社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2005－ＣＳＥＣ－30 (26)
2005／7／21

署名者数確認機能付きしきい値署名

藤崎英一郎† 鈴木幸太郎†

† 日本電信電話株式会社 〒239-0847 神奈川県横須賀市光の丘 1-1
E-mail: †{fujisaki.eiichiro,suzuki.koutarou}@lab.ntt.co.jp

あらまし 通常、t-out-of-n 分散署名では t 人以上の署名者が協力した時、正しい署名を作成することができる。本論文は、署名者の下限のみではなく、上限も指定できる分散署名方式（範囲署名と呼ぶことにする）を提案する。特に上限と下限が一致する場合、署名者数を確認できる署名方式となる。このような性質を目標にした署名方式は、文献[18], [19] で提案されている。ただし、これらの文献では、署名者数に関する定義がなされていない。本論文では、分散署名における真の署名者とはなにかを定義したうえで、範囲署名を定義し、さらに構成法を提案する。範囲署名の定式化、構成法は、文献[15] で我々が提案したものと異なっている。

キーワード 範囲署名、リング署名、否認不可署名、グループ署名。

Exact t-out-of-n Signer-Ambiguous Signature

Eiichiro FUJISAKI† and Koutarou SUZUKI†

† NTT Laboratories, 1-1 Hikarinooka Yokosuka-shi Kanagawa 239-0847 JAPAN
E-mail: †{fujisaki.eiichiro,suzuki.koutarou}@lab.ntt.co.jp

Abstract We propose a novel threshold ring signature scheme, where the verifier can be convinced that the signatures, as ordinary threshold (ring) signatures, were produced by the collaboration of at least t anonymous signers (among n possible signers), while he can be also convinced that the number of the collaborators was at most t' ($t \leq t'$). We formalize the notion of this threshold signature scheme, denoted a (t, t', n)-*ranged* threshold signature scheme. In case $t = t'$, we call it an exact t-out-of-n threshold signature scheme. Our scheme can be constructed based only on an ordinary discrete-logarithm setting. No stronger primitive such as pairing cryptosystems are necessary.

Key words threshold ranged signature, ring signature, undeniable signature, and group signature.

1. Introduction

A t-out-of-n threshold signature scheme is a protocol that allows the creation of valid signatures if and only if more than or equal to t (out of n) players participate in the protocol. In a threshold signature scheme, however, the verifier cannot be convinced of how many players exactly participated in the signature procedure, i.e., how many members exactly approved the document [1]. In some applications, it should be desirable that the signers can tell the verifier the exact number of them, while preserving anonymity among all possible signers.

To realize this property in a threshold signature scheme, however, we must settle an undefined issue. How to define the "real" signers? In a classic definition of the threshold signature scheme, it is not formalized.

We consider this problem in the setting of the ring signature — all the potential signers have their own public-keys for signatures. Hence, our proposal is categorized in a threshold version of ring signature, called a threshold ring signature scheme, in which the signers can choose an "arbitrary" set of possible signers, including themselves, to produce a threshold signature where the actual signers must cooperate each other but they don't need any information of the other members except for their public-keys.

We then introduce a language of partial signatures, say L (in $\mathcal{NP}$), where L is a language of a tuple of a set of

1) In the trivial threshold signature scheme, such as t signers produce normal signatures on the same message with respect to their own public-keys, it is obvious how many signers join the signing procedure, but we exclude such a "non-signer-ambiguous" construction from our target.

public-keys of all possible players, $pk_N = (pk_1, \ldots, pk_n)$, where $N = \{1, \ldots, n\}$, message m and signature σ_i on m with respect to public-key pk_i in pk_N, i.e.,

$$L = \{(pk_N, m, \sigma_i) | \sigma_i \text{ is a signature on } m, \text{ w.r.t. } pk_i \in pk_N.\}$$

where the witness is a pair of the corresponding sk_i and random coins r such that $\sigma_i = \mathrm{Sig}_{sk_i}(m; r)$.

Suppose that σ is the threshold ring signature on message m, among participants N. We let σ include $\sigma_N := (\sigma_1, \ldots, \sigma_n)$, in which, there are two types of partial signatures: For some $i \in N$, $(pk_N, m, \sigma_i) \in L$, whereas for some $i' \in N$, $(pk_N, m, \sigma_{i'}) \notin L$. Naturally, we define a participant $i \in N$ to be a real signer if $(pk_N, m, \sigma_i) \in L$. The decision problem on L must be hard to allow the protocol to remain "signer-ambiguous" – For instance, if L is a language for an undeniable signature scheme, the decision problem on L is hard.

Another new concept in this paper is ranged threshold signature. A (t, t', n)-ranged threshold signature scheme, $0 \leqq t \leqq t' \leqq n$, is a threshold ring signature scheme that can convince the verifier that the number of the collaboration of the real signers is at least t and at most t', whereas the verifier has no information about which of n possible signers participated in the protocol. In addition, the k signers can dynamically choose lower-bound t and upper-bound t' in range $0 \leqq t \leqq k \leqq t' \leqq n$, where n denotes the number of the possible signers. When $t = t'$, it is obvious that the verifier is convinced that exact t (out of n) anonymous signers approved the document and the signature is called an exact t-out-of n signature.

1.1 Our Results

We introduce a model and security definitions of the ranged threshold ring signature, which is signer-ambiguous, and makes it possible to specify both the lower-bound and the upper-bound of the signers who actually join the signing protocol. To rigorously capture the notion of this scheme, we define the notion of the real signers in a threshold ring signature scheme. We then suggest a construction based on an undeniable signature scheme [8]. We also prove our proposal secure in the security model we have introduced. In addition, it can be converted to a threshold group signature scheme, equipped with no group manager, no set-up protocol but an efficient revocation protocol.

1.2 Related Works

The notion of threshold signature has been widely studied. One of the earliest literature about this topic is found in [12]. Many other papers, e.g., [10], [11], [16], [17], [22], [23], have been published. In the literature, a threshold signature scheme is a regular digital signature scheme, such as El Gamal, RSA, and DSS, and allows a (static) group of n players who share a secret key among them with respect to one public-key, to sign messages via their collaboration.

Related notions to our topic include group and ring signatures.

