

# Secret Sharing and Applications

Pablo Azar<sup>†</sup>

Harvard University '09

Cambridge, MA 02138

azar@fas.harvard.edu

## 9.1 Shamir's Secret Sharing Scheme

In this column, I will discuss Shamir's **secret sharing protocol** [Sh]. The motivation for this protocol is the desire for individual privacy while computing an aggregate piece of data. For example, suppose that, to prevent corruption, no employee of a bank is allowed to access the security vault alone. Instead, each employee is given a piece of the password to the vault. When  $k$  employees get together, they can reconstruct the password and access the vault. However,  $k - 1$  of them will have insufficient information about the password, so that no  $k - 1$  corrupt employees can steal the bank's money.

You can encode a piece of information as a binary  $\ell$ -bit number  $a_{\ell-1}2^{\ell-1} + \dots + 2a_1 + a_0$  where  $a_i \in \{0, 1\}$ . If the message you want to share is too long, you can split it up into smaller messages so that each takes less than  $\ell$  bits to write. Without loss of generality, we may assume that the message can be written as an  $\ell$ -bit number.

Then we can interpret the message as an element of the finite field  $\mathbb{F}_{2^\ell}$ .<sup>1</sup> If you have a message  $x \in \mathbb{F}_{2^\ell}$ , you can compute any polynomial  $c_0 + c_1x + \dots + c_nx^n \in \mathbb{F}_{2^\ell}$  and use Lagrange's interpolation theorem: If  $\mathbb{F}$  is a field and  $(x_1, y_1), \dots, (x_n, y_n)$  are pairs of points in  $\mathbb{F}^2$ , then there exists a unique polynomial  $p(x) = c_{n-1}x^{n-1} + \dots + c_0$  of degree  $n - 1$  such that  $p(x_i) = y_i$  for  $i = 1, \dots, n$ .

What does this have to do with sharing secrets? Well, suppose that you encode a secret  $s$  as an  $\ell$ -bit number in  $\mathbb{F}_{2^\ell}$  and suppose that you want to distribute  $s$  among  $n$  people numbered  $1, \dots, n$ . You want them to be able to reconstruct  $s$  if  $k$  people get together and cooperate, but get no information if fewer than  $k$  people pool their knowledge. Now, generate  $k - 1$  random numbers  $c_1, \dots, c_{k-1}$  uniformly from  $\mathbb{F}_{2^\ell}$  and consider the polynomial  $f(x) = s + c_1x + \dots + c_{k-1}x^{k-1}$ . To each person  $i$ , give the **share**  $f(i)$ .

Players  $i_1, \dots, i_k$  can get together and pool their shares to obtain the set  $\{(i_1, f(i_1)), \dots, (i_k, f(i_k))\}$  of distinct points. With these  $k$  points, they can use Lagrange's interpolation theorem to reconstruct the polynomial  $f$ . Reconstructing  $f$  is equivalent to reconstructing its coefficients. In particular, all  $k$  players get knowledge of the secret  $s$ , which is the constant coefficient of  $f$ .

Furthermore, if  $k - 1$  players get together they do not learn anything about  $s$ . To see this, consider the following argument: given a set of coefficients  $(c_1, \dots, c_{k-1}) \in \mathbb{F}_{2^\ell}^{k-1}$  and a fixed secret  $s$ , we can generate the polynomial  $f(x) = s + c_1x + \dots + c_{k-1}x^{k-1}$  and the vector of

<sup>†</sup>Pablo Azar, Harvard 09, is an applied mathematics concentrator from Buenos Aires, Argentina. He is also enrolled in a concurrent masters program in computer science. He is a founding member of The HCMR and currently serves on The HCMRs staff.

