# Theory and Practice of Verifiable Secret Sharing

Rosario Gennaro

Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
Proposal for Thesis Research in Partial Fulfillment
of the Requirements for the Degree of
Doctor of Philosophy

## 1 Introduction

Secret Sharing is a fundamental notion for secure cryptographic design. In a Secret Sharing protocol a dealer shares a secret among $n$ servers so that subsets of $t+1$ or more servers can later reconstruct the secret, while subsets of $t$ or less servers have no information about it. This is what it is usually called a threshold Secret Sharing scheme.

A Verifiable Secret Sharing (VSS for short in the following) protocol adds fault-tolerance to the above tasks. Informally, as first introduced by Chor, Goldwasser,

Micali and Awerbuch in [4], a VSS protocol consists of a two-stage protocol. There are $n$ servers, $t$ of which may be bad and deviate from their prescribed instructions in any arbitrary way. One of the players, the dealer, possesses a value $s$ as a secret input. In the first stage, the dealer commits to a unique value $v$ (no matter what the bad players may do); moreover, $v = s$ whenever the dealer is honest. In the second stage, the already committed value $v$ will be recovered by all good players (no matter what the bad players might do).

VSS protocols have been shown in the past to be of crucial importance in the design of secure cryptographic protocols. Indeed there has been a very prolific production of VSS protocols in the literature, see for example [4, 1, 3, 5, 7]

# 2 The definition

Despite the centrality and the maturity of this concept (almost 10 years passed from its original introduction), we shall advocate that a stronger and better definition of a VSS is needed in order to achieve the very desirable property of reducibility for secure protocols. We shall then provide the first provably correct implementation of this stronger definition.

Reducibility is an essential tool for secure protocols design because it allows such protocols to be built in a modular way. In our case this means that one can first design a protocol P for a given task assuming that an "abstract" and "perfect" VSS protocol exists. Then one designs a correct (according to a given definition!) VSS

protocol Q. Finally one substitutes Q in place of the abstract sub-protocol in P. However this way of proceeding yields a secure protocol P if and only if the definition of VSS satisfied by Q enjoys the reducibility property. Unfortunately no prior notion of a VSS provably guarantees reducibility. Thus, one of the goals of this thesis is to provide such a definition and VSS protocols that satisfy it.

## 2.1 Previous work

Several definitions and protocols for VSS have been proposed in the past ten years (E.g., [4, 1, 3, 5, 7].) We contend, however, that these notions and these protocols are of *very limited use.* In fact, their security concerns *begin when the dealer's secret is committed, and end when it is recovered.* That is these notions do not concern themselves with what happens when VSS is used as a subprotocol. Because in many applications running a *single* VSS protocol is exactly what is wanted, these prior definitions and protocols are totally adequate in those scenarios. However, they are not adequate in more general scenarios, since it is by now a well-known phenomenon that protocols that are secure by themselves, cease to be secure when used as a sub-protocols. In these cases the security of the entire protocol must be proven *from scratch* (for instance, this is the case in [5] where they use VSS as a tool to reach Byzantine agreement) rather than in a more natural and elegant *modular way.*

A notion of security that guarantees reducibility has been presented by Micali and Rogaway [6] for the problem of *function evaluation*, but not for general multiparty

protocols.

## 2.2 Our plan

We plan to extend reducibility-guaranteeing notions of security to Verifiable Secret Sharing protocols and concretely exhibit VSS protocols that provably satisfy these notions. More precisely, we plan to achieve the following goals:

1. Propose a new definition of VSS casted as a special istance of secure function evaluation.

2. Compare our new notion with the previously proposed ones, and show that it is *strictly and inherently stronger.*

3. Modify earlier VSS protocols of [1, 7] and show that such protocols with our modifications are secure according to our new definition.

# 3 VSS in the access structure model

Traditional VSS protocols have *threshold security.* That is they maintain their security until a given threshold of the servers in the network correctly performs the protocol. For example, above we described the notion of VSS with threshold $t$. However, implicit in the notion of threshold security is the idea that all servers are identical and that an adversary can corrupt server A with the same effort that it would take to

corrupt server B. This is, generally, not the case. Different servers may have different security levels and corrupting one may be easier than breaking into another one.

