HOW TO SPLIT A SECRET INTO UNKNOWN SHARES

Ruxandra F. OLIMID

Department of Computer Science University of Bucharest, Romania
E-mail: ruxandra.olimid@fmi.unibuc.ro

Secret sharing allows a dealer to distribute a secret among multiple parties such that authorized coalitions can reconstruct the secret. Traditionally, the dealer knows the exact share each user holds. Grigoriev and Shpilrain recently considered secret sharing systems for which the dealer does not know 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 some simple to describe threshold secret sharing schemes that satisfy this property.
Key words: Secret sharing, Unknown shares, Black-box obfuscation, Oblivious polynomial evaluation.

1 Introduction

Blakley [2] and Shamir [14] independently introduced 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 among multiple parties; *authorized* sets of parties can reconstruct the secret, while *unauthorized* sets obtain no (in case of *perfect* secret sharing) or insufficient information about the secret.

Secret sharing can be naturally classified based on the structure of the authorized sets. An *all-or-nothing* scheme requires all shares for reconstruction, while a *threshold* scheme permits reconstruction from at least $t+1$ out of n shares ($t < n$).

1.1 Motivation and Related work

Traditionally, the dealer splits the secret into n shares and securely distributes one share to each participant, knowing the exact share each user holds. Recently, Grigoriev and Shpilrain considered secret sharing for which nobody (including the dealer) knows the share of a particular party [7]. Their study restricts to the case of all-or-nothing schemes, which motivates us to extend their work to the more general class of threshold secret sharing.

The property is somehow related to the notion of *cryptographic anonymous secret sharing* [6,8], which informally states that the owners of the submitted shares cannot be identified by an adversary who observes the shares. The dealer is never considered to be an adversary, so cryptographic anonymity does not aim to hide the shares from the dealer itself. This means that in the open reconstruction settings (i.e. shares are made public at reconstruction) the dealer could reveal who reconstructed the secret. This ability of the dealer can be undesirable in some scenarios, like for example one that illustrates a practical application of cryptographic anonymity - *controversial reconstruction* [8]: in this case there are strong argues whether the secret should be reconstructed or not; so, a group of parties that decides reconstruction wishes to keep the individual identities hidden. If the dealer knows the shares each participant owns and all shares are distinct, then it can easily identify the reconstruction group.

1.2 Contribution

We continue the work of Grigoriev and Shpilrain [7] and consider some simple ways to obtain threshold secret sharing schemes such that nobody (except the party itself) knows the share of any particular party. Similar to original paper, we do not consider coalitions among the dealer and the participants, we just consider the straightforward property that the dealer cannot reveal the shares owned by a particular share by itself.

Both constructions modify the Shamir's secret sharing [14]. To maintain the privacy of the shares with respect to the dealer, we disallow it to know both the polynomial and the value were the polynomial is evaluated to generate a share: (1) in the first construction, the dealer chooses the polynomial, but does not learn the values where the polynomial is evaluated in; (2) in the second construction, the dealer knows the values where the polynomial is evaluated in, but does not learn the polynomial itself.

The first construction is easy to describe and has the advantage of working over public channels only, but it is built from black-box obfuscation, which definitely makes it impractical. To overcome this issue, black-box obfuscation can be easily replaced by OPE (Oblivious Polynomial Evaluation). The second construction is self-contained and does not rely on any other primitive or protocol. It is theoretically secure, but requires secure communication channels between any two parties. All our constructions are inspired from existing works that aim to solve other problems or satisfy other properties.

We do not aim to get efficient constructions for the given property; our goal is to show that such constructions are possible for threshold secret sharing. So, we keep the constructions simple to describe and understand.

2 Preliminaries

2.1 Basic Definitions and Notations

Let p be a large prime number and $\mathbb{Z}_p = \{0, \dots, p-1\}$ the set of integers modulo p. All computations are performed in $\mathbb{Z}_p$, hence we do not explicitly mention this when it is evident from the context.

Let $S \in \mathbb{Z}_p$ be the secret a dealer $\mathcal{D}$ shares among a set of n parties $\{P_1, \dots, P_n\}$ using a secret sharing scheme, which usually consists in two phases: (1) *sharing phase*: the secret S is randomly split among the players $P_1, \dots, P_n$ and (2) *reconstruction phase*: an authorized set of parties collaboratively computes the secret S. Optionally, a *registration phase* could exist prior to the sharing phase to allow the players register or perform some prerequisites for the sharing.

A secret sharing scheme must satisfy (at least) two basic properties: (1) *correctness* - any authorized set of users can recover S and (2) *secrecy* - no unauthorized set of users can find S (except with negligible probability). Let $\mathcal{A}$ be a t-bounded adversary that can compromise at most

t out of n non-dealer parties with the goal to break secrecy property, i.e. to reveal the secret S. We work in the *semi-honest* (or *honest-but-curios*) adversarial model, for which the parties follow the protocol exactly. More precise, the adversary gains knowledge of the shares of up to t players with the goal to reveal S, but cannot coordinate the actions of the players.