<sup>1</sup>You can construct this field if you know an irreducible polynomial  $f(x)$  of degree  $\ell$  with coefficients in  $\mathbb{F}_2$ . The field is given by the quotient  $\mathbb{F}_{2^\ell} := \mathbb{F}_2[X]/f(X)$ . The elements of this field are polynomials of degree less than  $\ell$  with coefficients in  $\mathbb{F}_2$ , with all operations conducted modulo  $f(X)$ . Such polynomials require  $\ell$  bits to encode.

shares  $(f(i_1), \dots, f(i_{k-1})) \in \mathbb{F}_2^{k-1}$ . This gives us a map

$$M : \mathbb{F}^{k-1} \rightarrow \mathbb{F}^{k-1}$$

$$M(c_1, \dots, c_{k-1}) = (f(i_1), \dots, f(i_{k-1})).$$

This map is bijective by Lagrange's interpolation theorem: given a vector of shares  $(\alpha_{i_1}, \dots, \alpha_{i_{k-1}})$ , there exists a unique polynomial of degree  $k-1$  with constant coefficient  $s$  that interpolates the points  $\{(0, s), (i_1, \alpha_{i_1}), \dots, (i_{k-1}, \alpha_{i_{k-1}})\}$ . This polynomial is characterized by its non-constant coefficients  $c_1, \dots, c_{k-1}$ .

Therefore, there is a bijection between coefficients  $(c_1, \dots, c_{k-1})$  and shares  $(\alpha_1, \dots, \alpha_{k-1})$ . But remember that the dealer chose the coefficients  $c_1, \dots, c_{k-1}$  to be uniformly and independently distributed. This implies that the shares  $(\alpha_1, \dots, \alpha_{k-1})$  are also uniformly and independently distributed. Given the secret  $s$ , the players  $i_1, \dots, i_{k-1}$  can get any possible combination of  $k-1$  shares with equal probability. This shows that  $k-1$  shares do not reveal anything valuable about the secret.

## 9.2 Multi-Party Protocols, Corrupt Players and Corrupt Dealers

The scheme proposed above is very elegant, but the assumptions on the dealer and the honesty of the players may be too strong for applications. The first problem that arises is that there may be no dealer. In this case, each of the players may have a secret  $s_1, \dots, s_n$ , and all of them want to compute a function  $f(s_1, \dots, s_n)$  without revealing any information about their corresponding secrets besides what is known from  $f(s_1, \dots, s_n)$  [Ya]. Furthermore, some of the players may be malicious or faulty and give fake or incorrect shares to the other participants. To detect which players are being dishonest, the concept of information checking was introduced by Rabin and Ben-Or [RB]. Their work expands the secret sharing protocol so that, when more than half the players are honest and there are appropriate communication channels, any multiparty computation can be performed by the honest parties.

Another problem may be that of a corrupt dealer. That is, the dealer may be distributing shares  $s_1, \dots, s_n$  to the players so that when players  $i_1, \dots, i_k$  put their shares together, they get the secret  $s$ , but when players  $j_1, \dots, j_k$  put their shares together, they get the secret  $s' \neq s$ . A dealer is honest if and only if the secret reconstructed by any combination of  $k$  players is the same. In this case, we say that the players' shares are **consistent**.

To address this second problem, the concept of **Verifiable Secret Sharing** was introduced by Chor, Goldwasser, Micali and Awerbuch [CGMA]. In a Verifiable Secret Sharing scheme, the dealer can broadcast some information, revealing as little information as possible about the shares so that the players can verify that their shares are consistent. A particularly elegant scheme for doing this was introduced by Feldman [Fe]. In this scheme, the dealer takes a cyclic group  $G$  with publicly known generator  $g$ , such that obtaining the value of  $x$  if one knows  $g^x$  is computationally intractable. If  $|G|$  is a prime  $p$ , all shares in this scheme are in  $\mathbb{F}_p$ . If the dealer uses the polynomial  $f(x) = s + c_1x + \dots + c_{k-1}x^{k-1} \pmod p$ , she can post  $g, g^s, g^{c_1}, \dots, g^{c_{k-1}}, g^{f(i_1)}, \dots, g^{f(i_n)}$  on a bulletin board. This way, every player with share  $f(i_j)$  can check that  $g^{f(i_j)}$  as computed by them is equal to the posted  $g^{f(i_j)}$  and all players check that the posted  $g^{f(i_j)}$  equals  $g^s g^{c_1 i_j} \dots g^{c_{k-1} i_j^{k-1}}$ .<sup>2</sup>

## 9.3 Two-Party Protocols and Applications

These secret sharing and multi-party computation protocols lead to important applications. A toy example is the **salary problem**, in which  $n$  people learn their average salary without revealing anything about their own salaries except what is learned from knowing the sum of all the salaries.

