Threshold Ring Signatures in the Random Oracle Model (Extended Abstract)

Toshiyuki Isshiki Keisuke Tanaka*

Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology,
2-12-1 Ookayama, Meguro-ku, Tokyo 152-8552, Japan.
{isshiki9, keisuke}@is.titech.ac.jp.

Abstract

We improve on the Bresson–Stern–Szydlo threshold ring signature scheme which uses Shamir secret sharing scheme [6] by showing that the security can be proved under a strictly weaker assumption, that is the random oracle model rather than the ideal cipher model. Then we propose an efficient $(n-t)$-out-of-n threshold ring signature scheme which is efficient when t is small compared with n. Our scheme has a kind of dual structure of the Bresson–Stern–Szydlo threshold ring signature scheme [2], which is infeasible when t is small compared with n. In particular, we modify the trap-door one-way permutations in the ring signature scheme, and use a combinatorial notion called *fair partition*. Our scheme is provably secure in the random oracle model.

keywords

threshold ring signature, ideal cipher model, random oracle model

1 Introduction

Anonymity is required to ensure that information about the user is not revealed in some multi-user cryptographic applications. The notion of group signature was introduced by Chaum and van Heijst [3], allows a registered member of a predefined group to produce anonymous signatures on behalf of the group. However, this anonymity can be revoked by an authority if necessary. The distinct but related concept of ring signature has been formalized by Rivest, Shamir, and Tauman [5]. This concept is of particular interest when the members do not agree to cooperate since the scheme requires neither a group manager, nor a setup procedure, nor the action of a non-signing member.

A ring signature specifies a set of possible signers and a proof that is intended to convince any verifier that the author of the signature belongs to this set, while hiding her identity. The scheme is said to be signer ambiguous in the sense that the verifier cannot tell which user in this set actually produces the signature.

Assume that in order to create a certain signature at least t out of the n parties need to combine their knowledge. Combining the shares must not reveal the actual secret key. The correctness of the signature would be verifiable using the public keys. Any t out of the n parties can perform some cryptographic operation jointly, whereas it is infeasible for at most $t-1$ parties to do so.

Recently, Bresson, Stern, and Szydlo [2] and Kuwakado and Tanaka [4] independently proposed similar schemes which use Shamir secret sharing scheme [6] for threshold ring signature, which is

*Supported in part by NTT Information Sharing Platform Laboratories and Grant-in-Aid for Scientific Research, Ministry of Education, Culture, Sports, Science, and Technology, 14780190.

provably secure in the ideal cipher model. While the original scheme that proposed by Rivest, Shamir, and Tauman [5] geometrically makes a ring of individual signatures, both schemes make a curve of them.

In this paper, we improve on the Bresson–Stern–Szydlo scheme by showing that it holds under a strictly weaker assumption, that is the random oracle model rather than the ideal cipher model.

Bresson, Stern, and Szydlo [2] also proposed an efficient scheme for threshold ring signature, which is provably secure in the random oracle model. In particular, their construction is very efficient when threshold t is small. However, it is very inefficient when t is $\omega(\log n)$, and is infeasible when $t > n/2$ since the anonymity property does not hold in this case. In their scheme, there is the solution that the verifier adds the dummy members to the n ring members in a setup procedure in order to $t < n'/2$, however, this solution losses the property of ring signature that has no setup procedure.

Consider that a majority of members in some section of a company wishes to claim something for a director of the company. The previous scheme does not work for this simple case.

In this paper, we also propose a solution for this case, i.e., a threshold ring signature scheme which is efficient when threshold t is large compared with n, i.e. $t = n - k$ (k is small compared with n). Our scheme has a kind of dual structure of the Bresson–Stern–Szydlo threshold ring signature scheme. They used a structure of so-called *super-ring*, which has standard 1-out-of-n ring signatures as nodes. In our scheme, we use a set of ring signatures as a $(n-t)$-out-of-n signature. We still employ a simple structure of ring (not super-ring), and modify the trap-door one-way permutations for it.

