How to Split a Secret into Unknown Shares

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

September 12, 2014

Abstract

Grigoriev and Shpilrain recently considered secret sharing systems for which nobody (including the dealer) knows the share of a particular party and introduced a construction for the special case of all-or-nothing schemes. We extend their work and propose two threshold secret sharing schemes that satisfy this property.

1 Introduction

Blakley [B79] and Shamir [S79] independently introduced the idea of secret sharing as a solution to the key management problem. A secret sharing scheme 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 (in case of *perfect* secret sharing) or little information about the secret. An *all-or-nothing* scheme requires all shares for reconstruction, while a more general *threshold* scheme permits reconstruction from at least $t+1$ out of n shares, $t < n$.

Recently, Grigoriev and Shpilrain considered the construction of secret sharing schemes for which nobody (including the dealer) knows the share of a particular party [GS13]; through our work, we will call such systems *secret sharing schemes with unknown shares*. The study of Grigoriev and Shpilrain restricts to the particular case of all-or-nothing schemes, which motivates us to extend their research to the more general class of threshold secret sharing.

For the rest of this section, we explain our contribution: two threshold secret sharing schemes with unknown shares. Next section reviews the underlying notions and schemes we use for our constructions. Then, we present our two proposals and prove their properties.

1.1 Contribution

We continue the work of Grigoriev and Shpilrain [GS13] and propose two threshold secret sharing schemes with unknown shares: nobody (except the party itself) knows the shares of any particular party. Our constructions are built on Shamir's secret sharing [S79]. To maintain the privacy of the shares with respect to the dealer, we disallow him to know both the polynomial and the value were the polynomial is evaluated to generate a share: (1) in our first construction, the dealer chooses the polynomial, but he does not learn the values where the polynomial is evaluated; (2) in our second construction, the dealer knows the values where the polynomial is evaluated, but he does not learn the polynomial itself.

Our first proposal is elegant to expose and has the great advantage of working over public channels only; however, it is build from black-box obfuscation, which makes it impractical. To overcome this issue, we give a second construction that is theoretically secure; the drawback of this solution is that is requires secure communication channels between any two parties.

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

2 Preliminaries

2.1 Definitions and Notations

Let p be a prime large number. All computation are performed in $\mathbb{Z}_p$, hence we omit to explicitly mention this when it is evident from the context. We consider $S \in \mathbb{Z}_p$ the secret a dealer $\mathcal{D}$ shares among a set of parties $\{P_1, \ldots, P_n\}$.

Let $PRG : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ be a pseudorandom generator such that $2^{\lambda-1} \leq |\mathbb{Z}_p| \leq 2^\lambda$, where λ is the security parameter. Informally, a pseudorandom generator is secure if no polynomial-time distinguisher can make the difference between its output and a uniformly random sequence of the same length. As a consequence, given the output of the pseudorandom generator, it is computationally infeasible to determine the seed.

For the rest of our work, we consider a t-bounded adversary $\mathcal{A}$ that 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 players, but cannot coordinate their actions; moreover, 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).

2.2 Shamir's Secret Sharing

Shamir's secret sharing scheme is based on polynomial interpolation: a t-degree polynomial is uniquely defined by $t+1$ distinct points. We review next the construction [S79]:

Construction 2.1.

1. *Sharing Phase.*
   (a) $\mathcal{D}$ chooses uniformly at random a t-degree polynomial $f \in \mathbb{Z}_p[x]$ such that $f(0) = S$ and computes $f(i)$, $1 \leq i \leq n$;
   (b) $\mathcal{D}$ privately sends $f(i)$ to player P_i, for all $1 \leq i \leq n$;
2. *Reconstruction Phase.* Any $t+1$ (or more) parties $P_1, \ldots, P_{t+1}$ (without loss of generality after a possible reordering) recover $S = \sum_{i=1}^{t+1} f(i) \prod_{1 \leq j \leq t+1, i \neq j} \frac{j}{j-i}$.

Shamir's scheme satisfies two important properties: (1) it is *ideal*, since both the secret and the shares lie in $\mathbb{Z}_p$ and (2) it is *perfect* (when the exact degree of the polynomial is unknown), because t or less users obtain no information about the secret: for every value of the secret $S \in \mathbb{Z}_p$, the participants can reconstruct with the same probability a polynomial that passes through the t points they own and $(0, S)$ (we have considered the worst case scenario, when t participants cooperate).

