DIGITAL SIGNATURE DELEGATION

M.SC. BARTŁOMIEJ SŁOTA

AGENDA

- DIGITAL SIGNATURE DELEGATION
  - GENERAL DEFINITION
  - MATHEMATICAL DESCRIPTION
  - CLASSIFICATION
  - SECURITY REQUIREMENTS
  - ALGORITHMS
    - SCHNORR SIGNATURE SCHEME
    - MUO SCHEME
    - ZHANG SCHEME

GENERAL DEFINITION

- Enables original signer to delegate signing authority to a proxy signer
  - Temporal absence
  - Lack of time or computational power
- Proxy signer can compute a valid signature, that can be verified with the original signer's public key

MATHEMATICAL DEFINITION

PS = $(\mathcal{G}, \mathcal{K}, \mathcal{S}, \mathcal{V}, (\mathcal{D}, \mathcal{P}), \mathcal{PS}, \mathcal{PV}, \mathcal{ID})$

DS = $(\mathcal{G}, \mathcal{K}, \mathcal{S}, \mathcal{V})$ – digital signature scheme, where

- $\mathcal{G}$ – Randomized parameter-generation algorithm, which takes input 1^κ, where κ is the security parameter, and outputs some global public params
- $\mathcal{K}$ – Key generation algorithm – takes input global parameters and outputs a pair (pk, sk)
- $\mathcal{S}$ – Signature algorithm – takes input a secret key sk and a message $\mathcal{M} \in \{0,1\}^*$ and outputs a signature σ
- $\mathcal{V}$ – Deterministic verification algorithm – takes input a public key pk, a message $\mathcal{M}$, and a candidate signature σ, and outputs a bit (1 if signature is valid, 0 – otherwise)

MATHEMATICAL DEFINITION

PS = $(\mathcal{G}, \mathcal{K}, \mathcal{S}, \mathcal{V}, (\mathcal{D}, \mathcal{P}), \mathcal{PS}, \mathcal{PV}, \mathcal{ID})$

$(\mathcal{D}, \mathcal{P})$ – a pair of randomized algorithms forming proxy-designation protocol

$$skp \leftarrow [\mathcal{D}(pk_i, sk_i, j, pk_j, \omega), \mathcal{P}(pk_j, sk_j, pk_i)]$$

- $\mathcal{D}$ – takes input designator i public key pk_i, proxy signer j public key pk_j, the identity j of the proxy signer, and a message space descriptor ω for which user i wants to delegate its signing rights. Returns no output
- $\mathcal{P}$ – takes input designator i public key pk_i, proxy signer j public and private keys pk_j, sk_j and produces skp – proxy signing key, that user j uses to produce proxy signatures on behalf of user i

MATHEMATICAL DEFINITION

PS = $(\mathcal{G}, \mathcal{K}, \mathcal{S}, \mathcal{V}, (\mathcal{D}, \mathcal{P}), \mathcal{PS}, \mathcal{PV}, \mathcal{ID})$

($\mathcal{PS}$) - (possibly) randomized proxy signing algorithm, that takes input a proxy signing skp and a message $\mathcal{M} \in \{0,1\}^*$, and outputs a proxy signature $p\sigma \in \{0,1\}^* \cup \{\perp\}$

$$p\sigma \leftarrow \mathcal{PS}(skp, \mathcal{M})$$

MATHEMATICAL DEFINITION

$$\text{PS} = (\mathcal{G}, \mathcal{K}, \mathcal{S}, \mathcal{V}, (\mathcal{D}, \mathcal{P}), \mathcal{PS}, \mathcal{PV}, \mathcal{ID})$$

($\mathcal{PV}$) – deterministic proxy verification algorithm, that takes input a public key pk (corresponding to proxy secret key skp), a message $\mathcal{M} \in \{0,1\}^*$, and a proxy signature $p\sigma \in \{0,1\}^* \cup \{\perp\}$ and outputs 0 or 1. and outputs a proxy signature

$$\{0, 1\} \leftarrow \mathcal{PV}(pk, \mathcal{M}, p\sigma)$$

MATHEMATICAL DEFINITION

PS = $(\mathcal{G}, \mathcal{K}, \mathcal{S}, \mathcal{V}, (\mathcal{D}, \mathcal{P}), \mathcal{PS}, \mathcal{PV}, \mathcal{ID})$

($\mathcal{ID}$) – proxy identification algorithm which takes input a valid proxy signature $p\sigma$ and outputs an identity i ∈ $\mathcal{N}$ or $\perp$ in case of error

$$\mathcal{N} \cup \{\perp\} \leftarrow \mathcal{ID}(p\sigma)$$

CLASSIFICATION

- Full delegation
- Partial delegation
  - Proxy-unprotected proxy signature
  - Proxy-protected proxy signature
- Delegation by warrant
  - Delegate proxy
  - Bearer proxy
- Partial delegation with warrant
- Threshold delegation

