Redistributing secret shares to new access structures and its applications

Yvo Desmedt[*†]
Center for Cryptography Computer and Network Security
EE & CS, University of Wisconsin–Milwaukee
PO Box 784, Milwaukee, WI 53201, U.S.A.
Tel.: +1 (414) 229-6762, Fax: +1 (414) 229-6958
e-mail: desmedt@cs.uwm.edu

Sushil Jajodia[†]
Department of Information and Software Systems Engineering
School of Information Technology and Engineering,
George Mason University,
Fairfax, Virginia, U.S.A.
Tel.: +1 (703) 993-1653, Fax: +1 (703) 993-1638
e-mail: jajodia@isse.gmu.edu

Abstract

Proactive secret sharing deals with refreshing secret shares, *i.e.*, redistributing the shares of a secret to the *original* access structure. In this paper we focus on the general problem of redistributing shares of a secret key. Shares of a secret have been distributed such that access sets specified in the access structure Γ (*e.g.*, t-out-of-l) can access (or use) the secret. The problem is how to redistribute the secret, without recovering it, in such a way that those specified in the new access structure Γ' will be able to recover the secret.

We also adapt our scheme such that it can be used in the context of threshold cryptography and discuss its applications to secure databases.

1 Introduction

Since it invention, several improvements and variants of threshold schemes [6, 34] and general secret sharing [22] have been presented.

In proactive secret sharing schemes [30, 20] (see also [10]), shares of a secret are being *refreshed* by the participants to avoid a *mobile* attack in which some computers of some participants have been taken over temporarily (*e.g.*, by a computer virus). The main motivation is that in normal secret sharing once the shares of the secret have been distributed they remain fixed for the rest of time. Proactive secret sharing is characterized by the fact that the access structure before and after refresh remains (in essence) the same.

Another approach to a similar problem, called disenrollment, has been followed in [5], where some secret share is being disclosed and one wants to create new shares using a public broadcast channel.

In some applications a refresh is not necessarily sufficient. Let us consider the following scenarios.

[*] This research was carried out while the authors were visiting the Newton Institute on Computer Security, Cryptology and Coding Theory, Cambridge, UK.

[†] Part of this work was supported by NSF NCR-9508528.

- A secret has been distributed such that any 4-out-of-8 participants can recover the secret. An enemy *permanently* takes over (either physically or virtually) two participants. Proactive secret sharing can easily deal with such a situation. Indeed, one refreshes the shares of the participants and excludes during refresh to communicate to the participants whose shares have been corrupted. However, if the enemy is close to permanently take over two extra participants the refresh will be of no avail. Instead the participants may want to redistribute their shares such that any 2-out-of-4 can recover the secret. If more participants are being threatened they may want to redistribute the secret again.
- If in a democratic organization that uses a 50% plus one threshold, the number of members in the organization is sometimes increased. Proactive secret sharing is unable to give new appropriate shares. A good example of such an organization is the United Nations, where the number of members have increased by more than a factor 3 since its creation.

Other examples are given in Section 7.

To deal with such scenarios, we discuss how participants in the access structure can redistribute the secret. We assume that shares of a secret key have been given such that only the subsets of participants of $\mathcal{P}$ specified by the access structure Γ can recover the secret. The goal of the redistribution is that only the subsets of participants of $\mathcal{P}'$ given by the access structure Γ' will be able to recompute the secret, while those in $\Gamma \setminus \Gamma'$ can no longer. We will distinguish the case where insiders are assumed to be honest and the case they are not.

Of particular interest is the application of the redistribution of secret shares to threshold cryptography [8, 11, 16] in such contexts as ElGamal [14, 31], RSA [12, 17] and DSS [28, 18]. In threshold cryptography shares of a secret are (re)used in combination with a cryptosystem without leaking anything new about the secret to outsiders and unauthorized insiders.

In Section 2 we very briefly sketch some definitions. In Section 3 we discuss the background and notations we need to present our redistribution scheme. In Section 4 we present our main scheme. We adapt it to threshold cryptography in Section 5 and to robust threshold cryptography in Section 6.

2 Definitions

Due to space limitations, and to allow enough details in the proofs, and inspired by the threatment of definitions in [20], we only give an intuitive viewpoints to the definitions. Our results will be clearly understandable without formal definitions. Since we discuss unconditionally secure scenarios, as well as computationally secure ones, a good definition section would take excessive space.

We assume that the reader is familiar with secret sharing and refer the reader to the literature. Let $\mathcal{P}$ be the set of participants. Let Γ be a subset of the powerset of $\mathcal{P}$. Informally, a scheme is a secret sharing scheme if the following condition is satisfied. For all keys $k \in \mathcal{K}$ a tuple of $s_i \in \mathcal{S}_i$ $(i \in \mathcal{P})$ exists, such that given a sub-tuple of s_i where $i \in \mathcal{B} \in \Gamma$ one can reconstruct k.

We now briefly survey security. There are different types of security. In perfect secret sharing schemes the shareholders in $\tilde{\mathcal{B}} \notin \Gamma$ learn nothing about the secret in an information theoretical sense.

A secret sharing scheme is zero-knowledge if without knowing the key k one can simulate shares of a set $\tilde{\mathcal{B}} \notin \Gamma$ in expected polynomial time. Different levels of zero-knowledge exist. For details the reader is referred to [19, 15].

From now on we assume that the access structure Γ is monotone.

3 Background and notations

As mentioned in Section 1, we want our redistribution scheme to work for a variety of threshold cryptosystems. To avoid longwindedness we use a unified way to describe several secret sharing that have been used in the context of threshold cryptography.

We assume that the secret key k belongs to a set $\mathcal{K}$, called the keyspace, and that the share of participant i, called s_i belongs to $\mathcal{S}_i$, the i^{th} sharespace. As in [1], we assume that there are operations + defined on $\mathcal{K}$ and all $\mathcal{S}_i$. Without affecting the generality, regardless of $\mathcal{K}$, we use additive notation (as long as reasonable).

3.1 Linear secret sharing

An important class of secret sharing schemes [15, p. 676] (a generalization of [27, 29, 34]) is the one in which the key k can be written as:

$$k = \sum_{i \in \mathcal{B}} \psi_{i,\mathcal{B}}(s_i), \qquad \text{where } \mathcal{B} \in \Gamma \text{ and } \psi_{i,\mathcal{B}} \text{ is a homomorphism} \tag{1}$$

from $\mathcal{S}_i(+)$ to $\mathcal{K}(+)$, for any $i \in \mathcal{B}$. Note that $|\mathcal{S}_i| \geq |\mathcal{K}|$.

Although we do not require that $\mathcal{S}_i = \mathcal{S}_j$, we refer to this class, as the class of *linear secret sharing schemes.* We refer to the homomorphisms $\psi_{i,\mathcal{B}}$ as the scalars and (1) as a sum with weights $\psi_{i,\mathcal{B}}$. We do not need the more general class of secret sharing schemes, called multiplicative [13] in which $\psi_{i,\mathcal{B}}$ is a function.

It is clear that a linear secret sharing scheme is characterized by $(\Gamma_{\mathcal{P}}, \mathcal{K}(+), \boldsymbol{\mathcal{S}}, \boldsymbol{\psi})$, where $\boldsymbol{\mathcal{S}}$ is the tuple of $\mathcal{S}_i$ $(i \in \mathcal{P})$, and $\boldsymbol{\psi}$ is the tuple of $\psi_{i,\mathcal{B}}$ $(i \in B \in \Gamma)$. We silently assume an algorithm D used to distribute the shares with an appropriate distribution.

3.2 Linear subshares

It should be observed that several threshold schemes, *e.g.*, [34, 15, 13, 4] and Benaloh-Leichter's [2] general secret sharing scheme, generalized to any Abelian group $\mathcal{K}(+)$ in [15, p. 675] are linear. We now survey these schemes without going into details.

In all these secret sharing schemes the key space corresponds to an Abelian group, which can be viewed as a (left) R-module[1] $K(+)$ with (left) scalars from an appropriate *commutative* ring R. (We use the definition in which there is an identity element with respect to the multiplication in the ring [23].) Also, the share space $\mathcal{S}_i = \mathcal{K}^{a_i} = \mathcal{K} \times \mathcal{K} \times \cdots \times \mathcal{K}$, where a_i is an appropriate integer. We call a_i the i^{th} share-expansion (the rate [9] is $1/\max_i(a_i)$). In other words, a share $s_i \in \mathcal{S}_i$ can be viewed as a row (or column when appropriate) of *subshares*, so[2]

$$s_i = (k_{i,(0)}, k_{i,(1)}, \ldots, k_{i,(a_i-1)}) \qquad k_{i,(h)} \in K. \tag{2}$$

When there is no ambiguity we write $s_i = (k_{(0)}, k_{(1)}, \ldots, k_{(a_i-1)})_i$. For example, in Shamir's scheme $K(+)$ is a vector space $GF(q)$, which is a special type of module (any finite field is a vector space), and $a_i = 1$ if $l + 1 \geq q$.

