FHE-MPC Notes

Lecturer: Nigel Smart
Scribe: Joop van de Pol

Lecture #7
21st November 2011

Secret Sharing

Consider *n participants* $P_1, P_2, \ldots, P_n$ who want to distribute a *secret* s amongst themselves such that each participant P_i has a *share* $[s]_i$ of the secret. It should be possible to reconstruct the secret when in possession of 'enough' shares (i.e., the participants corresponding to these shares combined are *qualified* to access the secret) and impossible otherwise. This can be achieved by a *secret sharing scheme*, which consists of at least the following two protocols:

- The *distribution protocol*, where a sharer or *dealer* (who knows the secret s) creates and distributes the shares $[s]_i$ amongst the participants P_i.
- The *reconstruction protocol*, where a qualified set of participants recovers the secret by combining their shares.

Shamir Secret Sharing

Consider the following secret sharing scheme, which is due to Shamir [5]:

- Distribution: The dealer picks a random polynomial $f \in \mathbb{F}_p[x]$ of degree $t < n$ such that $f(0) = s \in \mathbb{F}_p$. He computes the values $[s]_i := f(i) \bmod p$ for $1 \leq i \leq n$ and sends each share $[s]_i$ to the corresponding participant P_i.
- Reconstruction: Any $t+1$ participants can reconstruct the polynomial f by applying Lagrange interpolation to the tuples $(i, [s]_i)$. They recover the secret by computing $f(0) \bmod p = s$.

This scheme is correct because any $t+1$ pairs (i, x_i) uniquely determine a polynomial g of degree t satisfying $g(i) = x_i$, thus interpolation on $t+1$ pairs $(i, f(i))$ must yield f. Since s can be easily obtained from f, f is sometimes denoted by $[s]$, the secret in its shared form. In this scheme, a set of participants is qualified (to reconstruct the secret) if it contains at least $t+1$ participants, regardless of which participants it contains. Such a scheme is called a $(t+1, n)$-threshold scheme, where $t+1$ is the threshold.

Furthermore, consider the case where an adversary has t shares $(i, f(i))$. Adding the imaginary share $(0, s)$ shows that the only polynomial corresponding to these shares which will yield the secret is f. Thus, guessing a correct share $(i', f(i'))$ is equivalent to guessing s, which can be done with probability $1/p$. In fact, adding any pair $(0, r)$ for $r \in \mathbb{F}_p$ to the t shares gives rise to a polynomial of degree t with unique values for the remaining $n-t$ shares. Thus, all values $r \in \mathbb{F}_p$ are equally likely to be the secret. This means that the adversary cannot get any information about the secret s when in possession of fewer than $t+1$ shares. Schemes which have this property (any unqualified set of participants do not gain any information about the secret) are called *perfect* secret sharing schemes.

Another interesting property of this scheme is that the secret and the shares are all of the same 'size', since they are all elements of $\mathbb{F}_p$. Consider the quantities

$$\rho_i = \frac{\#\text{ of bits in } s}{\#\text{ of bits in } [s]_i},$$

which consist of the ratio of bits in s to the bits in share $[s]_i$. Now, the information rate of a scheme is defined by $\rho = \min_i \rho_i$. In the case that the 'size' of the secret is the same as the size of each of the shares, the information rate $\rho = 1$. Secret sharing schemes that satisfy $\rho = 1$ are called ideal. Note that perfect secret sharing schemes satisfy $\rho \leq 1$.

Relation to error-correcting codes

Shamir Secret Sharing is related to error-correcting codes. In error-correcting codes, a message of length k is extended by $n-k$ 'redundant' bits. The resulting n bits are sent over a noisy channel, where the receiver might not correctly receive the value of all bits (although the order is unchanged). Then, the receiver uses the redundant information to repair the message. The original idea of Reed-Solomon codes was to oversample a polynomial of degree k at $n > k+1$ points and to use interpolation techniques to repair the message afterwards (although this view is not used in practice anymore). This is identical to Shamir Secret Sharing, but rather than reconstructing the secret from only partial information, the secret (polynomial) is used to reconstruct rest of the shares. As a result, Shamir Secret Sharing can handle the input of 'wrong' shares, as these correspond to wrongly transmitted bits in the error-correcting code setting. However, more shares are needed in this case, which leads to the condition $t < n/3$.

Access structures

It is also possible to construct general (non-threshold) secret sharing schemes. In this case there is a general *access structure* which consists of a pair of sets (Γ, Δ) such that $\Gamma, \Delta \subseteq 2^{[n]}$ and $\Gamma \cap \Delta = \emptyset$, where $[n]$ is the set of indices $\{1, \ldots, n\}$. An access structure is called complete if $\Gamma \cup \Delta = 2^{[n]}$. It is called monotone if for all sets $Q \in \Gamma$ all supersets of Q are also in Γ and likewise, for all sets $Q \in \Delta$ all subsets of Q are also in Δ.