A group signature scheme, originally due to [7] and followed by many other papers, e.g., [2]~[4], [6], allows a member of a (static) group to sign messages on the behalf of the group, while preserving his anonymity. A notable aspect of these group signature schemes is that there is a group manager who can revoke the anonymity of members if necessary. Usually, the group manager need set up a special type of key assignment protocol and hence, it is difficult to change the group dynamically.

A ring signature scheme, due to [21], is similar to a group signature scheme. The difference is that a ring signature scheme has no group manager, no set-up procedure, and allows the group to be dynamic, but has no revocation procedure. In these schemes, the Public-Key Infrastructure is assumed. The signer who wants to sign a message can choose an arbitrary set of possible signers, called the ring, including himself and he can sign messages only using his secret-key and the public-keys of the "ring" members.

Our proposals are a kind of the threshold version of the ring signature. We assume that all the potential signers have their own public-keys for signatures. We call these schemes "threshold ring signature schemes" to distinguish them from ordinary threshold signature schemes. Like a group/ring signature scheme, a threshold ring signature scheme is "signer-ambiguous", which means that the verifier cannot distinguish who participated in the protocol from those who didn't.

A few implementations of threshold ring signature schemes have been proposed [1], [5]. In addition, the witness indistinguishable proofs shown in [9] can be combined with the Fiat-Shamir technique to create a threshold ring signature scheme. Actually, our concrete proposals are produced based on their WI technique.

An intuitive notion of "exact t-out-of-n signature" is discussed in [19], although the definitions are not given. They also proposed a few heuristic implementations [18], [19], but no security analysis is given.

No known reference describes a definition capturing the notion of "exact t-out-of-n signature", and a threshold ring signature scheme which makes it possible to specify the exact number of the signers who actually participate in the protocol.

2. Ranged Threshold Signature: Model and Security Definitions

2.1 Notations and Syntax

Let X be a probability space on finite set $\mathcal{S}(\subset \{0,1\}^*)$.

We denote by $x \leftarrow X$ the operation of sampling an element of $\mathcal{S}$ according to the distribution of X, and assigning the result of this experiment to the variable x. We also write, for a finite set $\mathcal{S}$, $x \leftarrow_U \mathcal{S}$ to denote the operation of sampling an element of $\mathcal{S}$ uniformly, and assigning the result of this experiment to the variable x. We write $x := a$ to denote the operation of assigning the value of a to the variable x. Let A be a probabilistic algorithm. We write $y \leftarrow A(x_1, \ldots, x_n)$ to denote the experiment of running A for given $(x_1, \ldots, x_n)$, picking r uniformly from an appropriate domain, and assigning the result of this experiment to the variable y, i.e., $y := A(x_1, \ldots, x_n; r)$.

Let $\epsilon, \tau : \mathbb{N} \to [0,1] (\subset \mathbb{R})$ be positive $[0,1]$-valued functions. We say that $\epsilon(k)$ is negligible in k if, for any constant c, there exists a constant, $k_0 \in \mathbb{N}$, such that $\epsilon(k) < (1/k)^c$ for any $k > k_0$. We say that $\tau(k)$ is overwhelming in k if $\epsilon(k) \triangleq 1 - \tau(k)$ is negligible in k.

For a finite set S, we denote by a_S vector $(a_i)_{i \in S}$. Let $\#S$ be the number of the elements in S. For a subset $T \subset S$, we write $S - T$ to be the complement of T in S, i.e., $S \cap \overline{T}$.

Here, we define a ranged threshold signature scheme.

a) Syntax.

A ranged (t, t', n)-threshold signature scheme is a tuple of algorithms, Σ = (Gen, Sig, Comb, Ver, Rec), such that, for $k \in \mathbb{N}$,

- Gen, the key generation algorithm, is a probabilistic polynomial-time algorithm that takes security parameter k, and outputs (pk, sk), denoted a pair of a public-key and a secret-key.

$$(pk, sk) \leftarrow \mathrm{Gen}(1^k).$$

- Sig, the signing algorithm, is a probabilistic polynomial-time (in k) algorithm that takes range $[t, t']$, a vector of public keys, $pk_N \triangleq (pk_i)_{i \in N}$, where $N \triangleq \{1, ..., n\}$, one secret key, sk_i, where $i \in N$, and message $m \in \{0,1\}^*$, and that outputs (partial) signature σ_i.

$$\sigma_i \leftarrow \mathrm{Sig}_{sk_i}([t, t'], pk_N, m).$$

Here, note that σ_i is a (partial) signature that participant i who has possession of secret-key sk_i can generate on a message by himself.

- Comb, the signature combining algorithm, is a probabilistic polynomial-time (in k) algorithm that takes range $[t, t']$, a vector of public keys, pk_N, a vector of the secret keys, $sk_S \triangleq (sk_i)_{i \in S}$, where $S \subset N$, signatures $\sigma_S \triangleq (\sigma_i)_{i \in S}$, and message $m \in \{0,1\}^*$, and that outputs threshold signature σ.

$$\sigma \leftarrow \mathrm{Comb}_{sk_S}([t, t'], pk_N, \sigma_S, m).$$

- Ver, the signature verification algorithm, is a deterministic polynomial-time (in k) algorithm that takes range $[t, t']$, a vector of public-keys, pk_N, message $m \in \{0,1\}^*$, and threshold signature σ, and that outputs a bit.

$$1/0 \leftarrow \mathrm{Ver}([t, t'], pk_N, m, \sigma).$$

- Rec, the share recovering algorithm, is a deterministic polynomial-time (in k) algorithm that takes ranged $[t, t']$, a vector of public-keys, pk_N, message $m \in \{0,1\}^*$, and threshold signature σ, and that outputs a vector of strings, $\sigma_N \triangleq (\sigma_i)_{i \in N}$.

$$\sigma_N := \mathrm{Rec}([t, t'], pk_N, m, \sigma).$$

For simplicity, we write $(pk_S, sk_S) \leftarrow \mathrm{Gen}(1^k)$ to denote the experiment of $(pk_i, sk_i) \leftarrow \mathrm{Gen}(1^k)$ for $i \in S$ and assigning $(pk_S, sk_S) := (pk_i, sk_i)_{i \in S}$. Similarly, we write $\sigma_S \leftarrow \mathrm{Sig}_{sk_S}([t, t'], pk_N, m)$ to denote the experiment of $\sigma_i \leftarrow \mathrm{Sig}_{sk_i}([t, t'], pk_N, m)$ for $i \in S (\subset N)$ and assigning $\sigma_S := (\sigma_i)_{i \in S}$. In addition, we write $\sigma \leftarrow \mathrm{ThrSig}_{sk_S}([t, t'], pk_N, m)$ to denote the following experiment:

$\sigma_S \leftarrow \mathrm{Sig}_{sk_S}([t, t'], pk_N, m)$;

$\sigma \leftarrow \mathrm{Comb}_{sk_S}([t, t'], pk_N, \sigma_S, m)$;

return σ.