2 Preliminaries

In this paper, we follow the formalization proposed by Rivest, Shamir, and Tauman [5]. They proposed the notion of ring signature, which allows a member of an ad-hoc collection of users S to prove that a message is authenticated by a member of S without revealing which member actually produced the signature.

We assume that each user has received a public key PK_k, for which the corresponding secret key is denoted by SK_k. A ring signature scheme consists of the following algorithms.

- **Ring-sign**: A probabilistic algorithm outputs a ring signature σ for the message m, with input a message m, the public keys $PK_1, \ldots, PK_r$ of the r ring members, together with the secret key SK_s of a signer.
- **Ring-verify**: A deterministic algorithm outputs either "`ACCEPT`" or "`REJECT`" with input (m, σ) (where σ includes the public key of all the possible signers).

A ring signature scheme must satisfy the correctness (i.e. a correct ring signature should be accepted as valid with overwhelming probability) and unforgeability (i.e. it must be infeasible for any non-ring member to generate a valid ring signature, except with negligible probability). We also require anonymity that nobody should be able to guess the actual signer's identity with probability greater than $1/n + \epsilon$, where n is the number of the ring members, and ϵ is negligible.

2.1 Ring Signature Schemes by Bresson, Stern, and Szydlo [2]

Bresson, Stern, and Szydlo proposed a modification of the original Rivest–Shamir–Tauman ring signature scheme. In this section, we briefly review this modification proposed by Bresson, Stern, and Szydlo [2], based on the random oracle model, while the original Rivest–Shamir–Tauman scheme uses the ideal cipher model.

We denote by ℓ, ℓ_b, ℓ_0 three security parameters. We consider a hash function $\mathcal{H}$ that maps arbitrary strings on ℓ_b-bit strings. We assume that each user P_i uses a regular signature scheme built on a trapdoor one-way permutation f_i on $\mathbb{Z}_{n_i}^* : f_i(x) = x^{e_i} \bmod n_i$ where $|n_i| = \ell_b < \ell$.

$$g_i(x) = \begin{cases} q_i n_i + f_i(r_i) & \text{if } (q_i+1)n_i \leq 2^\ell \\ x & \text{otherwise} \end{cases} \tag{1}$$

where $x = q_i n_i + r_i$, and $0 \leq r_i < n_i$.

Generating a ring signature

Given the message m to be signed, her secret key SK_s, and the sequence of public keys $PK_1, PK_2, \ldots, PK_r$ of all the ring members, the signer computes a ring signature as follows.

1. **Choose a random seed:** The signer picks a random seed σ in $\{0,1\}^{\ell_b}$,and computes
$$v_{s+1} = \mathcal{H}(m, \sigma).$$
2. **Pick random x_i's:** The signer picks random x_i for all the other ring members $1 \leq i \leq r$, $i \neq s$ uniformly and independently from $\{0,1\}^{\ell_b}$, and computes for $i = s+1, s+2, \ldots, n, 1, 2, \ldots, s-1$,
$$v_{i+1} = \mathcal{H}(m, v_i \oplus g_i(x_i)).$$
where
$$g_i(x) = \begin{cases} q_i n_i + f_i(r_i) & \text{if } (q_i+1)n_i \leq 2^\ell \\ x & \text{otherwise} \end{cases}$$
with $x = q_i n_i + r_i$ and $0 \leq r_i < n_i$.
3. **Solve for x_s:** The signer solves the following equation for x_s by using her knowledge of trap-door permutation:
$$\sigma = v_s \oplus g_s(x_s).$$
4. **Output the signature:** The signer chooses at random an index $i_0 \in \{1, 2, \ldots, r\}$, then the signature on the message m is defined as the $(2r+2)$-tuple:
$$(PK_1, PK_2, \ldots, PK_r; i_0; v_{i_0}; x_1, x_2, \ldots, x_r).$$

Verifying a ring signature

A verifier can verify an alleged signature
$$(PK_1, PK_2, \ldots, PK_r; i_0; v_{i_0}; x_1, x_2, \ldots, x_r)$$
on the message m as follows.