The scalar $\psi_{i,\mathcal{C}}$ is a $1 \times a_i$ matrix

$$\psi_{i,\mathcal{C}} = (\omega_{i,\mathcal{C},(0)}, \omega_{i,\mathcal{C},(1)}, \ldots, \omega_{i,\mathcal{C},(a_i-1)}) \qquad \omega_{i,\mathcal{C},(h)} \in R \tag{3}$$

such that

$$\psi_{i,\mathcal{C}}(s_i) = \sum_{h=0}^{a_i-1} \omega_{i,\mathcal{C},(h)} \cdot k_{i,(h)} = (\omega_{i,\mathcal{C},(0)}, \omega_{i,\mathcal{C},(1)}, \ldots, \omega_{i,\mathcal{C},(a_i-1)}) \cdot (k_{i,(0)}, k_{i,(1)}, \ldots, k_{i,(a_i-1)})^T \tag{4}$$

where the first multiplication is the external (scalar) multiplication in the module $K(+)$ and the second one is similar to the scalar vector multiplication. If the share is a column of subshares, T indicates the transpose. If the share is a row of subshares, it corresponds to the identity transformation. Again, we refer to $\omega_{i,\mathcal{C},(h)}$ as scalars, and to (4) as a sum with weights $\omega_{i,\mathcal{C},(h)}$. Also, if there is no ambiguity we write $\psi_{i,\mathcal{C}} = (\omega_{(0)}, \omega_{(1)}, \ldots, \omega_{(a_i-1)})_{i,\mathcal{C}}$.

We call a linear secret sharing scheme $(\Gamma_{\mathcal{P}}, \mathcal{K}(+), \boldsymbol{\mathcal{S}}, \boldsymbol{\psi})$ satisfying (2–4) a sharing scheme with linear subshares. It is characterized by $(\Gamma_{\mathcal{P}}, \mathcal{K}(+), \mathbf{a}, \boldsymbol{\omega})$, where $\mathbf{a}$ is the tuple of a_i $(i \in \mathcal{P})$, and $\boldsymbol{\omega}$ specifies all $\omega_{i,\mathcal{C},(h)}$ $i \in \mathcal{C} \in \Gamma$ and $h \in Z_{a_i}$. To indicate the correspondence between $\boldsymbol{\mathcal{S}}$ and $\mathbf{a}$, and between $\boldsymbol{\psi}$ and $\boldsymbol{\omega}$ we write $(\Gamma_{\mathcal{P}}, \mathcal{K}(+), \boldsymbol{\mathcal{S}}, \boldsymbol{\psi}) \cong (\Gamma_{\mathcal{P}}, \mathcal{K}(+), \mathbf{a}, \boldsymbol{\omega})$.

For more details about each particular scheme consult Appendix A.

[1] A module has a similar definition as a vector space, but the scalars form a ring (instead of a finite field).

[2] We start with $k_{(0)}$ to be compatible with [15].

4 Main scheme

Theorem 1 *If*

1. *there exists a linear secret sharing scheme* $(\Gamma_{\mathcal{P}}, \mathcal{K}(+), \boldsymbol{\mathcal{S}}, \boldsymbol{\psi})$*, and each participant* P_i *in* $\mathcal{P}$ *has received (with the appropriate distribution) a share* $s_i \in \mathcal{S}_i$ *of a secret* $k \in \mathcal{K}$*,*
2. *for each* $i \in \mathcal{P}$ *there exists a linear secret sharing scheme* $(\Gamma'_{\mathcal{P}'}, \mathcal{S}_i(+), \widehat{\boldsymbol{\mathcal{S}}}_i, \widehat{\boldsymbol{\psi}}_i)$*,*
3. $\mathcal{K}(+)$ *is an Abelian group,*
4. *for each* i *and each* $\mathcal{C}$*, where* $i \in \mathcal{C} \in \Gamma$*, and for each* j *and each* $\mathcal{C}'$*, where* $j \in \mathcal{C}' \in \Gamma'$*, there exist appropriate homomorphisms* $\psi'_{j,\mathcal{C}'}$ *and* $\widehat{\psi}'_{j,(i,\mathcal{C})}$ *such that* $\psi_{i,\mathcal{C}}$ *and* $\widehat{\psi}_{i,(j,\mathcal{C}')}$ *pseudo-commute, i.e.,*
$$\psi_{i,\mathcal{C}} \circ \widehat{\psi}_{i,(j,\mathcal{C}')} = \psi'_{j,\mathcal{C}'} \circ \widehat{\psi}'_{j,(i,\mathcal{C})} \tag{5}$$

then any authorized set $\mathcal{B} \in \Gamma$ *of non-faulty participants can redistribute, without computing* k *and using secure channels, shares of* k *for the access structure* Γ' *with* $\mathcal{P}'$ *the set of participants. The resulting shares form a linear secret sharing scheme* $(\Gamma'_{\mathcal{P}'}, \mathcal{K}(+), \boldsymbol{\mathcal{S}}', \boldsymbol{\psi}')$*.*

To prove this theorem we first describe the redistribution protocol.

Redistribution protocol

Step 1 Each participant $i \in \mathcal{B}$ views his own share $s_i \in \mathcal{S}_i$ as a key and computes for each participant $j \in \mathcal{P}'$ a share $\hat{s}_{i,j} \in \widehat{\mathcal{S}}_{i,j}$ of the "key" s_i, using $(\Gamma'_{\mathcal{P}'}, \mathcal{S}_i(+), \widehat{\boldsymbol{\mathcal{S}}}_i, \widehat{\boldsymbol{\psi}}_i)$. Participant i sends $(i, \hat{s}_{i,j})$ to j via a secure channel. We call $\hat{s}_{i,j}$ a *temporary share.*

Step 2 Each participant $j \in \mathcal{P}'$, after having received $(i, \hat{s}_{i,j})$ from each $i \in \mathcal{B}$, computes as share:

$$s'_j = \sum_{i \in \mathcal{B}} \widehat{\psi}'_{j,(i,\mathcal{B})}(\hat{s}_{i,j}) \tag{6}$$

Step 3 Each participant $i \in \mathcal{P}$ erases s_i and all $\hat{s}_{i,j}$ $(j \in \mathcal{P}')$.

Proof. We now prove that the shares s'_j, $j \in \mathcal{P}'$ allow an authorized set $\mathcal{B}' \in \Gamma'$ to recompute the secret k. We claim that $k = \sum_{j \in \mathcal{B}'} \psi'_{j,\mathcal{B}'}(s'_j)$, what we now prove.

Since the share $s_i \in \mathcal{S}_i$ comes from the linear secret sharing scheme $(\Gamma_{\mathcal{P}}, \mathcal{K}(+), \boldsymbol{\mathcal{S}}, \boldsymbol{\psi})$, (1) is satisfied for each $\mathcal{B} \in \Gamma$. The shares $\hat{s}_{i,j}$ originate from the linear secret sharing scheme $(\Gamma'_{\mathcal{P}'}, \mathcal{S}_i(+), \widehat{\boldsymbol{\mathcal{S}}}_i, \widehat{\boldsymbol{\psi}}_i)$. So, for any $\mathcal{B}' \in \Gamma'$ we have $s_i = \sum_{j \in \mathcal{B}'} \widehat{\psi}_{i,(j,\mathcal{B}')}(\hat{s}_{i,j})$. Substituting the last formula into (1) we obtain:

$$\begin{aligned}
k &= \sum_{i \in \mathcal{B}} \psi_{i,\mathcal{B}} \left(\sum_{j \in \mathcal{B}'} \widehat{\psi}_{i,(j,\mathcal{B}')}(\hat{s}_{i,j}) \right) = \sum_{i \in \mathcal{B}} \sum_{j \in \mathcal{B}'} \psi_{i,\mathcal{B}} \left(\widehat{\psi}_{i,(j,\mathcal{B}')}(\hat{s}_{i,j}) \right) && (\psi_{i,\mathcal{B}} \text{ is a homomorphism}) \\
&= \sum_{j \in \mathcal{B}'} \sum_{i \in \mathcal{B}} \psi_{i,\mathcal{B}} \left(\widehat{\psi}_{i,(j,\mathcal{B}')}(\hat{s}_{i,j}) \right) && (K(+) \text{ is an Abelian group}) \\
&= \sum_{j \in \mathcal{B}'} \sum_{i \in \mathcal{B}} \psi'_{j,\mathcal{B}'} \left(\widehat{\psi}'_{j,(i,\mathcal{B})}(\hat{s}_{i,j}) \right) && (\psi_{i,\mathcal{C}} \text{ and } \widehat{\psi}_{i,(j,\mathcal{C}')} \text{ pseudo-commute}) \\
&= \sum_{j \in \mathcal{B}'} \psi'_{j,\mathcal{B}'} \left(\sum_{i \in \mathcal{B}} \widehat{\psi}'_{j,(i,\mathcal{B})}(\hat{s}_{i,j}) \right) && (\psi'_{j,\mathcal{B}'} \text{ is a homomorphism}) \\
&= \sum_{j \in \mathcal{B}'} \psi'_{j,\mathcal{B}'}(s'_j) && (\text{accordingly to the definition of } s'_j \text{ in (6)})
\end{aligned}$$

