Verifiable Secret Redistribution for Threshold Sharing Schemes

Theodore M. Wong Chenxi Wang[1] Jeannette M. Wing

February 2002

CMU-CS-02-114

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

We present a new protocol for verifiably redistributing secrets from an (m, n) threshold sharing scheme to an (m', n') scheme. Our protocol guards against dynamic adversaries. We observe that existing protocols either cannot be readily extended to allow redistribution between different threshold schemes, or have vulnerabilities that allow faulty old shareholders to distribute invalid shares to new shareholders. Our primary contribution is that in our protocol, new shareholders can verify the validity of their shares after redistribution between different threshold schemes.

[1]Department of Electrical and Computer Engineering, `chenxi@ece.cmu.edu`

This research is sponsored by the Defense Advanced Research Projects Agency (DARPA), Advanced Technology Office, under the title "Organically Assured and Survivable Information Systems (OASIS)" (Air Force Coop. Agreement no. F30602-00-2-0523).

Keywords: non-interactive verifiable secret redistribution, threshold sharing schemes, threshold cryptography

1 Introduction

Threshold cryptography protocols provide fundamental building blocks for secure distributed computation and the safeguarding of secrets. The area of threshold cryptography has been studied extensively since its introduction by Blakley and Shamir [Bla79, Sha79].

Two categories of threshold protocols, *proactive secret sharing* (PSS) protocols and *secret redistribution* protocols, provide enhanced protection against *dynamic adversaries* ([OY91]). PSS protocols [FGMY97a, FGMY97b, FMY99, FMY01, HJKY95, HJJ$^+$97, Rab98] protect against an adversary through periodic updating of the shares, which renders old shares obtained by the adversary useless. In general, PSS protocols retain the same threshold scheme before and after updating. Secret redistribution protocols protect against an adversary through periodic redistribution of shares from an (m, n) threshold sharing scheme to an (m', n') scheme [DJ97, FGMY97a], without requiring the intermediate reconstruction of the original secret.

To prevent faulty shareholders from corrupting the shares generated by a PSS or redistribution protocol, the shareholders must *verify* the *validity* of their shares after protocol execution (i.e., confirm that the shares can be used to reconstruct the original secret). In PSS protocols, shareholders obtain verification information during the initial distribution of shares, and update the information after updating the shares. In redistribution protocols, new shareholders obtain verification information from the old shareholders.

We observe that the verification mechanisms in existing protocols have the following shortcomings:

- The mechanisms in PSS protocols cannot be readily extended to allow "updates" between different threshold schemes or between disjoint sets of shareholders. Thus, these protocols cannot respond to the permanent removal or addition of a shareholder.
- The mechanisms in redistribution protocols have vulnerabilities that allow a faulty old shareholder to distribute invalid shares to new shareholders.

Our study is motivated by the application of redistribution protocols to survivable storage systems [WBS$^+$00, WBP$^+$01]. A survivable storage system distributes shares of files (secrets) across a set of storage servers. The system redistributes files to recover from the compromising of servers or to balance file access loads upon the addition of new servers.

We present a new protocol for *verifiable secret redistribution* (VSR) from an (m, n) threshold scheme to an (m', n') scheme. We base our protocol on Desmedt and Jajodia's redistribution protocol [DJ97], in which new shareholders generate shares from *subshares* of old shares. We extend their protocol with Feldman's verifiable secret sharing (VSS) scheme [Fel87] to enable new shareholders to verify the validity of their subshares (i.e., confirm that the subshares can be used to reconstruct old shares). However, we go beyond a naïve extension, which does not enable new shareholders to verify that they have received subshares of *valid* old shares. To achieve complete verification in our protocol, old shareholders broadcast a commitment to the secret to the new shareholders. We prove that the new shareholders can generate valid new shares if they can both verify the validity of the old shares and verify the validity of the subshares.

The primary contribution of our work is that in our protocol:

- *New shareholders can verify the validity of their shares after redistribution between different threshold schemes.*

2 Related work

Blakley and Shamir invented secret sharing schemes independently. In Shamir's (m, n) sharing scheme [Sha79], the interpolation of an $m - 1$ degree polynomial from m of n points yields a constant term in the polynomial that corresponds to the secret. In Blakley's scheme [Bla79], the intersection of m of n vector spaces yields a one-dimensional vector that corresponds to the secret. Desmedt surveys other sharing schemes [Des97].

Feldman's VSS scheme [Fel87] is one of several to catch a dealer that attempts to distribute invalid shares. Chor *et al* present a scheme in which the dealer and shareholders perform an interactive secure distributed computation [CGMA85]. Benaloh [Ben87], Gennaro and Micali [GJKR96, GM95], Goldreich *et al* [GMW87], and Rabin and Ben-Or [Rab94, RBO89] propose schemes in which the dealer and shareholders participate in an interactive zero-knowledge proof of validity; the scheme of Gennaro and Micali, and that of Rabin and Ben-Or, is information-theoretically secure. Pederson [Ped91] presents a scheme, like Feldman's, in which the dealer broadcasts a non-interactive zero-knowledge proof to the shareholders. Beth *et al* [BKO93] present a VSS scheme for monotone access structures based on finite geometries. Our VSR protocol differs from previous VSS schemes in that the multiple "dealers" of the new shares (the old shareholders) do not have the secret, and must use other information to generate a proof for the new shareholders. Also, each new shareholder verifies the validity of the subshares distributed by the old shareholders, and verifies the validity of the shares used by the old shareholders to generate the subshares.

Frankel *et al* [FGMY97b, FMY99, FMY01] and Rabin [Rab98] propose PSS protocols in which each shareholder periodically distributes a subshare of its share to each of the other shareholders. Each shareholder combines the received subshares to generate a new share. A drawback of these PSS protocols is that the shareholders rely on commitments received during the initial distribution of the secret to verify the validity that their generated shares, and thus one cannot redistribute between disjoint sets of n shareholders. Also, the commitments depend on m and n, and thus one cannot redistribute from an (m, n) to (m', n') threshold scheme. Lastly, the protocols build upon specific threshold schemes, and may not be applicable to a general class of schemes.

Desmedt and Jajodia [DJ97] present a secret redistribution protocol that does not require the intermediate reconstruction of the original secret. We present the details of their protocol in Sec. 3.2. Their protocol allows redistribution between different threshold schemes, and between disjoint sets of shareholders. Unfortunately, a compromised old shareholder in both protocols can undetectably distribute "subshares" of some random value instead of subshares of a valid old share. New shareholders that use these "subshares" will generate invalid new shares.

Frankel *et al* [FGMY97a], independently of Desmedt and Jajodia, present a (proactive) redistribution protocol for shares of a private key in a public key cryptosystem. The protocol involves redistribution of the key from an (m, n) to (m, m) threshold scheme, followed by redistribution to an (m', n') scheme. Each old shareholder broadcasts a commitment to its share when it distributes the subshares. New shareholders use the commitment to verify the validity of their subshares. However, nothing prevents a compromised old shareholder from broadcasting a "commitment" to some random value. Thus, the protocol ultimately suffers from the same shortcoming as that of Desmedt and Jajodia.

