PACKED SHAMIR SECRET SHARING

NAMAN KUMAR

ABSTRACT. Shamir Secret Sharing is a classical result in cryptography that finds applications in server privacy, distributed computing, MPC, security of P2P networks, and so on. A common variant of Shamir Secret Sharing is *Packed Secret Sharing* (henceforth referred to as PSS) which allows the sharing of *multiple* secrets with small overhead, introduced by [FY92] in the context of parallel invocations of MPC protocols. More recently, has been used to construct highly efficient MPC protocols in the low-security threshold setting. In this document, we will review the classical PSS scheme.

1. THRESHOLD SECRET SHARING

Shamir Secret Sharing is a *threshold* secret sharing scheme. Intuitively, a t-out-of-n threshold secret sharing scheme captures the setting in which n parties hold shares of a secret and *at least* t parties are required to reconstruct the secret. In particular, no $t-1$ colluding parties can gain any information about the secret. We capture this idea in the definition below.

Definition 1.1 (t-out-of-n threshold secret sharing scheme)**.** *A t-out-of-n threshold secret sharing scheme over a message space $\mathcal{M}$ is a tuple of algorithms* $(\mathsf{share}, \mathsf{reconstruct})$ *such that:*

- $(s_1, \ldots, s_n) \leftarrow \mathsf{share}(1^\lambda, m)$ *outputs an n-tuple of shares,*
- $x \leftarrow \mathsf{reconstruct}(\{y_{i_j}\}_{j\in[t]})$ *outputs a message $x \in \mathcal{M}$.*

The scheme satisfies the following properties:

(1) ***Correctness:*** *For all $m \in M$ and any subset $\{i_j\}_{j\in[t]} \subseteq (s_1, \ldots, s_n)$ of size t,*

$$\Pr[\mathsf{reconstruct}(\{s_{i_j}\}_{j\in[t]}) = m : (s_1, \ldots, s_n) \leftarrow \mathsf{share}(1^\lambda, m)] = 1.$$

(2) ***Perfect Security:*** *For all messages m, m' and subsets $S \subseteq (s_1, \ldots, s_n)$ such that $|S| < t$, for all PPT adversaries $\mathcal{A}$ it holds that*

$$\Pr[\mathcal{A}(1^\lambda, \{s_i : i \in S\}) = m] = \Pr[\mathcal{A}(1^\lambda, \{s_i : i \in S\}) = m']$$

where the probability is taken over $(s_1, \ldots, s_n) \leftarrow \mathsf{share}(1^\lambda, m)$.

2. SHAMIR SECRET SHARING

We begin by reviewing Shamir secret sharing, which works on the principle of *Lagrange interpolation*. Shamir secret sharing is a t-out-of-n secret sharing scheme for any $t \leq n$. The message space is any finite field $\mathbb{F}_q$.

t-out-of-n Shamir Secret Sharing

Parameters: A security parameter λ, a threshold t and n parties. Let $\mathbb{F}_q$ be the message space $\mathcal{M}$.

Protocol:

The share algorithm $\mathsf{share}(1^\lambda, m)$.

(1) The dealer samples $a_1, \ldots, a_{t-1} \leftarrow \mathbb{F}_q$ field elements at random and sets $a_0 := m$, constructing the polynomial $p(x) = a_0 + a_1 x + \cdots + a_{t-1}x^{t-1}$.

OREGON STATE UNIVERSITY

Date: January 17, 2025.

(2) The dealer computes $s_i := p(i)$ for each $i \in [n]$ and outputs $(s_1, \ldots, s_n)$.

The reconstruct algorithm $\mathsf{reconstruct}(\{y_{i_j}\}_{j\in[t]})$.

(1) The parties compute the Lagrange basis polynomials

$$L_{i_k}(x) = \frac{\prod_{j\neq k}(x - i_j)}{\prod_{j\neq k}(i_k - i_j)}$$

for each k.

(2) The parties compute the polynomial $p'(x) = \sum y_{i_j} L_{i_j}$ and output $p'(0)$.

We can now verify correctness and security. In case the shares are honest, the polynomial $p'(x)$ is the degree-$(t-1)$ polynomial interpolated at the points $\{s_{i_j}\}_{j\in[t]}$, which evaluates to s_{i_j} at i_j. Since p has degree-$(t-1)$, there is a unique polynomial which satisfies this requirement at all t different points. Hence, the interpolated polynomial is p, and the parties can obtain $m = p(0)$.

To verify security, suppose that $t-1$ colluding parties wish to recover m (in case there are less than $t-1$ complete $t-1$ parties by picking random shares). Then for each $m' \in \mathbb{F}_p$, there exists some y' such that $(y_{i_1}, \ldots, y_{i_{t-1}}, y')$ reconstructs a degree-$(t-1)$ polynomial with m' as the constant coefficient. Since the parties do not know a tth share, the polynomial and thus the secret cannot be recovered.

3. Packed Secret Sharing

At a high level, PSS is an extension of Shamir's secret sharing scheme that allows the sharing of k different secrets simultaneously. We introduce the scheme of [FY92], which is a $(t-k, t, k, n)$-packed secret sharing scheme, where k is the number of secrets, n is the number of parties, t parties are required to recover the secret, and no colluding group of less than $t-k$ parties can gain any information about the secrets. No security guarantee is made about m colluding parties if $t - k \leq m < t$.

Intuitively, the scheme uses additional properties of the polynomial to hide more than a single secret.

$(t-k, t, k, n)$ **Packed Shamir Secret Sharing**

Parameters: A security parameter λ, a threshold t, number of secrets $k < t$ and n parties. Let $\mathbb{F}_q$ be the message space $\mathcal{M}$. Let the secrets be $(m_1, \ldots, m_k)$ and let $(e_1, \ldots, e_k) \in \mathbb{F}_q^k$ and $(\alpha_1, \ldots, \alpha_n) \in \mathbb{F}_q^k$ be distinct public values where $\alpha_i \neq e_j$ for any i, j.

Protocol:

The share algorithm $\mathsf{share}(1^\lambda, (m_1, \ldots, m_k))$.

(1) The dealer samples any degree-$(t-1)$ polynomial p such that $p(e_i) = m_i$ for all $i \in [k]$. Note that there are at least q such polynomials since $t - 1 \geq k$.

(2) The dealer sets $s_i = p(\alpha_i)$ for all $i \in [n]$.

The reconstruct algorithm $\mathsf{reconstruct}(\{y_{i_j}\}_{j\in[t]})$.

(1) The parties compute the Lagrange basis polynomials

$$L_{i_k}(x) = \frac{\prod_{j\neq k}(x - \alpha_{i_j})}{\prod_{j\neq k}(\alpha_{i_k} - \alpha_{i_j})}$$

for each k.

(2) The parties compute the polynomial $p'(x) = \sum y_{i_j} L_{i_j}$ and output $(p'(e_1), \ldots, p'(e_k))$.

The correctness of the protocol follows from a similar argument to the correctness of Shamir Secret Sharing. For security, note that we can write $p(x) = q(x)\prod(x - e_i) + \sum m_i L_{e_i}(x)$ where $q(x)$ is a random degree-$(t-k)$ polynomial over $\mathbb{F}_q$. Thus $(t-k)$ or less colluding parties cannot find any information about

$q(x)$. However, more than $(t-k)$ colluding parties could obtain additional data points, for example $q(\alpha_j)$ for certain values of j in the colluding group.

References

[FY92] Matthew Franklin and Moti Yung. Communication complexity of secure computation (extended abstract). In *Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing*, STOC '92, page 699710, New York, NY, USA, 1992. Association for Computing Machinery.