Dealer-Leakage Resilient Verifiable Secret Sharing

Ruxandra F. Olimid*
ruxandra.olimid@fmi.unibuc.ro

September 19, 2014

Abstract

Verifiable Secret Sharing (VSS) guarantees that honest parties reconstruct a consistent secret even in the presence of a malicious dealer that distributes invalid shares. We empower the dishonest dealer and consider the case when he subliminally leaks information in valid shares, allowing an adversary to access the secret prior to the reconstruction phase. We define the concept of Dealer-Leakage Resilient Verifiable Secret Sharing (DLR-VSS) as a stronger notion of VSS that achieves security under this settings. We propose an efficient DLR-VSS and prove its properties in the semi-honest adversarial model.

1 Introduction

Secret sharing is a cryptographic primitive that allows a *dealer* to split a secret into multiple *shares* and distribute them to distinct parties; *authorized* sets of parties can reconstruct the secret, while the others obtain no information about the secret (in case of *perfect* secret sharing). Traditionally, the dealer is trusted; however, this does not always hold. *Verifiable Secret Sharing* (VSS) stands against a dealer who sends invalid shares to players with the goal to break the *consistency* of the scheme (i.e. distinct sets of *honest* players reconstruct different secrets). To achieve its objective, VSS allows the parties to verify the validity of their shares [CGMA85].

Although VSS captures a particular malicious behavior of the dealer, other dishonest strategies exist. In this paper, we consider a dealer that may subliminally send information in valid shares, giving the attacker a significant advantage to reconstruct the secret. At first, the approach might seem strange, since the dealer knows the secret and can simply send it over a secure communication channel placed outside the settings of the protocol. However, in practice this easily leads to dealer discard; the dealer's behavior must remain indistinguishable from genuine to anyone else except the attacker, otherwise his malicious behavior is revealed. To address this problem, we introduce the concept of *Dealer-Leakage Resilience Verifiable Secret Sharing* (DLR-VSS), a secret sharing that achieves both verifiability of shares and dealer-leakage resilience.

1.1 Related Work

VSS. Blakley [Bla79] and Shamir [Sha79] independently introduce secret sharing in 1979. Secret sharing permits a *dealer* to share a secret among n parties such that *authorized* sets of parties reconstruct the secret by combining their shares. The particular case when at least $t+1$ out of n parties are required for reconstruction is called $(n,t)-$*threshold secret sharing*. In 1985, Chor et al. extend ordinary secret sharing in the presence of *active* corruptions by allowing an adversary to corrupt at most t players (including the dealer) in an arbitrary way and define VSS (*Verifiable Secret Sharing*) [CGMA85]. Recall that the traditional secret sharing is secure in the presence of *passive* adversaries only (e.g. Tompa and Woll showed that Shamir's scheme is susceptible to cheating [TW88]). VSS allows the players to verify that their shares are *consistent* and permit *honest* parties to reconstruct a unique well-defined secret even in the presence of a *faulty* dealer. However, VSS aims secrecy only in the presence of a *trusted* dealer; to the best of our knowledge, no notion

*Department of Computer Science, University of Bucharest, Romania.

of secret sharing aims secrecy in case of a faulty dealer, although such attacks were recently revealed [Oli14]. The notion might seem contradictory at first, but we show that it is possible to achieve secrecy under the assumption that the dealer is only allowed to subliminally leak information and hence communication outside the settings of the protocol is forbidden.

Information leakage. Simmons is the first to consider *subliminal channels* as a way to achieve secure communication over insecure channels [Sim83] and include them in cryptographic algorithms [Sim84]. Young and Yung extend his idea and show that *black-box* model permits serious flaws in the cryptographic devices that are exhibited by information leakage [YY96]. They introduce SETUP (*Secretly Embedded Trapdoor with Universal Protection*), a malicious technique performed by the manufacturer of a cryptographic device to leak secret information. The goal of SETUP is to give an attacker overwhelming advantage to break the security, while it remains undetectable by other parties. This preserves the dishonest implementation hidden and hence avoids its replacement with a genuine one [YY96, YY97]. Recently, SETUP was introduced against secret sharing schemes [Oli14]. We remark that VSS are not SETUP resilient by definition (it is immediate that Pedersen's scheme remains susceptible to the same SETUP attack mounted against Shamir's scheme [Oli14]). We highlight the distinction from leakage-resiliency in the sense of the stronger concept that formally models the side-channels attacks in the current literature and allows the adversary to gain knowledge of arbitrary, but bounded information, by performing timings attacks, power consumption, electromagnetic radiation or fault attacks [MR04, SMY09, SPY$^+$10]. In this paper we restrict to the internal modification of the algorithm in the sense of SETUP and denote by *leakage-dishonest dealer* a malicious dealer that tries to leak information while he is modeled as a black-box with indistinguishable inputs and outputs from the original protocol, respectively by *dealer-leakage resilience* a scheme that remains secure in the presence of a leakage-dishonest dealer.