In words, this means the following: let $P_1, \ldots, P_n$ be the participants. Then the set Γ contains the 'qualified sets', i.e., all sets of indices $A \subseteq [n]$ such that the participants P_i for $i \in A$ are qualified to reconstruct the secret. Conversely, the set Δ contains all 'non-qualified sets'. The access structure is called complete if every set of participants is either qualified or non-qualified. An access structure is monotone if adding participants to an already qualified set of participants will not disqualify it and conversely, removing participants from a non-qualified set will not make it qualified. Thus, an access structure describes which sets of participants can or cannot reconstruct the secret when working together. In the following, the access structures will always be monotone and complete.

General construction

The following describes a construction, due to Brickell [1], gives rise to so-called linear secret sharing schemes. Given an access structure (Γ, Δ), a (public) matrix $A \in \mathbb{F}_p^{k \times n}$ that satisfies the following relation needs to be constructed:

$$Q \in \Gamma \Leftrightarrow \mathbf{b} \in \{A_Q \mathbf{c} : \mathbf{c} \in \mathbb{F}_p^{|Q|}\},$$

where $\mathbf{b}$ is a vector in $\mathbb{F}_p^k$ and A_Q is the $k \times |Q|$ matrix obtained by taking the columns of A indexed by Q. This means that Q is qualified if and only if $\mathbf{b}$ is in the linear span of the columns of A_Q. The scheme now consists of the following two phases:

- Distribution: The dealer takes a random column vector $\mathbf{r} \in \mathbb{F}_p^k$ such that $s = \mathbf{r} \cdot \mathbf{b}$. He now computes and distributes the shares $[s]_i := \mathbf{r} \cdot \mathbf{a}_i$, where $\mathbf{a}_i$ is the i'th column of A.
- Reconstruction: A qualified set of participants Q computes the $\mathbf{c}$ such that $A_Q\mathbf{c} = \mathbf{b}$. Denote by $[s]_Q$ the vector consisting of the shares of all participants indexed by Q. They then compute $[s]_Q \cdot \mathbf{c} = s$

The correctness of this scheme follows from the fact that $[s]_Q = \mathbf{r}A_Q$ and

$$[s]_Q \cdot \mathbf{c} = (\mathbf{r}A_Q)\mathbf{c} = \mathbf{r}(A_Q\mathbf{c}) = \mathbf{r} \cdot \mathbf{b} = s.$$

Once again, non-qualified sets of participants cannot learn any information about the secret. Let $Q \in \Delta$ be a non-qualified set and consider the following linear system of equations:

$$\begin{aligned} [s]_i &= \mathbf{r} \cdot \mathbf{a}_i, \quad i \in Q \\ s &= \mathbf{r} \cdot \mathbf{b}, \end{aligned}$$

which has $k+1$ unknowns (s and $\mathbf{r}$). Since Q is non-qualified, this means that by construction, $\mathbf{b}$ is linearly independent from the columns of A_Q. Therefore, this system has rank $d+1$, where d is the rank of A_Q. Thus, there are $k+1$ unknowns and $d+1$ equations. This implies that there exactly p^{k-d-1} solutions for every possible value of s, making them all equally likely. Thus, the scheme is perfect.

Consider the following example in $\mathbb{F}_2$:

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

Now, $\mathbf{a}_1 + \mathbf{a}_2 = \mathbf{b}$ and $\mathbf{a}_2 + \mathbf{a}_3 + \mathbf{a}_4 = \mathbf{b}$. Thus, $\Gamma = \{\{1,2\},\{1,2,3\},\{1,2,3,4\},\{2,3,4\}\}$ and $\Delta = \{\{1\},\{1,3\},\{1,3,4\},\{1,4\},\{2\},\{2,3\},\{2,4\},\{3\},\{3,4\},\{4\}\}$. Let the secret be $s = 1$, and let the corresponding vector be $\mathbf{r} = (1,0,1)$. The shares are given by $[s]_1 = [s]_3 = 1$ and $[s]_2 = [s]_4 = 0$. Now participants P_2, P_3 and P_4 pool their shares to reconstruct the secret. The corresponding $\mathbf{c} = (1,1,1)$, so they compute $[s]_{\{1,2,3\}} \cdot \mathbf{c} = (0,1,0) \cdot (1,1,1) = 1 = s$.

Pseudorandom Secret Sharing

One application of Shamir Secret Sharing is Pseudorandom Secret Sharing, due to Cramer, Damgård and Ishai [2]. The goal is to use initially distributed randomness to construct a virtually unlimited supply of Shamir secret sharings of pseudorandom values without further interaction. Let ψ_k be a keyed pseudorandom function that outputs an element of $\mathbb{F}_p$, where k is the key and let a be some common input that all participants agree on. Consider a maximal non-qualified set $B \in \Delta$, i.e., $|B| = t$ when doing Shamir Secret Sharing. Let A be the complement of B, which means that $|A| = n - t$. Now define a t-degree polynomial f_A such that

$$f_A(x) = \begin{cases} f(0) = 1 \\ f(i) = 0 \quad \forall i \in B \end{cases}$$