DIGITAL SIGNATURE DELEGATION

- Security requirements
  - Unforgeability
  - Proxy signer's deviation
  - Secret keys dependence
  - Verifiability
  - Distinguishability
  - Indentifiability
  - Undeniability

Only a designated proxy signer can generate a valid proxy signature

DIGITAL SIGNATURE DELEGATION

- Security requirements
  - Unforgeability
  - Proxy signer's deviation
  - Secret keys dependence
  - Verifiability
  - Distinguishability
  - Indentifiability
  - Undeniability

A proxy signer cannot generate a valid proxy signature which is not detected as generated by him

DIGITAL SIGNATURE DELEGATION

- Security requirements
  - Unforgeability
  - Proxy signer's deviation
  - Secret keys dependence
  - Verifiability
  - Distinguishability
  - Indentifiability
  - Undeniability

A proxy key should always be generated from the original signer's private key

DIGITAL SIGNATURE DELEGATION

- Security requirements
  - Unforgeability
  - Proxy signer’s deviation
  - Secret keys dependence
  - Verifiability
  - Distinguishability
  - Indentifiability
  - Undeniability

From given proxy signature, a verifier can be sure, that the original siger agreed on the signed message

DIGITAL SIGNATURE DELEGATION

- Security requirements
  - Unforgeability
  - Proxy signer’s deviation
  - Secret keys dependence
  - Verifiability
  - Distinguishability
  - Indentifiability
  - Undeniability

A proxy signature must be distinguishable from a normal signature of the original signer

DIGITAL SIGNATURE DELEGATION

- Security requirements
  - Unforgeability
  - Proxy signer’s deviation
  - Secret keys dependence
  - Verifiability
  - Distinguishability
  - Indentifiability
  - Undeniability

The original signer can conclude from a proxy signature who signed the message

DIGITAL SIGNATURE DELEGATION

- Security requirements
  - Unforgeability
  - Proxy signer’s deviation
  - Secret keys dependence
  - Verifiability
  - Distinguishability
  - Indentifiability
  - Undeniability

A proxy signer cannot disavow a proxy signature generated by him

SCHNORR SIGNATURE SCHEME

- Security based on DL problem

Key generation

- Take a large prime number p defining a finite field Zp
- Find g - generator of a multiplicative group Z^*_p
- compute random secret number x, $1 < x < p$
- compute $y = g^x \bmod p$
- {p, g, y} - public key
- x - private key

SCHNORR SIGNATURE SCHEME

Signature generation

- uses private key x, and cryptosystem parameters (p, g)
- select a random integer k, 1 < k < p-1 such that gcd(k, q-1) = 1
- compute r =g^k mod p
- compute e = h(m||r)
- compute s = xe + k mod q
- (s, e) - signature

Signature verification

- uses public key {p, g, y}
- compute e’ = h(m||$g^s y^{-e}$ mod p)
- the signature is valid if and only if e = e’

SCHNORR SIGNATURE SCHEME

Proof

$$g^s y^{-e} \bmod p = g^{xe + k} g^{-xe} \bmod p = g^k \bmod p = r$$

MUO SIGNATURE SCHEME

- Proposed by Mambo, Usuda and Okamoto
- The scheme shows how to create proxy signature algorithms
- Allows to use any DL-based signature scheme to compute proxy signature

MUO SIGNATURE SCHEME

p : a large prime number

q : a prime factor of $(p-1)$

g : an element of Z_p^* of order q

x : secret key of the original signer S, where $x \in_R Z_q$

y : public key of the original signer S, where $y = g^x \mod p$

MUO SIGNATURE SCHEME

Proxy Generation

1. (Key Generation)- The original signer, Alice, selects $k \in_R Z_q$, and computes

$$r = g^k \bmod p \quad \text{and} \quad s = x + kr \bmod q$$

The original signer secretly sends (s, r) to the proxy signer.

2. (Key verification)- The proxy signer checks the validity of the key (s, r) by checking, if

$$g^s = yr^r \bmod p$$

He accepts (s, r) as his secret key iff (s, r) satisfies this congruence.

MUO SIGNATURE SCHEME

Proxy Signing

When the proxy signer, Peter, signs a message m on behalf of the original signer, he computes a signature s_p using any original signature scheme (here we are using Schnorr scheme) and s as the secret key. Peter selects $k_p \in_R (1, q-1)$ and computes

$$r_p = g^{k_p} \bmod p \quad \text{and} \quad e = h(m||r_p)$$

$$s_p = se + k_p \bmod q$$

The pair (s_p, e) is Schnorr signature. (s_p, e, r) is the proxy signature.

Verification

To verify the proxy signature the verifier, Bob, computes

$$e' = h(m||g^{s_p}(yr^r)^{-e} \bmod p)$$

He accepts the signature if and only if $e' = e$.