Other researchers present secret redistribution protocols that do not involve the physical redistribution of shares. Blakley *et al* consider threshold schemes that *disenroll* (remove) shareholders from the access structure with broadcast messages [BBCM92]; the new shareholders are a subset of the old ones. Cachin proposes a secret sharing scheme that *enrolls* (adds) shareholders in the access structure after the initial sharing [Cac95]; the new shareholders are a superset of the old ones. Blundo *et al* presents a scheme in which the dealer uses broadcast messages to activate different, possibly disjoint, authorized subsets [BCSV96]. Blundo's scheme requires shareholders to have a share regardless of whether or not they are in the active authorized subset, in contrast to Desmedt and Jajodia's scheme. Our VSR protocol alters the threshold scheme by physical redistribution of shares, and allows new shareholders to verify that they have valid shares.

Herzberg *et al* [HJKY95, HJJ$^+$97] propose a PSS protocol for Shamir's sharing scheme [Sha79] in which each shareholder periodically distributes *update shares* to all other shareholders. Zhou, Schneider, and van Renesse propose a PSS protocol for asynchronous, wide-area networks, and employ it in an on-line certification authority [ZSvR00]. Our VSR protocol, unlike these PSS protocols, can redistribute shares to arbitrary access structures. However, we assume that there exist reliable broadcast communication channels among all participants and private channels between every pair of participants in our protocol, which Zhou *et al* avoid in their asynchronous protocol.

3 Cryptographic building blocks

In this section, we outline the cryptographic protocols that form the building blocks for our VSR protocol. We first summarize Desmedt and Jajodia's secret redistribution protocol [DJ97] for linear secret sharing schemes, and then summarize Feldman's VSS scheme [Fel87].

3.1 Mathematical notation

An (m, n) *linear threshold scheme* is an algorithm for the distribution of shares of a secret to a set of n shareholders such that the secret is a linear combination of the shares of any m shareholders. We define a secret k to be in set $\mathcal{K}$ of secrets, and each shareholder i to be in the set $\mathcal{P}$ ($|P| = n$) of shareholders. To distribute k, we generate a share s_i for each $i \in \mathcal{P}$ with a *polynomial* $a(i)$:

$$s_i = k + \sum_{l=1}^{m-1} a_l i^l \tag{1}$$

where s_i is in the set $\mathcal{S}_i$ of shares, and $\mathcal{S}_i$ is in the set $\mathcal{S}$ of share sets. For linear threshold schemes, $\mathcal{S}_i = \mathcal{S}_j$ for all $i, j \in \mathcal{P}$. To reconstruct k, we combine s_i from all i in an *authorized subset* $\mathcal{B}$ ($|\mathcal{B}| = m$) of $\mathcal{P}$:

$$k = \sum_{i \in \mathcal{B}} \psi_i(s_i) \tag{2}$$

ψ_i is a homomorphism from $\mathcal{S}_i$ to $\mathcal{K}$; we aggregate ψ_i into the set ψ of homomorphisms. For linear threshold schemes, the homomorphisms are multiplications by scalars ψ_i [DJ97]. All authorized subsets $\mathcal{B}$ are in the *access structure* $\Gamma_{\mathcal{P}}$. We represent linear threshold schemes with the tuple $\{\Gamma_{\mathcal{P}}, \mathcal{K}, \mathcal{S}, \psi\}$.

We utilize a homomorphic commitment function $C(x)$ [Ben87, Fel87] that maps from plain-text to cipher-text and is hard to invert. $C(x)$ is such that:

$$\begin{gathered} C(a+b) = C(a)C(b) \\ C(ax) = (C(a))^x \end{gathered} \tag{3}$$

3.2 Desmedt and Jajodia's secret redistribution protocol

Desmedt and Jajodia present a protocol for the redistribution of shares of secrets from threshold sharing schemes without requiring the intermediate reconstruction of the secret [DJ97]. For schemes that satisfy the conditions in Fig. 1, we can use the protocol in Fig. 2 to redistribute shares. Suppose we have a set $\mathcal{P}$ of shareholders i that have shares s_i of a secret k distributed with the scheme $(\Gamma_{\mathcal{P}}, \mathcal{K}, \mathcal{S}, \psi)$, and wish to redistribute to a set $\mathcal{P}'$ of shareholders j that have shares s'_j distributed with a different scheme $(\Gamma'_{\mathcal{P}'}, \mathcal{K}, \mathcal{S}', \psi')$.

To achieve this, we select an authorized subset $\mathcal{B} \in \Gamma_{\mathcal{P}}$ and use an intermediate scheme $(\Gamma'_{\mathcal{P}'}, \mathcal{S}_i, \hat{\mathcal{S}}_i, \hat{\psi}_i)$ to distribute subshares $\hat{s}_{ij}$ of each s_i of $i \in \mathcal{B}$ to each $j \in \mathcal{P}'$, where the set $\hat{\mathcal{S}}_i$ of sets of subshares is:

$$\hat{\mathcal{S}}_i = \left\{ \hat{\mathcal{S}}_{ij} : j \in \mathcal{B}', \mathcal{B}' \in \Gamma'_{\mathcal{P}'} \right\} \tag{4}$$

and the set $\hat{\psi}_i$ of homomorphisms from $\hat{\mathcal{S}}_i$ to $\mathcal{S}_i$ is:

$$\hat{\psi}_i = \left\{ \hat{\psi}_{ij} : j \in \mathcal{B}', \mathcal{B}' \in \Gamma'_{\mathcal{P}'} \right\} \tag{5}$$

If we treat $\hat{s}_{ij}$ as being distributed by another intermediate scheme $(\Gamma_{\mathcal{P}}, \mathcal{S}'_j, \hat{\mathcal{S}}'_j, \hat{\psi}'_j)$ (with $\hat{\mathcal{S}}_j$ and $\hat{\psi}_j$ defined similarly to $\hat{\mathcal{S}}_i$ and $\hat{\psi}_i$ in Eqns. (4) and (5)), we can generate s'_j for each j with the following equation:

$$s'_j = \sum_{i \in \mathcal{B}} \hat{\psi}'_{ji} \hat{s}_{ij} \tag{6}$$

The correctness of the protocol depends on a condition that the homomorphisms of the old, intermediate, and new schemes *pseudo-commute*. Homomorphisms ψ_i, $\hat{\psi}_{ij}$, ψ'_j, and $\hat{\psi}'_{ji}$ pseudo-commute if:

$$\psi_i \circ \hat{\psi}_{ij} = \psi'_j \circ \hat{\psi}'_{ji} \tag{7}$$

1. For a set $\mathcal{P}$ of shareholders, there exists a linear sharing scheme $(\Gamma_{\mathcal{P}}, \mathcal{K}, \mathcal{S}, \psi)$ such that each $i \in \mathcal{P}$ has received a share $s_i \in \mathcal{S}_i \in \mathcal{S}$ of $k \in \mathcal{K}$.
2. For each $i \in \mathcal{P}$ there exists an intermediate linear sharing scheme $(\Gamma'_{\mathcal{P}'}, \mathcal{S}_i, \hat{\mathcal{S}}_i, \hat{\psi}_i)$ for the distribution of subshares $\hat{s}_{ij}$ of s_i to each $j \in \mathcal{P}'$.
3. For all $x, y \in \mathcal{K}$, $x + y = y + x$.
4. For each $i \in \mathcal{B} \in \Gamma_{\mathcal{P}}$ and $j \in \mathcal{B}' \in \Gamma'_{\mathcal{P}'}$, there exist homomorphisms ψ_i, $\hat{\psi}_{ij}$, ψ'_j, and $\hat{\psi}'_{ji}$ that pseudo-commute:

$$\psi_i \circ \hat{\psi}_{ij} = \psi'_j \circ \hat{\psi}'_{ji}$$

Figure 1: Necessary conditions for the redistribution of shares from linear sharing schemes [DJ97].

