Shamir Secret Sharing and Reed-Solomon Codes

Notes by Linus Tang.

These notes have not been thoroughly reviewed. Any errors below are my own responsibility.

Sources:

- MIT 6.5610 lecture notes by Yael Kalai
  - https://65610.csail.mit.edu/2025/lec/l11-ss.pdf
- UWash Coding Theory lecture notes on Reed-Solomon codes by Anup Rao
  - https://homes.cs.washington.edu/~anuprao/pubs/codingtheory/lecture4.pdf

Secret sharing definition

Secret sharing schemes are good for distributing shares of an important secret to a group of people such that no individual person gains information about the secret, but sufficiently many people can communicate their shares to reveal the secret.

Formally, a t-out-of-n secret sharing scheme over finite message space $\mathcal{M}$ consists of two efficient algorithms:

- Share is a randomized algorithm that inputs a message $m \in \mathcal{M}$ and outputs n shares $(s_1, ..., s_n)$.
- $\mathsf{Reconstruct}$ is deterministic. It inputs t shares $\{(i, s_i)\}_{i \in I}$ for $|I| = t$ and outputs a message $m \in \mathcal{M}$.

They should satisfy the following properties:

- Correctness: **reconstruction recovers the original message** with probability 1, i.e.

$$\Pr_{(s_1,...,s_n) \leftarrow \mathsf{Share}(m)} \left[\mathsf{Reconstruct}\left(\{i, s_i\}_{i \in I}\right) = m\right] = 1$$

  for all $I \subseteq [n]$ with $|I| = t$.
- Security: **any collection of fewer than t shares reveals no information about the message**, i.e. for all $m, m' \in \mathcal{M}$ and $I \subseteq [n]$ with $|I| < t$,

$$(s_i)_{i \in I} \equiv (s'_i)_{i \in I}$$

  where $(s_i)_{i \in [n]} \leftarrow \mathsf{Share}(m)$ and $(s'_i)_{i \in [n]} \leftarrow \mathsf{Share}(m')$.

Construction by Shamir

We will reference many times the following important fact, which can be explained through Lagrange interpolation or the Vandermonde matrix/determinant:

Key fact.

Let d be a positive integer and points $(x_1, y_1), ..., (x_{d+1}, y_{d+1})$ be $d + 1$ pairs of elements of $\mathbb{F}_p$ with $x_1, ..., x_{d+1}$ distinct. Then there is a unique polynomial P of degree at most d such that $P(x_i) = y_i$ for $i = 1, ..., d + 1$.

A polynomial P of degree $t - 1$ over a finite field is uniquely determined by t points $(x, P(x))$ with distinct x.

Let $p > n$ be a prime and work over $\mathbb{F}_p$. (Actually, you can work over any finite field.)

We can write our message m in a fixed number of bits, so we can assume that $\mathcal{M} = \{0, 1\}$ and perform secret sharing on every bit of the message. (Alternatively we can encode m as a fixed-length sequence

of elements of $\mathbb{F}_p$ and perform secret sharing on every entry of the sequence, allowing us to assume $\mathcal{M} = \mathbb{F}_p$.)

Choose a random degree $t-1$ polynomial $f : \mathbb{F}_p \to \mathbb{F}_p$ such that $f(0) = m$. This can be done by setting the constant coefficient to m and choosing all coefficients independently and uniformly over $\mathbb{F}_p$.

The secret shares are given by $s_i = f(i)$ for all $1 \leq i \leq n$.

Reconstruction: Given any t shares $\{(i, s_i)\}_{i \in I}$ with $|I| = t$, we can recover the unique possible polynomial f of degree $d-1$ which satisfies $f(i) = s_i$ for all $i \in I$, by Lagrange interpolation. Then we can compute $f(0) = m$.

Security: Given $t' < t$ shares $\{(i, s_i)\}_{i \in I}$ with $|I| = t'$, there are $p^{t-t'}$ polynomials which satisfy $f(i) = s_i$ for all $i \in I$. Furthermore, exactly $p^{t-t'-1}$ of these polynomials achieve each possible evaluation at 0. (Exercise: use the key fact above to prove this!) Therefore, the distributions in the security definition are identical (because they are both uniform over $|I|$).

Shamir secret sharing is quite closely related to Reed-Solomon error correction codes.

Quick review on error correction codes

For a set S, an *error correction code* of length n over S with distance d is a subset C of S^n such that the Hamming distance between any two distinct elements of C (the number of coordinates at which they differ) is at least d.

If S is a finite field $\mathbb{F}_q$ and C is a subspace of $\mathbb{F}_q^n$, then C is called a *linear* error correction code.

One motivation for studying error correction codes: Alice can send a message $m \in C$ to Bob. The message is robust to up to $d-1$ erasures and $\left\lfloor \frac{d-1}{2} \right\rfloor$ errors. That is,

- If m gets corrupted to m' by hiding up to $d-1$ coordinates of m, then Bob, receiving m', can still uniquely determine the original message $m \in C$.
- If m gets corrupted to m' by changing up to $d-1$ coordinates of m to different values, then Bob, receiving m', can still uniquely determine the original message $m \in C$.