MUO SIGNATURE SCHEME

Proof

$$g^{Sp}(yr^{r})^{-e} \bmod p = g^{Sp}(g^{S})^{-e} \bmod p =$$
$$= g^{se + kp} g^{-se} \bmod p = g^{kp} \bmod p = r_p$$

MUO SIGNATURE SCHEME

- Proxy secret key is generated from original signer's private key
- Verifier can be convinced that signature comes from an authorised proxy signer
- Proxy signature is distinguishable from original signer's signature
- Knowledge of original signer's public key is sufficient to verify proxy signature

MUO SIGNATURE SCHEME

- Drawbacks
  - Does not provide non-reputation – both users know the proxy secret key
  - Requires secure channel to transport proxy key
- Remedy
  - Kan Zhang's proxy key generation algorithm

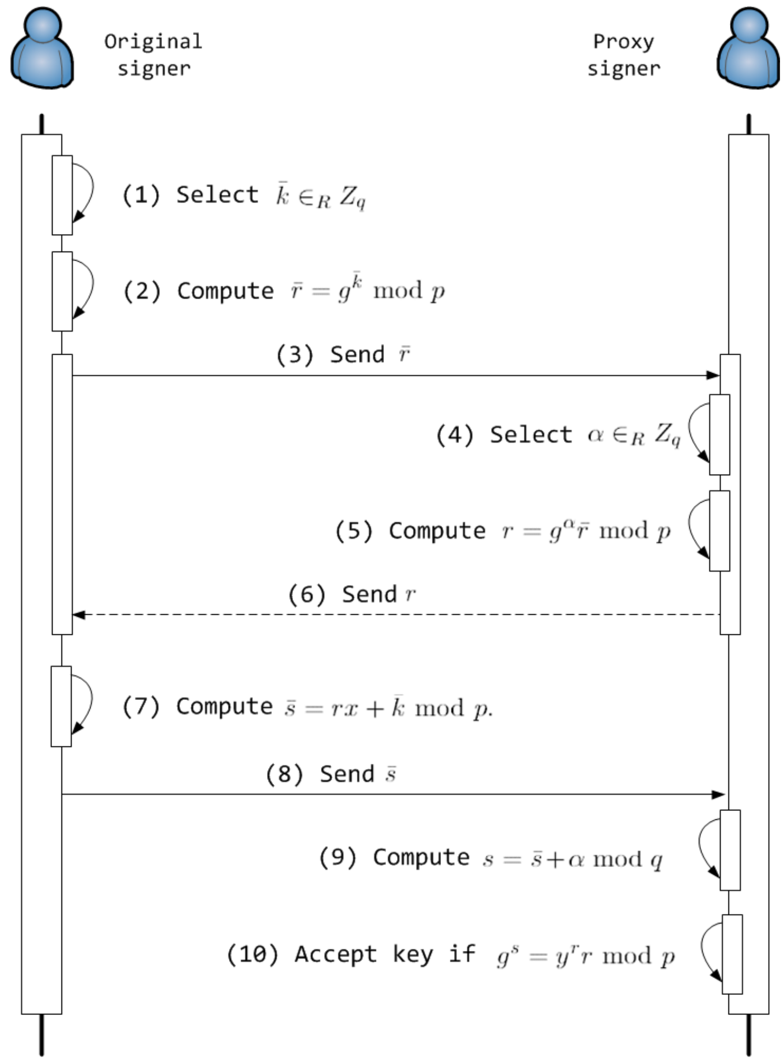
MUO SIGNATURE SCHEME

- Drawbacks
  - Does not provide non-reputation – both users know the proxy secret key
  - Requires secure channel to transport proxy key
- Remedy
  - Kan Zhang's proxy key generation algorithm

ZHANG'S ALGORITHM

1. (Key Generation)-
   (a) The original signer, Alice, selects a $\bar{k} \in_R Z_q$, and computes
   $$\bar{r} = g^{\bar{k}} \bmod p$$
   He sends $\bar{r}$ to the proxy signer through a public channel.
   (b) Proxy signer, Peter, selects $\alpha \in_R Z_q$, and computes
   $$r = g^{\alpha}\bar{r} \bmod p$$
   If $r \in Z_q^*$, he sends r to Alice, else repeats the process.
   (c) Alice computes $\bar{s} = rx + \bar{k} \bmod p$.
2. (Proxy Key Delivery)- Alice sends $\bar{s}$ to the proxy signer, Peter.
3. (Key verification)- After receiving the $\bar{s}$ the Peter modifies the key and obtain a new key $s = \bar{s} + \alpha \bmod q$. He accepts s as a valid proxy secret key iff (s, r) satisfies
   $$g^s = y^r r \bmod p$$

ZHANG'S ALGORITHM



ZHANG'S ALGORITHM

Proof

$$Left = g^{s} \bmod p = g^{\hat{s} + \alpha} \bmod p = g^{rx + \acute{k} + \alpha} \bmod p$$

$$Right = y^{r} r \bmod p = g^{rx} g^{\alpha} \acute{r} \bmod p = g^{rx + \alpha + \acute{k}} \bmod p$$

NEXT SEMINAR

- More proxy signature algorithms
- Digital signature restrictions
- Combination of delegated signature schemes and restrictions

THANK YOU!