Anonymous Proxy Signatures

G. Fuchsbauer D. Pointcheval
École normale supérieure
SCN '08
09/11/2008

Application: GRID computing

User authenticates herself and starts process which needs to authenticate to resources / start subprocesses
⇒ Delegation and re-delegation of signing rights
No need to know that it was not the user herself to be authenticated

Our Results

- Algorithm specifications
- Security definitions
- Proof of concept: existence assuming trapdoor permutations

Anonymous Consecutive Delegation of Signing Rights

Delegation A delegator delegates his signing rights to a proxy signer (or delegatee) who can then sign on the delegator's behalf

Consecutiveness A delegatee may re-delegate the received signing rights ⇒ intermediate delegators

Anonymity All intermediate delegators and the proxy signer remain anonymous

After verifying a proxy signature one knows that someone entitled signed but nothing more

Relation to Other Primitives

Anonymous proxy signatures are a generalization of

- Proxy signatures (consecutive delegation) formalized by [BPW03]
- Group signatures (anonymity) formalized by [BMW03, BSZ05]
  - dynamic (users can join after setup of group)
  - hierarchical (tree structure by consecutive delegations) [TW05]

and satisfy the respective security notions
Group Signatures

Group public key: $pk$

Algorithms for (Dynamic) Group Signatures

Algorithms
- **Setup** produces group public key, issuing key, opening key
- **Reg** registers new members joining the group using the issuing key
- **Sig** enables a group member to sign on behalf of the group
- **Ver** checks validity of a group signature using the group public key
- **Open** reveals the signer's identity using the opening key

Security Definitions for (Dynamic) Group Signatures

Security [BSZ05]
- **Anonymity**: no one except the opener can tell who produced a signature
- **Traceability**: every valid signature can be traced to its signer by the opener
- **Non-Frameability**: no one can produce a signature that opens to a member who did not sign

Proxy Signatures

Proxy Signatures
- **Delegator ($pk_D$)**
- **Delegatee/Signer**
- **Verify($pk_D$, msg)**
Proxy Signatures, Consecutive Delegations

Delegator \((pk_D)\)

Delegator 2

Delegator 3

ANONYMOUS

Proxy Signer

Opener \((ok_D)\)

open

Example: Redelegation of Reduced Task Set

\[\text{User } x (pk_x, sk_x)\]

\[\text{Del}(sk_x, (2,5,7), pk_y) \rightarrow \text{warr}\]

\[\text{User } y (pk_y, sk_y)\]

\[\text{Del}(sk_y, \text{warr}, (5,7), pk_y) \rightarrow \text{warr}'\]

\[\text{User } z (pk_z, sk_z)\]

\[\text{PVer}(pk_z, 5, M, \sigma) \rightarrow 1\]

\[\text{Open}(ok_x, 5, M, \sigma) \rightarrow \{pk_y, pk_y\}\]

Tasks

Delegation by Certificate

Delegator signs a warrant containing the proxy’s public key \(pk_p\)

Proxy signs message with her own signing key

\[\Rightarrow \text{Verify signature on warrant (w.r.t. } pk_D\text{) and message (w.r.t. } pk_p\text{).}\]

Delegation of Tasks

- possibility to delegate rights only for certain set of tasks
- re-delegate rights for restricted set of tasks

Delegation of \(TList\), a set of natural numbers representing tasks

Algorithms of Anonymous Proxy Signature Scheme \(\mathcal{PS}\)

\[\text{Issuer (ik)} \quad \cdots \quad \text{Reg} \quad \cdots \quad \text{Opener (ock)}\]

\[\text{User}\]

\[\text{Setup} \quad \rightarrow \quad pp \ ik \ ock\]

\[sk_x [\text{warr}_x \rightarrow x] \quad \text{TLiset} \quad pk_y\]

\[pk_x \quad \text{task} \quad M \quad \rightarrow \quad \text{PSig} \quad \rightarrow \quad \text{PVer} \quad \rightarrow \quad b \in \{0,1\}\]

\[ok_x \quad \text{task} \quad M \quad \rightarrow \quad \text{Open} \quad \rightarrow \quad \text{a list of users or } \perp \text{ (failure)}\]
Anonymity I

Idea:

- Adversary controls users and issuer
- produces 2 warrants
- one of them used to sign
- Adversary must decide which one

Restrictions:

- $U_1$ must be registered with the opener
- both warrants correctly formed
- both delegation chains of same length

Anonymity II

The experiment $\text{Exp}_{\text{PSA}}^\text{anon}(\ )$ returns 1 if

- $b = b'$
- no queries $\text{OK}(pk)$ and $\text{Open}(pk \text{ task } M)$ made

Definition

A proxy signature scheme is anonymous if for all p.p.t. adversaries $A$

$$\Pr[\text{Exp}_{\text{PSA}}^\text{anon}(\ ) = 1] - \frac{1}{2} = \text{negl}(\ )$$
Traceability I

Idea:

- Adversary can corrupt users and opener (which follows the protocol)
- gets $\text{SndToI}$ and $\text{SndToO}$ oracles for $\text{Reg}$ that return a transcript between them and opening key
- must produce signature that is valid but not openable

**Definition**

A proxy signature scheme is traceable if for all p.p.t. adversaries $A$

$$\Pr[\text{Exp}^{\text{trace}}_{PSA}(\cdot) = 1] = \operatorname{negl}(\cdot)$$

Non-Frameability I

**Exp_{PSA}^{\text{frame}}(\cdot)**

<table>
<thead>
<tr>
<th>$pp$</th>
<th>$\text{Adversary}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ik$</td>
<td>$\text{Oracles}$:</td>
</tr>
<tr>
<td>$ock$</td>
<td>$\text{SndToI}$</td>
</tr>
<tr>
<td></td>
<td>$\text{SndToO}$</td>
</tr>
</tbody>
</table>

The experiment keeps a list $HU$ of honest users with whom $A$ interacts via $\text{ISndToU}$ and $\text{OSndToU}$. $\text{SK}$ returns the secret key of a user after deleting her from $HU$.

Traceability II

**Exp_{PSA}^{\text{frame}}(\cdot)**

<table>
<thead>
<tr>
<th>$pp$, $ik$, $ock$</th>
<th>Setup($\cdot$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$pp$</td>
<td>$\text{Adversary}$</td>
</tr>
<tr>
<td>$pk$, task $M$</td>
<td>$\text{Oracles}$:</td>
</tr>
<tr>
<td></td>
<td>$\text{SndToI}$</td>
</tr>
<tr>
<td></td>
<td>$\text{SndToO}$</td>
</tr>
</tbody>
</table>

return 1 if $\text{PVer}(pk$, task $M) = 1$ and $\text{Open}(\text{OK}(pk)$ task $M) = 1$
else return 0

Non-Frameability II

The experiment $\text{Exp}_{PSA}^{\text{frame}}(\cdot)$ returns 1 if is valid and its opening reveals

- either a delegation by an honest user which was not queried via $\text{Del}$
- or an honest proxy signer who was not queried via $\text{PSig}$

**Definition**

A proxy signature scheme is non-frameable if for all p.p.t. adversaries $A$

$$\Pr[\text{Exp}^{\text{frame}}_{PSA}(\cdot) = 1] = \operatorname{negl}(\cdot)$$
Generic Construction

using
- Digital signatures (EUF-CMA)
- Public-key encryption (IND-CCA)
- NIZK (simulation sound)
(follow from trapdoor permutations)

Conclusion

- Defined new primitive encompassing group and proxy signatures (satisfies rigorous security notions of both)
- Non-frameable dynamic hierarchical group signatures

Open Problem

- Efficient implementation