□

An important observation is that due to Step 1, $\hat{s}_{i,j}$ is not only a share of s_i for $(\Gamma'_{\mathcal{P}'}, \mathcal{S}_i(+), \widehat{\boldsymbol{\mathcal{S}}}_i, \widehat{\boldsymbol{\psi}}_i)$, but that (6) implies that $\hat{s}_{i,j}$ is also a share of s'_j using the access structure Γ and participants $\mathcal{P}$.

We discuss the security (*e.g.*, perfectness) in Section 4.2. First we give examples that satisfy the conditions in Theorem 1.

4.1 Class of examples

Theorem 2 *If $K(+)$ is an Abelian group, the t-out-of-l threshold schemes in [34, 15, 13, 4] and Benaloh-Leichter's [2] general secret sharing scheme, generalized to any Abelian group $\mathcal{K}(+)$ in [15, p. 675] satisfy the conditions in Theorem 1.*

Proof.[3] We verify that the conditions of Theorem 1 are satisfied. Condition 1 is trivially satisfied by using $(\Gamma_{\mathcal{P}}, \mathcal{K}(+), \boldsymbol{\mathcal{S}}, \boldsymbol{\psi}) \cong (\Gamma_{\mathcal{P}}, \mathcal{K}(+), \mathbf{a}, \boldsymbol{\omega})$ as sharing scheme with linear subshares, using an appropriate (depending on Γ) scheme from the cited ones. When k is the secret key, we let the resulting shares be s_i with subshares $(k_{(0)}, k_{(1)}, \ldots, k_{(a_i-1)})^T_i$, where T is the matrix[4] transpose.

We now construct the sharing scheme with linear subshares $(\Gamma'_{\mathcal{P}'}, \mathcal{S}_i(+), \widehat{\boldsymbol{\mathcal{S}}}_i, \widehat{\boldsymbol{\psi}}_i) \cong (\Gamma'_{\mathcal{P}'}, \mathcal{S}_i(+), \widehat{\mathbf{a}}_i, \widehat{\boldsymbol{\omega}}_i)$, where $\mathcal{S}_i(+) = \mathcal{K}^{a_i}(+)$ and $i \in \mathcal{P}$, in such a way that Condition 4 will be satisfied. We use as a building block $(\Gamma'_{\mathcal{P}'}, \mathcal{K}(+), \boldsymbol{\mathcal{S}}', \boldsymbol{\psi}') \cong (\Gamma'_{\mathcal{P}'}, \mathcal{K}(+), \mathbf{a}', \boldsymbol{\omega}')$ an appropriate aforementioned sharing scheme with linear subshares to construct the share $\hat{s}_{i,j} \in \widehat{\mathcal{S}}_{i,j}$, as we now explain. For each h $(0 \leq h \leq a_i - 1)$, participant $i \in \mathcal{P}$ uses $(\Gamma'_{\mathcal{P}'}, \mathcal{K}(+), \mathbf{a}', \boldsymbol{\omega}')$ to give shares of the "key" $k_{(h)} = k_{i,(h)} \in K$, where $k_{i,(h)}$ is a subshare of s_i. We let the j^{th} $(j \in \mathcal{P}')$ share of the "key" $k_{i,(h)}$ be $(k_{i,j,(h,0)}, k_{i,j,(h,1)}, \ldots, k_{i,j,(h,a'_j-1)})$, denoted as $(k_{(h,0)}, k_{(h,1)}, \ldots, k_{(h,a'_j-1)})_{i,j}$ when there is no ambiguity. Since this is done for each h $(0 \leq h \leq a_i - 1)$, we obtain a two-dimensional array of subshares, which we call the share $\hat{s}_{i,j}$. So,

$$\begin{aligned}
\hat{s}_{i,j} &= \begin{pmatrix} k_{(0,0)} & k_{(0,1)} & \cdots & k_{(0,a'_j-1)} \\ k_{(1,0)} & k_{(1,1)} & \cdots & k_{(1,a'_j-1)} \\ \vdots & \vdots & \ddots & \vdots \\ k_{(a_i-1,0)} & k_{(a_i-1,1)} & \cdots & k_{(a_i-1,a'_j-1)} \end{pmatrix}_{i,j} \\
&= \left(\begin{pmatrix} k_{(0,0)} \\ k_{(1,0)} \\ \vdots \\ k_{(a_i-1,0)} \end{pmatrix} \begin{pmatrix} k_{(0,1)} \\ k_{(1,1)} \\ \vdots \\ k_{(a_i-1,1)} \end{pmatrix} \begin{pmatrix} \cdots \\ \cdots \\ \ddots \\ \cdots \end{pmatrix} \begin{pmatrix} k_{(0,a'_j-1)} \\ k_{(1,a'_j-1)} \\ \vdots \\ k_{(a_i-1,a'_j-1)} \end{pmatrix} \right)_{i,j}
\end{aligned}$$

where $k_{(h,m)} \in K$ $(0 \leq h \leq a_i - 1, 0 \leq m \leq a'_j - 1)$. This defines $\widehat{\mathcal{S}}_{i,j} = \mathcal{S}_i^{a'_j} = (K^{a_i})^{a'_j} = K^{a_i * a'_j}$ and $\hat{s}_{i,j}$ is a typical element of $\widehat{\mathcal{S}}_{i,j}$. So $\widehat{\mathbf{a}}_i = \mathbf{a}'$ (*i.e.*, the tuple of expansions a'_j $(j \in \mathcal{P}')$), as is easy to verify. Finally we define $\widehat{\boldsymbol{\omega}}_i$, by specifying that

$$\widehat{\omega}_{i,(j,\mathcal{C}'),(m)}\left((k_{(0,m)}, \ldots, k_{(a_i-1,m)})^T_{i,j}\right) = \left(\omega'_{j,\mathcal{C}',(m)}(k_{(0,m)}), \ldots, \omega'_{j,\mathcal{C}',(m)}(k_{(a_i-1,m)})\right)^T_{i,j}. \qquad (7)$$

Since $(\Gamma'_{\mathcal{P}'}, \mathcal{S}_i(+), \widehat{\boldsymbol{\mathcal{S}}}_i, \widehat{\boldsymbol{\psi}}_i) \cong (\Gamma'_{\mathcal{P}'}, \mathcal{S}_i(+), \widehat{\mathbf{a}}_i, \widehat{\boldsymbol{\omega}}_i)$, (7) defines $\widehat{\psi}_{i,(j,\mathcal{C}')}$.

Condition 3 is assumed, so nothing needs to be proven. Before explaining why Condition 4 is satisfied, we emphasize that in $(\Gamma'_{\mathcal{P}'}, \mathcal{S}_i(+), \widehat{\mathbf{a}}_i, \widehat{\boldsymbol{\omega}}_i)$ we view $\hat{s}_{i,j}$ as a share of s_i, the *columns* of $\hat{s}_{i,j}$ as subshares, and the scalar action of $\widehat{\psi}_{i,(j,\mathcal{C}')}$ on $\hat{s}_{i,j}$ as the weighted sum of the *columns* in $\hat{s}_{i,j}$. We will essentially prove that, using the access structure Γ and participants $\mathcal{P}$, one can view $\hat{s}_{i,j}$ as a

[3] A *much* shorter proof could have been given using bimodules and tensor product of modules [24]. Since the theory of modules might not be so well known to the reader, we avoided to threat the subject this way.

[4] Very strictly speaking, entries to matrices must belong to a ring. To facilitate the reading we speak loosely of two dimensional arrays as matrices.

share of s'_j in which the *rows* of $\hat{s}_{i,j}$ are the subshares and the $\widehat{\psi}'_{j,(i,\mathcal{C})}(\hat{s}_{i,j})$ is defined as the weighted sum of the *rows* in $\hat{s}_{i,j}$.