We call this imaginary algorithm the threshold signing algorithm.

b) Correctness.

For every enough large $k \in \mathbb{N}$, Σ must satisfy the following **correctness** condition: For every integer n, t, t', $0 \leqq t \leqq t' \leqq n$, for every $S \subset N = \{1, \ldots, n\}$, $t \leqq \#S \leqq t'$, and for every $m \in \{0,1\}^*$, if $(pk_N, sk_N) \leftarrow \mathrm{Gen}(1^k)$, $\sigma_S \leftarrow \mathrm{Sig}_{sk_S}([t, t'], pk_N, m)$, $\sigma \leftarrow \mathrm{Comb}_{sk_S}([t, t'], pk_N, \sigma_S, m)$, and $\sigma'_N := \mathrm{Rec}\ ([t, t'], pk_N, m, \sigma)$, then it always holds that

$$\mathrm{Ver}([t, t'], pk_N, m, \sigma) = 1 \text{ and } \sigma_S = \sigma'_S,$$

where $\sigma_S := (\sigma_i)_{i \in S}$ and $\sigma'_S := (\sigma'_i)_{i \in S}$.

c) Partial signature.

We say that σ_i is the partial signature of i with respect to $([t, t'], pk_N, m, \sigma)$ if $\sigma'_N := \mathrm{Rec}([t, t'], pk_N, m, \sigma)$ and $\sigma_i = \sigma'_i$.

d) Language of partial signatures.

Here, we define language L in $\mathcal{NP}$, derived from Σ as follows,

$$L \triangleq \left\{ ([t, t'], pk_N, m, \sigma_i) \;\middle|\; \begin{array}{l} \exists r, r', \text{ and } \exists pk_i \in pk_N \text{ s.t.} \\ (pk_i, sk_i) = \mathrm{Gen}(1^k; r) \quad \wedge \\ \sigma_i = \mathrm{Sig}_{sk_i}([t, t'], pk_N, m; r'). \end{array} \right\}.$$

We denote by $L([t, t'], pk_N, m)$ the language of the (partial) signature σ_i on $([t, t'], pk_N, m)$, namely $\sigma_i \in L([t, t'], pk_N, m)$ iff $([t, t'], pk_N, m, \sigma_i) \in L$.

2.2 Security Definitions

In the following, we define security requirements for a ranged threshold signature scheme — "signer-ambiguity", "soundness", and "exculpability".

e) (Computational) Signer-Ambiguity.

We consider the following game of Σ against adversary D with respect to parameters, N, and s, where $s \leqq n(= \#N)$. At the beginning of the game, (pk_N, sk_N) are generated by $\mathrm{Gen}(1^k)$ and a random subset S whose order is s is chosen from N. Then, D takes (pk_N, s) and submits any sequence of queries to threshold signing oracle $\mathrm{ThrSig}_{sk_S}([\cdot,\cdot], pk_N, \cdot)$, where we denote by $\mathrm{ThrSig}_{sk_S}([\cdot,\cdot], pk_N, \cdot)$ the threshold signature oracle that takes range $[t,t']$, where $t \leqq s \leqq t'$, and message $m \in \{0,1\}^*$, and that returns σ, with respect to sk_S. At the end of the game, D outputs $i \in N$. We say that D wins (the game) if $i \in S$. We define the advantage of D against Σ with respect to N and s, as $\mathrm{Adv}^{\mathrm{ambi}}(\Sigma, D, N, s)(k) =$

$$\Pr\left[\begin{array}{l}(pk_N, sk_N) \leftarrow \mathrm{Gen}(1^k);\quad S \subset_U N;\\ i \leftarrow D^{\mathrm{ThrSig}_{sk_S}([\cdot,\cdot],pk_N,\cdot)}(pk_N, s)\end{array} : i \in S\right] - \frac{s}{n},$$

where the probability is taken over the choice of subset S from N, the coin tosses of Gen, Sig, and Comb in Σ and those of D.

[Definition 1] (Signer-Ambiguity) We say that Σ is signer-ambiguous if, for every probabilistic polynomial-time (in k) adversary D, every N, and every $s(\leqq n)$, the advantage $\mathrm{Adv}^{\mathrm{ambi}}(\Sigma, D, N, s)(k)$ is negligible in k.

f) Soundness.

We define the soundness condition for a ranged threshold signature scheme. Let A be a probabilistic algorithm (adversary) that takes 1^k and outputs $([t,t'], pk_N, m, \sigma)$. We define the advantage of A against Σ, $\mathrm{Adv}^{\mathrm{sound}}(\Sigma, A)(k)$, as the probability that A outputs $([t,t'], pk_N, m, \sigma)$ such that $\mathrm{Ver}([t,t'], pk_N, m, \sigma) = 1$ and, letting σ_i be the partial signature of i with respect to $([t,t'], pk_N, m, \sigma)$, the number of σ_i's that belong to $L([t,t'], pk_N, m)$ does NOT lies in $[t,t']$, where the probability is taken over the coin tosses of A.

[Definition 2] (Soundness) We say that Σ is sound if for any probabilistic polynomial-time (in k) algorithm A, $\mathrm{Adv}^{\mathrm{sound}}(\Sigma, A)(k)$ is negligible in k.

g) Exculpability.

