Ring Signature

A ring singture allows users to sign oin behalf on a group *without revealing the signer's identity.*

Def. Ring signature consists of $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$:

- $(vk_1, sk_1), \ldots, (vk_n, sk_n) \leftarrow \mathsf{KeyGen}(1^\lambda, n)$;
  - Input: a security parameter λ and a number of ring users n
  - Output: a verification key vk_i and a signing key sk_i for each ring users
- $\sigma \leftarrow \mathsf{Sign}(m, vk_1, \ldots, vk_n, sk_i)$ for some $1 \leq i \leq n$;
  - Input: a message m, all verification keys $vk_1, \ldots, vk_n$ and a signing key sk_i for some user
  - Output: a signature σ of m
- $b \leftarrow \mathsf{Verify}(m, \sigma, vk_1, \ldots, vk_n)$;
  - Input: a message m, a signature σ and all verification keys $vk_1, \ldots, vk_n$
  - Output: a bit $b = 1$ if σ is a valid siganture of m signed by sk_i for $1 \leq i \leq n$

Before describing [RST01] ring signature scheme, I will introduce a combining function which is a main technique of the scheme.

Combining Function

Let E_k be a symmetric encryption with a secret key k.

Let $C_{k,v}(y_1, \ldots, y_r) = E_k(y_r \oplus E_k(y_{n-1} \oplus \cdots \oplus E_k(y_1 + v)) \cdots))$

Then,

- $C_{k,v}$ is a one-to-one mapping from y_s to z for $1 \leq s \leq r$ and fixed $y_i, i \neq s$.
- For $1 \leq s \leq r$ and $y_i, i \neq s$, it is possible to efficiently find y_s such that $C_{k,v}(y_1, \ldots, y_s, \ldots, y_r) = z$
- Given k, z and v, it is hard to solve $C_{k,v}(g_1(x_1), \ldots, g_r(x_r)) = z$ for $x_1, \ldots, x_r$ if g_i's are one-way function.
  - Define $g_i(x) = x^{e_i} \mod n_i$ which is actually an encryption of RSA.
  - One can easily obtain m if he/she has d_i such that $e_i d_i \equiv 1 \bmod n_i$.
  - However, it is hard to obtain m without such d_i.

[RST01] @ Asiacrypt'01

- $\mathsf{KeyGen}(1^\lambda, r)$;
  - Each member executes $\mathsf{RSA.KeyGen}(1^\lambda)$
  - Output $vk_i = \{n_i, e_i\}$ and $sk_i = \{p_i, q_i, d_i\}$ for all i
- $\mathsf{Sign}(m, vk_1, \ldots, vk_r, sk_s)$ for a signer s;
  - Compute $k := H(m)$ where H is a cryptographic hash function

  - Choose $v \leftarrow \{0,1\}^b$ and $x_i \leftarrow \{0,1\}^b$ for $1 \le i \le r, i \ne s$
  - Compute $y_i := g(x_i) = x_i^{e_i} \bmod n_i$
  - Solve the equation $C_{k,v}(y_1, \ldots, y_s, \ldots, y_r) = v$ for y_s
  - Compute $x_s := g_s^{-1}(y_s) = y_s^{d_s} \bmod n_i$
  - Output $\sigma := (v, x_1, \ldots, x_r)$
- $\mathsf{Verify}(m, \sigma, vk_1, \ldots, vk_r)$;
  - Compute $y_i := g_i(x_i)$ for all i
  - Compute $k := H(m)$
  - Compute $\sigma' := C_{k,v}(y_1, \ldots, y_r)$
  - If $\sigma = \sigma'$, output 1.
  - Otherwise, output 0.

Remark

- In Monero (XMR), they use a *linkable* ring signatures
  - anyone can efficiently verify that the signature were generated by *the same* signer without learning who the signer is.

Group Signature

A group signature allows a member of a group to *anonymously sign a message* on behalf of the group.

There is a group manager who is in charge of adding group members and has ability to reveal the original signer.

Def. Group signature consists of $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Open})$:

- $(vk, msk, sk_1, \ldots, sk_n) \leftarrow \mathsf{KeyGen}(1^\lambda, n)$;
  - Input: a security parameter λ and a number of group users n
  - Output: a verification key vk, a master secret key msk, a signing key sk_i for each group users
- $\sigma \leftarrow \mathsf{Sign}(m, sk_i)$ for some $1 \le i \le n$;
  - Input: a message m and a signing key sk_i
  - Output: a signature σ of m
- $b \leftarrow \mathsf{Verify}(m, \sigma, vk)$;
  - Input: a message m, a signature σ and a verification key vk
  - Output: a bit $b = 1$ if σ is a valid signature of m signed by sk_i for $1 \le i \le n$
- $i \leftarrow \mathsf{Open}(m, \sigma, msk)$;
  - Input: a message m, a signature σ and a master secret key msk
  - Output: a user i or $\perp$

A construction of a group signature will be given after dealing with zero-knowledge proof.

Other Signatures

- **Threshold Signature**
- **Multisignature** is a scheme a certain number of signers signs a given message.
  - much shorter than than the set of individual signatures
- **Proxy Signature** allows a delegator to give partial signing rights to other parties called proxy signer.