So, we define the sharing scheme with linear subshares $(\Gamma_{\mathcal{P}}, \mathcal{S}'_j(+), \widehat{\boldsymbol{\mathcal{S}}'}_j, \widehat{\boldsymbol{\psi}'}_j) \cong (\Gamma_{\mathcal{P}}, \mathcal{S}'_j(+), \widehat{\mathbf{a}}'_j, \widehat{\boldsymbol{\omega}'}_i)$, where $\widehat{\boldsymbol{\mathcal{S}}'}_j = \widehat{\boldsymbol{\mathcal{S}}}_i$ but viewing the rows as subshares, $\widehat{\mathbf{a}}'_j = \mathbf{a}$, and

$$\widehat{\omega}'_{j,(i,\mathcal{C}),(h)}\left((k_{(h,0)}, \ldots, k_{(h,a_i-1)})_{i,j}\right) = \left(\omega_{i,\mathcal{C},(h)}(k_{(h,0)}), \ldots, \omega_{i,\mathcal{C},(h)}(k_{(h,a'_j-1)})\right)_{i,j}. \tag{8}$$

This naturally defines $\widehat{\psi}'_{i,(j,\mathcal{C}')}$.

To prove Condition 4, using (3) and (4) for $(\Gamma'_{\mathcal{P}'}, \mathcal{S}_i(+), \widehat{\mathbf{a}}_i, \widehat{\boldsymbol{\omega}}_i)$ and (7), and dropping indices $i, j, \mathcal{C}, \mathcal{C}'$ in $k_{i,j,(h,m)}, \omega_{i,\mathcal{C},(h)}$ and in $\omega'_{j,\mathcal{C}',(m)}$ when there is no ambiguity, we obtain

$$\begin{aligned}
\psi_{i,\mathcal{C}}\left(\widehat{\psi}_{i,(j,\mathcal{C}')}\left(\hat{s}_{i,j}\right)\right) &= \psi_{i,\mathcal{C}}\left(\left(\sum_{m=0}^{a'_j-1}\omega'_{(m)}(k_{(0,m)}), \sum_{m=0}^{a'_j-1}\omega'_{(m)}(k_{(1,m)}), \ldots, \sum_{m=0}^{a'_j-1}\omega'_{(m)}(k_{(a_i-1,m)})\right)^T_{i,j}\right) \\
&= \sum_{h=0}^{a_i-1}\left(\omega_{(h)} \cdot \sum_{m=0}^{a'_j-1}\left(\omega'_{(m)} \cdot k_{(h,m)}\right)\right) && \text{(due to (4) for } (\Gamma_{\mathcal{P}}, \mathcal{K}(+), \boldsymbol{\mathcal{S}}, \boldsymbol{\psi})) \\
&= \sum_{h=0}^{a_i-1}\sum_{m=0}^{a'_j-1}\left(\omega_{(h)} \cdot \left(\omega'_{(m)} \cdot k_{(h,m)}\right)\right) && (\omega_{(h)} \in R \text{ and } K(+) \text{ is an } R\text{-module}) \\
&= \sum_{m=0}^{a'_j-1}\sum_{h=0}^{a_i-1}\left(\left(\omega_{(h)} \cdot \omega'_{(m)}\right) \cdot k_{(h,m)}\right) && (K(+) \text{ is an } R\text{-module}) \\
&= \sum_{m=0}^{a'_j-1}\sum_{h=0}^{a_i-1}\left(\left(\omega'_{(m)} \cdot \omega_{(h)}\right) \cdot k_{(h,m)}\right) && (R \text{ is commutative}) \\
&= \sum_{m=0}^{a'_j-1}\left(\omega'_{(m)} \cdot \sum_{h=0}^{a_i-1}\left(\omega_{(h)} \cdot k_{(h,m)}\right)\right) && (\omega'_{(m)} \in R \text{ and } \mathcal{K} \text{ is an } R\text{-module}) \\
&= \psi'_{j,\mathcal{C}'}\left(\left(\sum_{h=0}^{a_i-1}\omega_{(h)} \cdot k_{(h,0)}, \sum_{h=0}^{a_i-1}\omega_{(h)} \cdot k_{(h,1)}, \ldots, \sum_{h=0}^{a_i-1}\omega_{(h)} \cdot k_{(h,a'_j-1)}\right)_{i,j}\right) \\
&= \psi'_{j,\mathcal{C}'}\left(\widehat{\psi}'_{j,(i,\mathcal{C})}\left(\hat{s}_{i,j}\right)\right).
\end{aligned}$$

The last equation is due to (8), and the last but one is due to (3) and (4) for $(\Gamma'_{\mathcal{P}'}, \mathcal{K}(+), \boldsymbol{\mathcal{S}}',$ $\boldsymbol{\psi}') \cong (\Gamma'_{\mathcal{P}'}, \mathcal{K}(+), \mathbf{a}', \boldsymbol{\omega}')$. □

The attentive reader will have observed that the proof technique used in this proof is very similar to the proof of Theorem 1. Here, the fact that R is commutative basically replaces, in the proof of Theorem 1, the fact that $\psi_{i,\mathcal{C}}$ and $\widehat{\psi}_{i,(j,\mathcal{C}')}$ pseudo-commute.

4.2 Security

The following corollaries of Theorem 1 are easy to prove. Due to space limitation we state these without formal details and we will defer the proofs to the final paper.

Corollary 1 *If the linear sharing schemes $(\Gamma'_{\mathcal{P}'}, \mathcal{S}_i(+), \widehat{\boldsymbol{\mathcal{S}}}_i, \widehat{\psi}_i)$ are zero-knowledge, then the joint view of (informally, information received by) the participants in $\tilde{\mathcal{B}}' \notin \Gamma'$ in the redistribution protocol of Section 4, can be simulated. This implies that the participants in $\tilde{\mathcal{B}}' \notin \Gamma'$ have no information about k.*

This corollary can be extended to address the information theoretical aspects.

Corollary 2 *If the linear sharing schemes* $(\Gamma_{\mathcal{P}}, \mathcal{K}(+), \boldsymbol{\mathcal{S}}, \boldsymbol{\psi})$ *and all* $(\Gamma'_{\mathcal{P}'}, \mathcal{S}_i(+), \widehat{\boldsymbol{\mathcal{S}}}_i, \widehat{\boldsymbol{\psi}}_i)$ *are minimum knowledge, then the joint view of (informally, information received by) the participants in* $\mathcal{B}' \in \Gamma'$ *in the redistribution algorithm of Section 4, can be simulated. This implies that the participants in* $\mathcal{B}' \in \Gamma'$ *obtain no new information, besides* k.

We remind the reader that the dual of a monotone function f is obtained by interchanging the logical AND and OR in f. Since a monotone access structure Γ corresponds to a monotone function f, we call the access structure corresponding to the dual of f the dual of Γ, and denote it as $\overline{\Gamma}$. Observe that if Γ corresponds to a t-out-of-l access structure, then $\overline{\Gamma}$ corresponds to a $(l-t+1)$-out-of-l threshold.

Corollary 3 *When the linear sharing schemes* $(\Gamma_{\mathcal{P}}, \mathcal{K}(+), \boldsymbol{\mathcal{S}}, \boldsymbol{\psi})$ *and* $(\Gamma'_{\mathcal{P}'}, \mathcal{S}_i(+), \widehat{\boldsymbol{\mathcal{S}}}_i, \widehat{\boldsymbol{\psi}}_i)$ *are zero-knowledge, it is sufficient that all* i *in a set* $\overline{\mathcal{B}} \in \overline{\Gamma}$ *erase* s_i *and* $\hat{s}_{i,j}$ *(for all* j *in a set* $\overline{\mathcal{B}}' \in \overline{\Gamma}'$*), to guarantee that no subset* $\mathcal{B} \subseteq (\mathcal{P} \setminus \mathcal{P}')$ *can recover the secret* k.

5 Threshold cryptography variant

It is rather straightforward to use above redistribution algorithm in the context of threshold cryptography, when the Abelian group $\mathcal{K}$ to which the secret belongs, is public. This is for example the case in a discrete log setting, as in ElGamal [14, 31] and DSS [28, 18]. However, in the case of RSA [33] the participants do not know the group $\mathcal{K}$, being $Z_{\phi(n)}(+)$, as mentioned in [12]. Therefore, we focus on such a scenario.