Desmedt and Jajodia's Secret Redistribution protocol:
To redistribute k from the (m, n) scheme $\{\Gamma_{\mathcal{P}}, \mathcal{K}, \mathcal{S}, \psi\}$ to the (m', n') scheme $\{\Gamma_{\mathcal{P}'}, \mathcal{K}, \mathcal{S}', \psi'\}$:

1. Select an authorized subset $\mathcal{B} \in \Gamma_{\mathcal{P}}$. Use the intermediate scheme $(\Gamma'_{\mathcal{P}'}, \mathcal{S}_i, \hat{\mathcal{S}}_i, \hat{\psi}_i)$ to distribute subshares $\hat{s}_{ij}$ of each share s_i of $i \in \mathcal{B}$ to each $j \in \mathcal{P}'$.
2. For each $j \in \mathcal{P}'$, treat $\hat{s}_{ij}$ as if distributed with another intermediate scheme $(\Gamma_{\mathcal{P}}, \mathcal{S}'_j, \hat{\mathcal{S}}'_j, \hat{\psi}'_j)$, and generate s'_j:

$$s'_j = \sum_{i \in \mathcal{B}} \hat{\psi}'_{ji}(\hat{s}_{ij})$$

Figure 2: Desmedt and Jajodia's secret redistribution protocol for linear sharing schemes [DJ97].

3.3 Feldman's VSS scheme

Feldman presents a scheme that shareholders of a secret can use to verify the validity of their shares [Fel87]. Feldman assumes that there exists a homomorphic commitment function $C(x)$ that is hard to invert. Given the threshold scheme $\{\Gamma_{\mathcal{P}}, \mathcal{K}, \mathcal{S}, \psi\}$, the dealer of the secret $k \in \mathcal{K}$, in addition to sending shares $s_i \in \mathcal{S}_i$ to each $i \in \mathcal{P}$, broadcasts $C(k)$ and $C(a_1) \ldots C(a_{m-1})$ (commitments of the coefficients of the polynomial $a(i)$ used to generate s_i). Each i then verifies that s_i is a valid share of k with the following equation:

$$C(s_i) \equiv C(k) \prod_{l=1}^{m-1} C(a_l)^{i^l} \tag{8}$$

Eqn. (8) follows from Eqn. (1) and the homomorphic properties of $C(x)$ in Eqn. (3). Since $C(x)$ is hard to invert, no i can learn k from the broadcast of $C(k)$. We summarize Feldman's scheme in Fig. 3.

4 The VSR protocol

We present our verifiable secret redistribution protocol for secrets distributed with linear threshold schemes. We represent the (m, n) and (m', n') schemes with $\{\Gamma_{\mathcal{P}}, \mathcal{K}, \mathcal{S}, \psi\}$ and $\{\Gamma_{\mathcal{P}'}, \mathcal{K}, \mathcal{S}', \psi'\}$ respectively. We assume that there exists a homomorphic commitment function $C(x)$ that is hard to invert, and that there exist reliable broadcast communication channels among all participants and private channels between every pair of participants. We also assume that there are at most $n - m$ faulty old shareholders, that $m > \frac{n}{2}$, and that there are n' non-faulty new shareholders.

The initial distribution of a secret (INITIAL in Fig. 4) proceeds as in Feldman's VSS scheme [Fel87]. The dealer of secret $k \in \mathcal{K}$ distributes shares $s_i \in \mathcal{S}_i$ to each shareholder $i \in \mathcal{P}$, using the polynomial $a(i)$ (INITIAL step 1). The dealer also broadcasts $C(k), C(a_1) \ldots C(a_{m'-1})$, which each i uses in Eqn. (8) to verify the validity of s_i (INITIAL steps 2 and 3). If Eqn. (8) holds, i stores s_i and $C(k)$ (INITIAL step 4).

Redistribution of the secret (REDIST in Fig. 4) proceeds as in Desmedt and Jajodia's protocol [DJ97]. Each i in an authorized subset $\mathcal{B} \in \Gamma_{\mathcal{P}}$ uses an intermediate scheme $\{\Gamma_{\mathcal{P}'}, \mathcal{S}_i, \hat{\mathcal{S}}_i, \hat{\psi}'\}$ (with the polynomial $a'_i(j)$) to distribute subshares $\hat{s}_{ij} \in \hat{\mathcal{S}}_i$ of s_i to each shareholder $j \in \mathcal{P}'$ (REDIST step 1). Each j then generates the new share s'_j (Eqn. (6), which is REDIST step 4). We may redistribute k an arbitrary number of times before we reconstruct it.

Feldman's Verifiable Secret Sharing scheme:
To use the (m, n) threshold scheme $\{\Gamma_{\mathcal{P}}, \mathcal{K}, \mathcal{S}, \psi\}$ to distribute a secret $k \in \mathcal{K}$:

1. For each $i \in \mathcal{P}$, use the polynomial $a(i) = k + a_1 i + \ldots + a_{m-1} i^{m-1}$ to compute the share $s_i = a(i)$ of k, and send s_i to i over a private channel.
2. For each $i \in \mathcal{P}$, use commitment function $C(x)$ to generate $C(k), C(a_1), \ldots, C(a_{m-1})$, and broadcast them to all i.
3. For each $i \in \mathcal{P}$, verify that:

$$C(s_i) \equiv C(k) \prod_{l=1}^{m-1} C(a_l)^{i^l}$$

If the condition holds, i broadcasts a "commit" message. Otherwise, i broadcasts an "abort" message.

Figure 3: Feldman's VSS scheme for an (m, n) threshold scheme [Fel87].

Verifiable Secret Redistribution protocol:
INITIAL: To use the (m, n) linear threshold scheme $\{\Gamma_{\mathcal{P}}, \mathcal{K}, \mathcal{S}, \psi\}$ to distribute a secret $k \in \mathcal{K}$:

1. For each $i \in \mathcal{P}$, use the polynomial $a(i) = k + a_1 i + \ldots + a_{m-1} i^{m-1}$ to compute the share s_i of k, and send s_i to i over a private channel.
2. Use commitment function $C(x)$ to generate $C(k), C(a_1), \ldots, C(a_{m-1})$, and send them to all $i \in \mathcal{P}$ over the broadcast channel.
3. For each $i \in \mathcal{P}$, verify that:

$$C(s_i) \equiv C(k) \prod_{l=1}^{m-1} C(a_l)^{i^l}$$

   If the condition holds, i broadcasts a "commit" message. Otherwise, i broadcasts an "abort" message.
4. If all $i \in \mathcal{P}$ agree to commit, each i stores s_i and $C(k)$. Otherwise, they abort the protocol.