Reed-Solomon error correction codes

The Reed-Solomon error correction code has a very simple definition. Let $\mathbb{F}_q$ be a finite field. Consider the set P_k of polynomials $f : \mathbb{F}_q \to \mathbb{F}_q$ of degree at most $k-1$, which is a k-dimensional vector space over $\mathbb{F}_q$.

Let γ be a generator of $\mathbb{F}_q^\times$ so that the elements of $\mathbb{F}_q$ are $\{0, \gamma^0, \gamma^1, ..., \gamma^{q-2}\}$.

The map $\mathsf{Eval} : P_k \to \mathbb{F}_q^q$ given by $f \mapsto (f(0), f(\gamma^0), f(\gamma^1), ..., f(\gamma^{q-2}))$ is injective and we let its image be C_k, which is the Reed-Solomon code. By the key fact, C_k has distance $q-k+1$, which is the maximum possible distance of a k-dimensional code in a q-dimensional space.

Encoding efficiently

If we let $f(X) = f_0 + f_1 X + \cdots + f_{k-1} X^{k-1}$ and consider Eval as a linear transformation

$$(f_0, ..., f_{k-1}) \mapsto (f(0), f(\gamma^0), f(\gamma^1), ..., f(\gamma^{q-2})),$$

then the transformation has matrix

$$G = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & \gamma^0 & \gamma^0 & \dots & \gamma^0 \\ 1 & \gamma^1 & \gamma^2 & \dots & \gamma^{k-1} \\ 1 & \gamma^2 & \gamma^4 & \dots & \gamma^{2(k-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \gamma^{q-1} & \gamma^{2(q-2)} & \dots & \gamma^{(k-1)(q-2)} \end{pmatrix}.$$

The encoding becomes computing the matrix-vector multiplication

$$G \cdot \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{k-1} \end{pmatrix},$$

which can be done in $O(q \log k)$ field operations using FFT.

Decoding efficiently

Decoding from up to $q-k$ erasures is identical to reconstruction from Shamir secret sharing. Simply choose k non-erased points and interpolate to recover the original polynomial f.

Figuring out how to efficiently decode from up to $\frac{q-k}{2}$ errors is a challenging and fun puzzle!

Setup/problem: Let $\alpha_1, \dots, \alpha_q$ be the elements of $\mathbb{F}_q$ and $(\beta_1, \dots, \beta_q)$ be a noised message. In particular, we may assume that there exists a polynomial f of degree $\leq k-1$ such that $f(\alpha_i) = \beta_i$ holds for all but at most $\left\lfloor \frac{q-k}{2} \right\rfloor$ values i. Because the code C_k has distance $q-k+1$, it follows that f is unique. How can we efficiently determine f?

Solution:

Consider the set $\mathcal{X}$ of pairs of polynomials (e, g) of degrees at most $\left\lfloor \frac{q-k}{2} \right\rfloor$ and $\left\lfloor \frac{q+k}{2} - 1 \right\rfloor$, respectively, such that

$$h(X, Y) = e(X) \cdot Y - g(X)$$

vanishes on all (α_i, β_i).

This is a vector space over $\mathbb{F}_q$ because the vanishing constraints are linear constraints in the coefficients of e and g.

Note that $\mathcal{X}$ has at least one element (e, g) with e nonzero, as shown by the example

$$e(X) = \prod_{i : f(\alpha_i) \neq \beta_i} (X - \alpha_i), \quad g(X) = e(X) f(X).$$

We now prove that all $(e, g) \in \mathcal{X}$ must satisfy $g = ef$.

Indeed, if we define $u = ef - g$, then at least $\left\lceil \frac{q+k}{2} \right\rceil$ field elements α_i satisfy

$$\begin{aligned} u(\alpha_i) &= e^*(\alpha_i) f(\alpha_i) - g^*(\alpha_i) \\ &= e^*(\alpha_i) \beta_i - g^*(\alpha_i) \\ &= h(\alpha_i, \beta_i). \end{aligned}$$

Now u is a polynomial of degree at most $\left\lfloor \frac{q+k}{2} - 1 \right\rfloor$ which vanishes on at least $\left\lceil \frac{q+k}{2} \right\rceil$ points, which implies that u is the zero polynomial, so $g = ef$ as desired.

Thus, it suffices to find any element $(e, g) \in \mathcal{X}$ with e nonzero. We efficiently (complexity quadratic in q if I'm not mistaken) obtain a basis for $\mathcal{X}$ by writing the linear constraints imposed by the vanishing condition on the coefficients of g and e and performing Gaussian elimination. After finidng a basis, choosing such (e, g) is straightforward, and we can determine $f = \frac{g}{e}$, as desired.

Remarks:

- The linked notes on Reed Solomon codes mention a "near linear" time solution by using polynomial interpolation to find nontrivial $(e, g) \in \mathcal{X}$. I haven't been able to figure out how this works.
- I'm not very sure how to motivate the above solution. I believe that a useful intuition is that everything about the Reed Solomon code is very "linear" in a sense, so we want to solve for the coefficients of f by solving some linear system. The error pairs with $f(\alpha_i) \neq \beta_i$ get in the way of this, so we can "mask" over these pairs by introducing the error-locator polynomial

$$e(X) = \prod_{i : f(\alpha_i) \neq \beta_i} (X - \alpha_i).$$