5.1 RSA scenario

In the RSA scenario the secret key corresponds to $d \in Z^*_{\phi(n)} \subset Z_{\phi(n)}(+)$. In [12] the distributor of the shares knows $\phi(n)$. However, in Step 1 of the redistribution protocol the participant $i \in \mathcal{B}$ gives shares of s_i but does not know $\phi(n)$, and should not know it. Therefore, the distribution algorithm described in [12] does not work.

If we can present a sharing scheme with linear subshares for the case the secret (*i.e.*, the subshare of d) is in $Z_{\phi(n)}$ but the distributor does not know $\phi(n)$, then due Theorem 2, we are able to redistribute shares. Therefore, from now on in this section, we focus solely on this problem without rediscussing the context.

As in [17], we view the subshares of the secret d as integers. We observe that *from an algebraic viewpoint* the sharing schemes with linear subshares in [15, 13, 4, 2] also work when $K = Z$, the integers. But, as proven in [7], the sharing scheme cannot be perfect. However, in our context an upperbound on the secret key d is known, namely the RSA public modulus n. Also if the shares of d came from a distributor knowing $\phi(n)$, the subshares will be less than n. This allows us to present a solution.

5.2 Sharing a positive integer with known upperbound

We use as a primitive a $(\Gamma_{\mathcal{P}}, \mathcal{K}(+), \mathbf{a}, \boldsymbol{\omega})$, satisfying the conditions in Lemma 1.

Lemma 1 *Let* $(\Gamma_{\mathcal{P}}, \mathcal{K}(+), \mathbf{a}, \boldsymbol{\omega})$ *be a secret sharing scheme with linear subshares in which* $R = Z$*. If:*

1. *the values of* $\mathbf{a}$ *and* $\boldsymbol{\omega}$ *are independent of* $\mathcal{K}$,
2. *for any finite Abelian group* $\mathcal{K}$ *(1)–(4) are valid, and* $(\Gamma_{\mathcal{P}}, \mathcal{K}(+), \mathbf{a}, \boldsymbol{\omega})$ *is a secret sharing scheme,*

then for all $i \in \mathcal{P}$ *and all* h $(0 \leq h \leq a_i - 1)$ *the subshare* $k_{i,(h)}$ *is equal to:*

$$k_{i,(h)} = u'_{i,(h)} \cdot k + \sum_{j=1}^{b} u''_{i,(h),j} \cdot r_j \tag{9}$$

where $k \in K$ is the secret key, $r_j \in K$, and $u'_{i,(h)}, u''_{i,(h),j}$ are integers which are independent of k and r_j. Also, if the binary length of $\omega_{i,\mathcal{C},(h)}$ and the value of $a_\Sigma = \sum_{i \in \mathcal{P}} a_i$, the total number of subshares, are polynomially bounded in $l = |\mathcal{P}|$, then the lengths of $u'_{i,(h)}, u''_{i,(h),j}$ are also polynomially bounded[5] *in l.*

Proof. Rather straightforward, see Appendix B. □

We now explain how to give shares of a secret k using the primitive in which we change the distribution of the shares. Let $p(|n|)$ be an appropriate polynomial in the length of n. We give concrete values for $p(|n|)$ in the final paper. When $t = 2$ the scheme in [13] allows that $p(|n|)$ is as low as $p(|n|) = 2$ (asymptotically $1 + c'$, where $c' > 0$ is sufficient).

Share distribution algorithm

Assume k is the secret and it can be viewed as an integer in the interval $[0, n-1]$.

Step 1 For each j $(1 \leq j \leq b)$, choose in the interval $[0, n^{p|n|} - 1]$ an integer r_j uniformly random.

Step 2 Give each shareholder $i \in \mathcal{P}$ subshares $k_{i,(h)}$ defined by k and the chosen r_j using (9).

It is rather straightforward to see that if the conditions in Lemma 1 are satisfied we obtain a secret sharing scheme for a key in the interval $[0, n-1]$. We now need to address the security of this scheme.

Theorem 3 *Let $(\Gamma_{\mathcal{P}}, \mathcal{K}(+), \mathbf{a}, \boldsymbol{\omega})$ satisfy the conditions in Lemma 1. If the number of shareholders, l, is bounded by a polynomial in $|n|$, the binary length of $\omega_{i,\mathcal{C},(h)}$ and the value a_Σ are polynomially bounded in l, and $(\Gamma_{\mathcal{P}}, \mathcal{K}(+), \mathbf{a}, \boldsymbol{\omega})$ is perfect for any finite Abelian group $\mathcal{K}$, then the shares obtained in the above share distribution algorithm are statistically zero-knowledge (i.e., they are close to perfect).*

Proof. See Appendix C. □

The zero-knowledge t-out-of-l threshold schemes in [15, 13, 4] satisfy these conditions. For some access structures, Benaloh-Leichter's [2] general secret sharing scheme, generalized to any Abelian group $\mathcal{K}(+)$ in [15, p. 675] also satisfies.

Above can easily be generalized to any bounded interval and to a more general setting than strictly RSA.

Note

Observe that each time the redistribution protocol is used, the size of the shares grows. Indeed, the first time the subshares are $|n|$ bits long, but the subshares in $s_{i,j}$ are at least $p(|n|)$ times longer. This implies that the subshares in s'_j will also be at least $p(|n|))$ times longer. So, if the algorithm is used the next time, the upperbound n can no longer be used, so the bound n must be replaced by the higher value, making the new shares even larger.

If the set $\mathcal{K}$ is known, as in DSS or ElGamal, the shares do not have such a memory effect. One can wonder whether it is possible to avoid this memory effect in an RSA context.

6 Robust variant

Due to the linear nature, our results easily extend to robust threshold cryptography which is discrete log based. In several verifiable secret sharing schemes (*e.g.*, [31, 32]) and their application to robust threshold cryptography based on discrete log [18], what is being communicated to verify the share is $f(s_i)$, where f is a *homomorphism* from S_i to an appropriate group, in general f is applied on each subshare. This implies that we can talk about a more general share consisting of the secret s_i

[5] Even if $|\Gamma|$ is superpolynomial in l, b independent rows will exist in W, implying a polynomial time description of the form (9).

and the public $f(s_i)$, *i.e.*, regard the share as $(s_i, f(s_i))$. Due to the linearity of our schemes and the fact that f is a homomorphism, all formulas apply. This implies that the shareholders in the redistribution algorithm can give verifiable shares $\hat{s}_{i,j}$ of the verifiable shares s_i. Since s'_j is a linear combination of $\hat{s}_{i,j}$, this results in verifiable shares s'_j for the new access structure Γ', using [32, p. 137]. In the limited context of verifiable secret sharing, the results are not restricted to a discrete log setting.

We also observe that the interactive method to achieve robust threshold RSA in [17] continues to work after redistribution.

7 Application to secure databases

In distributed systems, there is often a need to enforce mutual exclusion which requires that at any given time there is at most one group that can execute some critical operation. One method to achieve mutual exclusion is *static* voting [3] which assigns a vote[6] to each node in the system; a node can perform the critical operation if it can acquire a majority of votes from the nodes in the distributed system. A drawback of this scheme is that failures can occur in such a way that no node can perform the critical operation until these failures are repaired. A popular generalization of the static voting scheme is *dynamic* voting [25] which reduces the chances of system halting in this fashion. Unlike static voting, dynamic voting adjusts the necessary quorum each time the critical operation is performed. The requirements of dynamic voting are identical to those considered by us in this paper. Whenever a group of nodes decides to change its votes, the votes are adjusted such that nodes outside the group can no longer form a majority.

Acknowledgements

The first author thanks Gus Simmons for having mentioned the problem of redistribution of shares of a secret at an Oberwolfach workshop in the early 1990's and Shafi Goldwasser for a discussion on statistically zero-knowledge. The authors also thank the Newton Institute on Computer Security, Cryptology and Coding for its support.

Appendices

A Specific secret sharing schemes

We very briefly survey some of the secret sharing schemes mentioned in Section 3 and explain how they fit in the general description which we use in this text.

In Shamir t-out-of-l threshold scheme [34] $\mathcal{K}(+)$ corresponds to the additive group of a finite field $GF(q)$, $a_i = 1$ when $q \geq l+1$, and $\psi_{i,\mathcal{B}}$ is equal to the Lagrange coefficients

$$\psi_{i,\mathcal{B}} = \prod_{\substack{j\in\mathcal{B} \\ j\neq i}} \frac{0 - x_j}{x_i - x_j}. \tag{10}$$

In its generalization [15] to any zero-knowledge homomorphic sharing scheme in which $\mathcal{K}(+)$ is an Abelian group, $a_i = a \geq l$, where $a+1$ is a prime. The homomorphisms $\psi_{i,\mathcal{B}} = F_0 \circ \bar{\psi}_{i,\mathcal{B}}$, where $\bar{\psi}_{i,\mathcal{B}}$ is similar as in (10) and F_0 maps $(k_0, k_1, \ldots, k_{a-1})$ into $k_0 \in \mathcal{K}$ (for details consult [15]).

