## Anonymous Proxy Signatures

**G. Fuchsbauer**  
**D. Pointcheval**  
*École normale supérieure*

**SCN ’08**  
09/11/2008

---

### Delegation

- **Delegation**: A **delegator** delegates his signing rights to a **proxy signer** (or **delegatee**) who can then sign on the delegator’s behalf.

### Consecutiveness

- **Consecutiveness**: A delegatee may **re-delegate** the received signing rights to **intermediate delegates**.

### Anonymity

- **Anonymity**: All intermediate delegates and the **proxy signer** remain **anonymous**.

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

---

### Application: GRID computing

- User authenticates herself and starts a 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

---

### 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: \( pk \)

- **Issuer** (\( ik \))
- **Opener** (\( ok \))
- **Reg**
- **Group members** (\( sk_i \))

Signatures can be signed on behalf of the group, and the signer's identity can be revealed using the opening key.

Algorithms for (Dynamic) Group Signatures

- **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

- **Delegator** (\( pk_D \))
- **Delegatee/Signer**

Verifying a proxy signature involves checking the signature against the delegator's public key and the message.
Proxy Signatures, Consecutive Delegations

Delegator \((pk_D)\)

Delegator 2

Delegator 3

ANONYMOUS

Proxy Signer

Opener \((ok_D)\)

open

Proxy Signatures, Consecutive Delegations

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

Proxy signs message with her own signing key

⇒ Verify signature on warrant (w.r.t. \(pk_D\)) and message (w.r.t. \(pk_P\)).

Delegation by Certificate

Delegation of Tasks

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

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

Example: Redelegation of Reduced Task Set

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

\(\text{Del}(sk_x, (2,5,7), pk_x) \rightarrow \text{warr}\)

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

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

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

\(\text{PSig}(sk_z, \text{warr}', 5, M) \rightarrow \sigma\)

\(\text{PVer}(pk_x, 5, M, \sigma) \rightarrow 1\)

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

Tasks

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

Issuer \((ik)\)

\(\text{Reg} \rightarrow \text{Opener}(ok)\)

\(\lambda \rightarrow \text{Setup} \rightarrow pp, ik, ok\)

\(sk_x, [\text{warr} \rightarrow \ldots] \rightarrow \text{TLst}, pk_y\)

\(\text{Del} \rightarrow \text{warr} \rightarrow \text{TLst} \rightarrow \text{y} \rightarrow \text{x} \rightarrow \text{y}\)

\(sk_y, \text{warr}, \ldots, \text{y}, \text{task}, M \rightarrow \text{PSig} \rightarrow \sigma\)

\(pk_x, \text{task}, M, \sigma \rightarrow \text{PVer} \rightarrow b \in \{0,1\}\)

\(ok_x, \text{task}, M, \sigma \rightarrow \text{Open} \rightarrow \text{a list of users or } \bot \text{ (failure)}\)
Security for Anonymous Proxy Signatures

**Anonymity** intermediate delegators and proxy signer remain anonymous

~ **BUT:** the number of delegations may not remain hidden (if no restriction on number of delegations)

**Traceability** every valid signature can be traced to its intermediate delegators and proxy signer

**Non-Frameability** no one can produce a signature that, when opened, wrongfully reveals a delegator or signer

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 $\text{Exp}_{\text{PS}, A}^{\text{anon}}(\lambda)$ returns 1 if
- $b = b'$
- no queries $\text{OK}(pk)$ and $\text{Open}(pk, task, M, \sigma)$ made

Definition

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

$$\Pr\left[\text{Exp}_{\text{PS}, A}^{\text{anon}}(\lambda) = 1\right] - \frac{1}{2} = \text{negl}(\lambda)$$
Traceability I

Idea:
- Adversary can corrupt users and opener (which follows the protocol)
- gets SndToI and SndToO oracles for 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 $A$

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

Non-Frameability I

$$\text{Exp}_{\text{PSA}}^{\text{n-frame}}(\lambda)$$

The experiment keeps a list $HU$ of honest users with whom $A$ interacts via ISndToU and OSndToU.

$SK$ returns the secret key of a user after deleting her from $HU$.

Traceability II

$$\text{Exp}_{\text{PSA}}^{\text{trace}}(\lambda)$$

Adversary

Oracles:
- SndToI
- SndToO

The experiment $\text{Exp}_{\text{PSA}}^{\text{n-frame}}(\lambda)$ returns 1 if $\sigma$ is valid and its opening reveals
- either a delegation by an honest user which was not queried via Del
- or an honest proxy signer who was not queried via PSig

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

$$\Pr[\text{Exp}_{\text{PSA}}^{\text{n-frame}}(\lambda) = 1] = \text{negl}(\lambda)$$
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