We consider the following game against an adversary F with respect to (N, S'), where $S' \subset N$. At the beginning of the game, (pk_N, sk_N) are generated by $\mathrm{Gen}(1^k)$. Then, F takes pk_N, and $sk_{S'}$, and submits any sequence of queries to threshold signing oracle ThrSig, where we denote by ThrSig the threshold signature generation oracle that takes the vector of public-keys, pk_N, the set of the signers, $\tilde{S}$, range $[\tilde{t}, \tilde{t'}]$, where $\tilde{t} \leqq \tilde{s} \leqq \tilde{t'} \leqq n$, and message $\tilde{m} \in \{0,1\}^*$, and that returns $\tilde{\sigma}$ on message $\tilde{m}$, with respect to $sk_{\tilde{S}}$. Finally, F outputs $([t,t'], m, \sigma)$. We say that F entraps a participant if the following condition holds:

(1) $\mathrm{Ver}([t,t'], pk_N, m, \sigma) = 1$,

(2) m was not submitted to ThrSig for signing, and

(3) There is $\sigma_i \in L([t,t'], pk_N, m)$ such that $i \notin S'$ and σ_i is the partial signature of i with respect to $([t,t'], pk_N, m, \sigma)$.

We define the advantage of F against Σ with respect to N and S' as the probability that F entraps some participant. Namely, $\mathrm{Adv}^{\mathrm{entrap}}(\Sigma, F, N, S')(k) = \Pr[(pk_N, sk_N) \leftarrow \mathrm{Gen}(1^k) : F^{\mathrm{ThrSig}}(pk_N, sk_{S'})$ entraps a participant.], where the probability is taken over the coin tosses of Gen, Sig, and Comb in Σ and those of F.

[Definition 3] (Exculpability) We say that Σ is exculpable if, for any probabilistic polynomial-time adversary F and any $S' \subset N$, $\mathrm{Adv}^{\mathrm{entrap}}(\Sigma, F, N, S')(k)$ is negligible in k.

We say that Σ is "strongly" exculpable if, Σ is exculpable in the same condition except that sentence 2 above is replaced by the sentence that the same $([t,t'], pk_N, m, \sigma)$ did not appear in the query/answer list between F and ThrSig.

3. Proposed Scheme, Protocol 1

In this section, we present our proposal, Protocol 1.

Protocol 1 is intuitively constructed as follows: Each real signer produces partial signature σ_i on message m, by using an undeniable signature scheme. Then, by collaborating each other, the real signers make fake partial signatures, $\sigma_{i'}$'s, for $i' \notin S$, where S denotes the set of the real signers. Since σ_i is an undeniable signature, nobody except them can see whether σ_i is valid or not. Then, they collaborate and produce two non-interactive zero-knowledge proofs: One proves that the number of σ_i's such that $(pk_N, m, \sigma_i) \in L$ is at least t, whereas the other proves the number of σ_i's such that $(pk_N, m, \sigma_i) \in \overline{L}$ is at least t' (Namely, # of $(pk_N, m, \sigma_i) \in L$ is "at most" t').

Now let us start the concrete description of Protocol 1. Let G be multiplicative group of prime order q and let g be generator of G. Let $H : \{0,1\}^* \to G$, $H' : \{0,1\}^* \to G$, and $H'' : \{0,1\}^* \to \mathbb{Z}_q$ be distinct hash functions. We denote by $N = \{1, \ldots, n\}$ the set of n players. Let (y_i, x_i) be the public/secret key pair of participant $i \in N$ such that $y_i = g^{x_i} \in G$. Let $y_N = (y_1, \ldots, y_n)$.

h) **Signing procedure :**

Let $S(\subset N)$ be a set of the signers who collaborate with each other to generate a signature on message m.

(1) All participants of S collaborate to select randomly $T, T' \subset_U N$ s.t. $\#T = t, \#T' = t'$ and $T \subset S \subset T'$, and collaborate to select randomly $r \leftarrow_U \{0,1\}^k$.

(2) Each participant $i \in S$ computes $\sigma_i = h^{x_i}$, using his own secret key $x_i \in \mathbb{Z}_q$, where $h = H([t,t'], y_N, m, r)$, and distributes σ_i to the other members of S.

(3) All participants of S collaborate to select randomly $\sigma_i \leftarrow_U G$ for $i \in T' - S$ (If $T' - S = \emptyset$, just ignore this step).

(4) Each participant in S sets $\sigma_i = \prod_{j=0}^{t'}(A_j)^{i^j} \in G$ for all $i \in N - T'$, where $A_0 = H'([t,t'], y_N, m, r) \in G$, and $A_j = \prod_{k \in T'}(\sigma_k / A_0)^{m_{jk}} \in G$ for $j = 1, ..., t'$, where

$$\begin{pmatrix} m_{1,k_1} & \cdots & m_{1,k_{t'}} \\ \vdots & \ddots & \vdots \\ m_{t',k_1} & \cdots & m_{t',k_{t'}} \end{pmatrix} = \begin{pmatrix} {k_1}^1 & \cdots & {k_1}^{t'} \\ \vdots & \ddots & \vdots \\ {k_{t'}}^1 & \cdots & {k_{t'}}^{t'} \end{pmatrix}^{-1}$$

is the inverse matrix of van der Monde matrix and $T' = \{k_1, ..., k_{t'}\}$.

Notice that there exists a polynomial of degree t', $\alpha(x) \in \mathbb{Z}_q[x]$, such that $A_0 = h^{\alpha(0)} \in G$ and $\sigma_i = h^{\alpha(i)} \in G$ for $i \in N$.

(5) Each participant $i \in S$ selects randomly $w_i \leftarrow_U \mathbb{Z}_q$, sets $a_i = g^{w_i}, b_i = h^{w_i} \in G$, and distributes a_i, b_i to the other members of S.

(6) All participants of S collaborate to select randomly $z_i, c_i \leftarrow_U \mathbb{Z}_q$, and set $a_i = g^{z_i} y_i^{c_i}, b_i = h^{z_i} \sigma_i^{c_i} \in G$ for $i \in N - S$.

(7) All participants of S collaborate to select randomly $c_i \leftarrow_U \mathbb{Z}_q$ for $i \in S - T$ (If $S - T = \emptyset$, just ignore this step).

(8) Each participant in S sets a polynomial of degree $(n-t)$ in $\mathbb{Z}_q[x]$, $\beta(x)$, such that

$$\beta(0) = H''([t,t'], y_N, m, r, h, (A_i)_{i=0}^{t'}, a_N, b_N) \in \mathbb{Z}_q, \text{ and}$$
$$\beta(i) = c_i \in \mathbb{Z}_q \quad \text{for } i \in N - T,$$

where $(A_i)_{i=0}^{t'} = (A_0, A_1, \ldots, A_{t'})$, $a_N = (a_1, \ldots, a_n)$, and $b_N = (b_1, \ldots, b_n)$.