In addition to correctness and privacy, we ask that an *honest-but-curios* dealer $\mathcal{D}$ does not know the share of any particular party. So, a dealer that follows the protocol exactly should be unaware of the shares the parties own. In the proofs, we will refer to this property as *unknown shares.*

2.2 Shamir's Secret Sharing

Shamir's threshold secret sharing scheme is based on polynomial interpolation: a t-degree polynomial $f(s)$ is uniquely defined by $t+1$ distinct pairs $(s, f(s))$. We review next the construction [14]:

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 $S = \sum\limits_{i=1}^{t+1} f(i) \prod\limits_{1 \leq j \leq t+1, i \neq j} \frac{j}{j-i}$.

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 learn 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 scheme [11]:

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 $S_n = S - \sum\limits_{i=1}^{n-1} S_i$;
   (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 $S = \sum\limits_{i=1}^{n} S_i$.

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 about the secret: for every value of the secret $S \in \mathbb{Z}_p$, the last share is computed with the same probability as $S_{n-1} = S - \sum\limits_{i=1}^{n-1} S_i$ (we have considered the worst case scenario, when $n-1$ participants cooperate).

2.4 Pseudo-Random Generator

A pseudo-random generator PRG is a deterministic algorithm that on input a random seed outputs a longer pseudo-random sequence. Informally, a pseudo-random generator is secure if no efficient algorithm can make the difference between its output and a uniformly random sequence of the same length. As a consequence, given the output of the pseudo-random generator, it is computationally infeasible to determine the seed.

2.5 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 [1]. Barak and al. showed the impossibility of a universal black-box obfuscator that could be applied for any program [1]; subsequent work extended the impossibility results [3]. However, candidates for simple programs do exist [5,12,15].

2.6 Oblivious Polynomial Evaluation

OPE (Oblivious Polynomial Evaluation) [9,13] is a protocol between two parties, a *sender* and a *receiver* such that the sender inputs a polynomial f and the receiver inputs a value s; at the end of the protocol, the sender gains no information about s, while the receiver gains $f(s)$, the evaluation of the polynomial in s and nothing else. Hence, OPE allows the receiver to get the value $f(s)$ for any s without learning anything else about the polynomial f and without revealing to the sender any information about s.

3 First Construction

3.1 Construction

We present a secret sharing scheme from black-box obfuscation and pseudo-random generators that keeps the shares hidden from the dealer. The construction is inspired by the Boneh-Zhandry NIKE [4]. While their construction only needs indistinguishability obfuscation (to solve a long time open problem), our construction considers the stronger notion of black-box obfuscation, which is unacceptable for practical reasons. However, we present the construction as a first and simple to expose solution to our problem.

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 pseudo-random 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, $1 \le i \le n$ 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 $S = \sum\limits_{i=1}^{t+1} f(s_i) \prod\limits_{1\le j\le t+1, i\ne j} \frac{s_j}{s_j - s_i}$.

In comparison to Shamir's scheme, the construction: (1) remains ideal; (2) maintains the same reconstruction algorithm; (3) looses perfect secrecy, but achieves computational secrecy from black-box obfuscation and pseudo-random 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, $1 \le i \le n$ 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 pseudo-random generator and $\mathcal{O}$ is a secure black-box obfuscator, then a t-bounded adversary $\mathcal{A}$ does not 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 $(s_i, f(s_i))$, $1 \le i \le t$ (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 pseudo-random 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 pseudo-random generator, then $\mathcal{D}$ does not reveal any information about the share of a player P_i, $1 \le i \le n$ (except with negligible probability).*

Proof. If PRG is a secure pseudo-random 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. □

3.3 Practical solution based on OPE

The previous scheme is not practical because it uses black-box obfuscation. To overcome this issue, we keep the same underlying idea (that the dealer chooses the polynomial, but he does not learn the values where the polynomial is evaluated) and replace obfuscation by OPE. This solves the problem, because OPE efficient protocols do exist [9,13].

More precisely, 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 runs a OPE protocol with each party. In this way, each party evaluates f in a value, while the dealer does not learn the value where the polynomial is evaluated in.

Construction 3.2.

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 $f(0) = S$;
   (b) $\mathcal{D}$ runs OPE with each party P_i, $1 \leq i \leq n$ on input f, respectively s_i; at the end of the OPE, each P_i obtains $f(s_i)$;
3. *Reconstruction Phase.* Any $t+1$ (or more) parties $P_1, \ldots, P_{t+1}$ with distinct $s_1, \ldots, s_{t+1}$ (without loss of generality after a possible reordering) recover $S = \sum_{i=1}^{t+1} f(s_i) \prod_{1 \leq j \leq t+1, i \neq j} \frac{s_j}{s_j - s_i}$.