REDIST: To redistribute k from the (m, n) scheme $\{\Gamma_{\mathcal{P}}, \mathcal{K}, \mathcal{S}, \psi\}$ to the (m', n') scheme $\{\Gamma_{\mathcal{P}'}, \mathcal{K}, \mathcal{S}', \psi'\}$:

1. For each $i \in \mathcal{B}$ ($\mathcal{B} \in \Gamma_{\mathcal{P}}$), use the polynomial $a'_i(j) = s_i + a'_{i1} j + \ldots + a'_{i(m'-1)} j^{m'-1}$ to compute the subshares $\hat{s}_{ij}$ of s_i, and send $\hat{s}_{ij}$ to the corresponding $j \in \mathcal{P}'$ over a private channel.
2. For each $i \in \mathcal{P}$, use the commitment function $C(x)$ generate $C(s_i), C(a_{i1}), \ldots, C(a_{i(m'-1)})$, and send them to all $j \in \mathcal{P}'$ over the broadcast channel.
3. For each $j \in \mathcal{P}'$, verify that:

$$\forall i \in \mathcal{B} : C(\hat{s}_{ij}) \equiv C(s_i) \prod_{l=1}^{m'-1} C(a_{il})^{j^l}$$

   and:

$$C(k) = \prod_{i \in \mathcal{B}} C(s_i)^{\psi_i}$$

   If the conditions hold, j broadcasts a "commit" message. Otherwise, j broadcasts an "abort" message.
4. If all $j \in \mathcal{P}'$ agree to commit, each j generates s'_j:

$$s'_j = \sum_{i \in \mathcal{B}} \hat{\psi}'_{ji} \hat{s}_{ij}$$

   and stores s'_j and $C(k)$. Otherwise, they abort the protocol.

Figure 4: Verifiable secret redistribution protocol for the redistribution of shares from an (m, n) to (m', n') threshold scheme.

For the new shareholders to verify that their shares of the secret are valid after redistribution, we require that two conditions, SHARES-VALID and SUBSHARES-VALID, hold. Recall that for linear threshold schemes, homomorphisms ψ_i are multiplications by scalars ψ_i. When all $i \in \mathcal{B}$ ($\mathcal{B} \in \Gamma_{\mathcal{P}}$) redistribute s_i to each $j \in \mathcal{P}'$, all s_j are valid shares of k if:

SHARES-VALID:
$k = \sum_{i \in \mathcal{B}} \psi_i s_i$

SUBSHARES-VALID:
$\forall i \in \mathcal{B}, \mathcal{B}' \in \Gamma_{\mathcal{P}'} : s_i = \sum_{j \in \mathcal{B}'} \hat{\psi}'_{ij} \hat{s}_{ij}$

We define a NEW-SHARES-VALID condition. The condition holds if new shareholders have valid shares of the secret. We prove in Sec. 4.3 that NEW-SHARES-VALID holds if SHARES-VALID and SUBSHARES-VALID hold. The definition of NEW-SHARES-VALID follows from Eqn. (2) for $\{\Gamma_{\mathcal{P}'}, \mathcal{K}, \mathcal{S}', \psi'\}$:

NEW-SHARES-VALID:
$k = \sum_{j \in \mathcal{B}'} \psi'_j s'_j$

We use Feldman's VSS scheme to verify that SUBSHARES-VALID holds in our protocol. The distribution of $\hat{s}_{ij}$ from s_i (REDIST step 1) is an application of the scheme $\{\Gamma_{\mathcal{P}'}, \mathcal{S}_i, \hat{\mathcal{S}}_i, \hat{\psi}'\}$. Thus, each $i \in \mathcal{B}$ broadcasts $C(s_i)$ and $C(a_{i1}) \ldots C(a_{i(m-1)})$, which each j uses to verify the validity of $\hat{s}_{ij}$ (REDIST step 2).

The key insight embodied in our VSR protocol is that the naïve extension of Desmedt and Jajodia's protocol with Feldman's scheme does not in itself allow the new shareholders to verify that NEW-SHARES-VALID holds. The difficulty arises because Feldman's scheme only verifies that SUBSHARES-VALID holds, which in the absence of SHARES-VALID is insufficient to verify that NEW-SHARES-VALID holds. Although Desmedt and Jajodia observe that the linear properties of their protocol and the properties of $C(x)$ ensure that each j generates valid shares [DJ97], they implicitly assume that each $i \in \mathcal{B}$ distributes subshares of valid s_i. The VSS scheme simply allows $i \in \mathcal{B}$ shareholder to prove that it distributed valid $\hat{s}_{ij}$ of some value. However, i may have distributed "subshares" of some random value instead of $\hat{s}_{ij}$ of s_i. Thus, we require a sub-protocol for i to prove that it distributed $\hat{s}_{ij}$ of s_i to $j \in \mathcal{P}'$.

To allow the new shareholders to verify that SHARES-VALID holds, which together with SUBSHARES-VALID verifies that NEW-SHARES-VALID holds, the old shareholders in our protocol broadcast a commitment to the secret. $i \in \mathcal{B}$ must therefore store $C(k)$ (received during INITIAL) and later broadcast it to $j \in \mathcal{P}'$. Recall that each j receives s_i from each i to verify that SUBSHARES-VALID holds. Once each j receives $C(k)$, it verifies that s_i is a valid share of k with the following equation:

$$C(k) = \prod_{i \in \mathcal{B}} C(s_i)^{\psi_i} \tag{9}$$

Eqn. (9) follows from Eqn. (2) and the homomorphic properties of $C(x)$ in Eqn. (3). Since $C(x)$ is hard to invert, no j can learn k from the broadcast of $C(k)$.

4.1 Assumptions about faulty shareholders

When we redistribute a secret from the scheme $\{\Gamma_{\mathcal{P}}, \mathcal{K}, \mathcal{S}, \psi\}$ to the scheme $\{\Gamma_{\mathcal{P}'}, \mathcal{K}, \mathcal{S}', \psi'\}$ with our VSR protocol, we assume that at least m of the n shareholders in $\mathcal{P}$ and all n' of the shareholders in $\mathcal{P}'$ are non-faulty, and up to $n - m$ of the remaining shareholders in $\mathcal{P}$ may be faulty. We denote faulty shareholders, and the values they distribute, with over-bars. A non-faulty shareholder $i \in \mathcal{P}$ distributes valid subshares

$\hat{s}_{ij}$ of its share s_i to all shareholders $j \in \mathcal{P}'$ and broadcasts $C(k)$ corresponding to secret $k \in \mathcal{K}$. A faulty shareholder $\bar{i} \in \mathcal{P}$ may distribute invalid subshares $\overline{\hat{s}_{\bar{i}j}}$ or broadcast $\overline{C(k)}$ not corresponding to k.