1. **Apply the trapdoor permutations:** For $i = i_0+1, i_0+2, \ldots, n, 1, 2, \ldots, i_0-1$, the verifier computes
$$v_i = \mathcal{H}(m, v_{i-1} \oplus g_{i-1}(x_{i-1})).$$
2. **Verify the equation:** The verifier checks that the v_i's satisfy the equation:
$$v_{i_0} = \mathcal{H}(m, v_{i_0-1} \oplus g_{i_0-1}(x_{i_0-1})).$$
If this equation is satisfied, the verifier outputs "`ACCEPT`", otherwise "`REJECT`".

2.2 Formulation of Threshold Ring Signature

In [2], Bresson, Stern, and Szydlo introduced the definition and the security requirements for threshold ring signature. Here, we briefly review them:

A t-out-of-n threshold ring signature scheme consists of the following algorithms:

- **T-ring-sign:** A probabilistic algorithm outputs a t-out-of-n threshold ring signature σ on the message m (where σ includes the value of t as well as the n public keys of all ring members), with input a message m, the public keys $PK_1, \ldots, PK_n$ of the n ring members, together with the t secret keys $\mathrm{SK}_{i_1}, \ldots, SK_{i_t}$ of t signers.
- **T-ring-verify:** A deterministic algorithm outputs either "`ACCEPT`" or "`REJECT`" with input (m, σ).

The adversary $\mathcal{A}$ is given the public keys $PK_1, \ldots, PK_n$ of the n ring members, and can access to the hash function $\mathcal{H}$. Also, $\mathcal{A}$ is given access to a signing oracle. We define that t-forger against a threshold ring signature is a probabilistic polynomial-time Turing machine $\mathcal{A}$, that can sign a message on behalf of t users, with up to $t-1$ corrupted users, under the adaptive chosen message attack.

Definition 1 *We say a t-out-of-n threshold ring signature scheme is t-CMA-secure if no t-forger $\mathcal{A}$ can succeed to forge a signature with non-negligible probability.*

We require the signature to have anonymity (i.e. nobody should be able to guess the actual signer's identity) and unforgeability (i.e. the scheme is t-CMA-secure).

2.3 The Previous Threshold Ring Signature Schemes

2.3.1 The Scheme using Secret Sharing by Bresson, Stern, and Szydlo [2]

In this paper, we improve on the Bresson–Stern–Szydlo threshold ring signature scheme which uses Shamir secret sharing [6] by showing that it holds under a strictly weaker assumption, that is the random oracle model rather than the ideal cipher model. Here, we briefly review the Bresson–Stern–Szydlo threshold ring signature scheme using secret sharing. Their idea is to use Shamir secret sharing scheme [6] to perform a threshold proof. In such a proof, the "challenge" is shared in order to prove knowledge of a minimum number of secrets. The challenge to share depends on the group on behalf of which the signature is produced.

Let m be a message, and t be the number of sign-members. For simplicity, we index the sign-members with numbers $1, \ldots, t$. We denote $P_1, \ldots, P_n$ the public keys of all ring members. Here, we assume that the existence of public collision-resistant hash functions $\mathcal{H}$ which is mapping $\{0,1\}^*$ to $\{0,1\}^\ell$ and computed by random oracle. We consider an encryption scheme E using ℓ_0-bit length keys as well as an additional parameter $i \in [1, n]$. We prefer to use the notation $E_{k,i}(\cdot)$.

Signing algorithm.