(9) Each participant $i \in S$ sets $z_i = w_i - \beta(i) x_i \in \mathbb{Z}_q$ using secret key $x_i \in \mathbb{Z}_q$, and distributes z_i to the other members of S.

(10) All participants of S output signature $\sigma = (r, (A_i)_{i=1}^{t'}, \beta(x), z_N)$, where $(A_i)_{i=1}^{t'} = (A_1, \ldots, A_{t'})$ and $z_N = (z_1, \ldots, z_n)$.

[Remark 1] In Step 1, 3, 6, and 7, all participants in S need collaborate with each other to pick up some random elements. In the other steps, no interaction is required.

i) **Verification procedure:**

The verifier is given parameter $[t,t']$, public keys y_N, message m, and signature $\sigma = (r, (A_i)_{i=1}^{t'}, \beta(x), z_N)$.

(1) The verifier checks $r \in \{0,1\}^k$, and $h, A_0, A_1, \ldots, A_{t'} \in G$, where

$$h = H([t,t'], y_N, m, r) \text{ and } A_0 = H'([t,t'], y_N, m, r).$$

(2) The verifier checks that $z_i \in \mathbb{Z}_q$ and $\sigma_i \in G$ for all $i \in N$, where $\sigma_i = \prod_{j=0}^{t'}(A_j)^{i^j}$.

(3) The verifier checks that $\beta(x) \in \mathbb{Z}_q[x]$, $\deg \beta(x) = (n-t)$, and $\beta(0) = H''([t,t'], y_N, m, r, h, (A_i)_{i=0}^{t'}, a'_N, b'_N)$, where $(A_i)_{i=0}^{t'} = (A_0, \ldots, A_{t'})$, $a'_N = (a'_1, \ldots, a'_N)$ and $b'_N = (b'_1, \ldots, b'_N)$ such that $a'_i = g^{z_i} y_i^{\beta(i)} \in G$ and $b'_i = h^{z_i} \sigma_i^{\beta(i)} \in G$ for $i \in N$.

(4) If all checks above are passed, the verifier accepts, otherwise rejects.

[Remark 2] The partial signature of i with respect to $([t,t'], y_N, m)$ is (r, σ_i) and the language of (partial) signature L is

$$L \triangleq \left\{ ([t,t'], y_N, m, (r, \sigma_i)) \;\middle|\; \begin{array}{l} \exists x_i \in \mathbb{Z}_q, \exists r \in \{0,1\}^k, \\ \exists y_i \in y_N \text{ s.t.} \\ \quad y_i = g^{x_i} \text{ and} \\ \sigma_i = H([t,t'], y_N, m, r)^{x_i}. \end{array} \right\},$$

where $y_N = (y_1, \ldots, y_n)$.

Notice that (a part of) the partial signature, σ_i, of participant $i \in S$ who really joined the signature procedure has the form of $\sigma_i = h^{x_i}$, which can be seen an undeniable signature proposed by Chaum [8].

j) **Confirmation procedure:**

There is an efficient 4-move zero-knowledge interactive proof on language L for confirmation, see [8], in which participant i can prove that $\log_g (y_i) = \log_h (\sigma_i)$.

k) **Disavowal procedure:**

There is an efficient 4-move zero-knowledge interactive proof on language $\overline{L}$ for disavowal, see [8], in which participant i can prove that $\log_g (y_i) \neq \log_h (\sigma_i)$.

4. Security Analyses

In this section, we provide security analyses of the proposed scheme.

[Theorem 1] (Signer-Ambiguity) Protocol 1 is computationally signer-ambiguous under the decisional Diffie-Hellman assumption in the random oracle model.

Proof. Suppose for contradiction that there is an adversary D with advantage ϵ, which means that, by definition, D can correctly guess $i \in S$ with probability $\frac{s}{n} + \epsilon$. We now construct an algorithm A to solve the decisional Diffie-Hellman problem. Let (g, g', u, v) be a given instance, where $g, g', u, v \in G$. When (g, g', u, v) is a DDH tuple, $\log_g (u) = \log_{g'} (v)$ holds. We construct A as follows:

(1) Pick up at random $S \subset_U N$ and $i \leftarrow S$.

(2) Pick up at random $x_j \in \mathbb{Z}_q$ for $j \in S - \{i\}$.

(3) Set $y_i := u$ and $y_j := g^{x_j}$ for $j \in S - \{i\}$.

(4) Pick up at random $y_{i'} \leftarrow G$ for $i' \in N - S$.

(5) Feed (t, t', s, y_N) to D, where $s = \#S$.

(6) In case D submits a fresh query to random oracles, H' and H'', pick up a random element in G to reply with. Store the query/answer pairs in the lists, $Q_{H'}$ and $Q_{H''}$, respectively.

(7) In case D submits a fresh query to random oracle H, pick up random $\alpha \leftarrow_U \mathbb{Z}_q$ and returns g'^{α}. Store these

value in the query/answer list Q_H, as well as α.

(8) In case D submits a signing request on $\{[t,t'], y_N, m\}$, first, pick up random $r \leftarrow_U \mathbb{Z}_q$. Second, set g'^{α} as $H([t,t'], y_N, m, r)$ and $\sigma_i = v^{\alpha}$, where $\alpha \leftarrow_U \mathbb{Z}_q$ (In case $H([t,t'], y_N, m, r)$ has already been booked in Q_H, use the corresponding α). Generate the (threshold) signature σ, with respect to $sk_{S-\{i\}}$, following the signature procedure in Protocol 1.

(9) Finally, when D outputs i', output 1 if $i' = i$, otherwise flip a coin $b \in \{0,1\}$ to output.

Here, notice that if (g, g', u, v) is a DDH tuple, σ is identical to the real signature using sk_S, otherwise if it is a random tuple, σ is identical to the real signature using $sk_{S-\{i\}}$.