We also assume that we do not know which m of the n shareholders in $\mathcal{P}$ are non-faulty. Suppose we include a faulty shareholder $\bar{i}$ in our selection of authorized subset $\mathcal{B} \in \Gamma_{\mathcal{P}}$ to participate in redistribution (REDIST in Fig. 4). However, if $\bar{i}$ distributes $\overline{\hat{s}_{\bar{i}j}}$, one of the j will detect the presence of $\bar{i}$ since one of Eqns. (8) or (9) will not hold. Alternatively, if $\bar{i}$ broadcasts $\overline{C(k)}$, all j will detect the discrepancy when non-faulty old shareholders broadcast $C(k)$. Thus, $\bar{i}$ must participate in the protocol without fault or risk detection. If we detect the presence of $\bar{i}$, we must restart redistribution with another set of m old shareholders. Unfortunately, we cannot identify $\bar{i}$ with our protocol.

The assumption that we do not know which m shareholders in $\mathcal{P}$ are non-faulty bounds the relative values of m and n. We assume that we can detect discrepancies between $\overline{C(k)}$ and $C(k)$ broadcast by faulty and non-faulty shareholders in $\mathcal{P}$ respectively. However, if we were to select a group of m faulty shareholders $\bar{i}$ inadvertently, then we would be unable to detect discrepancies if all $\bar{i}$ broadcast $\overline{C(k)}$. We therefore require that $m > \frac{n}{2}$ so each $\mathcal{B}$ contains at least one non-faulty shareholder; if $m \leq \frac{n}{2}$, $n - m$ faulty shareholders in $\mathcal{P}$ could conspire to reconstruct k or deceive shareholders in $\mathcal{P}'$.

The requirement that all n' shareholders in $\mathcal{P}'$ are non-faulty is reasonable if we view the purpose of our VSR protocol as one of detecting faulty behavior by shareholders in $\mathcal{P}$. This is analogous to one of the assumptions underlying Feldman's VSS scheme [Fel87], in which the shareholders are implicitly trusted to store valid shares (and reject invalid shares) of a secret.

4.2 Computational cost

The computational cost for each new shareholder of verification in our VSR protocol (REDIST Step 3 in Fig. 4) is $\mathrm{O}(mm')$ multiplications and $\mathrm{O}(mm')$ exponentiations, exclusive of the cost of the commitment function $C(x)$. Consider redistribution from the scheme $\{\Gamma_{\mathcal{P}}, \mathcal{K}, \mathcal{S}, \psi\}$ to the scheme $\{\Gamma_{\mathcal{P}'}, \mathcal{K}, \mathcal{S}', \psi'\}$. Each new shareholder $j \in \mathcal{P}'$ performs $m - 1$ multiplications ($\mathcal{B} \in \Gamma_{\mathcal{P}}$; $|\mathcal{B}| = m$) and m exponentiations to verify that SHARES-VALID holds (Eqn. (9)), for a total cost of $\mathrm{O}(m)$; we do not include the (small) cost of computing the powers of i. Each j also performs $m' - 1$ multiplications ($\mathcal{B}' \in \Gamma_{\mathcal{P}'}$; $|\mathcal{B}'| = m'$) and $m' - 1$ exponentiations for m old shareholders $i \in \mathcal{B}$ to verify that SUBSHARES-VALID holds (Eqn. (8)), for a total cost of $\mathrm{O}(mm')$. Thus, the total cost for each j to verify that both conditions hold is $\mathrm{O}(mm')$ multiplications and $\mathrm{O}(mm')$ exponentiations, exclusive of the cost of $C(x)$.

4.3 Correctness

We prove that NEW-SHARES-VALID holds after share redistribution if SHARES-VALID and SUBSHARES-VALID hold. We also show that Eqns. (8) and (9) verify that SUBSHARES-VALID and SHARES-VALID hold.

Lemma 1 SUBSHARES-VALID *holds if Eqn. (8) holds.*

PROOF: Proved by Feldman [Fel87]. □

Lemma 2 SHARES-VALID *holds if Eqn. (9) holds.*

PROOF: Assume that Eqn. (9) holds. It then follows that SHARES-VALID holds from Eqn. (2) and the homomorphic properties of the commitment function $C(x)$. □

Theorem 1 (VSR theorem) *For the (m, n) linear threshold scheme $\{\Gamma_{\mathcal{P}}, \mathcal{K}, \mathcal{S}, \psi\}$ and the (m', n') scheme $\{\Gamma_{\mathcal{P}'}, \mathcal{K}, \mathcal{S}', \psi'\}$, for all secrets $k \in \mathcal{K}$, and for all authorized subsets $\mathcal{B} \in \Gamma_{\mathcal{P}}$, $\mathcal{B}' \in \Gamma_{\mathcal{P}'}$,* NEW-SHARES-VALID *holds after redistribution of k with the VSR protocol if* SHARES-VALID *and* SUBSHARES-VALID *hold.*

PROOF: Assume that both SHARES-VALID and SUBSHARES-VALID hold. Then:

$$
\begin{aligned}
k &= \sum_{i \in \mathcal{B}} \psi_i s_i \qquad \text{(SHARES-VALID)} \\
&= \sum_{i \in \mathcal{B}} \psi_i \left(\sum_{j \in \mathcal{B}'} \hat{\psi}_{ij} \hat{s}_{ij} \right) \qquad \text{(SUBSHARES-VALID)} \\
&= \sum_{i \in \mathcal{B}} \sum_{j \in \mathcal{B}'} \psi_i \hat{\psi}_{ij} \hat{s}_{ij} \qquad (\psi_i \text{ is a homomorphism}) \\
&= \sum_{i \in \mathcal{B}} \sum_{j \in \mathcal{B}'} \psi'_j \hat{\psi}'_{ji} \hat{s}_{ij} \qquad \text{(pseudo-commutativity of homomorphisms (Eqn. (7)))} \\
&= \sum_{j \in \mathcal{B}'} \sum_{i \in \mathcal{B}} \psi'_j \hat{\psi}'_{ji} \hat{s}_{ij} \qquad (\forall x, y \in \mathcal{K} : x + y = y + x) \\
&= \sum_{j \in \mathcal{B}'} \psi'_j \left(\sum_{i \in \mathcal{B}} \hat{\psi}'_{ji} \hat{s}_{ij} \right) \qquad (\psi'_j \text{ is a homomorphism}) \\
&= \sum_{j \in \mathcal{B}'} \psi'_j s'_j \qquad \text{(Eqn. (6))}
\end{aligned}
$$

□

Our correctness proof mirrors that for Desmedt and Jajodia's secret redistribution protocol [DJ97].

5 Specialization of the VSR protocol for Shamir's sharing scheme

We present the specialization of our VSR protocol for Shamir's sharing scheme [Sha79]. We first summarize Shamir's scheme, and then specialize our protocol for Shamir's scheme. We present the specialization to demonstrate the practical application of our VSR protocol, and to emphasize the need for new shareholders to obtain the commitment to the secret for verification of their shares.

