- Key Generation: $KG() = (s_A, p_A)$
- Re-Encryption Key Generation: $RKG(s_A, p_B) = r_{A\rightarrow B}$
- First-level Encryption: $E_1(m, p_A) = c_1$
- Second-level Encryption: $E_2(m, p_A) = c_2$
- Re-Encryption: $R(c_2, r_{A\rightarrow B}) = c_1$
- First-level Decryption: $D_1(c_1, s_A) = m$
- Second-level Decryption: $D_2(c_2, s_A) = m$
Proxy Re-Encryption: AFGH scheme

\[ c_1 \in G_2 \times G_2 \quad \text{and} \quad c_2 \in G_1 \times G_2 \]

\[ m \in G_2 \]

\[ E_1 \quad D_1 \quad D_2 \quad E_2 \]

\[ R \]

Figure: Transformations between plaintext and ciphertext spaces

Properties:
- Unidirectional
- Unihop
- Collusion-resistant
Privacy-preserving IDaaS system: overview

Outline
- Introduction
- Support technologies
- Privacy-preserving IDaaS system
- Conclusions

Privacy-preserving IDaaS system

- OpenID Provider
  - Encrypted attributes
  - Re-encryption
- OpenID Consumer
  - Decryption
- User
  - Identity Provider
  - Service Provider
  - Encryption

Equations:
- \( E(p_U, a_1), \ldots, E(p_U, a_n) \)
- \( (p_U, s_U) \)
- \( r_{U \rightarrow SP} \)
- \( E(p_{SP}, a_i) \)
Privacy-preserving IDaaS system: assumptions

- **Honest-but-curious provider**: The cloud provider will respect protocol fulfillment, but will try to read users’ data.

- Existing trust relationship between users and requesters.
Privacy-preserving IDaaS system: main interactions

1. Requests access
2. Asks for authn. and attributes
3. User authenticates
4. Retrieves ciphered attributes and re-encrypts them
5. Sends authn. result and re-encrypted attributes
6. Decrypts attributes
Instantiation with OpenID AX

Figure: Modified OpenID sequence
Implementation details

We have implemented:

- OpenID Provider and Consumer using the OpenID4Java library\(^1\)
- AFGH Proxy Re-Encryption scheme using Java Pairing-Based Cryptography library (jPBC)\(^2\)

---

\(^1\) [http://code.google.com/p/openid4java](http://code.google.com/p/openid4java)

\(^2\) A. D. Caro, [http://gas.dia.unisa.it/projects/jpbc](http://gas.dia.unisa.it/projects/jpbc)
Economic analysis

- Most of proposals do not analyze their economic impact.
- Cryptographic operations have an economic cost due to computation, communication, etc.
  \[ \Rightarrow \text{Cloud provider incurs in expenses due to energy consumption, personnel, ...} \]
- Our estimations are based on a research from Chen & Sion\(^3\)
  \[ \Rightarrow \text{They give estimations for computation, storage and communication costs, expressed in picocents (1 picocent} \quad \text{=} \quad 10^{-12} \text{USD cent)} \]
- We estimate the number of CPU cycles to give an approximation of the costs.

---

\(^3\)Y. Chen and R. Sion, “On securing untrusted clouds with cryptography” in Proc. 9th annual ACM workshop on Privacy in the Electronic Society.
### Economic analysis: time measurements

**Table**: Performance results for the main operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Time (ms)</th>
<th>Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generation of global parameters</td>
<td>7279.98</td>
<td>1.94E+10</td>
</tr>
<tr>
<td>Generation of a secret key</td>
<td>0.01</td>
<td>1.86E+04</td>
</tr>
<tr>
<td>Generation of a public key</td>
<td>20.05</td>
<td>5.33E+07</td>
</tr>
<tr>
<td>Generation of re-encryption key</td>
<td>139.66</td>
<td>3.72E+08</td>
</tr>
<tr>
<td>Encryption</td>
<td>23.31</td>
<td>6.20E+07</td>
</tr>
<tr>
<td>Re-encryption</td>
<td>90.09</td>
<td>2.40E+08</td>
</tr>
<tr>
<td>Decryption</td>
<td>14.28</td>
<td>3.80E+07</td>
</tr>
</tbody>
</table>
Economic analysis: costs

**Table**: Costs in picocents for the main operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Cost per operation</th>
<th>Operations per cent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encryption</td>
<td>4.34E+08</td>
<td>2304</td>
</tr>
<tr>
<td>Re-encryption</td>
<td>4.79E+08</td>
<td>2087</td>
</tr>
<tr>
<td>Decryption</td>
<td>5.70E+08</td>
<td>1755</td>
</tr>
</tbody>
</table>
Economic analysis: example scenario

- IDaaS provider that handles 1 million attribute requests per day $\Rightarrow$ 1 million re-encryptions per day

- Approx. 2000 USD per year

- Reasonable cost for an average-sized company, considering that their information is encrypted at the cloud provider
Conclusions

- IDaaS is a promising paradigm for organizations

- Cloud providers are in a privileged position to gain information about their users

- We need technical safeguards, such as those based in cryptography, to ensure users’ privacy
Conclusions

- In this work, we describe an IDaaS system that handles encrypted attributes and still provides an identity service.

- Our system is based in OpenID Attribute Exchange and Proxy Re-Encryption.

- The cloud identity provider transforms encrypted attributes from the original users to ciphertexts for the requesters using re-encryption.

- Implementation and economic analysis is provided.
Future work

- More secure and efficient proxy re-encryption schemes
- Improve trust and assurance
- Other identity management protocols (e.g., SAML)
- Evaluation in a real cloud setting
Thank you!