2.3 All-or-Nothing Secret Sharing

We review next a simple and efficient all-or-nothing secret sharing [K83]:

Construction 2.2.

1. *Sharing Phase.*
   (a) $\mathcal{D}$ chooses uniformly at random $S_i \in \mathbb{Z}_p$, $1 \leq i \leq n-1$ and computes $S_n = S - \sum_{i=1}^{n-1} S_i$;
   (b) $\mathcal{D}$ privately sends S_i to player P_i, for all $1 \leq i \leq n$;

2. *Reconstruction Phase.* All parties $P_1, \ldots, P_n$ recover $S = \sum_{i=1}^{n} S_i$.

The scheme satisfies the same two properties as Shamir's scheme: (1) it is *ideal*, since both the secret and the shares lie in $\mathbb{Z}_p$ and (2) it is *perfect*, because less than n users obtain no information regarding the secret: for every value of the secret $S \in \mathbb{Z}_p$, the last share is computed with the same probability as $S_{n-1} = S - \sum_{i=1}^{n-1} S_i$ (we have considered the worst case scenario, when $n-1$ participants cooperate).

2.4 Obfuscation

Obfuscation is a technique used to hide the content of a program or implementation by making it hard to understand and analyze. An obfuscator $\mathcal{O}$ is an efficient algorithm that on input a program P outputs another program $\mathcal{O}(P)$ with the same functionality as P (i.e. for the same input x, $\mathcal{O}(P(x)) = P(x)$) that keeps private the internal implementation. Ideally, a *black-box obfuscator* should behave as a *virtual black-box* in the sense that it only allows access to the inputs and outputs of the system: anything that can be efficiently computed from $\mathcal{O}(P)$ can be efficiently computed given oracle access to P [BG01]. Barak and al. showed the impossibility of a universal black-box obfuscator that could be applied for any program [BG01]; subsequent work extended the impossibility results [BCC14]. However, candidates for simple programs do exist [L04, W05, CV13].

3 First Proposal

3.1 Construction

We build a secret sharing scheme with unknown shares from black-box obfuscation and pseudorandom generators, inspired by the Boneh-Zhandry NIKE [BZ14]. The idea is the following: each party P_i generates a seed s_i and publishes $x_i = PRG(s_i)$, where PRG is a pseudorandom generator. The dealer $\mathcal{D}$ chooses a t-degree polynomial f as in Shamir's scheme, but instead of distributing the shares to the parties, he builds and publishes an obfuscated program that allows the parties to compute the shares by themselves. To operate, the obfuscated program requires a valid seed; in this way, each of the parties can only evaluate f in a single point and hence the protocol maintains the threshold. Anyone else (including the dealer) does not know any of the seeds and therefore is unable to compute the corresponding shares.

The construction is as follows:

Construction 3.1.

1. *Registration Phase.* Each party P_i, $1 \leq i \leq n$ chooses uniformly at random a private seed $s_i \in \mathbb{Z}_p^*$ and publishes $x_i = PRG(s_i)$.

2. *Sharing Phase.*

   (a) $\mathcal{D}$ chooses uniformly at random a t-degree polynomial $f \in \mathbb{Z}_p[x]$ such that $f(0) = S$ and builds the program $Prg_{f;\{x_1,\ldots,x_n\}}$ as follows:

   Input: $s \in \mathbb{Z}_p^*$
   Constants: f, $\{x_1, \ldots, x_n\}$, PRG

   i. if $PRG(s) \in \{x_1, \ldots, x_n\}$, then output $f(s)$
   ii. otherwise, output $\perp$

   (b) $\mathcal{D}$ makes public $\mathcal{O}(Prg_{f;\{x_1,\ldots,x_n\}})$, the black-box obfuscated version of $Prg_{f;\{x_1,\ldots,x_n\}}$;

   (c) Each party P_i, $1 \leq i \leq n$ runs $\mathcal{O}(Prg_{f;\{x_1,\ldots,x_n\}})$ on input s_i to obtain $f(s_i)$;

3. *Reconstruction Phase.* Any $t+1$ (or more) parties $P_1, \ldots, P_{t+1}$ with distinct $x_1, \ldots, x_{t+1}$ (without loss of generality after a possible reordering) recover $S = \sum_{i=1}^{t+1} f(s_i) \prod_{1 \leq j \leq t+1, i \neq j} \frac{s_j}{s_j - s_i}$.

In comparison to Shamir's scheme, our first proposal: (1) remains ideal; (2) maintains the same reconstruction algorithm; (3) looses perfect secrecy, but achieves computational secrecy from black-box obfuscation and pseudorandom generators; (4) requires no secure communication.