1.2 Motivations and Contributions

VSS only aims secrecy in the presence of a trusted dealer. Our work extends VSS and considers a *leakage-dishonest dealer* as a faulty dealer that may subliminally hide information in valid shares, based on a strategy implemented by the adversary previous to the protocol execution.

First, we define DLR-VSS (*Dealer-Leakage Resilient Verifiable Secret Sharing*) as a VSS that aims security in the presence of a *leakage-dishonest dealer*; we say that such a scheme achieves *dealer-leakage resilience*. To prevent trivial win, any communication between the dealer and the players outside the settings of the protocol is forbidden and the dealer's malicious behavior must remain indistinguishable from honest in the black-box model. Our definition considers the SETUP settings [YY97]: the sharing mechanism is modeled as a black-box that performs secret sharing and distributes the shares to the players; the device is contaminated in advance and the adversary gains no direct information about the input at runtime [Oli14]. We highlight that our definition of DLR-VSS does not claim to be leakage-resilient in the sense of [MR04,SMY09,SPY$^+$10] and subsequent work.

Second, we propose a DLR-VSS scheme for $n \geq 2t + 1$ in the synchronous communication settings and prove its properties in the semi-honest security model. Our construction requires three rounds in the sharing phase and it is efficient in terms of communication messages.

Our work considers DLR-VSS due to the importance of the VSS as a cryptographic primitive. Naturally, defining *Dealer-Leakage Resilient Secret Sharing* (DLR-SS) as an extension of ordinary secret sharing is immediate - we introduce the notion for completeness.

VSS lies at the core of a multitude of cryptographic protocols (e.g. secure multi-party computation). Replacing the underlying VSS by its enhanced analogous DLR-VSS should lead to stronger security. In the following, we motivate our work by its applicability to a practical use case scenario: secure data storage.

Use case. Storage systems can provide long-time data privacy by using secret sharing instead of encryption [WBS$^+$00, SB05, SGMV09]. A client-side application uses threshold secret sharing to spread the data to a set of independent servers that provide persistent storage of shares. The client first performs sharing and then distributes the shares to the storage nodes. The usage of threshold secret sharing provides availability

and data recovery. The system relies on multiple points of trust, since the data remains secure as long as the compromised number of servers does not reach the threshold. To access the stored information, secret reconstruction is performed: the client requests the shares from the storage nodes and reconstructs the data. The client application communicates with the storage servers to read and write information and thus it keeps decentralization hidden from the user.

In the following, we consider a contaminated client-side application. It apparently acts genuine, so when the user accesses the client, data is split into shares and stored to a set of independent servers. However, it subliminally leaks secret information in shares that are distributed to a subset of distinguished servers. The security of the system reduces to breaking the servers that contains the leaked information. Depending on the scheme that is used, this may lead to only two points of trust, independently of the given threshold [Oli14].

VSS safeguards against malicious servers that send invalid shares to the client-side application for data recovery; similarly, DLR-VSS may serve as a solution to prevent the decreasing in the number of points of trust and maintain the original threshold. Implementing such solutions and analyzing their efficiency in practice are subjects of future work.

1.3 Organization

Section 2 describes the communication settings and introduces the definitions of the underlying notions. Section 3 introduces the concept of leakage-dishonest dealer and defines DLR-VSS as an extension of VSS. Section 4 presents a 3-round DLR-VSS for $n \geq 2t + 1$ in the semi-honest adversarial model. Section 5 concludes.