In Benaloh-Leichter's general secret sharing scheme (which can be generalized to any key space $\mathcal{K}(+)$ as observed in [15, p. 675]) and the threshold schemes in [13] a secret share corresponds to several subshares and a_i is an appropriate integer. In these schemes $\psi_{i,\mathcal{B}}$ corresponds to selecting one subshare. So $\psi_{i,\mathcal{B}}$ maps $(k_{i,(0)}, k_{i,(1)}, \ldots, k_{i,(a_i-1)})$ into $k_{i,(f_i(\mathcal{B}))}$, where f_i is a function from $\{\mathcal{C} \mid i \in C \in \Gamma\}$ to Z_{a_i}. In other words $\psi_{i,\mathcal{B}} = (\omega_{(0)}, \omega_{(1)}, \ldots, \omega_{(a_i-1)})_{i,\mathcal{B}}$ where all $\omega_{i,\mathcal{B},(h)}$ are zero, except one which has the value one.

[6] It is possible to assign different weights to nodes.

From an algebraic viewpoint, the $\psi_{i,\mathcal{B}}$ in the secret sharing scheme in [4] are a combination of these in [15] and [13].

B Proof of Lemma 1

All subshares for all participants in a $(\Gamma_{\mathcal{P}}, \mathcal{K}(+), \mathbf{a}, \boldsymbol{\omega})$ scheme can be viewed as one large column of σ_j where $1 \le j \le a_\Sigma$, $a_\Sigma = \sum_{i\in\mathcal{P}} a_i$, so that if we rename $\mathcal{P} = \{1, \ldots, l\}$, we have

$$\sigma_j = k_{i,(h)}, \qquad \text{where } j = h + 1 + \sum_{m=1}^{i} a_m. \tag{11}$$

Then Equations (1)–(4) and (11) imply that

$$\begin{pmatrix} k \\ k \\ \vdots \\ k \end{pmatrix} = W \cdot \begin{pmatrix} \sigma_1 \\ \vdots \\ \sigma_{a_\Sigma} \end{pmatrix}, \quad \text{where } W_{|\Gamma|\times a_\Sigma} = (w_{\mathcal{B},j}) \text{ such that } w_{\mathcal{B},j} = \omega_{i,\mathcal{B},(h)} \text{ and } j = h+1+\sum_{m=1}^{i} a_m, \tag{12}$$

and $\mathcal{B} \in \Gamma$.

We assume that the first row in W corresponds to the set $\mathcal{B}_1$ and that $\Gamma = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_{|\Gamma|}\}$. If $R = Z$, *i.e.*, $w_{\mathcal{B},j} \in Z$, then $VWU = W'$, where W' is Smith's normal form of W [21, pp. 384-387] and $V = (v_{i',j})$ and $U = (u_{i',j})$ are invertible in the matrix rings $M_{|\Gamma|}(Z)$ and $M_{a_\Sigma}(Z)$, respectively. So, W' is a diagonal matrix, with diagonal elements the invariant factors $(d_1, d_1 \cdot d_2, d_1 \cdot d_2 \cdots d_m, 0 \ldots, 0)$, where $d_{i'} > 0$. We first prove that $(d_1 \cdots d_{i'}) \mid \sum_{j=1}^{|\Gamma|} v_{i',j}$. Similarly as in [21, pp. 387-389], one obtains:

$$\begin{pmatrix} \sum_{j=1}^{|\Gamma|} v_{1,j}\cdot k \\ \sum_{j=1}^{|\Gamma|} v_{2,j}\cdot k \\ \vdots \\ \sum_{j=1}^{|\Gamma|} v_{m,j}\cdot k \\ \sum_{j=1}^{|\Gamma|} v_{m+1,j}\cdot k \\ \vdots \\ \sum_{j=1}^{|\Gamma|} v_{|\Gamma|,j}\cdot k \end{pmatrix} = \begin{pmatrix} d_1 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ 0 & d_1\cdot d_2 & \ldots & 0 & 0 & \ldots & 0 \\ \ldots & \ldots & \ddots & \ldots & \ldots & \ddots & \ldots \\ 0 & 0 & \ldots & d_1\cdots d_m & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ \ldots & \ldots & \ddots & \ldots & \ldots & \ddots & \ldots \\ 0 & 0 & \ldots & 0 & 0 & \ldots & 0 \end{pmatrix} \cdot U^{-1} \cdot \begin{pmatrix} \sigma_1 \\ \vdots \\ \sigma_{a_\Sigma} \end{pmatrix}. \tag{13}$$

Since, $(\Gamma_{\mathcal{P}}, \mathcal{K}(+), \mathbf{a}, \boldsymbol{\omega})$ is a secret sharing scheme for any finite Abelian group $\mathcal{K}$, one can take $K = Z_{d_1 \cdots d_{i'}}(+)$. Accordingly to the right hand side the row i' in (13) gives $\sum_{j=1}^{|\Gamma|} v_{i',j} \cdot k = 0$, regardless of k. Indeed, $(\Gamma_{\mathcal{P}}, \mathcal{K}(+), \mathbf{a}, \boldsymbol{\omega})$ is a secret sharing scheme, so for all $k \in K$ we have $\sum_{j=1}^{|\Gamma|} v_{i',j} \cdot k = 0$, implying $d_1 \cdots d_{i'} \mid \sum_{j=1}^{|\Gamma|} v_{i',j}$.

The rest follows easily solving the equation (13) in the unknowns $\sigma_{i'}$ from [21, pp. 387-389], using the transpose. Defining $b = a_\Sigma - m$, this gives:

$$\sigma_{i'} = \left(\sum_{j=1}^{m} \left(u_{i',j} \frac{\sum_{h=1}^{|\Gamma|} v_{j,h}}{\prod_{h=1}^{j} d_h} \right) \right) \cdot k + \sum_{j=1}^{b} u_{i',m+j} \cdot r_j \tag{14}$$

where r_j can be any element in K. So, we obtain (9) in which $u''_{i,(h),j}$ is clearly an integer. $u'_{i,(h)}$ is also an integer, since the fractions in (14) are integers. The size of the integers follows from [26]. □

C Proof of Theorem 3

We first prove a technical lemma, which can be skipped in a first reading. Let $\{D_1(x)\}$ and $\{D_2(x)\}$ be two families of random variables. When applying the same (deterministic) function on these random variables, we denote the obtained families of random variables $\{f(D_1)(x)\}$ and $\{f(D_2)(x)\}$ respectively.

Lemma 2 *If the families of random variables $\{D_1(x)\}$ and $\{D_2(x)\}$ are statistically indistinguishable [19] (in function of x), then $\{f(D_1)(x)\}$ and $f(D_2)(x)$ are also statistically indistinguishable.*

Proof. Statistically indistinguishable means that

$$\sum_{\alpha\in\{0,1\}^*} |\mathrm{prob}(D_1(x)=\alpha)-\mathrm{prob}(D_2(x)=\alpha)| \leq \varepsilon(|x|), \tag{15}$$

where $\varepsilon(|x|)$ is a negligible function [19], *e.g.*, $2^{-|x|}$. Now

$$\begin{aligned}
&\sum_{\beta\in\{0,1\}^*} |\mathrm{prob}(f(D_1(x))=\beta)-\mathrm{prob}(f(D_2(x))=\beta)| \\
&\quad= \sum_{\beta\in\{0,1\}^*} \left| \sum_{\substack{\alpha \\ f(\alpha)=\beta}} (\mathrm{prob}(D_1(x)=\alpha)-\mathrm{prob}(D_2(x)=\alpha)) \right| && \text{(theorem of total probability)} \\
&\quad\leq \sum_{\beta\in\{0,1\}^*} \sum_{\substack{\alpha \\ f(\alpha)=\beta}} |\mathrm{prob}(D_1(x)=\alpha)-\mathrm{prob}(D_2(x)=\alpha)| && (|a+b|\leq|a|+|b|) \\
&\quad\leq \sum_{\alpha\in\{0,1\}^*} |\mathrm{prob}(D_1(x)=\alpha)-\mathrm{prob}(D_2(x)=\alpha)| \quad\leq\quad \varepsilon(|x|) && \text{(due to (15))}.
\end{aligned}$$

