Compact and Anonymous Role-Based Authorization Chains

Danfeng (Daphne) Yao
Department of Computer Science
Rutgers University, New Brunswick

Joint work with Roberto Tamassia, Brown University

Outline

- Motivation for anonymity and aggregation
- Construction of Anonymous-Signer Aggregate Signature Scheme
- Security properties of the scheme
- Applications
Digital credential

- Digital credential is signed by the issuer with a digital signature scheme
  - To certify the credential holder
- Digital signature scheme
  - Signing uses the private key
  - Verification uses the public key

Bob is a university professor
University's signature

Bob's credential can be verified against university's public key

Bank cashiers

Motivation: Anonymous authorization

- Group signature schemes
  - [Chaum van Heijst 91, Ateniese Camenisch Joye Tsudik 00, Boneh Boyen Shacham 04, Camenisch Lysyanskaya 04]
  - Support anonymity
**Motivation: Aggregation**

1. Request
2. Authorization
3. Authorization
4. Authorization

[Boneh Gentry Shacham Lynn 03]

**Our goal: Aggregate anonymous signatures**

- Signing anonymity
- Signature aggregation

[Signatures] → Aggregate → [Aggregate Signature]
Anonymous authorization chain

1. Request
2. Authorization
3. Authorization
4. Authorization

Our anonymous-signer aggregate (ASA) signature scheme

- Goals: (1) role member signs anonymously (2) signature aggregation
- Properties
  - Aggregation: Bob’s signature can be added with Alice’s
  - Unforgeability: No one can forge a valid signature without being a role member
  - Anonymity: No one can tell that a signature is signed by Bob
  - Unlinkability: No one can tell that two signatures are from the same signer
  - Exculpability (non-framing): No one can sign on behalf of Bob
  - Traceability: The role manager can revoke Bob’s anonymity
  - Collusion-resistance: Collusion does not affect the security
- Our approach: one-time signing key of Bob is a randomized long-term private key of his
  - Based on BGLS aggregate signature [Boneh Gentry Shacham Lynn 03]
Aggregate signature scheme

- Aggregate signature scheme [Boneh Gentry Shacham Lynn 03]
  - The size of signatures and public keys 170 bits with security comparable to 1024 bit RSA and 320 bit DSA schemes
- Verification is linear in the number of individual signatures

The size of signatures and public keys 170 bits with security comparable to 1024 bit RSA and 320 bit DSA schemes

Verification is linear in the number of individual signatures

How to make the aggregate signature scheme support anonymity?

An attempt to support anonymity using the existing aggregate signatures

- Signers sign with certified one-time signing keys

Does not satisfy the non-framing requirement!
Our solution: anonymous-signer aggregate signature scheme

- Signing key has two parts
  - Long term public key certified by CA
  - Random one time secret
  - Combined to become the signing key

- Supports
  - Signature aggregation
  - Anonymous authorization

- Based on the aggregate signature scheme [Boneh Gentry Shacham Lynn 03]

- Standard assumptions for pairing-based cryptography

Overview: Anonymous-signer aggregate signature scheme

- Long term public key
- Trusted third party
  - Certifies with aggregate signature
- Public key certificate
- One-time secret
- Combine
- One time member certificate
- Bank admin
  - Certifies with aggregate signature
  - Cannot frame others
- Signs with
- Please sign my check
- Aggregates $S_a + S_m$
- Verifies with signing key
Entities and Operations in Our Scheme

- **Entities**
  - Role manager (cashier in this talk)
  - Role member (bank admin in this talk)

- **Setup**: Each entity chooses long-term public/private key pair
- **Join**: A user becomes a role member
  - Obtains *membership certificates*
- **Sign**: An entity signs on behalf of the role
  - Operation Sign produces a *role signature*
- **Aggregate**: Multiple role signatures are aggregated
- **Verify**: Aggregate role signatures are verified
- **Open**: A role manager revokes the anonymity of a signer by revealing his or her identity

Some math about the operations

- **π** Public parameter

- Private key $s_u$
- Public key $P_u = s_u \pi$
- One time signing secret $x_u$
- One time signing public key $s_u x_u \pi$

- Private key $s_a$
- Public key $P_a = s_a \pi$
- Certifies $s_a H(m)$

<table>
<thead>
<tr>
<th>$s_a$</th>
<th>Signature $s_a x_a H(m)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_a$</td>
<td>+ $s_m$</td>
</tr>
<tr>
<td>$s_a$</td>
<td>Role signature; may be aggregated further with others</td>
</tr>
</tbody>
</table>

Framing is hard – equivalent to computational Diffie-Hellman Problem
Security
Our anonymous signer aggregate signature scheme satisfies the following requirements:
- correctness,
- unforgeability,
- anonymity,
- unlinkability,
- traceability,
- non-framing,
- coalition-resistance,
and aggregation
assuming
random oracle model, bilinear map, and gap groups.