2 Preliminaries

2.1 Synchronous Communication Model

We consider the synchronous communication model. A protocol consists of several rounds; during one round, each party can perform local computation and send one or more messages to other players; also, by the end of the round all participants receive the messages sent during that round.

For our construction in Section 4, we consider that each party communicates with the dealer via a unidirectional secure channel (our proposal only requires secure communication from the players to the dealer); no other secure links are required. In addition, a public broadcast channel allows communications between all parties, including the dealer.

2.2 Verifiable Secret Sharing

Let $\mathcal{P} = \{P_1, \ldots, P_n\}$ be a set of parties and $\mathcal{D}$ a privileged party called the *dealer*. In the following, we denote by VSS-Share the *sharing phase*, during which the dealer $\mathcal{D}$ splits a secret S into multiple shares and distributes them to the non-dealer parties in $\mathcal{P}$ and by VSS-Rec the *reconstruction phase*, during which the players cooperate to reconstruct S. We further assume that the secret S lies in a finite field $\mathbb{F}$; in Section 4 we particularize the finite field to a subset of $\mathbb{Z}_p^*$.

Definition 1. *A (n, t)-VSS is a 2-phases protocol (VSS-Share, VSS-Rec) between a set of parties $\mathcal{P} = \{P_1, \ldots, P_n\} \cup \mathcal{D}$, one of whom is a distinguished party $\mathcal{D}$ called the dealer that holds an initial input $S \in \mathbb{F}$ (the secret), that satisfies the following properties for a t-bounded adversary $\mathcal{A}$ (i.e. $\mathcal{A}$ can corrupt up to t non-dealer parties):*

1. *Secrecy: If $\mathcal{D}$ is honest, then $\mathcal{A}$ gains no information about S prior to VSS-Rec;*
2. *Correctness: If $\mathcal{D}$ is honest, the joint view of the honest parties output S at the end of VSS-Rec;*
3. *Commitment: If $\mathcal{D}$ is dishonest, the joint view of the honest parties output $S^* \in \mathbb{F}$ at the end of VSS-Rec.*

We stick to the above definition, also weaker or stronger notions of commitment exist [BKP11, GGOR13, RBO89, FGG+06].

A VSS is *efficient* if the total computation and communication performed by all parties is polynomial in the n and the size of the secret. With respect to the synchronous communication model, we assume that the round complexity is given by the number of communication rounds performed during the sharing phase [FGG+06, GIKR01]. In general, multiple rounds are possible during reconstruction and hence the round complexity should be defined as the total number of rounds in both the sharing and reconstruction phases [PCRR09, Agr12]. However, our protocol requires a single round during reconstruction, so throughout this paper by round complexity we mean the number of rounds in the sharing phase, as in [BKP11]. Because the broadcast channel is considered to be an expensive resource, a refinement introduces *broadcast complexity* to make distinction between unicast and broadcast communication rounds [GGOR13]. We ignore this distinction for the current work.

We base our proposal in Section 4 on Pedersen's scheme, a popular and widely used VSS [Ped91]. We skip here the description of Pedersen's VSS, but invite the reader to address the original work. We give next the definition of a *commitment scheme*, since we will need it later on this work.

Definition 2. *A commitment scheme is a protocol between two parties called committer and verifier that consists in two phases:*

1. *Commit: The committer holds a value x, computes $y = \mathsf{Commit}(x)$ and publishes y as the commitment that binds the value x (the binding property) without revealing it (the hidden property);*
2. *Open: The committer opens the commitment y by revealing an opening z to a verifier that knowing z can check if x is consistent with the commitment y.*

2.3 SETUP

Young and Yung showed that a cryptosystem implememented as a black-box (i.e. the inputs and outputs are externally accessible, while its internal algorithm or implementation remain inaccessible) can be designed in a way that leaks secret information to an attacker, while it remains indistinguishable from genuine to anyone else [YY96, YY97]. We review the definition of SETUP [YY97]

Definition 3. *Assume that C is a black-box cryptosystem with a publicly known specification. A (regular) SETUP mechanism is an algorithmic modification made to C to get C' such that:*

1. *The input of C' agrees with the public specifications of the input of C;*
2. *C' computes efficiently using the attacker's public encryption function e (and possibly other functions as well), contained within C';*
3. *The attacker's private decryption function d is not contained within C' and is known only by the attacker;*
4. *The output of C' agrees with the public specifications of the output of C;*
5. *The output of C and C' are polynomially indistinguishable to everyone except the attacker;*
6. *After the discovery of the specifics of the SETUP algorithm and after discovering its presence in the implementation (e.g. reverse-engineering of hardware tamper-proof device), users (except the attacker) cannot determine past (or ideally, future) keys.*

A cryptosystem that implements SETUP is called *contaminated* [YY96].