3.2 Proofs

Theorem 3.1 (Correctness)**.** *If $\mathcal{D}$ is honest, then at the end of the reconstruction phase any set of at least $t+1$ parties $P_1, \ldots, P_{t+1}$ with distinct $x_1, \ldots, x_{t+1}$ output the correct secret.*

Proof. By construction, at the end of the sharing phase all parties P_i, $1 \leq i \leq n$ hold valid points $(s_i, f(s_i))$ on the polynomial f such that $f(0) = S$. Note that the seeds $s_1, \ldots, s_{t+1}$ are distinct for distinct values $x_1, \ldots, x_{t+1}$. Now, the correctness of the scheme reduces to the correctness of Shamir's scheme. □

Theorem 3.2 (Secrecy)**.** *If $\mathcal{D}$ is honest, PRG is a secure psudorandom generator and $\mathcal{O}$ is a secure black-box obfuscator, then a t-bounded adversary $\mathcal{A}$ cannot reveal any information about the secret (except with negligible probability).*

Proof. By definition, a t-bounded adversary $\mathcal{A}$ gains access to the knowledge of up to t parties, i.e. pairs $(s_i, f(s_i))$, $1 \leq i \leq t$ (without loss of generality, after a possible reordering). The proof reduces to Shamir's secrecy if $\mathcal{A}$ gains no additional information. First, if PRG is a secure pseudorandom generator, then $\mathcal{A}$ cannot compute s_i from the public x_i, except with negligible probability; hence, $\mathcal{A}$ can query $\mathcal{O}(Prg_{f;\{x_1,\ldots,x_n\}})$ in at most t points $s_1, \ldots, s_t$. Second, if $\mathcal{O}$ is a secure black-box obfuscator, $\mathcal{O}(Prg_{f;\{x_1,\ldots,x_n\}})$ reveals no information about the polynomial f. Hence, the scheme is computationally secure. □

Theorem 3.3 (Unknown shares)**.** *If PRG is a secure psudorandom generator, then $\mathcal{D}$ cannot reveal any information about the share of a player P_i, $1 \leq i \leq n$ (except with negligible probability).*

Proof. If PRG is a secure pseudorandom generator, then $\mathcal{D}$ cannot compute s_i from the public x_i, except with negligible probability. Since $\mathcal{D}$ does not know the input s_i on which the player runs $\mathcal{O}(Prg_{f;\{x_1,\ldots,x_n\}})$, the output $f(s_i)$ remains hidden. □

4 Second Proposal

4.1 Construction

We build a theoretically secure secret sharing scheme with unknown shares. The idea is the following: the dealer $\mathcal{D}$ does not choose the polynomial, but instead he allows each party P_i to select a t-degree polynomial f_i such that their sum f is a valid polynomial for the given secret (i.e. $f(0) = \sum_{i=1}^{n} f_i(0) = S$). To guarantee the correctness of f, $\mathcal{D}$ previously distributes $f_i(0)$ to P_i as a share in an all-or-nothing scheme. Except the free coefficient, $\mathcal{D}$ knows nothing about f_i; in consequence, he does not learn f and therefore is unable to compute the corresponding shares.

The construction is as follows:

Construction 4.1.

1. *Sharing Phase.*

   (a) $\mathcal{D}$ chooses uniformly at random $S_i \in \mathbb{Z}_p$, $1 \leq i \leq n-1$ and computes $S_n = S - \sum_{i=1}^{n-1} S_i$;

(b) $\mathcal{D}$ privately sends S_i to player P_i, for all $1 \leq i \leq n$;

(c) Each party P_i, $1 \leq i \leq n$ uses Shamir's scheme to share S_i among the players, i.e. P_i chooses uniformly at random a t-degree polynomial $f_i(x) \in \mathbb{Z}_p[x]$ such that $f_i(0) = S_i$ and privately distributes sub-share $f_i(j)$ to party P_j, $1 \leq j \leq n$;

(d) Each party P_i, $1 \leq i \leq n$ computes the share $f(i) = \sum_{j=1}^{n} f_j(i)$;

2. *Reconstruction Phase.* Any $t+1$ (or more) parties $P_1, \ldots, P_{t+1}$ (without loss of generality after a possible reordering) recover $S = \sum_{i=1}^{t+1} f(i) \prod_{1 \leq j \leq t+1, i \neq j} \frac{j}{j-i}$.