5.1 Shamir's sharing scheme

Shamir presents an (m, n) sharing scheme based on polynomial interpolation [Sha79]. The secret k is in $\mathbb{Z}_p$ (p prime; $p > n$), and each shareholder i is in the set $\mathcal{P}$ ($|\mathcal{P}| = n$). All mathematical operations are in the finite field $\mathbb{Z}_p$. To distribute k, we select a polynomial $a(i)$ with degree $m-1$ and constant term k, and generate a share s_i for each i in $\mathcal{P}$ with $a(i)$:

$$s_i = k + a_1 i + \ldots a_{m-1} i^{m-1} \tag{10}$$

where $s_i \in \mathbb{Z}_p$. To reconstruct k, we recover m coordinate pairs (i, s_i) of all $i \in \mathcal{B}$, (where $|\mathcal{B}| = m$ and $\mathcal{B} \in \Gamma^S_{\mathcal{P}}$), and use the pairs in the Lagrange interpolation formula:

$$k = \sum_{i \in \mathcal{B}} b_i s_i \quad \text{where} \quad b_i = \prod_{l \in \mathcal{B}, l \neq i} \frac{l}{(l - i)} \tag{11}$$

We represent Shamir's scheme with the tuple $\{\Gamma^S_{\mathcal{P}}, \mathbb{Z}_p, \{\mathbb{Z}_p\}, \psi^S\}$, where $\psi_i = b_i$ and $\psi_i \in \psi^S$.

5.2 The VSR protocol for Shamir's scheme

We present our VSR protocol for secrets distributed with Shamir's sharing scheme [Sha79]. We represent the (m, n) scheme with $\{\Gamma^S_{\mathcal{P}}, \mathbb{Z}_p, \{\mathbb{Z}_p\}, \psi^S\}$, and the (m', n') scheme with $\{\Gamma^S_{\mathcal{P}'}, \mathbb{Z}_p, \{\mathbb{Z}_p\}, {\psi'}^S\}$. We assume that the computation of discrete logs in a finite field is intractable. As for the general VSR protocol, we assume there exist reliable broadcast communication channels among all participants and private channels between every pair of participants. We assume that there are at most $n - m$ faulty old shareholders, that $m > \frac{n}{2}$, and that there are n' non-faulty new shareholders. We summarize the protocol in Fig. 5.

Redistribution of the secret (REDIST in Fig. 5) proceeds as follows. Each i in an authorized subset $\mathcal{B} \in \Gamma^S_{\mathcal{P}}$ uses an intermediate scheme $\{\Gamma^S_{\mathcal{P}'}, \mathbb{Z}_p, \{\mathbb{Z}_p\}, {\psi'}^S\}$ (with the polynomial $a'_i(j)$) to distribute subshares $\hat{s}_{ij} \in \mathbb{Z}_p$ of their share s_i of secret $k \in \mathbb{Z}_p$ to each shareholder $j \in {\mathcal{P}'}^S$ (REDIST step 1). Each j then generates the new share s'_j (REDIST step 4):

$$s'_j = \sum_{i \in \mathcal{B}} b_i \hat{s}_{ij} \tag{12}$$

To allow the new shareholders to verify that SHARES-VALID and SUBSHARES-VALID hold, the old shareholders use the commitment function:

$$C(x) = g^x \tag{13}$$

where g is a generator for $\mathbb{Z}_p$:

$$\forall b \in \{1, \ldots, p-1\}\ \exists a \in \{1, \ldots, p-1\} : g^a \equiv b \bmod p \tag{14}$$

The old shareholders $i \in \mathcal{B}$ ($\mathcal{B} \in \Gamma^S_{\mathcal{B}}$) broadcast the commitment to the secret g^k, shares g^{s_i}, and coefficients of the polynomial $g^{a_{i1}} \ldots g^{a_{i(m'-1)}}$ (REDIST Step 2 in Fig. 5). The new shareholders $j \in \mathcal{P}'$ then verify that (REDIST Step 3):

$$g^{\hat{s}_{ij}} \equiv g^{s_i} \prod_{l=1}^{m'-1} (g^{a'_{il}})^{j^l} \tag{15}$$

for each $i \in \mathcal{B}$, and

$$g^k \equiv \prod_{i \in \mathcal{B}} (g^{s_i})^{b_i} \quad \text{where} \quad b_i = \prod_{l \in \mathcal{B}, l \neq i} \frac{l}{(l-i)} \tag{16}$$

5.3 Discussion

To emphasize the shortcomings in the naïve extension of Desmedt and Jajodia's redistribution protocol [DJ97] by Feldman's VSS scheme [Fel87], we present an alternative verification mechanism for secret redistribution for Shamir's scheme [Sha79] that still requires the new shareholders to obtain the commitment to the secret. Consider redistribution of a secret k from the scheme $\{\Gamma^S_{\mathcal{P}}, \mathbb{Z}_p, \{\mathbb{Z}_p\}, \psi^S\}$to the scheme $\{\Gamma^S_{\mathcal{P}'}, \mathbb{Z}_p, \{\mathbb{Z}_p\}, {\psi'}^S\}$. Suppose we knew the shares s_i of the old shareholders $i \in \mathcal{B}$ ($\mathcal{B} \in \Gamma^S_{\mathcal{P}}$) and the coefficients

Verifiable Secret Redistribution protocol for Shamir's sharing scheme:
INITIAL: To use the (m, n) scheme $\{\Gamma^S_{\mathcal{P}}, \mathbb{Z}_p, \{\mathbb{Z}_p\}, \psi^S\}$ to distribute a secret $k \in \mathbb{Z}_p$:

1. For each $i \in \mathcal{P}$, use the polynomial $a(i) = k + a_1 i + \ldots + a_{m-1} i^{m-1}$ to compute the shares s_i of k, and send s_i to $i \in \mathcal{P}$ over a private channel.
2. Use g to generate $g^k, g^{a_1} \ldots g^{a_{m-1}}$, and send them to all $i \in \mathcal{P}$ over the broadcast channel.
3. For each $i \in \mathcal{P}$, verify that:

$$g^{s_i} \equiv g^k \prod_{l=1}^{m-1} (g^{a_l})^{i^l}$$

If the condition holds, i broadcasts a "commit" message. Otherwise, i broadcasts an "abort" message.