The signature algorithm performs the following steps:

- **Compute the symmetric key for E:** The signer computes $k = \mathcal{H}(m)$.
- **Compute value at origin:** The signer computes $v = \mathcal{H}(P_1, \ldots, P_n)$.
- **Choose random seeds:** For each $i = t+1, \ldots, n$, the signer chooses $x_i \in \{0,1\}^\ell$ and sets $y_i = g_i(x_i)$.
- **Compute a sharing polynomial:** The signer computes a polynomial f over $GF(2^\ell)$ s.t. $\deg(f) = n - t$, $f(0) = v$ and for each $i = 1, \ldots, n : f(i) = E_{k,i}(y_i)$.
- **Solve the remaining equations:** For each $i = 1, \ldots, t$, the signer computes

$$x_i = g^{-1}(E_{k,i}^{-1}(f(i))).$$

- **Output the signature:**

$$(m, P_1, \ldots, P_n, v, x_1, \ldots, x_n, f).$$

Verification algorithm.

On receiving a tuple $(m, P_1, \ldots, P_n, v, x_1, \ldots, x_n, f)$, the verifying algorithm performs the following steps:

- **Recover the symmetric key:** The verifier computes $k = \mathcal{H}(m)$.
- **Recover y_i's:** For each $i = 1, \ldots, n$, the verifier computes $y_i = g_i(x_i)$.
- **Verify the equations:** The verifier computes $f(0) = \mathcal{H}(P_1, \ldots, P_n)$ and for each $i = 1, \ldots, n$, checks the equations:

$$f(i) = E_{k,i}(y_i).$$

If the signature is correct, the verifier accepts it as a t-out-of-n signature, where $t = n - \deg(f)$.

2.3.2 The Scheme using Fair Partitions by Bresson, Stern, and Szydlo [2]

In this paper, we propose an $(n-t)$-out-of-n ring signature scheme, where t is small compared with n. In our scheme, we use a kind of dual structure of the Bresson–Stern–Szydlo threshold ring signature scheme, and employ a combinatorial notion called *fair partition* that is used in the Bresson–Stern–Szydlo threshold ring signature scheme. Here, we briefly review its definition and (n,t)-complete partitioning systems introduced in [2] (see also [1]).

Let $\pi = (\pi^1, \cdots, \pi^t)$ a partition of $[1,n]$ in t subsets and $I = \{i_1, \cdots, i_t\}$ a set of t indices in $[1,n]$. If all integers in I belongs to t different subsets, we say that π is a *fair partition for* I.

Definition 2 *Let* $\pi = (\pi^1, \cdots, \pi^t)$ *a partition of* $[1,n]$ *in* t *subsets and* $I = \{i_1, \cdots, i_t\}$ *a set of* t *indices in* $[1,n]$*. We say that* π *is a* ***fair partition*** *for* I *if for all* $j \in [1,t]$*,*

$$\#(I \cap \pi^j) = 1.$$

Here, $\#(A)$ *denotes the number of elements of* A*.*

To ensure anonymity, we need to provide a set Π of partitions such that there exists a fair partition for any set I of t indices in $[1,n]$.

Definition 3 *Let* $t < n$*. We say that a set of* Π *of partitions of* $[1,n]$ *is an* (n,t)***-complete partitioning system*** *if for any set* I *of cardinality* t*, there exists a fair partition in* Π *for* I*.*

A perfect hash function for a set I is a mapping $h : [1,n] \to [1,t]$ which is 1-1 on I. We say H is an (n,t)-family of perfect hash functions if for any I of size t, there exists $h \in H$ which is perfect on I. It is clear that defining a partition in t sub-groups for each member of an (n,t)-family makes an (n,t)-complete partitioning system. In [1], Alon, Yuster, and Zwick has been proved that there exists an (n,t)-family of perfect hash function which has size of $2^{O(t)} \log n$. Moreover each of these functions is efficiently computable.

Here, we briefly describe the idea of the threshold ring signature scheme proposed by Bresson, Stern, and Szydlo [2]. Consider a ring of n members, and among them t users who want to sign for a message. Let $I = \{i_1, \ldots, i_t\}$ a set of t indices in $[1,n]$ such that $P_{i_1}, \ldots, P_{i_t}$ are signers. The idea is to split the group into t disjoint sub-groups regard to a fair partition for I, and to show that each of these sub-groups contains one signer by producing sub-rings. However doing so may compromise perfect anonymity since such split restricts the anonymity of each user to a sub-ring. To ensure anonymity, their scheme needs to split the group regard to an (n,t)-complete partitioning system for which any t users are in different sub-rings. Then all of these splits are used as nodes in a super-ring. The super-ring proves that at least one split has been solved.

