Secret sharing: Suppose we have a secret and want to distribute it among $n$ parties such that any $t$ of them can subsequently recover the secret and any $(t-1)$ subset cannot [e.g., Board of directors at Coca-Cola want to protect Coca-Cola recipe.]

Two algorithms: Share($m$) $\rightarrow$ \{s_i\}_{i=1}^n : takes a message $m$ and outputs a collection of $n$ shares

- $\text{Reconstruct}(\{s_i\}) = m / 1$ : takes a set of shares and reconstructs the message

Recall $S_i$ : secret shares $s_1, s_2, \ldots, s_n$ are defined so that $s_i \in \{0, 1\}^m$.
A little more detail... how to construct the polynomial $f$. Lagrange interpolation.

Let $(x_0, y_0), \ldots, (x_t, y_t)$ be a collection of $t+1$ points. To find the polynomial of degree $t$ that interpolates these points, we can write

$$f(x) = a_0 + a_1 x + \cdots + a_t x^t,$$

where $a_0, \ldots, a_t \in \mathbb{Z}_p$

Then, we can write

$$f(x_0) = a_0 + a_1 x_0 + \cdots + a_t x_0^t = y_0$$

$$f(x_t) = a_0 + a_1 x_t + \cdots + a_t x_t^t = y_t$$

Interpolating a polynomial over $\mathbb{Z}_p$ just corresponds to solving a linear system over $\mathbb{Z}_p$. A unique solution exists as long as the Vandermonde matrix is invertible. It turns out that you can show (via linear algebra) that

$$\det \begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^t \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_t & x_t^2 & \cdots & x_t^t \end{pmatrix} = \prod_{0 \leq j < k \leq t} (x_j - x_k)$$

If $x_i, x_j$ are all distinct, and we work over a finite field (e.g., there are no non-trivial divisors of 0), then this matrix is invertible and we can interpolate efficiently.

Let us now analyze the properties of Shamir’s secret sharing scheme:

**Correctness:** Follows by uniqueness of interpolating polynomial (e.g., $t$ shares uniquely define a polynomial of degree $t-1$)

**Security:** Given any subset of $(t-1)$ shares $(i_1, y_1), \ldots, (i_r, y_r)$, and any message $m \in \mathbb{Z}_p$, there is a unique polynomial $f$ of degree $t-1$ where

$$f(i_1) = y_1, \ldots, f(i_r) = y_r$$

and $f(0) = m$.

Thus, any $(t-1)$ shares can be consistent with secret-sharing of any message $m$ => information-theoretic security.

**Efficiency:** Both share-generation and share-reconstruction consist of polynomial evaluation and interpolation, both of which are efficiently computable (see above).

Secret sharing is very useful for building threshold cryptosystems. Here, we describe one example with threshold RSA signatures:

**Setup:** sample primes $p, q$, set $N = pq$

choose $e, d$ such that $ed = 1$ (mod $\varphi(N)$)

$$\mapsto \text{sk: (N, e)}$$

(Useful for protecting signing keys)

$$\text{vk: (N, d)}$$

**Sign(sk, m):** Output $\sigma \leftarrow H(m)^d$

**Verify(vk, m, \sigma):** Output 1 if $H(m) = \sigma^e$

Can apply $n$-out-of-$n$ secret sharing to signing key $d$: sample $d_1, \ldots, d_n \in \mathbb{Z}_p$ such that $\sum_{i=1}^n d_i = d$ (mod $\varphi(N)$):

$$\begin{array}{c}
\text{Pi: (d_i)} \\
\text{H(m)}^d_i \\
\text{m} \\
\text{client (m)} \\
\text{H(m)}^d \\
\end{array}
\begin{array}{c}
\text{Pi: (d_i)} \\
\text{m} \\
\text{client (m)} \\
\text{H(m)}^d \\
\end{array}
\begin{array}{c}
\text{m} \\
\text{H(m)}^d \\
\end{array}
\begin{array}{c}
\text{Combine signatures:} \\
\prod_{i=1}^n H(m)^d_i = H(m)^{\sum_{i=1}^n d_i} = H(m)^d = \sigma
\end{array}$$

Can apply $n$-out-of-$n$ secret sharing to signing key $d$: sample $d_1, \ldots, d_n \in \mathbb{Z}_p$ such that $\sum_{i=1}^n d_i = d$ (mod $\varphi(N)$):