The main goal of a SETUP attack against secret sharing is to allow the attacker to reconstruct the secret previous to the reconstruction phase or even if reconstruction never takes place. Clearly, this ruins the

security of the scheme. To achieve its goal, the contaminated dealer hides secret information in apparently genuine shares. To reconstruct the secret under these settings, the attacker only needs access to the shares that contain the leaked information. SETUP was successfully mounted against secret sharing schemes that employ enough randomness; applied to Shamir's secret scheme, the secret can be leaked in two distinct valid shares, regardless the threshold [Oli14].

3 Dealer-Leakage Resilient Verifiable Secret Sharing

3.1 Leakage-Dishonest Dealer

We extend the capabilities of the malicious dealer and define a *leakage-dishonest* dealer as a dealer that subliminally leaks secret information in valid shares. Unlike a traditional cheating dealer that distributes inconsistent shares, the goal of a leakage-dishonest dealer is to preserve consistency and to allow the adversary to recover the secret from the leaked information, previous to the reconstruction phase or even when no reconstruction takes place.

Our model straightness the ordinary cheating dealer in two ways:

1. The shares that contain the leaked information allow the adversary to reconstruct the secret (or at least parts of the secret); clearly, this breaks the secrecy of the scheme;

2. The malicious behavior of the dealer remains hidden (as long as the shares are indistinguishable from genuine for anyone except the attacker); this does not hold for a malicious dealer that distributes invalid shares, since his malicious behavior is usually revealed (note that this always holds in the semi-static security model).

The adversary can coordinate the dealer's actions such that they follow a predefined strategy, but only previous to the sharing phase. More precise, the dealer's behavior is modeled as a black-box that can be internally modified by the adversary previous to the execution of the protocol as long as the inputs and outputs remain polynomially indistinguishable to everyone except the attacker. This means that the adversary gains no direct knowledge of the shared secret. To prevent trivial win, all communication between the dealer and the adversary outside the settings of the protocol is forbidden (i.e. the dealer is not allowed to simply send the secret to the adversary). To achieve indistinguishability from genuine, the dealer must send well-formed shares. To resume, the dealer adversarial model satisfies SETUP definition [YY96, YY97].

Use case. For clearness, we exemplify the leakage-dishonest dealer within the secure data storage use case presented in Subsection 1.2. Before it is goes into production, an adversary (e.g. the manufacturer) contaminates the client-side application to implement leakage. But once the application is up and running, he gains no direct access on the input data: the contaminated client application cannot send information directly to the attacker since the additional communication can be detected and the system replaced with a trusted one. Nevertheless, the scheme becomes insecure: there is no need to break into all points of trust, since the attacker recovers the data once he gains access to the servers that stores the leaked information.

3.2 DLR-VSS Definition

VSS is a cryptographic primitive that achieves security against cheating participants, including the dealer [CGMA85]. However, it provides secrecy only when the dealer is trusted. We extend VSS and define the notion of DLR-VSS (*Dealer-Leakage Resilience - Verifiable Secret Sharing*) that achieves security in the presence of a leakage-dishonest dealer.

Definition 4. *A (n,t)-DLR-VSS is a (n,t)-VSS that satisfies the property of Dealer-Leakage Resilience: If $\mathcal{D}$ is leakage-dishonest, then $\mathcal{A}$ gains no information about S prior to* VSS-Rec *or the attack is revealed prior to* VSS-Rec.

We emphasize that dealer-leakage resilience models leakage in the theoretical settings of the protocol only and does not aim to stand against side-channels attacks or solve the general problem of leakage resilience. Its scope is to stand against internal modifications of the protocols settings. SETUP against secret sharing exemplifies such an attack [Oli14]; hence, DLR-VSS is defined to resist against a SETUP attack.