The size of the signature in this scheme is $\mathcal{O}(\ell 2^t n \log n)$, the cost is t inversions of the signers' one-way functions and $\mathcal{O}(2^t n \log n)$ computations in the easy direction.

It should be pointed out that when $t > n/2$, there exist some partitions which consist only one element in a fair partition for I. Therefore, this scheme cannot be used for such t since the anonymity property does not hold.

3 Our scheme using Secret Sharing

In this section, we explain how to significantly improve the scheme using secret sharing by Bresson, Stern and Szydlo [2] by removing the assumption of an ideal-cipher. Here we use the random permutation oracle over $\{0,1\}^\ell$ which assumes that all the parties have access to oracles that provides truly random answers to new queries for E, E^{-1}, F, and F^{-1}. Here, we assume that the existence of public collision-resistant hash functions $\mathcal{H}$ and $\mathcal{H}'$ where $\mathcal{H}$ which computed by random oracle is mapping $\{0,1\}^*$ to $\{0,1\}^\ell$ and $\mathcal{H}'$ is mapping $\{0,1\}^*$ to $\{0,1\}^\ell$.

Signing algorithm.

The signature algorithm performs the following steps:

- **Compute value at origin:** The signer computes

$$k = \mathcal{H}(m), \text{ and}$$
$$v = \mathcal{H}'(P_1, \ldots, P_n).$$

- **Choose random seeds:** For each $i = t+1, \ldots, n$, the signer chooses $x_i \in \{0,1\}^\ell$ and sets $y_i = g_i(x_i)$.
- **Compute a sharing polynomial:** The signer computes a polynomial f over $GF(2^\ell)$ s.t. $\deg(f) = n-t$, $f(0) = v$ and for each $i = 1, \ldots, n : f(i) = F(E(y_i) \oplus k)$.
- **Solve the remaining equations:** For each $i = 1, \ldots, t$, the signer computes

$$x_i = g^{-1}(E^{-1}(F^{-1}(f(i)) \oplus k)).$$

- **Output the signature:**

$$(m, P_1, \ldots, P_n, v, x_1, \ldots, x_n, f).$$

Verification algorithm.

On receiving a tuple $(m, P_1, \ldots, P_n, v, x_1, \ldots, x_n, f)$, the verifying algorithm performs the following steps:

- **Recover value at origin:** The verifier computes $k = \mathcal{H}(m)$.
- **Recover y_i's:** For each $i = 1, \ldots, n$, the verifier computes $y_i = g_i(x_i)$.
- **Verify the equations:** The verifier computes $f(0) = \mathcal{H}(P_1, \ldots, P_n)$ and for each $i = 1, \ldots, n$, checks the equations:

$$f(i) = F(E(y_i) \oplus k).$$

  If the signature is correct, the verifier accepts it as a t-out-of-n signature, where $t = n - \deg(f)$.

3.1 Security Analysis

We prove that the above scheme has the required property of threshold ring signature in random oracle model. The proofs are in the full version of this paper.

3.2 Efficiency

We discuss the efficiency of our scheme. Let n to be the number of members and t to be the number of sign-members. The size of threshold ring signature is $(2n - t + 2) \times \ell$-bit. Here, the public key P_i is ignored because it is public. The time complexity of signing is t inversions of the g's functions, $n - t$ computations in the easy direction, and n polynomial evaluations. Verifying such a signature requires n computations of g's and n polynomial evaluations.

4 Our Scheme using Fair Partitions

In this section, we propose an efficient $(n-t)$-out-of-n threshold ring signature scheme where the number of non-signer t is small compared with the number of ring members n. Let $\Pi_n^{t+1} = \{\pi_1, \ldots, \pi_p\}$ ($p = 2^{t+1} \log n$) to be an $(n, t+1)$-complete partitioning system.