The advantage of A against the DDH problem is defined as

$$\Pr[A(g, g'.u, v) = 1 | (g, g', u, v) \in \mathrm{DDH}] - \Pr[A(g, g', u, v) = 1 | (g, g', u, v) \notin \mathrm{DDH}].$$

Here, $\Pr[A(g, g'.u, v) = 1 | (g, g', u, v) \in \mathrm{DDH}] = \Pr[i' = i] + \Pr[i' \neq i] \cdot \Pr[b = 1 | i \neq i]$

$$= \left(\frac{1}{n} + \frac{\epsilon}{s}\right) + \left(1 - \left(\frac{1}{n} + \frac{\epsilon}{s}\right)\right) \cdot \frac{1}{2} = \frac{1}{2} \cdot \left(\frac{\epsilon}{s} + \frac{1}{n}\right) + \frac{1}{2},$$

where $\Pr[i' = i] = \Pr[i' \in S] \cdot \Pr[i' = i | i' \in S] = \left(\frac{s}{n} + \epsilon\right) \cdot \frac{1}{s} = \frac{1}{n} + \frac{\epsilon}{s}$.

On the other hand, $\Pr[A(g, g'.u, v) = 1 | (g, g', u, v) \notin \mathrm{DDH}] = \Pr[i' = i] + \Pr[i' \neq i] \cdot \Pr[b = 1 | i' \neq i]$

$$= \left(\frac{1}{n} - \frac{\epsilon}{n - s + 1}\right) + \left(1 - \left(\frac{1}{n} - \frac{\epsilon}{n - s + 1}\right)\right) \cdot \frac{1}{2} = \frac{1}{2}\left(\frac{1}{n} - \frac{\epsilon}{n - s + 1}\right) + \frac{1}{2},$$

where $\Pr[i' = i] = \Pr[i' \notin S \backslash \{i\}] \cdot \Pr[i' = i | i' \notin S \backslash \{i\}] = \left(1 - \left(\frac{s-1}{n} + \epsilon\right)\right) \cdot \frac{1}{n-s+1}$.

Therefore, the advantage of A becomes $\frac{1}{2}\left(\frac{1}{s} + \frac{1}{n-s+1}\right) \cdot \epsilon$. To suppress the advantage of A to be negligible in k, ϵ must be negligible in k. □

[Theorem 2] (Soundness) Protocol 1 is sound in the random oracle model.

Proof. Let A be an adversary that fools the verifier, after it accessed the random oracles $q(k)$ times in total.

First, consider Case 1: $\#\{i \in N | (r, \sigma_i) \in L([t,t'], pk_N, m)\} < t$.

$\mathrm{Ver}([t,t'], pk_N, m, \sigma) = 1$ implies that $a_i = g^{z_i} y_i^{\beta(i)} \in G$ and $b_i = h^{z_i} \sigma_i^{\beta(i)} \in G$ for $i \in N$, which means that $\log_g(a_i) = z_i + x_i \beta(i)$ and $\log_h(b_i) = z_i + x_i' \beta(i)$ for $i \in N$. Note that if $x_i \neq x_i'$, $\beta(i)$ is determined. Hence, Case 1 implies that there are at least $(n - t + 1)$ determined $\beta(i)$'s, meaning polynomial $\beta(x)$ of degree $(n - t)$ is uniquely determined. Since H'' is a random oracle, for any given $([t,t'], y_N, m, r, h, (A_i)_{i=0}^{t'}, a'_N, b'_N)$, the probability that

$$H''([t,t'], y_N, m, r, h, (A_i)_{i=0}^{t'}, a'_N, b'_N) = \beta(0)$$

is at most q^{-1}, where the probability is taken over the choice of H''. Therefore, for any A with at most $q_{H''}$ queries to random oracle H'', the verifier accepts the signature with a probability at most $\frac{q_{H''}}{q}$.

Next, consider Case 2: $\#\{i \in N | (r, \sigma_i) \in L([t,t'], pk_N, m)\} > t'$.

Since $\mathrm{Ver}([t,t'], pk_N, m, \sigma) = 1$, we have $\sigma_i = \prod_{j=0}^{t'} (A_j)^{i^j} \in G$ for $i \in N$. Hence, there is a polynomial of degree t', $\alpha(x) \in \mathbb{Z}_q[x]$, such that $\alpha(i) = \log_h(\sigma_i)$ for $i \in N$. Case 2 implies that at least $(t' + 1)$ x_i's are on polynomial $\alpha(x)$, which determines the polynomial. Since H' a is random oracle, for any given $([t,t'], y_N, m, r)$, the probability that "$\log_h(H'([t,t'], y_N, m, r)) = \alpha(0)$" is at most q^{-1}, where the probability is taken over the choice of H'. Thus, for any adversary A with at most $q_{H'}$ number of queries to random oracle H', the verifier accepts the signature with a probability at most $\frac{q_{H'}}{q}$.

To sum up, for any invalid instance, and any adversary with at most $q(k)$ queries to the random oracles, the verifier can be fooled only with probability at most $\frac{q(k)}{q}$. □

[Theorem 3] (Exculpability) Protocol 1 is exculpable under the computational Diffie-Hellman assumption in the random oracle model.

Proof. Suppose for contradiction that there is subset $S' \subset N$ and adversary F that, given S', entraps a signer against Protocol 1. We then show that we can construct A that given a random instance, (g, Y, Y'), where $g, Y, Y' \in G$ outputs Y'^{y} such that $Y = g^{y}$. We construct A as follows.

(1) Pick up $x_i \leftarrow_U \mathbb{Z}_q$ as sk_i and compute $y_i = g^{x_i} \in G$ for $i \in S'$. Pick up $r_i \leftarrow_U \mathbb{Z}_q$ to set $y_i = Y^{r_i} \in G$ for $i \in N - S'$.

(2) Feed pk_N and $sk_{S'}$ to adversary F.

(3) In case F submits a fresh query to random oracles, H' and H'', pick up a random element in G to reply with. Then store the query/answer pairs in the lists, $Q_{H'}$ and $Q_{H''}$, respectively.

(4) In case F submits a fresh query to random oracle H, pick up random $u \leftarrow_U \mathbb{Z}_q$ and return Y'^{u}. Then store the query/answer pair in the list Q_H.

(5) In case F submits query $([\tilde{t}, \tilde{t}'], \tilde{S}, pk_N, m)$, where $\tilde{t} \leq \tilde{s} (= \#\tilde{S}) \leq \tilde{t}' \leq n$, to the signing oracle ThrSig, return σ as follows.

(a) Pick up at random $\tilde{T}, \tilde{T}' \subset N$ s.t. $\#\tilde{T} = \tilde{t}, \#\tilde{T}' = \tilde{t}'$ and $\tilde{T} \subset \tilde{S} \subset \tilde{T}'$.