We propose a simple and general idea to construct DLR-VSS from VSS: restrict the dealer to employ randomness. This leads to the impossibility of the dealer to hide secret information in his messages, since they are all predetermined. We exemplify our approach in Subsection 4.2 when obtain a DLR-VSS from Pedersen's VSS.

Naturally, the dealer-leakage resilience property can also be considered an extension of a traditional (non-verifiable) secret sharing. For completeness, we introduce next the concept of *Dealer-Leakage Resilient Secret Sharing* (DLR-SS) for which the underlying secret sharing is not necessarily verifiable.

Definition 5. *A DLR-SS is a (non-verifiable) secret sharing scheme that satisfies the property of Dealer-Leakage Resilience: If $\mathcal{D}$ is leakage-dishonest, then $\mathcal{A}$ gains no information about S prior to* SS-Rec *or the attack is revealed prior to* SS-Rec*.*

4 A 3-Round DLR-VSS

4.1 Adversarial Model

The adversary is t-bounded (i.e. he can compromise at most t out of n non-dealer parties). We work in the *semi-honest* adversarial model, a particular case of malicious adversarial model for which the parties follow the protocol exactly. More precise, the adversary gains the knowledge of up to t *corrupt* parties but cannot coordinate their actions. A non-corrupted party is called *honest*. Moreover, we assume that the adversary is *adaptive* (i.e. he can corrupt any non-dealer party during the protocol execution as long as the number of corrupted parties is bounded by t) - and *rushing* (i.e. he can wait the honest parties to send their messages and afterwards send his corrupt messages during the same round).

In addition, the adversary implements a leakage-dishonest dealer previous to the execution of the protocol. Note that this means that the adversary can predefine the dealer's behavior, but looses all control once the protocol starts; in particular, the adversary gains no direct access to the secret or to the shares of honest parties.

We work in the computational security settings: the adversary is computational bounded and the black-box implementation of the leakage-dishonest dealer is distinguished from genuine with negligible probability.

4.2 Our proposal

We present a 3-round sharing, 1-round reconstruction DLR-VSS secure against a t-bounded adversary in the semi-honest security model for $n \geq 2t+1$. The protocol is similar to Pedersen's VSS [Ped91], except that we disallow the dealer to choose the polynomials and give this ability to the non-dealer parties.

Overview. Let p and q be two large primes such that q divides $p-1$ and G a subgroup of $\mathbb{Z}_p^*$ of order q with g as generator. Let $h \in G$ such that nobody knows x that satisfies $g^x = h$ in G and the discrete logarithm problem is difficult (i.e. the probability to obtain x such that $g^x = h$ in G is negligible).

Without loss of generality, we use Pedersen's commitment function for our construction: the committer commits to $a \in \mathbb{Z}_q$ by choosing $b \in \mathbb{Z}_q$ at random and computing $\mathsf{Commit}(a,b) = g^a h^b$; the commitment is later opened by revealing a and b [Ped91].

Fig.1 describes our proposal. Let $S = (S_1, S_2) \in \mathbb{Z}_q^2$ be the secret to be shared. The dealer $\mathcal{D}$ initiates the protocol by committing to S. In response, each party privately sends to $\mathcal{D}$ a pair of values (a_i', b_i') and publicly commits to his selection. These values, together with (S_1, S_2) uniquely determine the two polynomials $f(x)$ and $r(x)$. By the end of Round 3, each user P_i can compute a correct point (f_i, r_i) on the committed polynomials, where $f_i = f(i)$ and $r_i = r(i)$. Note that the dealer masks the points by adding them to the private values (a_i', b_i') shared with each user to achieve privacy over the public broadcast channel.

VSS – Share.

Round 1.

1. $\mathcal{D}$ sets $\tilde{a}_0 = S_1, \tilde{b}_0 = S_2$, and broadcasts $\mathsf{Com_0} = \mathsf{Commit}(\tilde{a}_0, \tilde{b}_0)$.

Round 2.