We describe formally our $(n-t)$-out-of-n ring signature scheme where t is small.

We denote by ℓ a security parameter. We denote an $(n, t+1)$-complete partitioning system $\Pi_n^{t+1} = \{\pi_1, \ldots, \pi_p\}$ where $p = 2^{t+1} \log n$, and each partition $\pi_i = (\pi_i^1, \ldots, \pi_i^{t+1})$, where each π_i^j is a set of indices. Let $\{P_1, \ldots, P_n\}$ be a set of n ring members for a message m.

For each i, j, we denote by q_i^j the number of elements of π_i^j, and by Q the maximum number of q_i^j. Let $\pi_i^j = \{p_i^{j,1}, \ldots, p_i^{j,q_i^j}\}$.

We assume that for all integer n and $t \leq n$, an $(n, t+1)$-complete partitioning system is publicly available, and that each user P_i uses a regular signature scheme built on a trapdoor one-way permutation g_i over $\{0,1\}^\ell$. We say π_i^j is *legal* if for all $k \in \pi_i^j$, P_k is a signer.

We consider that a hash function $\mathcal{H}$ that maps $(Q \times \ell)$-bit strings. For each partition π_i^j, we define a trapdoor one-way permutation G_i^j:

If $q_i^j = Q$, then let $S_i^j = \pi_i^j$, else let $S_i^j = \{\pi_i^j \cup \{p_i^{j,q_i^j+1}, p_i^{j,q_i^j+2}, \ldots, p_i^{j,Q}\}\}$, where $p_i^{j,q_i^j+1} = p_i^{j,q_i^j+2} = \ldots = p_i^{j,Q} = p_i^{j,q_i^j}$.

$$G_i^j(x_1, \ldots, x_Q) = g_{p_i^{j,1}}(x_1)|| \cdots ||g_{p_i^{j,Q}}(x_Q).$$

Thus, each G_i^j has a Q-tuple of ℓ-bit strings as input and outputs a $(Q \times \ell)$-bit string, since each $g_{p_i^{j,k}}$ $(k = 1, 2, \ldots, Q)$ is a trapdoor one-way permutation of $P_{p_i^{j,k}}$ over $\{0,1\}^\ell$. The trapdoor of G_i^j is a set of all $g_{p_i^{j,k}}$'s trapdoors. It is clearly that G_i^j is a trapdoor one-way permutation since if one can invert G_i^j, then he do invert all $g_i^{j,k}$'s. For example, we assume that each $g_{p_i^{j,k}}$ is an extended RSA permutation (1) in Section 2.1, and let $(n_{p_i^{j,k}}, e_{p_i^{j,k}})$ to be the public key of $P_{p_i^{j,k}}$ and $d_{p_i^{j,k}}$ to be the secret key of $P_{p_i^{j,k}}$. Then, the trapdoor of G_i^j is $(d_{p_i^{j,1}}, \ldots, d_{p_i^{j,Q}})$.

Signing algorithm.

The signer executes the following steps for each π_i $(i = 1, 2, \ldots, p)$. Here, we assume that S_i^j is legal.

- **Choose random seeds:** The signer chooses random seeds $s^1, \ldots, s^Q \in \{0,1\}^\ell$ randomly and computes
$$v_{j+1} = \mathcal{H}(m, s^1, \ldots, s^Q).$$
- **Pick random x's:** For each $k = j+1, \ldots, t+1, 1, 2, \ldots, j-1$, the signer chooses $x_k^1, \ldots, x_k^Q \in \{0,1\}^\ell$ at random, and computes
$$v_{k+1} = \mathcal{H}(m, v_k \oplus G_i^k(x_k^1, \ldots, x_k^Q)).$$
- **Invert the legal S_i^j's trapdoor permutation:** The signer uses his knowledge of trapdoors of each $g_{p_i^{j,k}}$ in order to invert G_i^j to obtain $x_j^1, \ldots, x_j^Q$ such that $v_{j+1} = \mathcal{H}(m, v_j \oplus G_i^j(x_j^1, \ldots, x_j^Q))$.
- **Output the signature for π_i:** The signer chooses at random an index $i_0 \in \{1, 2, \ldots, t+1\}$, then the signature on the message m for π_i is defined as the $(2(t+1)Q+2)$-tuple:
$$\sigma_i = (PK_{p_i}^{1,1}, \ldots, PK_{p_i^{1,Q}}, PK_{p_i^{2,1}}, \ldots, PK_{p_i^{t+1,Q}};$$
$$i_0; v_{i_0}; x_1^1, \ldots, x_1^Q, x_2^1, \ldots, x_{t+1}^Q).$$