(b) Pick up at random $r \leftarrow_U \{0,1\}^k$ and $v \leftarrow_U \mathbb{Z}_q$, to define hash value $H([\tilde{t}, \tilde{t}'], y_N, m, r)$ to be g^{v}. In case $H([\tilde{t}, \tilde{t}'], y_N, m, r)$ has already been booked in Q_H, A halts. Otherwise, set $\sigma_i = y_i^{v}$ for all $i \in \tilde{S}$. Here, notice that since

$\sigma_i = h^{\log_g y_i}$ and $h = g^v$, σ_i is identical to the real partial signature using the secret key x_i such that $y_i = g^{x_i}$.

(c) Pick up at random $\sigma_i \leftarrow_U G$ for all $i \in \tilde{T}' - \tilde{S}$.

(d) Set $A_0 = H'([\tilde{t}, \tilde{t}'], y_N, m, r) \in G$ as follows: If $H'([\tilde{t}, \tilde{t}'], y_N, m, r)$ has already been booked in $Q_{H'}$, set the value as A_0; otherwise, select A_0 at random.

(e) Compute $A_j = \prod_{k \in \tilde{T}'} (\sigma_k / A_0)^{m_{jk}} \in G$ for $j = 1, ..., \tilde{t}'$, following Step 4 in the signature procedure in Protocol 1.

(f) Compute $\sigma_i = \prod_{j=0}^{\tilde{t}'} (A_j)^{i^j} \in G$ for $i \in N - \tilde{T}'$.

(g) Pick up a random polynomial of degree $n - \tilde{t}$, $\beta(x) \in \mathbb{Z}_q[x]$.

(h) Pick up random $z_i \leftarrow_U \mathbb{Z}_q$ to compute $a_i = g^{z_i} y_i^{\beta(i)}$ and $b_i = h^{z_i} \sigma_i^{\beta(i)}$ for all $i \in N$.

(i) Set $H''([\tilde{t}, \tilde{t}'], y_N, m, r, h, (A_i)_{i=0}^{\tilde{t}'}, a_N, b_N) = \beta(0)$. In case it has already been booked in $Q_{H''}$, A halts.

(j) Return $\sigma = (r, (A_i)_{i=1}^{\tilde{t}'}, \beta(x), z_N)$.

(k) Store the query/answer pair in the list Q_{TS}.

(6) Finally, F outputs (x, σ), where $x = ([t, t'], pk_N, m)$, such that $\mathrm{Ver}(x, \sigma) = 1$ and $m \notin Q_{TS}$. Then, retrieve $\sigma_{N-S'}$ from σ, pick up at random $\sigma_i \leftarrow_U \sigma_{N-S'}$, and output σ_i^{1/ur_i}.

Here, notice that, since $\sigma_i = h^{\log_g y_i}$ and $h = g^v$, σ_i is identical to the real partial signature using the secret key x_i such that $y_i = g^{x_i}$. Hence, $\log_g(a_i) = \log_h(b_i)$ for $i \in S$. It implies that the simulation of A for signing with respect to S is identical to the real signature, unless h or $\beta(0)$ has already been booked in the query/answer lists. Hence, the probability of failure is at most $\frac{q_H}{2^k} + \frac{q_{H''}}{q^n}$, where q_H and $q_{H''}$ denote the numbers of queries of F to the random oracles, H and H'', respectively.

If F entraps a signer, then there is at least a $\sigma_i \in \sigma_{N-S'}$ such that $\sigma_i = h^{x_i}$ where $x_i = r_i \log_g Y$ and $h = Y'^u$, which implies $\sigma_i = Y'^{ur_i y}$ such that $Y = g^y$. Although we don't see who is entrapped, but with probability at least $\frac{1}{n-s'}$, we can simply guess the entrapped player $i \notin S'$. Note that, since $m \notin Q_{TS}$ (by definition), the value h above does not lie in the list Q_H at most with probability $\frac{q_H}{q}$.

Hence, if F entraps a signer with probability ε after it accesses the random oracles, H and H'', q_H and $q_{H''}$ times, respectively, A can solve the computational DH problem with probability at least $\frac{1}{n-s'} \left(\varepsilon - \frac{q_H}{2^k} - \frac{q_H}{q} - \frac{q_{H''}}{q^n} \right)$. □

[Theorem 4] (Exculpability in case $t = t'$) If $t = t'$, Protocol 1 is "strongly" exculpable under the discrete logarithm assumption in the random oracle model.

Proof. Suppose for contradiction that there are subset $S' \subset N$, where $\#S' = t$, and adversary F such that F takes $sk_{S'}$ and existentially forge a signature of some participant $i \notin S'$. Then, we show that we can construct algorithm A that solves the discrete logarithm problem. Let $g, Y \in G$ be a given instance of discrete logarithm problem. The goal of A is to output $\log_g Y$.

A runs F as same as in case of $t < t'$ except that: In case F submits a fresh query to random oracle H, pick up a random element in G to reply with, instead of Y'^u. Then store the query/answer pair in the list Q_H.

Suppose that F outputs a valid message/signature pair, (x, σ), such that $(x, \sigma) \notin Q_{TS}$, and there is a σ_i in σ such that $i \in N - S'$. This happens with probability ϵ, where ϵ denotes the advantage of F.

Then, A rewinds F on the same input including the same random coins to F, following the standard technique of knowledge extractor in [13], [20].

In the replay, A behaves as same as in the first play except that, letting $x = ([t, t'], pk_N, m)$ and $\sigma = (r, (A_i)_{i=0}^{t}, \beta(x), z_N)$, when F submits $([t, t'], pk_N, m, r, h, (A_i)_{i=0}^{t}, a_N, b_N)$ to H'', where $a_i = g^{z_i} y_i^{\beta(i)}$ and $b_i = h^{z_i} \sigma_i^{\beta(i)}$, pick up at random $\hat{\beta}(0)$ to reply with as $H''([t, t'], pk_N, m, r, h, (A_i)_{i=0}^{t}, a_N, b_N)$. If $\hat{\beta}(0) = \beta(0)$, A halts. However, it happens only with probability q^{-1}.