1. Every P_i, $i = 1, \ldots, n$ selects a pair $(a'_i, b'_i) \in \mathbb{Z}_q^2$, sends it securely to $\mathcal{D}$ and broadcasts $\mathsf{SCom_i} = \mathsf{Commit}(a'_i, b'_i)$.

Round 3.

1. Sort $\{\mathsf{SCom_i}, i = 1, \ldots, n\}$ and let $\mathcal{P}_S$ be the ordered set of parties. After reordering, let a_i, b_i and $\mathsf{Com_i}$ be the values sent by the i-th party in the ordered set $\mathcal{P}_S$, $i = 1, \ldots, n$.
2. $\mathcal{D}$ computes, for $i = 1 \ldots t$:
$$\tilde{a}_i = a_i + \cdots + a_{i+t}$$
$$\tilde{b}_i = b_i + \cdots + b_{i+t}$$
and obtains $f(x) = \tilde{a}_0 + \tilde{a}_1 x + \ldots \tilde{a}_t x^t$ and $r(x) = \tilde{b}_0 + \tilde{b}_1 x + \ldots \tilde{b}_t x^t$.
3. For every P_i, $\mathcal{D}$ broadcasts (f'_i, r'_i), where $f'_i = a'_i + f_i$ and $r'_i = b'_i + r_i$ with $f_i = f(i)$ and $r(i) = r_i$.

Local Computation. Every party discards $\mathcal{D}$ and halts the execution if $\mathcal{D}$ broadcasts (f'_i, r'_i) such that $\mathsf{Commit}(f'_i, r'_i) \neq \mathsf{SCom_i} \cdot \mathsf{Com_0} \prod_{j=1}^{t} (\prod_{k=j}^{j+t} \mathsf{Com_k})^{i^j}$.

VSS – Rec.

1. Party P_i is said to be confirmed if $\mathsf{Commit}(f_i, r_i) = \mathsf{Com_0} \prod_{j=1}^{t} (\prod_{k=j}^{j+t} \mathsf{Com_k})^{i^j}$.
2. Any t confirmed parties interpolate $f(x)$ to output $S_1 = f(0)$ and $r(x)$ to output $S_2 = r(0)$.

Figure 1: 3-Round (n, t)-DLR-VSS, $n \geq 2t + 1$

To ensure correctness, $\mathcal{D}$ is discarded if the broadcast messages are not consistent with the commitments sent in Round 2.

In the reconstruction phase, any $t + 1$ confirmed players recover the polynomials $f(x)$ and $r(x)$ to find $S_1 = f(0)$ and $S_2 = r(0)$.

Discussion. The main difference between our construction and Pedersen's scheme is that we no longer allow the dealer to choose the polynomials $f(x)$ and $r(x)$ on his own wish. Since the dealer employ no randomness, he cannot subliminally hide information in his messages and therefore we obtain dealer-leakage resilience.

Our proposal requires 3 rounds during sharing phase and 1 round in the reconstruction phase. The scheme remains efficient in terms of communication: it uses unidirectional private channels (although in the opposite direction than Pedersen's) and the sharing phase requires $\mathcal{O}(n)$ messages both on the secure channels and the broadcast channel.

4.3 Proofs

Theorem 4.1. *The construction in Fig.1 is a (n,t)-DLR-VSS in the semi-honest adversarial model for $n \geq 2t+1$.*

Proof. We prove the properties a DLR-VSS must satisfy:

Secrecy. In Rounds 1 and 2 the adversary gains knowledge on all $\mathsf{Com_i}$, $i = 0, \ldots, n$. From the hiding property of Pedersen's commitment function, the coefficients a_i and b_i remain secure. We remark the particular case of $\mathsf{Com_0}$ that reveals no information about $\tilde{a}_0 = S_1$ and $\tilde{b}_0 = S_2$. In Round 3, the adversary eavesdrops on $f'_i = a'_i + f_i$ and $r'_i = b'_i + r_i$. We note that a'_i and b'_i are randomly chosen and secure, hence f_i and r_i remain as secure as prior to the broadcast of (f'_i, r'_i).