As this definition fixes $t+1$ points, it uniquely determines the polynomial f_A of degree t. Now, the dealer distributes a random value r_A to all participants in A. This needs to be done independently for every set A that is the complement of a maximal non-qualified set B, i.e., for all sets A of size $n-t$ when doing Shamir Secret Sharing. Consider the polynomial f defined as

$$f(x) = \sum_{A \subset [n]:|A|=n-t} \psi_{r_A}(a) f_A(x),$$

and take $s = f(0) = \sum \psi_{r_A}(a)$. Now, the value s is supposed to be pseudorandom, as the values r_A are all random and ψ is a pseudorandom function. Furthermore, f is the sum of t-degree polynomials, and thus the degree of f is t. The polynomial f is now shared amongst the participants as follows: Each player computes his own share as

$$[s]_j = \sum_{A \subset [n]:|A|=n-t, j \in A} \psi_{r_A}(a) f_A(j) = f(j),$$

without knowing the actual polynomial f. The last equality holds because $f_A(j) = 0$ for all A such that $j \notin A$. The polynomial f can be reconstructed from the shares $[s]_j$ using interpolation as before. While this description focuses on Shamir Secret Sharing, it is possible to extend this procedure to linear schemes for general (non-threshold) access structures.

Homomorphic Secret Sharing

As with encryption, it is possible for secret sharing schemes to have homomorphic properties, i.e., for operations on the secret (plaintext), there are corresponding operations on the shares (ciphertext) that preserve the relation between secret and shares.

Consider the Shamir scheme once more. Let s and t be two secrets with polynomials f and g, respectively. Now consider the sum of the two secrets, $s+t$. Since $s+t = f(0)+g(0) = (f+g)(0)$, the t-degree polynomial $f+g$ is a good candidate for the shared secret $[s+t]$. Conversely, adding the shares $[s]_i$ and $[t]_i$ gives $[s]_i + [t]_i = f(i) + g(i) = (f+g)(i)$, which correspond to the i'th share of $[s+t]$. Thus $[s]_i + [t]_i = [s+t]_i$, which provides the required corresponding operation on the shares.

By the above, the additive homomorphic property of Shamir's scheme is "free" in the sense that everything can be can be computed locally, but what about multiplication? Attempting the same gives $s * t = f(0) * g(0) = (f * g)(0)$ and $[s]_i * [t]_i = f(i) * g(i) = (f * g)(i) = [s * t]_i$. However, the polynomial $f * g$ is of degree $2t$ rather than t, which means that $2t+1$ shares are needed for interpolation of $f * g$. However, needing $2t+1$ shares to reconstruct the secret seems cumbersome. It turns out that it is possible using t shares, as will be shown for general linear secret sharing schemes in the following.

Consider linear secret sharing schemes for a general access structure. Cramer, Dåmgard and Maurer [3] showed that it is possible that, under certain restrictions on the access structure, such schemes can also be made multiplicative. As with Shamir's scheme, the addition follows from the linear properties of the scheme and can be done locally. However, the multiplication needs some extra work.

A Linear Secret Sharing Scheme is called multiplicative if for all secrets x and y, there exists a "recombination" vector $\mathbf{r} = (r_1, \ldots, r_n)$ of length n such that $x*y = \sum_i r_i * [x]_i * [y]_i$, where $*$ signifies the Schur product if the shares consist of vectors. An access structure (or rather, its *adversary structure*) is called Q_2 if there are no non-qualified sets $S_1, S_2 \in \Delta$ such that $S_1 \cup S_2 = [n]$. Note that, for threshold schemes such as Shamir's scheme, this is equivalent to $2t < n$. The result of Cramer, Dåmgard and Maurer states that any linear secret sharing scheme for an adversary structure that is Q_2 can be made multiplicative (and secure against passive attackers).

Multiplication is performed as follows in a multiplication linear secret sharing scheme. Consider the participants $P_1, \ldots, P_n$, where participant P_i has shares $[s]_i$ for secret s and $[t]_i$ for secret t. Now, to get the shares for $[s*t]$, each participant needs to compute $[s]_i * [t]_i = \tilde{c}_i$. Then, the participant P_i secret-shares his computed value $\tilde{c}_i$, resulting in shares $[\tilde{c}_i]_j$ for $1 \leq j \leq n$. Participant P_j now computes shares $[s * t]_j = \sum_{i=1}^{n} r_i [\tilde{c}_i]_j$ for $[s * t]$, where the vector $\mathbf{r} = (r_1, \ldots, r_n)$ is the recombination vector that follows from the multiplicative property of the scheme.

References

[1] E. Brickell, Some ideal secret sharing schemes, In *Advances in Cryptology–EUROCRYPT '98*, pages 468–475. Springer, 1990.

[2] R. Cramer, I. Damgå and Y. Ishai. Share conversion, pseudorandom secret-sharing and applications to secure computation. *Theory of Cryptography*, pages 342–362, 2005.

[3] R. Cramer, I. Damgård and U. Maurer. General secure multi-party computation from any linear secret-sharing scheme. In *Advances in Cryptology–EUROCRYPT '00*, pages 316–334. Springer, 2000.

[4] B. Schoenmakers, Secret Sharing. 2009. Slides available at `http://www.win.tue.nl/~berry/2WC11/`.

[5] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.