In comparison to Shamir's scheme, our second proposal: (1) remains ideal; (2) maintains the same reconstruction algorithm; (3) preserves perfect secrecy (4) requires secure communication between any two parties.

4.2 Proofs

Theorem 4.1 (Correctness)**.** *If $\mathcal{D}$ is honest, then at the end of the reconstruction phase any set of at least $t+1$ honest parties $P_1, \ldots, P_{t+1}$ output the correct secret.*

Proof. By construction, $f(0) = \sum_{i=1}^{n} f_i(0) = \sum_{i=1}^{n} S_i = S$; hence $f(x) = \sum_{i=1}^{n} f_i(x)$ is a valid polynomial that can be used to share S using Shamir's secret sharing. At the end of the sharing phase, each player holds $f(i) = \sum_{j=1}^{n} f_j(i)$ a valid point on f. Now, the correctness of the scheme reduces to the correctness of Shamir's scheme. □

Theorem 4.2 (Secrecy)**.** *If $\mathcal{D}$ is honest, then a t-bounded adversary $\mathcal{A}$ cannot reveal any information about the secret (i.e. the scheme is perfect).*

Proof. By definition, a t-bounded adversary $\mathcal{A}$ gains access to the knowledge of up to t parties, i.e. values S_i, polynomials f_i and pairs $(i, f(i))$, $1 \leq i \leq t$ (without loss of generality, after a possible reordering). The proof reduces to Shamir's secrecy if $\mathcal{A}$ gains no additional information from S_i and f_i, $1 \leq i \leq t$; this holds from the perfect secrecy of the all-or-nothing scheme (not that f_i can be also seen as shares of f). Hence, the scheme remains perfectly secure.

□

Theorem 4.3 (Unknown shares)**.** *$\mathcal{D}$ cannot reveal any information about the share of a player P_i, $1 \leq i \leq n$.*

Proof. By construction, the polynomial $f(x) = \sum_{i=1}^{n} f_i(x)$ remains hidden to $\mathcal{D}$, except the free coefficient $f(0) = S$. Hence, $\mathcal{D}$ cannot reveal any information about the shares $f(i)$, $1 \leq i \leq n$. □

5 Conclusions and Future Work

We proposed two threshold secret sharing schemes that have the property that nobody, including the dealer, knows the shares of any particular players. Our first proposal has the great advantage of working over public channels only, but it is build from black-box obfuscation, which makes it impractical. Our second proposal is theoretically secure, but requires secure communication channels between any two parties. Further work might consider decreasing the number of secure channels. An interesting open problem is to extend the construction to general access structure schemes.

References

[BG01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai , Salil P. Vadhan and Ke Yang. On the (im)possibility of obfuscating programs. In *Advances in Cryptology - CRYPTO 2001*, pages 1–18, 2001.

[BCC14] Nir Bitansky, Ran Canetti, Henry Cohn, Shafi Goldwasser, Yael Tauman Kalai, Omer Paneth and Alon Rosen. The impossibility of obfuscation with auxiliary input or a universal simulator. In *Advances in Cryptology - CRYPTO 2014*, pages 71–89, 2014.

[B79] G. R. Blakley. Safeguarding cryptographic keys. In *Proceedings of the 1979 AFIPS National Computer Conference*, pages 313–317, 1979.

[BZ14] Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In *Advances in Cryptology - CRYPTO 2014*, pages 480–499, 2014.

[CV13] Ran Canetti and Vinod Vaikuntanathan. Obfuscating branching programs using black-box pseudo-free groups. IACR Cryptology ePrint Archive, Report 2013/500, 2013. `http://eprint.iacr.org`

[GS13] Dima Grigoriev and Vladimir Shpilrain. Secrecy without one-way functions. Cryptology ePrint Archive, Report 2013/055, 2013. `http://eprint.iacr.org`

[K83] Ehud D. Karnin, Jonathan W. Greene and Martin E. Hellman. On secret sharing systems. In *IEEE Transactions on Information Theory 29*, pages 35–41, 1983.

[L04] Ben Lynn, Manoj Prabhakaran and Amit Sahai. Positive results and techniques for obfuscation. In *Advances in Cryptology - EUROCRYPT 2004*, pages 20–39, 2004.

[S79] Adi Shamir. How to share a secret. In *Commun. ACM 22*, pages 612–613, 1979.

[W05] Hoeteck Wee. On obfuscating point functions. In *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing* pages 523–532, 2005.