Linkable Ring Signature Algorithm

Reference: "Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups" by Liu , Wei et al. [section 4.1 and 4.2]

June 25, 2012

1 Signature Generation

Let $G = \langle g \rangle$ be a group of large prime order (preferably $safeprime$) q. Let $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q$ and $H_2 : \{0,1\}^* \rightarrow G$ be independent hash functions. Each user i $\{i = 1, ..., n\}$ has a distinct public key y_i and a private key x_i such that $y_i = g^{x_i}$. Let L = $\{y_1, ..y_n\}$ be the list of n public keys.

Input: $m\epsilon\{0,1\}^*$, list of public keys L = $\{y_1, ..y_n\}$, private key x_π corresponding to y_π $1 \leq \pi \leq n$.

1: Compute $h = H_2(L)$ and $\tilde{y} = h^{x_\pi}$.
2: Pick $u\epsilon_R\mathbb{Z}_q$ and compute $c_{i+1} = H_1(L, \tilde{y}, m, g^u, h^u)$.
3: For $i = \pi + 1, ...n, 1, ..., \pi - 1$ pick $s_i\epsilon_R\mathbb{Z}_q$ and compute $c_{i+1} = H_1(L, \tilde{y}, m, g^{s_i}y_i^{c_i}, h^{s_i}\tilde{y}^{c_i})$
4: Compute $s_\pi = u - x_\pi c_\pi modq$

The signature is $\sigma_L(m) = (c_1, s_1, ..., s_n, \tilde{y})$.

2 Signature Verification

Input: $\sigma_L(m) = (c_1, s_1, ..., s_n, \tilde{y})$ and list of public keys L = $\{y_1, ..y_n\}$.

1: Compute $h = H_2(L)$ and for $i = 1, ..., n$, compute $z_i' = g^{s_i}y_i^{c_i}$ and $z_i'' = h^{s_i}\tilde{y}^{c_i}$ and then $c_{i+1} = H_1(L, \tilde{y}, m, z_i', z_i'')$ if $i \neq n$.
2: Check $c_1 \stackrel{?}{=} H_1(L, \tilde{y}, m, z_n', z_n'')$. If yes, accept signature else reject.

3 Linkability Testing

Given: Two signatures $\sigma_L'(m') = (c_1', s_1', ..., s_n', \tilde{y}')$ and $\sigma_L''(m'') = (c_1'', s_1'', ..., s_n'', \tilde{y}'')$ corresponding to messages m' and m''.

1: Verify signatures : $\sigma_L'(m')$ and $\sigma L''(m"")$
2: Check if $\tilde{y}' = \tilde{y}''$. If so, signatures are created by the same signer. Otherwise, signatures are generated by different signers.