<sup>2</sup>In practice, such groups are constructed by taking primes  $p, q$  such that  $q = 2p + 1$  and taking  $G = \text{Sq}(\mathbb{Z}_q^\times)$ , the group of all non-zero squares in the finite field of order  $q$ . It is conjectured that there are an infinite number of primes of the form  $q = 2p + 1$  where  $p$  is prime. Such primes are called **Sophie Germain primes**.

Many applications consider computations with only two parties. Since the multi-party protocol relying on secret sharing needs more than half the players to be honest, it does not apply to two-party computation. Some of these applications (described below), rely on a two-party primitive called **Oblivious Transfer** [Ra, NP], which was introduced by Michael Rabin in 1981.<sup>3</sup> In the Oblivious Transfer protocol, there is one sender and one receiver. The sender has  $N$  messages, and the receiver chooses one of them. The protocol is designed so that the sender does not know which message was chosen, and the receiver does not learn anything about any of the other  $N - 1$  messages.

One important example is **private querying** of databases. Say Alice has a large database, which Bob pays to use on a per-query basis. However, Bob does not want Alice to know what he is querying. Furthermore, since Alice derives her profit from Bob's queries, she does not want anything revealed to Bob except the results of his query.

Another application is **privacy preserving data mining**. Suppose that two rival companies have datasets  $D_1$  and  $D_2$ , on which they want to perform data mining. However, they want to reveal as little as possible about their proprietary data to their rival. Lindell and Pinkas [LP] suggest such a protocol, showing that one can get aggregate data about  $D_1 \cup D_2$  revealing as little information as possible about  $D_1$  or  $D_2$  individually. An important lesson from their work is that theoretical protocols may not be the most efficient and that they may need to be modified to accommodate resource constraints. When the databases in question are large, one may want to minimize communication between the parties so as to limit the amount of bandwidth used. The reader interested in the practical applications of multi-party computations is encouraged to look at the report by Du and Atallah [DA], where many interesting problems—including these—are presented.

Shamir's original secret sharing scheme is both simple and applicable. While the generalizations and applications depend on some difficult concepts, the basic secret sharing scheme relies solely on linear algebra in finite fields. It is yet another example demonstrating that mathematics can be modern, elegant, and useful.

## Acknowledgements

The vast majority of the information presented here was demonstrated to me in some form by Professor Michael O. Rabin of Harvard University. I would also like to thank Shrenik Shah, for his extensive edits. The exposition of Shamir's secret sharing scheme owes much to his guidance.

## References

- [CGMA] Benny Chor, Shafi Goldwasser, Silvio Micali, Baruch Awerbuch: Verifiable Secret Sharing in the Presence of Faults, *Proc. 26th IEEE Symp. on Foundations of Computer Science* (1985).
- [DA] Wenliang Du, Mikhail J. Atallah: Secure Multi-Problem Computation Problems and Their Applications: A Review and Open Problems, *CERIAS Tech Report 2001-51* (2001).
- [Fe] Paul Feldman: A practical scheme for non-interactive verifiable secret sharing. *Proceedings of the 28th IEEE Symposium on the Foundations of Computer Science* (1987).
- [LP] Yehuda Lindell, Benny Pinkas: Privacy Preserving Data Mining. *J. Cryptology* **15** (2002), 177–206.
- [NP] Moni Naor, Benny Pinkas: Oblivious transfer and polynomial evaluation, *Proceedings of the thirty-first annual ACM symposium on Theory of computing* (1999).
- [Ra] Michael O. Rabin: How to exchange Secrets with Oblivious Transfer, *Technical Report TR-81* Harvard University: Aiken Computation Lab (1981).

---

<sup>3</sup>The version I am presenting is given by Naor and Pinkas [NP]

- [RB] Tal Rabin, Michael Ben-Or: Verifiable secret sharing and multiparty protocols with honest majority, *Proceedings of the twenty-first annual ACM symposium on Theory of computing* (1989).
- [Sh] Adi Shamir: How to share a secret, *Communications of the ACM* (1979).
- [Ya] Andrew C. Yao: Protocols for secure computations, *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science* (1982), 160–164.