Historically this is why the notion of access structure has been introduced for Secret Sharing protocols. Given the set $\mathcal{S}$ of $n$ servers, a (monotone) access structure $\mathcal{ACC}$ on $\mathcal{S}$ is a family of subsets of $\mathcal{S}$

$$\mathcal{ACC} \subseteq 2^{\mathcal{S}}$$

with the following monotonicity property:

$$\forall S' \supset S \; S \in \mathcal{ACC} \implies S' \in \mathcal{ACC}$$

A basis of $\mathcal{ACC}$ is the minimal set of authorized subsets $\{S_1, \ldots, S_k\}$ with the following property:

$$\forall S' \in \mathcal{ACC}, \; \exists S_i \text{ such that } S' \supseteq S_i$$

Given a specific access structure $\mathcal{ACC}$, the dealer shares the secret in such a way that only subsets of players in $\mathcal{ACC}$ (usually called authorized subsets) can actually reconstruct the secret, while non-authorized subsets have no information about it.

Secret Sharing over monotone access structures has been studied in depth in the past. This research culminated in the work of Benaloh and Leichter [2] that showed how to share a secret over *any* monotone structure.

We plan to adapt the notion of Verifiable Secret Sharing to the access structure model.

In particular, we plan to extend the definition for threshold VSS to general access structure VSS. Since our definition of VSS is based on the more general definition of secure function evaluation, we will also develop a definition of secure computation over monotone access structures.

## 3.1 Admissible adversaries

The main question raised by the above task is the following. Since the notion of threshold is missing, what kind of adversaries will be tolerated by the protocols? In the traditional threshold model we know that adversaries who corrupt at most $t$ players are tolerable. Here it is not clear what kind of adversaries we should aim to tolerate.

In order to overcome this problem we define the notion of admissible adversary. Such adversary is allowed to corrupt an unlimited number of servers, as long as some basic integrity property of the access structure is preserved. Let us illustrate this notion by means of an example. Consider the access structure that is generated by a threshold $t$ when the set of participants is $n = 2t + 1$. The basis of such access structure includes all the subsets of cardinality $t + 1$. Any adversary that corrupts at most $t$ players satisfies the following integrity condition:

> Admissibility Condition: There exists a subset $S_i$ in the basis for which no server has been corrupted and for all the other subsets $S_j$ in the basis $(j \neq i)$, at least one server in $S_j$ has not been corrupted.

By drawing an analogy with the case of $t = n/2$ for threshold schemes, we can define an admissible adversary for a generic access structure as an adversary that satisfies the above admissibility condition.

We will prove that such admissible adversary is the most powerful adevrsary that we can tolerate. That is we will prove that if an adversary infringes the above condition then VSS is impossible to obtain. Also we will prove, by combining the protocol of Benaloh and Leichter [2] with the techniques of Rabin [7], that provably secure VSS protocols can be constructed over any monotone access structure against an admissible adversary.

**Different notions of admissibility:** The above admissibility condition mimics the notion of honest majority in threshold secure computation. Due to negative results in this area, we know that in order to achieve our VSS protocols against an admissible adversary we have to augment the model with a broadcast channel (otherwise we face the impossibility result on Byzantine Agreement with more than $n/3$ faults) and allow a negligible probability of failure.

However in threshold computation we know that we can do without such assumptions if we assume that the adversary can corrupt at most one third of the players. We plan to extend the above analogy to less powerful admissible adversaries and try to see if VSS protocols, without broadcast or probability of failure, are possible against such adversaries.

# 4 Applications

In the final part of the thesis we plan to present applications of VSS to the practical issue of robust shared signatures. Although maintaining rigorous formal treatment, special emphasis will be given to the efficiency of the protocols.

Shared signatures allow a group of servers to sign a document with a secret key that is shared among them. We plan to present efficient threshold signature schemes for the Digital Signature Standard and the RSA Signature Algorithm. The protocols are fully robust, that is they tolerate the presence of a threshold of malicious servers who may try to impede the signature process.

# References


[1] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for fault-tolerant distributed computing. In *Proc. 20th ACM Symp. on Theory of Computing*, pages 1–10, Chicago, 1988. ACM.

[2] J. Benaloh and J. Leichter. Generalized secret sharing and monotone functions. In S. Goldwasser, editor, *Proc. CRYPTO 88*, pages 27–36. Springer-Verlag, 1988. Lecture Notes in Computer Science No. 403.

[3] David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols. In Carl Pomerance, editor, *Proc. CRYPTO 87*, pages 462–462. Springer-Verlag, 1988. Lecture Notes in Computer Science No. 293.

[4] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults. In *Proc. 26th IEEE Symp. on Foundations of Comp. Science*, pages 383–395, Portland, 1985. IEEE.

[5] Paul Feldman and Silvio Micali. An optimal probabilistic protocol for synchronous byzantine agreement. In *20th ACM Symposium on Theory of Computing*, 1988. Final version to appear in the SIAM J. of Computing.

[6] S. Micali and P. Rogaway. Secure computation. In J. Feigenbaum, editor, *Proc. CRYPTO 91*, pages 392–404. Springer, 1992. Lecture Notes in Computer Science No. 576.

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