Notice that all coefficients $\tilde{a}_i$, $\tilde{b}_i$, $i = 1, \ldots, n$ remain secure for any coalition of t or less parties. Nevertheless, the adversary has access to their commitments: $\mathsf{Commit}(\tilde{a}_i, \tilde{b}_i) = \prod_{j=i}^{i+t} \mathsf{Com}_j$; by definition, a t-bounded adversary also gains access to t points on $f(x)$, respectively $r(x)$. To conclude, the security follows directly from Pedersen's scheme.

Correctness. $\mathcal{D}$ is honest by definition, so he is never discarded and he always distributes correct pairs (f'_i, r'_i) to participants. In the semi-honest adversarial model all parties follow the protocol, so they always compute proper shares (f_i, r_i) and make them public in the reconstruction phase. To conclude, any $t+1$ or more parties interpolate their points to compute $S_1 = f(0)$ and $S_2 = r(0)$.

Commitment. If $\mathcal{D}$ is discarded, then all parties may assume a predefined value as the secret (S_1, S_2). Hence, we consider the case when $\mathcal{D}$ is not discarded. Under this scenario, $f(x)$ and $r(x)$ are t-degree polynomials and each party P_i holds a pair of points (f_i, r_i) at the end of Round 3. To conclude, we have to show the correctness of (f_i, r_i). Since $\mathcal{D}$ is not discarded, then $\mathsf{Commit}(f'_i, r'_i) = \mathsf{SCom_i} \cdot \mathsf{Com_0} \prod_{j=1}^{t} (\prod_{k=j}^{j+t} \mathsf{Com_k})^{i^j}$. P_i is honest (i.e. $\mathsf{SCom_i} = \mathsf{Commit}(a'_i, b'_i)$), so $\mathsf{Commit}(f_i, r_i) = \mathsf{Com_0} \cdot \prod_{j=1}^{t} (\prod_{k=j}^{j+t} \mathsf{Com_k})^{i^j}$ holds. This implies that the pair (f_i, r_i) is correct, unless corrupted $\mathcal{D}$ had broken the biding property of the commitment function.

Dealer-Leakage Resilience. The proof is complete if the *leakage-dishonest* dealer $\mathcal{D}$ cannot leak secret information about S without being discarded. Note that $\mathcal{D}$ is not allowed to communicate with the parties outside the settings of the protocol and he must preserve his messages indistinguishable from genuine. Hence, we consider all messages originating from $\mathcal{D}$.

First, it is immediate that $\mathcal{D}$ cannot use $\mathsf{Com_0}$ to leak information in Round 1 because no value is on his own choice: (S_1, S_2) is the secret to be shared and g, h are public parameters.

Second, we show that $\mathcal{D}$ cannot use (f'_i, r'_i) to leak information in Round 3. $\mathcal{D}$ is not discarded, so $\mathsf{Commit}(f'_i, r'_i) = \mathsf{SCom_i} \cdot \mathsf{Com_0} \prod_{j=1}^{t} (\prod_{k=j}^{j+t} \mathsf{Com_k})^{i^j}$ holds. This implies that $f'_i = a'_i + f_i$ and $r'_i = b'_i + r_i$, unless corrupted $\mathcal{D}$ had broken the biding property of the commitment function.

Both a'_i and f_i are independent of the dealer's strategy: P_i had selected a'_i in Round 2 and $f_i = f(i)$, where $f(x)$ is uniquely defined by its coefficients $\tilde{a}_0 = S_1$ and $\tilde{a}_i$, $i = 1, \ldots, t$. Therefore, f'_i is fixed. The same reasoning applies to r'_i. □

5 Conclusions and Future Work

We introduced the concept of DLR-VSS as a VSS that restricts a leakage-dishonest dealer to subliminally send secret information. We proposed a 3-round DLR-VSS scheme that is efficient in terms of message communication and proved its properties in the semi-honest security model. Future work includes the study of the lower bound for the number of communication rounds in the synchronous model and the construction

of a DLR-VSS in the asynchronous communication model. An interesting open problem to investigate is the construction of a DLR-VSS in the fully malicious adversarial model.

Acknowledgments

We thank IACR Cryptology ePrint Archive editor Tal Rabin for her comments on a previous version of the protocol.
This work was supported by the strategic grant POSDRU/159/1.5/S/137750, "Project Doctoral and Postdoctoral programs support for increased competitiveness in Exact Sciences research" cofinanced by the European Social Found within the Sectorial Operational Program Human Resources Development 2007-2013.

