**Zero-Knolwedge Proof in Group Signature**

proposed by David Chaum (Fourth Group Signature Scheme)

- KeyGen$(1^\lambda, n)$;
  - Choose a prime $p$
  - Choose a generator $g \leftarrow \mathbb{Z}_p^*$
  - Each member $i$ chooses $s_i \leftarrow \mathbb{Z}_{p-1}^*$ and compute $y_i \equiv g^{s_i} \mod p$
  - Output $pk = \{p, g, \{y_i\}\}$ and $sk_i = \{s_i\}$
- Sign$(m, pk, sk_s)$;
  - Computes $\sigma \equiv m^{s_s} \mod p$
  - Output $(m, \sigma)$

Note that $\sigma$ is a valid signature of $m$ if and only if

$$\{\sigma \equiv m^{s_s} \mod p \wedge g^{s_s} \in \{y_i\}\}$$

In order to prove this statement, zero-knowledge proof is used.

A signer has to give a zero-knowledge proof for this statement together with a signature.

## Zero-Knowledge Proof

- allows Provers to convince Verifier that a certain fact is true without giving any information
- involves a number of challenge-response communication rounds between Prover and Verifier
  1. [Announcement] Prover $\Rightarrow$ Verifier
  2. [Challenge] Verifier $\Rightarrow$ Prover
  3. [Response] Prover $\Rightarrow$ Verifier
  4. [Verification]

**Example 1. (Proof of Knowledge for the Square Root)**

Let $n = pq$ be the product of two large primes.

Let $y$ be a square $\mod n$ with $\gcd(y, n) = 1$, i.e., $x^2 \equiv y \bmod n$ for some $x$.

Prover claims *to know a square root $x$ of $y$.*

1. [Announcement];
   - Choose a random $s \leftarrow \mathbb{Z}_n$
   - Compute $a \equiv s^2 \mod n$
   - Sends $a$ to Verifier
2. [Challenge];
   - Choose $c \in \{0, 1\}$

    - Sends $c$ to Prover
3. [Response];
    - If $c = 0$, then $r \equiv s \bmod n$
    - If $c = 1$, then $r \equiv xs \bmod n$
    - Sends $r$ to Verifier
4. [Verification];
    - Compute $r^2 \bmod n$
    - If $c = 0$, check $r^2 \equiv a \bmod n$
    - If $c = 1$, check $r^2 \equiv ya \bmod n$
    - If this is true, then Verifier accpets
    - Otherwise, Verifier rejects

**Remark**

- Finding square root $\bmod n$ is equivalent to factoring $n$ which is a hardness problem of RSA.
- In verification phase, since $r \equiv x^c s \bmod n$,
    - $r^2 \equiv x^{2c} s^2 \equiv y^c a \bmod n$
- If $y$ is not a squre, then only one $a$ or $ay$ is a squre modulo $n$.
    - the probability $p$ that Prover will not be able to answer is 50%
    - repeat $k$ times, then $p = \frac{1}{2^k}$

**Example 2. (Proof of Knowledge for Discrete Logarithm)**

Let $p$ be a DL-secure prime and $g \leftarrow \mathbb{Z}_p^*$.

Let $y = g^x \bmod p$.

Prover claims _to know a discrete logarithm of $y$_, i.e., $x = \log_g y$.

1. [Announcement]
    - Choose $s \leftarrow \mathbb{Z}_p^*$
    - Compute $a = g^s \bmod p$
    - Send $a$ to Verifer
2. [Challenge]
    - Choose $c \leftarrow \mathbb{Z}_p^*$
    - Send $c$ to Prover
3. [Response]
    - Compute $r = s + cx \bmod p - 1$
    - Send $r$ to Verifier
4. [Verification]
    - If $g^r = a \cdot y^c \bmod p$, then Verifier accepts
    - Otherwise, Verifier rejects

**Example 3. (Proof of Equality of Discrete Logarithm over Different Groups)**

Let $p$ be a DL-secure prime and $g_1, g_2 \leftarrow \mathbb{Z}_p^*$.

Let $y_1 = g_1^x \bmod p$ and $y_2 = g_2^x \bmod p$.

Prover claims that *two discrete logarithm over different groups are the same.*

1. [Announcement]
    - Choose $s \leftarrow \mathbb{Z}_p^*$
    - Computet $a_1 = g_1^s \bmod p$ and $a_2 = g_2^s \bmod p$
    - Send $a_1$ and $a_2$ to Verifier
2. [Challenge]
    - Choose $c \leftarrow \mathbb{Z}_p^*$
    - Send $c$ to Prover
3. [Response]
    - Compute $r = s + cx \bmod p - 1$
    - Send $r$ to Verifier
4. [Verification]
    - If $g_1^r = a_1 \cdot y_1^c$ and $g_2^r = a_2 \cdot y_2^c$, then Verifier accepts
    - Otherwise, Verifier rejects