4. If all $i \in \mathcal{P}$ agree to commit, each i stores s_i and g^k. Otherwise, they abort the protocol.

REDIST: To redistribute k from the (m, n) scheme $\{\Gamma^S_{\mathcal{P}}, \mathbb{Z}_p, \{\mathbb{Z}_p\}, \psi^S\}$ to the (m', n') scheme $\{\Gamma^S_{\mathcal{P}'}, \mathbb{Z}_p, \{\mathbb{Z}_p\}, {\psi'}^S\}$

1. For each $i \in \mathcal{B}$ ($\mathcal{B} \in \Gamma^S_{\mathcal{P}}$), use the polynomial $a'_i(j) = s_i + a'_{i1} j + \ldots + a'_{i(m'-1)} j^{m'-1}$ to compute the subshares $\hat{s}_{ij}$ of s_i, and send $\hat{s}_{ij}$ to the corresponding $j \in \mathcal{P}'$ over a private channel.
2. For each $i \in \mathcal{P}$, use g to generate $g^{s_i}, g^{a'_{i1}} \ldots g^{a'_{i(m'-1)}}$, and send them to all $j \in \mathcal{P}'$ over the broadcast channel.
3. For each $j \in \mathcal{P}'$, verify that:

$$\forall i \in \mathcal{B} : g^{\hat{s}_{ij}} \equiv g^{s_i} \prod_{l=1}^{m'-1} (g^{a'_{il}})^{j^l}$$

and:

$$g^k \equiv \prod_{i \in \mathcal{B}} (g^{s_i})^{b_i} \quad \text{where} \quad b_i = \prod_{l \in \mathcal{B}, l \neq i} \frac{l}{(l-i)}$$

If the conditions hold, j broadcasts a "commit" message. Otherwise, j broadcasts an "abort" message.

4. If all $j \in \mathcal{P}'$ agree to commit, each j generates s'_j:

$$s'_j = \sum_{i \in \mathcal{B}} b_i \hat{s}_{ij}$$

and stores s'_j and g^k. Otherwise, they abort the protocol.

Figure 5: Verifiable secret redistribution protocol for the redistribution of shares from Shamir's (m,n) sharing scheme [Sha79] to Shamir's (m', n') scheme.

of the polynomial $a_i(j)$ used by i to distribute the subshares $\hat{s}_{ij}$ of s_i. We could then interpolate the $m'-1$ degree polynomial that a central dealer could have used to distribute shares s'_j of k to new shareholders $j \in \mathcal{P}'$ directly:

$$
\begin{aligned}
s'_j &= \sum_{i \in \mathcal{B}} b_i \hat{s}_{ij} \qquad \text{(Eqn. (6))} \\
&= \sum_{i \in \mathcal{B}} b_i \left(s_i + a'_{i1} j + \ldots + a'_{i(m'-1)} j^{m'-1} \right) \qquad \text{(REDIST Step 1 in Fig. 5)} \\
&= \sum_{i \in \mathcal{B}} b_i s_i + \sum_{i \in \mathcal{B}} b_i a'_{i1} j + \ldots + \sum_{i \in \mathcal{B}} b_i a'_{i(m'-1)} j^{m'-1} \qquad \text{(Eqn. (11))} \\
&= k + \left(\sum_{i \in \mathcal{B}} b_i a'_{i1} \right) j + \ldots + \left(\sum_{i \in \mathcal{B}} b_i a'_{i(m'-1)} \right) j^{m'-1} \qquad \text{(Eqn. (11))}
\end{aligned}
\tag{17}
$$

We might be tempted to use a new check similar to that in Feldman's VSS scheme to verify the validity of the shares held by new shareholders. Suppose each $i \in \mathcal{B}$ broadcasts the same information as they did in the specialized VSR protocol (REDIST Step 2 in Fig. 5). Each $j \in \mathcal{P}'$ then verifies that s'_j is a valid share of k with the following equation:

$$
g^{s'_j} = g^k g^{\left(\sum_{i \in \mathcal{B}} b_i a'_{i1}\right) j} \ldots g^{\left(\sum_{i \in \mathcal{B}} b_i a'_{i(m'-1)}\right) j^{m'-1}} \tag{18}
$$

Eqn. (18) follows from Eqn. (17) and the homomorphic properties of exponentiation. Since finding discrete logs is intractable, no j can learn k from the broadcast of g^k.

Even though the new check in Eqn. (18) appears similar to that of Feldman's VSS scheme in Eqn. (8) (with $C(x) = g^x$), it is subtly different from our use of Feldman's scheme to verify that SUBSHARES-VALID holds. More specifically, in our use of Feldman's scheme a single old shareholder $i \in \mathcal{B}$ proves to the n' new shareholders $j \in \mathcal{P}'$ that it distributed valid subshares. In the new check suggested by Eqn. (18), the m shareholders $i \in \mathcal{B}$ prove that they distributed valid subshares of valid shares to the n' new shareholders $j \in \mathcal{P}'$. To use Feldman's scheme, we require that each i broadcast only the commitments to the shares g^{s_i} and coefficients of the polynomial $g^{a_{i1}} \ldots g^{a_{i(m'-1)}}$. For j to use the new check, we require that each i broadcast in addition the commitment to the secret g^k (as required in our VSR protocol in Sec. 4).

6 Summary and future work

We have presented a protocol to verifiably redistribute shares of secrets between different threshold schemes. We proved that new shareholders have valid shares after redistribution if SHARES-VALID and SUBSHARES-VALID hold, and have given the corresponding verifications. We showed that our protocol guards against faulty behavior by up to $n - m$ of the old shareholders provided that $m > \frac{n}{2}$. In our presentation, we assumed that there exist commitment functions that are hard to invert, and that there exist reliable broadcast communication channels among all participants and private channels between every pair of participants. The primary contribution of our work is that in our protocol, new shareholders can verify the validity of their shares after redistribution between different threshold schemes.

As part of our future work, we will investigate ways to identify faulty old shareholders during redistribution, and to relax the bounds on the number of non-faulty new shareholders. We are currently implementing our protocol as part of the Carnegie Mellon PASIS survivable storage system [WBP+01, WBS+00] to evaluate its performance costs.

References

[BBCM92] Bob Blakley, G. R. Blakley, Agnes H. Chan, and James L. Massey. Threshold schemes with disenrollment. In Ernest F. Brickell, editor, *Proc. of CRYPTO 1992, the 12th Ann. Intl. Cryptology Conf.*, volume 740 of *Lecture Notes in Computer Science*, pages 540–548. Intl. Assoc. for Cryptologic Research, Springer-Verlag, August 1992.

[BCSV96] Carlo Blundo, Antonella Cresti, Alfredo De Santis, and Ugo Vaccaro. Fully dynamic secret sharing schemes. *Theoretical Computer Science*, 165(2):407–440, October 1996.

[Ben87] Josh Cohen Benaloh. Secret sharing homomorphisms: Keeping shares of a secret secret. In Andrew M. Odlyzko, editor, *Proc. of CRYPTO 1986, the 6th Ann. Intl. Cryptology Conf.*, volume 263 of *Lecture Notes in Computer Science*, pages 213–222. Intl. Assoc. for Cryptologic Research, Springer-Verlag, 1987.

[BKO93] Thomas Beth, Hans-Joachim Knobloch, and Marcus Otten. Verifiable secret sharing for monotone access structures. In *Proc. of the 1st ACM Intl. Conf. on Computer and Communications Security*, pages 189–194. ACM SIGSAC, November 1993.

[Bla79] G. R. Blakley. Safeguarding cryptographic keys. In *Proc. of the Natl. Computer Conf.*, volume 48 of *American Federation of Information Processing Societies Proceedings*, 1979.

[Cac95] Christian Cachin. On-line secret sharing. In Colin Boyd, editor, *Proc. of the 5th IMA Conf. on Cryptography and Coding*, volume 1025 of *Lecture Notes in Computer Science*, pages 90–198. The Inst. of Mathematics and its Applications, Springer-Verlag, December 1995.