References

[Agr12] Shashank Agrawal. Verifiable secret sharing in a total of three rounds. *Inf. Process. Lett.*, 112(22):856–859, 2012.

[BKP11] Michael Backes, Aniket Kate, and Arpita Patra. Computational verifiable secret sharing revisited. In *ASIACRYPT*, pages 590–609, 2011.

[Bla79] G.R. Blakley. Safeguarding cryptographic keys. In *Proceedings of the 1979 AFIPS National Computer Conference*, pages 313–317. AFIPS Press, 1979.

[CGMA85] Benny Chor, Shafi Goldwasser, Silvio Micali, and Baruch Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract). In *FOCS*, pages 383–395, 1985.

[FGG+06] Matthias Fitzi, Juan A. Garay, Shyamnath Gollakota, C. Pandu Rangan, and K. Srinathan. Round-optimal and efficient verifiable secret sharing. In *TCC*, pages 329–342, 2006.

[GGOR13] Juan A. Garay, Clint Givens, Rafail Ostrovsky, and Pavel Raykov. Broadcast (and round) efficient verifiable secret sharing. In *ICITS*, pages 200–219, 2013.

[GIKR01] Rosario Gennaro, Yuval Ishai, Eyal Kushilevitz, and Tal Rabin. The round complexity of verifiable secret sharing and secure multicast. In *STOC*, pages 580–589, 2001.

[MR04] Silvio Micali and Leonid Reyzin. Physically observable cryptography (extended abstract). In *TCC*, pages 278–296, 2004.

[Oli14] Ruxandra F. Olimid. Setup in secret sharing schemes using random values. Cryptology ePrint Archive, Report 2014/184, 2014. `http://eprint.iacr.org/`.

[PCRR09] Arpita Patra, Ashish Choudhary, Tal Rabin, and C. Pandu Rangan. The round complexity of verifiable secret sharing revisited. In *CRYPTO*, pages 487–504, 2009.

[Ped91] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *CRYPTO*, pages 129–140, 1991.

[RBO89] Tal Rabin and Michael Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In *STOC*, pages 73–85, 1989.

[SB05] Arun Subbiah and Douglas M. Blough. An approach for fault tolerant and secure data storage in collaborative work environments. In *StorageSS*, pages 84–93, 2005.

[SGMV09] Mark W. Storer, Kevin M. Greenan, Ethan L. Miller, and Kaladhar Voruganti. Potshards - a secure, recoverable, long-term archival storage system. *TOS*, 5(2), 2009.

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

[Sim83] Gustavus J. Simmons. The prisoners' problem and the subliminal channel. In *CRYPTO*, pages 51–67, 1983.

[Sim84] Gustavus J. Simmons. The subliminal channel and digital signature. In *EUROCRYPT*, pages 364–378, 1984.

[SMY09] François-Xavier Standaert, Tal Malkin, and Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. In *EUROCRYPT*, pages 443–461, 2009.

[SPY+10] François-Xavier Standaert, Olivier Pereira, Yu Yu, Jean-Jacques Quisquater, Moti Yung, and Elisabeth Oswald. Leakage resilient cryptography in practice. In Ahmad-Reza Sadeghi and David Naccache, editors, *Towards Hardware-Intrinsic Security*, Information Security and Cryptography, pages 99–134. Springer Berlin Heidelberg, 2010.

[TW88] Martin Tompa and Heather Woll. How to share a secret with cheaters. *J. Cryptology*, 1(2):133–138, 1988.

[WBS+00] Jay J. Wylie, Michael W. Bigrigg, John D. Strunk, Gregory R. Ganger, Han Kiliççöte, and Pradeep K. Khosla. Survivable information storage systems. *Computer*, 33(8):61–68, 2000.

[YY96] Adam L. Young and Moti Yung. The dark side of "black-box" cryptography, or: Should we trust capstone? In *CRYPTO*, pages 89–103, 1996.

[YY97] Adam L. Young and Moti Yung. Kleptography: Using cryptography against cryptography. In *EUROCRYPT*, pages 62–74, 1997.