Non-framing property
- Our scheme protects a cashier from being framed by anyone including bank admin
- Consider a simple attack by an admin
  - Picks random \( x^* \) and \( s^* \) and uses \( x^*s^* \) to sign
- Admin cannot misattribute a signature to a cashier \( u \)
  - \( u \) with pub key \( P_u = s_u \pi \)
  - \( e(s^*x^*x^*, \pi) \neq e(P_u, x^*x^*) \)
- In general, framing is equivalent to
  - Computing \( b \pi \), given \( q, a \pi, \) and \( c \pi \) such that
    \[
    ab \equiv c \mod q
    \]
known equivalence to CDH problem [Chen Zhang Kim 03]
Anonymous-signer aggregate (ASA) signature summary

- **Assumptions**: computation Diffie Hellman problem is hard, decision Diffie Hellman problem is easy; existence of an admissible pairing.
- **Theorem** Join takes $O(k)$, where $k$ is the number of one time signing keys certified. Verify takes $O(n)$, where $n$ is the number of signatures aggregated.
- **Theorem** Our ASA signature scheme is as secure as the BGLS aggregate signature scheme against existential forgery attacks.
- **Theorem** Our ASA signature scheme from bilinear pairings in gap groups preserves anonymity, traceability, and exculpability in the random oracle model.
- Unlinkability and collusion-resistance follow as corollaries.

An application: Anonymous role-based delegation

- University prof. can access
- The access to the digital library at a hospital is controlled
- Bob can access
- Need to access
- Bob is a university professor and can access
- Collaborate
- Researchers at a company collaborate with Bob
- Collaborate
- Engineers at a lab collaborate with researchers
- Need to access
Another application: Protecting whistleblower

- Protects the identity of whistleblowers
  - The verifier only knows that the whistleblower is a certified FBI agent or a New York Times reporter
- Supports efficiently certification of a series of reports

Signed reports of whistleblower(s)
- Enron scandal: day 101 $S_1$
- Enron scandal: day 102 $S_2$
- Enron scandal: day 103 $S_3$
- Aggregated signature $S$ 

Some other IBE related work that I did:

Forward-Secure Hierarchical ID-Based Encryption Scheme

Joint work with Nelly Fazio (IBM Research), Yevgeniy Dodis (NYU), Anna Lysyanskaya (Brown University)
Why need forward-secure Hierarchical IBE?

- In HIBE, exposure of parent private keys compromises children's keys
- Forward secure HIBE mitigates key exposure
- Forward security
  - [Gunther 89] [Diffie Oorschot Wiener 92] [Anderson 97] [Bellare Miner 99] [Abdalla Reyzin 00] [Malkin Micciancio Miner 02] [Canetti Halevi Katz 03]
  - Secret keys are evolved with time
  - Compromising current key does NOT compromise past communications

Overview of our fs-HIBE scheme

- Based on HIBE [Gentry Silverberg 02] and fs PKE [Canetti Halevi Katz 03] schemes
- Scalable, efficient, and provable secure
  - Forward security
  - Dynamic joins
  - Joining time obliviousness
  - Collusion resistance
- Security based on Bilinear Diffie-Hellman assumption [BF 01] and random oracle model [Bellare Rogaway 93]
  - Chosen-ciphertext secure against adaptive chosen (ID, tuple, time) adversary
Security of fs-HIBE

- Security definitions
  - Secure for past communications of compromised nodes
  - Secure for ancestor nodes
  - Secure for sibling nodes
- Security based on hardness of BDH problem and random oracle model
- **Theorem** Suppose there is an adaptive adversary A
  - \( \varepsilon \): advantage against one way secure fs HIBE
  - \( h \): level of some target ID tuple
  - \( l = \log_2 N \) and \( N \) is the total number of time periods
  - \( H_1, H_2 \): random oracles
  - \( q_{H_1} \): number of hash queries made to hash function \( H_1 \)
  - \( q_{H_2} \): number of hash queries made to lower level setup queries
  - then there exists an algorithm B that solves BDH problem with advantage

\[
\varepsilon \left( \frac{h + l}{\sqrt{2q_{H_1} + h + l}} \right)^{(h+l)/2} = \frac{1}{2^{h/2}} \sqrt{q_{H_2}}
\]

References