Suppose F outputs a valid message/signature pair, $(x', \sigma') \notin Q_{TS}$, such that $x = x'$ and $\sigma' = (r, (A_0, \hat{A}_1, \ldots, \hat{A}_t), \hat{\beta}(x), \hat{z}_N)$. This event happens with probability $\frac{1}{q_{H''}} \epsilon$, following [20].

Since the degree of $\hat{\beta}(x)$ is also $n - t$, there are at least t i's, $i \in N$, such that $\beta(i) \neq \hat{\beta}(i)$. Therefore, A obtains at least t pairs of $(a_i, b_i, \beta(i), z_i)$ and $(a_i, b_i, \hat{\beta}(i), \hat{z}_i)$ such that $\beta(i) \neq \hat{\beta}(i)$ and hence, it can compute at least t logs, $\log_g(y_i)$'s, as $\frac{z_i - \hat{z}_i}{\hat{\beta}(i) - \beta(i)}$'s.

Consider the case that every i above lies in S'. Then, the number of σ_i's that belongs to L must be at least $t + 1$, because F has entrapped at least a participant i' (i.e., $\sigma_{i'}$ belongs to L), and every $\sigma_i \in \sigma_{S'}$ belongs to L, because $\log_g(y_i) = \log_h(\sigma_i)$ holds. By theorem 2, the probability is at most $\frac{q_{H'}}{q}$. Hence, with an overwhelming probability, A can extract at least one secret key $\log_g(y_i)$ for $i \in N - S'$, and hence it can output $\log_g Y$ dividing by r_i.

To sum up, the success probability of A is at least $\frac{\epsilon^2}{q_{H''}} - \frac{1}{q} - \frac{q_H}{q} - \frac{q_H}{2^k} - \frac{q_{H'}}{q} - \frac{q_{H''}}{q^n}$. □

5. Applications

A possible application of these protocols is to a group signature scheme. Our proposal can provide a threshold group signature scheme which has no group manager, no set-up protocol but an efficient revocation protocol (by using efficient confirmation and disavowal protocols). Another possible application is to anonymous petition or simple anonymous voting, in which the signers can tell everyone exactly how many people approve or disapprove of an issue, while preserving anonymity among all possible signers.

6. Extension

By definitions above, it is clear that the ranged threshold signature schemes are at best computationally signer-ambiguous. In some application, however, it might be more desirable that signer-ambiguity is unconditional. To realize this property, we have a definition of the unconditionally-signer-ambiguous ranged threshold signature and another construction. Due to the space limitation, however, it has been described in [14].

References

[1] Masayuki Abe, Miyako Ohkubo, and Koutarou Suzuki. Efficient threshold signer-ambiguous signatures from variety of keys. *IEICE Trans. Fund.*, vol.E87-A, no.2:471–479, 2004.

[2] Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *CRYPTO 2000*, pages 255–270, 2000.

[3] Giuseppe Ateniese and Gene Tsudik. Group signatures la carte. In *SODA '99: Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, pages 848–849. Society for Industrial and Applied Mathematics, 1999.

[4] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *EUROCRYPT 2003*, pages 614–629, 2003.

[5] E. Bresson, J. Stern, and M. Szydlo. Threshold ring signatures and applications to ad-hoc groups. In Moti Yung, editor, *Advances in Cryptology — CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 465–480. Springer-Verlag, 2002.

[6] J. Camenisch and M. Stadler. Efficient group signature schemes for large groups. In B. S. Kaliski Jr., editor, *Advances in Cryptology — CRYPTO'97*, volume 1294 of *Lecture Notes in Computer Science*, pages 410–424. Springer-Verlag, 1997.

[7] D. Chaum and E. Van Heyst. Group signatures. In D. W. Davies, editor, *Advances in Cryptology — EUROCRYPT '91*, volume 547 of *Lecture Notes in Computer Science*, pages 257–265. Springer-Verlag, 1991.

[8] David Chaum. Zero-knowledge undeniable signatures. In *EUROCRYPT 1990*, pages 458–464, 1990.

[9] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO 1994*, pages 174–187, 1994.

[10] Y. Desmedt and Y. Frankel. Threshold cryptosystems. In G. Brassard, editor, *Advances in Cryptology — CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*, pages 307–315. Springer-Verlag, 1990.

[11] Y. Desmedt and Y. Frankel. Shared generation of authenticators and signatures. In J. Feigenbaum, editor, *Advances in Cryptology — CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 457–469. Springer-Verlag, 1992.

[12] Yvo Desmedt. Society and Group Oriented Cryptography: A New Concept. In C. Pomerance, editor, *Advances in Cryptology — CRYPTO '87*, volume 293 of *Lecture Notes in Computer Science*, pages 120–127. Springer-Verlag, 1987.

[13] U. Feige, A. Fiat, and A. Shamir. Zero-knowledge proofs of identity. *Journal of Cryptology*, 1:77–94, 1988.

[14] Eiichiro Fujisaki and Koutarou Suzuki. Exact t-out-of-n signer-ambiguous signature (extended abstract). Unpublished Manuscript, 2005.

[15] Eiichiro Fujisaki and Koutarou Suzuki. Just t-out-of-n signature (in japanese). In *SCIS2005 3E4-4*, pages 1639–1644, 2005.

[16] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Robust threshold DSS signatures. In U. Maurer, editor, *Advances in Cryptology — EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 354–371. Springer-Verlag, 1996.

[17] L. Harn. Group-oriented (t, n) threshold digital signature scheme and multisignature. In *IEE Proceedings - Computers and Digital Techniques*, volume 141(5), pages 307–313, 1994.

[18] Tatsumi Higuchi, Takeshi Okamoto, and Eiji Okamoto. Extension of k-out-of-n signature (in japanese). In *SCIS2004 4D1-4*, pages 1441–1446, 2004.

[19] Takeshi Okamoto and Eiji Okamoto. "just" k-out-of-n signature (in japanese). In *CSEC 22-21*, pages 151–156, 2003.

[20] D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 2000.

[21] R. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In C. Boyd, editor, *Advances in Cryptology - Asiacrypt 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565. Springer-Verlag, 2001.

[22] A. De Santis, Y. Desmedt, Y. Frankel, and M. Yung. How to share a function securely. In *Proceedings of the 26th annual ACM Symposium on Theory of Computing*, pages 522–533, 1994.

[23] Victor Shoup. Practical threshold signatures. In *EUROCRYPT 2000*, pages 207–220, 2000.