□

Corollary 4 *When the family of random variables $\{D(x)\}$ is statistically zero-knowledge and f is a function that can be computed in expected polynomial time (in function of $|x|$), then the family of distributions $\{f(D)(x)\}$ is statistically zero-knowledge.*

Proof. $\{D(x)\}$ being statistically zero-knowledge, implies that there exists an expected polynomial time simulator that can generate an $\{D'(x)\}$ which is statistically indistinguishable from $D(x)$. Then to simulate $\{f(D)(x)\}$, use the simulator and apply f on the output of the simulator. □

Proof of Theorem 3

We denote the column $(r_1,\ldots,r_b)$ as $\bar{r}$. We now organize all the subshares of the participants in a set $\mathcal{B}$ as a long column and call it $\bar{\sigma}_{\mathcal{B}}$. Using a similar ordering, we organize all the corresponding integers $u'_{i,(h)}$ (see (9)) where $i\in\mathcal{B}$ in a column and call it $\bar{u}'_{\mathcal{B}}$. For each j $(1\leq j\leq b)$, we proceed in the same way with the integers $u''_{i,(h),j}$ (see (9)) to form a column $\bar{u}''_{j,\mathcal{B}}$. We define $U''_{\mathcal{B}}$ to be the matrix with columns $\bar{u}''_{j,\mathcal{B}}$ $(1\leq j\leq b)$ and $U'_{\mathcal{B}}$ have the same columns as $U''_{\mathcal{B}}$ but augmented by the column $\bar{u}'_{\mathcal{B}}$. So, (9) gives:

$$\bar{\sigma}_{\mathcal{B}} = \bar{u}'_{\mathcal{B}}\cdot k + U''_{\mathcal{B}}\cdot\bar{r}. \tag{16}$$

We first prove that, *if* $\tilde{\mathcal{B}}\notin\Gamma$ and if the conditions in Lemma 1 are satisfied and if $(\Gamma_{\mathcal{P}},\mathcal{K}(+),\mathbf{a},\boldsymbol{\omega})$ is perfect for any finite Abelian group $\mathcal{K}$, *then* $U''_{\tilde{\mathcal{B}}}$ and $U'_{\tilde{\mathcal{B}}}$ have the same invariant factors. The prove is similar as the one in Appendix B. Indeed, there exist invertible matrices (over Z) V_1 and V_2 such that $V_1U''_{\tilde{\mathcal{B}}}V_2 = U'''_{\tilde{\mathcal{B}}}$, where $U'''_{\tilde{\mathcal{B}}}$ is the Smith normal form of $U''_{\tilde{\mathcal{B}}}$, with diagonal elements the invariant factors $(d_1, d_1\cdot d_2, d_1\cdot d_2\cdots d_m, 0\ldots,0)$, where $d_{i'}>0$. Then (16) can be transformed into:

$$V_1\cdot\bar{\sigma}_{\tilde{\mathcal{B}}} = V_1\cdot\bar{u}'_{\tilde{\mathcal{B}}}\cdot k + U'''_{\tilde{\mathcal{B}}}\cdot V_2^{-1}\cdot\bar{r}. \tag{17}$$

Let $K(+) = Z_{d_1}$, then the first element in $V_1 \cdot \bar{u}'_{\tilde{\mathcal{B}}}$ must be divisible by d_1, otherwise the scheme is clearly not perfect. Similar arguments are valid for the next elements of the column $V_1 \cdot \bar{u}'_{\tilde{\mathcal{B}}}$. This easily implies that $V_1 \cdot \bar{u}'_{\tilde{\mathcal{B}}} = U'''_{\tilde{\mathcal{B}}} \cdot \bar{z}_{\tilde{\mathcal{B}}}$, where $\bar{z}_{\tilde{\mathcal{B}}}$ is a column of integers. Now it is straightforward to see that $U''_{\tilde{\mathcal{B}}}$ and $U'_{\tilde{\mathcal{B}}}$ have the same invariant factors, using a similar argument as in [21, pp. 387-389].

Now, there exists an $\bar{z}'_{\tilde{\mathcal{B}}}$, a column of integers, such that $\bar{z}_{\tilde{\mathcal{B}}} = V_2^{-1} \cdot \bar{z}'_{\tilde{\mathcal{B}}}$. So, (17) can be rewritten as:

$$\begin{aligned} V_1 \cdot \bar{\sigma}_{\tilde{\mathcal{B}}} &= U'''_{\tilde{\mathcal{B}}} \cdot V_2^{-1} \cdot \left(\bar{r} + \bar{z}'_{\tilde{\mathcal{B}}} \cdot k\right), \qquad \text{or} \\ \bar{\sigma}_{\tilde{\mathcal{B}}} &= U''_{\tilde{\mathcal{B}}} \cdot \left(\bar{r} + \bar{z}'_{\tilde{\mathcal{B}}} \cdot k\right). \end{aligned} \tag{18}$$

If $K = Z$ and we do not bound the choice of $\bar{r}$, it is possible to obtain the same tuple of subshares $\bar{\sigma}_{\tilde{\mathcal{B}}}$, regardless of the value of k. We are now ready to discuss the simulator when each integer r_j is chosen in the interval $[0, n^{p|n|} - 1]$, *i.e.*, prove that $\bar{\sigma}_{\tilde{\mathcal{B}}}$ can be simulated in expected polynomial time and this for each $\tilde{\mathcal{B}} \notin \Gamma$. Note that, due to our conditions in the theorem, the length of the integers in $U''_{\tilde{\mathcal{B}}}$ are polynomially bounded in $|n|$ (see Lemma 1). So, $U''_{\tilde{\mathcal{B}}}$ is a polynomial time function. This implies, due to Corollary 4, that, to prove the theorem it is sufficient to prove that: $\bar{\rho} = (\bar{r} + \bar{z}'_{\tilde{\mathcal{B}}} \cdot k)$ can be simulated by a simulator who does not know k (*i.e.*, the same simulator is used regardless of k). We now prove that this sufficient condition is satisfied.

The simulator will choose ρ'_j $(1 \leq j \leq b)$ as an integer chosen in the interval $[0, n^{p|n|} - 1]$ and return $\bar{\rho}' = (\rho'_1, \ldots, \rho'_b)^T$ (as a simulation for $\bar{\rho}$).

Let $D(n)$ be the actual distribution of $\bar{\rho}$ and $D'(n)$ be the distribution of the simulated $\bar{\rho}'$. We prove that these are statistically indistinguishable. Informally, we demonstrate that most of the time $\bar{\rho}' = \bar{\rho}$. The set of possible $\bar{\rho}$ and $\bar{\rho}'$ are generalized cubes $\mathcal{R}$ and $\mathcal{R}'$. Crucial to the proof is that, if $\bar{\rho}$ is not in $\mathcal{R}'$ then there is at least one ρ_j for which it is *not* true that $0 \leq \rho_j \leq n^{p|n|} - 1$. Similar, if $\bar{\rho}'$ is not in $\mathcal{R}$ then there is at least one ρ'_j for which it is *not* true that $(z'_{\tilde{\mathcal{B}},j} \cdot k) \leq \rho'_j \leq (n^{p|n|} - 1) + (z'_{\tilde{\mathcal{B}},j} \cdot k)$. So, if $z'_{\tilde{\mathcal{B}},j} \cdot k$ is positive and for one j: $0 \leq \rho_j < z'_{\tilde{\mathcal{B}},j}$ or $n^{p|n|} \leq \rho_j \leq (n^{p|n|} - 1) + (z'_{\tilde{\mathcal{B}},j} \cdot k)$ then $\bar{\rho}$ does not belong to $\mathcal{R} \cap \mathcal{R}'$. We have a similar condition when $z'_{\tilde{\mathcal{B}},j} \cdot k$ is negative. Note that, due to our conditions and [26], the binary length of the integers in $\bar{z}'_{\tilde{\mathcal{B}}}$ is polynomial in $|n|$, let say less than $|n|^{c_1}$, so $z'_{\tilde{\mathcal{B}},j} \leq n^{|n|^{c_1 - 1}}$. We then obtain:

$$\begin{aligned} \sum_{\alpha \in \{0,1\}^*} |\text{prob}(D_1(x) = \alpha) - \text{prob}(D_2(x) = \alpha)| &\leq \sum_{i=1}^{b} \binom{b}{i} \frac{\left(n^{|n|^{c_1-1}+1}\right)^i \left(n^{p(|n|)}\right)^{b-i}}{\left(n^{p(|n|)}\right)^b} \\ &\leq \sum_{i=1}^{b} \binom{b}{i} \left(\frac{n^{|n|^{c_1-1}+1}}{n^{p(|n|)}}\right)^i \\ &= -1 + \left(1 + \frac{n^{|n|^{c_1-1}+1}}{n^{p(|n|)}}\right)^b \end{aligned}$$

which is negligible when $p(|n|) \geq |n|^{c_1 - 1} + 2$ (asymptotically).