Thus, the signature on the message m is defined to be the p-tuple:

$$\sigma = (\sigma_1, \ldots, \sigma_p).$$

Verifying algorithm.

The verifier can verify each σ_i $(i = 1, 2, \ldots, p)$ on the message m as follows.

- **Apply the trapdoor permutations:** For $k = i_0+1, i_0+2, \ldots, t+1, 1, 2, \ldots, i_0-1$, the verifier computes
$$v_k = \mathcal{H}(m, v_{k-1} \oplus G_i^j(x_k^1, \ldots, x_k^Q)),$$
and checks the equation:
$$v_{i_0} = \mathcal{H}(m, v_{i_0-1} \oplus G_i^j(x_{i_0-1}^1, \ldots, x_{i_0-1}^Q)).$$
If this equation is satisfied, then σ_i is "`TRUE`", otherwise "`FALSE`".

If for all $k = 1, 2, \ldots, p$, then σ_i is `TRUE`, then the verifier outputs "`ACCEPT`", otherwise "`REJECT`".

4.1 Security Analysis

We prove that the above scheme has the required property of threshold ring signature. The proofs are in the full version of this paper.

4.2 Efficiency

We discuss the efficiency of our scheme. The size of an $(n-t)$-out-of-n threshold ring signature is $2^t \log n \times \{2\,((t+1) \times Q \times \ell) + \ell \times Q\} = \mathcal{O}(\ell 2^t n \log n)$. The time complexity of singing is $Q \times 2^t \log n$ inversions of the g's functions and $Q \times (t+1) \times 2^t \log n = \mathcal{O}(2^t n \log n)$ computations in the easy direction. Our scheme is clearly more efficient than generic solution such that making ring signatures for all subgroups cardinality $n-t+1$ since this would lead to $\binom{n}{t-1} = \mathcal{O}(n^{t-1})$ size.

References

[1] Alon, N., Yuster, R., and Zwick, U. Color-coding. *Electronic Colloquium on Computational Complexity (ECCC) 1*, 009 (1994). Full paper appears in J.ACM 42:4, July 1995, 844-856.

[2] Bresson, E., Stern, J., and Szydlo, M. Threshold Ring Signatures and Applications to Ad-hoc Groups. In *Advances in Cryptology – CRYPTO 2002* (Santa Barbara, California, USA, August 2002), M. Yung, Ed., vol. 2442 of *Lecture Notes in Conputer Science*, Springer-Verlag, pp. 465–480.

[3] Chaum, D., and van Heijst, E. Group Signatures. In *Advances in Cryptology – EUROCRYPT '91* (Brighton, UK, April 1991), D. Davies, Ed., vol. 547 of *Lecture Notes in Conputer Science*, Springer-Verlag, pp. 257–265.

[4] Kuwakado, H., and Tanaka, H. Threshold Ring Signature Scheme Based on the Curve. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences E86–A*, 10 (2003), 2146–2154.

[5] Rivest, R. L., Shamir, A., and Tauman, Y. How to Leak A Secret. In *Advances in Cryptology – ASIACRYPT 2001* (Gold Coast, Australia, December 2001), C. Boyd, Ed., vol. 2248 of *Lecture Notes in Conputer Science*, Springer-Verlag, pp. 552–565.

[6] Shamir, A. How to share a secret. *Commun. ACM 22*, 11 (1979), 612–613.