It is easy to see that the properties of the scheme follow directly from the ones of the OPE, similar to the results in Section 3.2. Theorem 3.1 (*Correctness*) and Theorem 3.3 (*Unknown Shares*) remain unchanged, while Theorem 3.2 (*Secrecy*) becomes:

Theorem 3.4 (Secrecy)**.** *If $\mathcal{D}$ is honest, PRG is a secure pseudo-random generator and OPE is secure, then a t-bounded adversary $\mathcal{A}$ does not reveal any information about the secret (except with negligible probability).*

To avoid repetition, we only highlight the differences in the proofs: *correctness* follows from the correctness of the OPE, since all parties P_i, $1 \leq i \leq n$ hold valid points $(s_i, f(s_i))$ at the end of the sharing phase; *secrecy* uses the fact that the adversary $\mathcal{A}$ cannot gain additional information on f due to the security of OPE; *unknown shares* property holds because the dealer $\mathcal{D}$ cannot find the player's input s_i or its output $f(s_i)$, as follows directly from the security of OPE.

Note that the registration phase becomes optional for the construction based on OPE: correctness holds for any set of t parties with distinct s_i's and collisions on s_i can become negligible for a large enough p. However, we keep it for similarity to the previous construction.

4 Second Construction

4.1 Construction

We present a theoretically secure secret sharing scheme, inspired by the proactive secret sharing scheme of Herzberg et al.[10]. Proactive secret sharing consists in periodically renewing the shares in a way that the information revealed to the adversary before renewal becomes useless; it differs from the property we want to achieve, but the construction is similar in the sense that the players distribute sub-shares between themselves.

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, the construction: (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 $\mathcal{A}$ does not 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 (note that f_i can be also seen as shares of f). Hence, the scheme remains perfectly secure.

□

Theorem 4.3 (Unknown shares). *$\mathcal{D}$ does not 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 $f(i)$, $1 \leq i \leq n$. $\square$

5 Acknowledgment

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.

6 References

1. B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai , S.P. Vadhan, K. Yang, On the (im)possibility of obfuscating programs. In *Advances in Cryptology - CRYPTO 2001*, pp. 1–18, 2001.

2. G. R. Blakley, Safeguarding cryptographic keys. In *Proceedings of the 1979 AFIPS National Computer Conference*, pp. 313–317, 1979.

3. N. Bitansky, R. Canetti, H. Cohn, S. Goldwasser, Y.T. Kalai, O. Paneth and A. Rosen, The impossibility of obfuscation with auxiliary input or a universal simulator. In *Advances in Cryptology - CRYPTO 2014*, pp. 71–89, 2014.

4. D. Boneh, M. Zhandry, Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In *Advances in Cryptology - CRYPTO 2014*, pp. 480–499, 2014.

5. R. Canetti, V. Vaikuntanathan, Obfuscating branching programs using black-box pseudo-free groups. Cryptology ePrint Archive, Report 2013/500, 2013. `http://eprint.iacr.org`

6. C. Gehrmann, Topics in Authentication Theory. Lund University, 1997.

7. D. Grigoriev, V. Shpilrain, Secrecy without one-way functions. Cryptology ePrint Archive, Report 2013/055, 2013. `http://eprint.iacr.org`

8. M. Guillermo, K. M. Martin, C. M. O'Keefe, Providing anonymity in unconditionally secure secret sharing schemes. Designs, Codes and Cryptography, 28(3), pp.227–245, 2003.

9. C. Hazay, Y. Lindell, Efficient Oblivious Polynomial Evaluation with Simulation-Based Security. Cryptology ePrint Archive, Report 2009/459, 2009. `http://eprint.iacr.org`

10. A. Herzberg, S. Jarecki, H. Krawczyk, M. Yung, Proactive Secret Sharing Or: How to Cope With Perpetual Leakage. In *Advances in Cryptology - CRYPTO 1995*, pp. 339-352, 1995.

11. E.D. Karnin, J.W. Greene, M.E. Hellman, On secret sharing systems. In *IEEE Transactions on Information Theory 29*, pp. 35–41, 1983.

12. B. Lynn, M. Prabhakaran, A. Sahai, Positive results and techniques for obfuscation. In *Advances in Cryptology - EUROCRYPT 2004*, pp. 20–39, 2004.

13. M. Naor, B.Pinkas, Oblivious Polynomial Evaluation. SIAM J. Comput 35, pp.1254–1281, 2006.

14. A. Shamir, How to share a secret. In *Commun. ACM 22*, pp. 612–613, 1979.

15. H. Wee, On obfuscating point functions. In *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing* pp. 523–532, 2005.