□

References

[1] J. C. Benaloh. Secret sharing homomorphisms: Keeping shares of a secret secret. In A. Odlyzko, editor, *Advances in Cryptology, Proc. of Crypto '86 (Lecture Notes in Computer Science 263)*, pp. 251–260. Springer-Verlag, 1987. Santa Barbara, California, U.S.A., August 11–15.

[2] J. C. Benaloh and J. Leichter. Generalized secret sharing and monotone functions. In S. Goldwasser, editor, *Advances in Cryptology, Proc. of Crypto'88 (Lecture Notes in Computer Science 403)*, pp. 27–35. Springer-Verlag, 1990. Santa Barbara, California, U.S.A., August 11–15.

[3] P. A. Bernstein, V. Hadzilacos, and N. Goodman. *Concurrency Control and Recovery in Database Systems.* Addison-Wesley, Reading, MA, 1987.

[4] S. R. Blackburn, M. Burmester, Y. Desmedt, and P. R. Wild. Efficient multiplicative sharing schemes. In U. Maurer, editor, *Advances in Cryptology — Eurocrypt '96, Proceedings (Lecture Notes in Computer Science 1070)*, pp. 107–118. Springer-Verlag, 1996. Zaragoza, Spain, May 12–16.

[5] B. Blakley, G. R. Blakley, A. H. Chan, and J. Massey. Threshold schemes with disenrollment. In E. F. Brickell, editor, *Advances in Cryptology — Crypto '92, Proceedings (Lecture Notes in Computer Science 740)*, pp. 540–548. Springer-Verlag, 1993. Santa Barbara, California, U.S.A., August 16–20.

[6] G. R. Blakley. Safeguarding cryptographic keys. In *Proc. Nat. Computer Conf. AFIPS Conf. Proc.*, pp. 313–317, 1979. vol.48.

[7] G. R. Blakley and L. Swanson. Infinite structures in information theory. In D. Chaum, R.L. Rivest, and A. T. Sherman, editors, *Advances in Cryptology. Proc. of Crypto'82*, pp. 39–50. Plenum Press N. Y., 1983. Crypto '82, Santa Barbara, CA, August 1982.

[8] C. Boyd. Digital multisignatures. In H. Beker and F. Piper, editors, *Cryptography and coding*, pp. 241–246. Clarendon Press, 1989. Royal Agricultural College, Cirencester, December 15–17, 1986.

[9] E. F. Brickell and D. R. Stinson. Some improved bounds on the information rate of perfect secret sharing schemes. *Journal of Cryptology*, 5, pp. 153–166, 1992. Preliminary version appeared in "Advances in Cryptology – CRYPTO '90", A. J. Menezes and S. A. Vanstone, eds., Lecture Notes in Computer Science 537 (1991), 242-252.

[10] Ran Canetti and Amir Herzberg. Maintaining security in the presence of transient faults. In Yvo G. Desmedt, editor, *Advances in Cryptology — Crypto '94, Proceedings (Lecture Notes in Computer Science 839)*, pp. 425–439. Springer-Verlag, 1994. Santa Barbara, California, U.S.A., August 22–25.

[11] R. A. Croft and S. P. Harris. Public-key cryptography and re-usable shared secrets. In H. Beker and F. Piper, editors, *Cryptography and coding*, pp. 189–201. Clarendon Press, 1989. Royal Agricultural College, Cirencester, December 15–17, 1986.

[12] A. De Santis, Y. Desmedt, Y. Frankel, and M. Yung. How to share a function securely. In *Proceedings of the twenty-sixth annual ACM Symp. Theory of Computing (STOC)*, pp. 522–533, May 23–25, 1994. Montréal, Québec, Canada.

[13] Y. Desmedt, G. Di Crescenzo, and M. Burmester. Multiplicative non-abelian sharing schemes and their application to threshold cryptography. In J. Pieprzyk and R. Safavi-Naini, editors, *Advances in Cryptology — Asiacrypt '94, Proceedings (Lecture Notes in Computer Science 917)*, pp. 21–32. Springer-Verlag, 1995. Wollongong, Australia, November/December, 1994.

[14] Y. Desmedt and Y. Frankel. Threshold cryptosystems. In G. Brassard, editor, *Advances in Cryptology — Crypto '89, Proceedings (Lecture Notes in Computer Science 435)*, pp. 307–315. Springer-Verlag, 1990. Santa Barbara, California, U.S.A., August 20–24.

[15] Y. G. Desmedt and Y. Frankel. Homomorphic zero-knowledge threshold schemes over any finite abelian group. *SIAM Journal on Discrete Mathematics*, 7(4), pp. 667–679, November 1994.

[16] Y. Desmedt. Society and group oriented cryptography : a new concept. In C. Pomerance, editor, *Advances in Cryptology, Proc. of Crypto '87 (Lecture Notes in Computer Science 293)*, pp. 120–127. Springer-Verlag, 1988. Santa Barbara, California, U.S.A., August 16–20.

[17] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Robust and efficient sharing of RSA functions. In N. Koblitz, editor, *Advances in Cryptology — Crypto '96, Proceedings (Lecture Notes in Computer Science 1109)*, pp. 157–172. Springer-Verlag, 1996. Santa Barbara, California, U.S.A., August 18–22.

[18] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Robust threshold DSS signatures. In U. Maurer, editor, *Advances in Cryptology — Eurocrypt '96, Proceedings (Lecture Notes in Computer Science 1070)*, pp. 354–371. Springer-Verlag, 1996. Zaragoza, Spain, May 12–16.

[19] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1), pp. 186–208, February 1989.

[20] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung. Proactive secret sharing. In D. Coppersmith, editor, *Advances in Cryptology — Crypto '95, Proceedings (Lecture Notes in Computer Science 963)*, pp. 339–352. Springer-Verlag, 1995. Santa Barbara, California, U.S.A., August 27–31.

[21] Hua. *Introduction to Number Theory.* Springer, New York, 1982.

[22] M. Ito, A. Saito, and T. Nishizeki. Secret sharing schemes realizing general access structures. In *Proc. IEEE Global Telecommunications Conf., Globecom'87*, pp. 99–102. IEEE Communications Soc. Press, 1987.

[23] N. Jacobson. *Basic Algebra I.* W. H. Freeman and Company, New York, 1985.

[24] N. Jacobson. *Basic Algebra II.* W. H. Freeman and Company, New York, 1989.

[25] Sushil Jajodia and David Mutchler. Dynamic voting algorithms for maintaining the consistency of a replicated database. *ACM Trans. on Database Systems*, 15(2), pp. 230–280, june 1990.

[26] R. Kannan and A. Bachem. Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM J. Comput.*, 8(4), pp. 499–507, November 1979.

[27] E. D. Karnin, J. W. Greene, and M. Hellman. On secret sharing systems. *IEEE Tr. Inform. Theory*, 29(1), pp. 35–41, January 1983.

[28] S. K. Langford. Threshold DSS signatures without a trusted party. In D. Coppersmith, editor, *Advances in Cryptology — Crypto '95, Proceedings (Lecture Notes in Computer Science 963)*, pp. 397–409. Springer-Verlag, 1995. Santa Barbara, California, U.S.A., August 27–31.

[29] R. J. McEliece and D. V. Sarwate. On sharing secrets and Reed-Solomon codes. *Comm. ACM*, 24(9), pp. 583–584, September 1981.

[30] R. Ostrovsky and M. Yung. How to withstand mobile virus attacks. In *Proceedings of the 10-th Annual ACM Symp. on Principles of Distributed Computing*, pp. 51–60, August 19–21, 1991. Montreal, Quebec, Canada.

[31] T. P. Pedersen. A threshold cryptosystem without a trusted party. In D. W. Davies, editor, *Advances in Cryptology, Proc. of Eurocrypt '91 (Lecture Notes in Computer Science 547)*, pp. 522–526. Springer-Verlag, April 1991. Brighton, U.K.

[32] T. P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In J. Feigenbaum, editor, *Advances in Cryptology — Crypto '91, Proceedings (Lecture Notes in Computer Science 576)*, pp. 129–140. Springer-Verlag, 1992. Santa Barbara, California, U.S.A., August 12–15.

[33] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public key cryptosystems. *Commun. ACM*, 21, pp. 294–299, April 1978.

[34] A. Shamir. How to share a secret. *Commun. ACM*, 22, pp. 612–613, November 1979.