[CGMA85] Benny Chor, Shafi Goldwasser, Silvio Micali, and Baruch Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults (Extended abstract). In *Proc. of the 26th IEEE Ann. Symp. on Foundations of Computer Science*, pages 383–395. IEEE, October 1985.

[Des97] Yvo Desmedt. Some recent research aspects of threshold cryptography. In E. Okamoto, G. Davida, and M. Mambo, editors, *Proc. of the 1st Intl. Information Security Workshop*, volume 1396 of *Lecture Notes in Computer Science*, pages 158–173. Springer-Verlag, September 1997.

[DJ97] Yvo Desmedt and Sushil Jajodia. Redistributing secret shares to new access structures and its applications. Technical Report ISSE TR-97-01, George Mason University, Fairfax, VA, July 1997.

[Fel87] Paul Feldman. A practical scheme for non-interactive verifiable secret sharing. In *Proc. of the 28th IEEE Ann. Symp. on Foundations of Computer Science*, pages 427–437. IEEE, October 1987.

[FGMY97a] Yair Frankel, Peter Gemmell, Philip D. MacKenzie, and Moti Yung. Optimal resilience proactive public-key cryptosystems. In *Proc. of the 38th IEEE Ann. Symp. on Foundations of Computer Science*, pages 384–393. IEEE, October 1997.

[FGMY97b] Yair Frankel, Peter Gemmell, Philip D. MacKenzie, and Moti Yung. Proactive RSA. In Burton S. Kaliski Jr, editor, *Proc. of CRYPTO 1997, the 17th Ann. Intl. Cryptology Conf.*, volume 1294 of *Lecture Notes in Computer Science*, pages 440–454. Intl. Assoc. for Cryptologic Research, Springer-Verlag, August 1997.

[FMY99] Yair Frankel, Philip D. MacKenzie, and Moti Yung. Adaptively-secure optimal-resilience proactive RSA. In Kwok-Yan Lam, Eiji Okamoto, and Chaoping Xing, editors, *Proc. of ASIACRYPT1999, the 5th Intl. Conf. on the Theory and Application of Cryptology and Information Security*, volume 1716 of *Lecture Notes in Computer Science*, pages 180–194. Intl. Assoc. for Cryptologic Research, Springer-Verlag, November 1999.

[FMY01] Yair Frankel, Philip D. MacKenzie, and Moti Yung. Adaptive security for the additive-sharing based proactive RSA. In Kwangjo Kim, editor, *Proc. of PKC 2001, the 4th Intl. Workshop on Practice and Theory in Public Key Cryptography*, volume 1992 of *Lecture Notes in Computer Science*, pages 240–263. Springer-Verlag, Febraury 2001.

[GJKR96] Rosario Gennaro, Stanislaw Jarecki, Hugo Krawczyk, and Tal Rabin. Robust threshold DSS signatures. In Ueli M. Maurer, editor, *Proc. of EUROCRYPT 1996, the Intl. Conf. on the Theory and Application of Cryptographic Techniques*, volume 1070 of *Lecture Notes in Computer Science*, pages 354–371. Intl. Assoc. for Cryptologic Research, Springer-Verlag, May 1996.

[GM95] Rosario Gennaro and Silvio Micali. Verifiable secret sharing as secure computation. In Louis C. Guillou and Jean-Jacques Quisquater, editors, *Proc. of EUROCRYPT 1995, the Intl. Conf. on the Theory and Application of Cryptographic Techniques*, volume 921 of *Lecture Notes in Computer Science*, pages 168–182. Intl. Assoc. for Cryptologic Research, Springer-Verlag, May 1995.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to prove all NP statements in zero-knowledge and a methodology of cryptograhpic protocol design. In Andrew M. Odlyzko, editor, *Proc. of CRYPTO 1986, the 6th Ann. Intl. Cryptology Conf.*, volume 263 of *Lecture Notes in Computer Science*, pages 171–185. Intl. Assoc. for Cryptologic Research, Springer-Verlag, 1987.

[HJJ+97] Amir Herzberg, Markus Jakobsson, Stanislaw Jarecki, Hugo Krawczyk, and Moti Yung. Proactive public key and signature systems. In *Proc. of the 4th ACM Intl. Conf. on Computer and Communications Security*, pages 100–110. ACM SIGSAC, April 1997.

[HJKY95] Amir Herzberg, Stanislaw Jarekci, Hugo Krawczyk, and Moti Yung. Proactive secret sharing or: How to cope with perpetual leakage. In Don Coppersmith, editor, *Proc. of CRYPTO 1995, the 15th Ann. Intl. Cryptology Conf.*, volume 963 of *Lecture Notes in Computer Science*, pages 339–352. Intl. Assoc. for Cryptologic Research, Springer-Verlag, August 1995.

[OY91] Rafail Ostrovsky and Moti Yung. How to withstand mobile virus attacks. In *Proc. of the 10th Ann. ACM Symp. on Principles of Distributed Computing*, pages 51–59. ACM SIGACT and ACM SIGOPS, August 1991.

[Ped91] Torben Pryds Pederson. Non-iteractive and information-theoretic secure verifiable secret sharing. In Joan Feigenbaum, editor, *Proc. of CRYPTO 1991, the 11th Ann. Intl. Cryptology Conf.*, volume 576 of *Lecture Notes in Computer Science*, pages 129–140. Intl. Assoc. for Cryptologic Research, Springer-Verlag, August 1991.

[Rab94] Tal Rabin. Robust sharing of secrets when the dealer is honest or cheating. *Journal of the ACM*, 41(6):1089–1109, November 1994.

[Rab98] Tal Rabin. A simplified approach to threshold and proactive RSA. In Hugo Krawczyk, editor, *Proc. of CRYPTO 1998, the 18th Ann. Intl. Cryptology Conf.*, volume 1462 of *Lecture Notes in Computer Science*, pages 89–104. Intl. Assoc. for Cryptologic Research, Springer-Verlag, August 1998.

[RBO89] Tal Rabin and Michael Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proc. of the 21st Symp. on the Theory of Computing*, pages 73–85. Assoc. for Computing Machinery, May 1989.

[Sha79] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, November 1979.

[WBP+01] Jay J. Wylie, Mehmet Bakkaloglu, Vijay Pandurangan, Michael W. Bigrigg, Semih Oguz, Ken Tew, Cory Williams, Gregory R. Ganger, and Pradeep K. Khosla. Selecting the right data distribution scheme for a survivable storage system. Technical Report CMU-CS-01-120, Sch. of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, May 2001.

[WBS+00] Jay J. Wylie, Michael W. Bigrigg, John D. Strunk, Gregory R. Ganger, Han Kiliççöte, and Pradeep K. Khosla. Survivable information storage systems. *IEEE Computer*, pages 61–68, August 2000.

[ZSvR00] Lidong Zhou, Fred B. Schneider, and Robbert van Renesse. COCA: A secure distributed on-line certification authority. Technical Report TR2000-1828, Dept. of Computer Science, Cornell University, Ithaca, NY 14853, December 2000.