Secret Sharing
Principle

- There is a set of parties $\mathbf{P} = \{P_1, \ldots, P_n\}$.
- There is some (secret) value $v$.
  - Shares of $v$ are distributed among $P_1, \ldots, P_n$.
- There is a set of subsets of parties $\mathcal{S} \subseteq \mathcal{P}(\mathbf{P})$.
  - $\mathcal{S}$ is upwards closed — if $P_1 \in \mathcal{S}$ and $P_1 \subseteq P_2$, then also $P_2 \in \mathcal{S}$.
  - $\mathcal{S}$ is called an access structure.
  - Let us call the elements of $\mathcal{S}$ privileged sets.
- Certain parties $P_{i_1}, \ldots, P_{i_k}$ have come together and are trying to find out $v$.
- They must succeed only if $\{P_{i_1}, \ldots, P_{i_k}\} \in \mathcal{S}$.
General solution

- Let $v$ be an element of some (additive) group $G$.
- Express $\varphi$ as a propositional formula $\overline{\varphi}(x_1, \ldots, x_n)$, such that for each $Q \subseteq P$

  \[
  \overline{\varphi}(P_1 \in Q, \ldots, P_n \in Q) \text{ iff } Q \in \varphi.
  \]

  - Use only operations AND and OR (of arbitrary arity) in $\overline{\varphi}$.
- Define a share for each node in the syntax tree of $\overline{\varphi}$:
  - The share of the root node is $v$.
  - If the share of an OR-node is $x$, then the shares of all its immediate descendants are $x$, too.
  - If the share of an AND-node of arity $m$ is $x$, then generate $r_1, \ldots, r_{m-1} \in_R G$ and put $r_m = x - \sum_{i=1}^{m-1} r_i$. The shares of the immediate descendants are $r_1, \ldots, r_m$.
- Give the party $P_i$ the shares of all leaf nodes marked with $x_i$. 

Example

Let $\mathbf{P} = \{P_1, P_2, Q_1, Q_2, Q_3\}$.

- Let $P_1$ and $P_2$ be allowed to know the secret.
- Let two $Q$-s be allowed to replace one of the $P$-s.

\[
\overline{\varphi}(P_1, P_2, Q_1, Q_2, Q_3) = P_1 \land P_2 \lor P_1 \land (Q_1 \land Q_2 \lor Q_1 \land Q_3 \lor Q_2 \land Q_3) \lor P_2 \land (Q_1 \land Q_2 \lor Q_1 \land Q_3 \lor Q_2 \land Q_3)
\]
Example

\[
\begin{array}{c}
\lor \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
& \\
\end{array}
\]

\[
\begin{array}{c}
\land \quad \land \quad \land \quad \land \quad \land \\
P_1 \quad P_2 \\
P_1 \\
P_2 \\
Q_1 \quad Q_2 \\
Q_1 \quad Q_3 \\
Q_2 \quad Q_3 \\
Q_1 \quad Q_2 \\
Q_1 \quad Q_3 \\
Q_2 \quad Q_3 \\
\end{array}
\]
Example
Example
Example
Example
Example
Example

- We generate the values \( r_1, \ldots, r_9 \in \mathbb{R} \) and give the following values to following parties:
  - \( P_1 \) learns \( s_{11} = v - r_1 \) and \( s_{12} = v - r_2 \);
  - \( P_2 \) learns \( s_{21} = r_1 \) and \( s_{22} = v - r_3 \);
  - \( Q_1 \) learns \( t_{11} = r_4, t_{12} = r_5, t_{13} = r_7 \) and \( t_{14} = r_8 \);
  - \( Q_2 \) learns \( t_{21} = r_2 - r_4, t_{22} = r_6, t_{23} = r_3 - r_7 \) and \( t_{24} = r_9 \);
  - \( Q_3 \) learns \( t_{31} = r_2 - r_5, t_{32} = r_2 - r_6, t_{33} = r_3 - r_8 \) and \( t_{34} = r_3 - r_9 \).

- When a privileged set of parties meet then they figure out which of the values to add up to recover \( v \).
- A non-privileged set gets no information about \( v \).
The components

- Number of parties $n$.
- The secret $v$.
- The parties $P_1, \ldots, P_n$ holding the shares of $v$, and the dealer $D$ that originally knows $v$.
- The access structure $\mathcal{P}$.
  - $\mathcal{P}$ is a $t$-threshold structure if all minimal elements in $\mathcal{P}$ have the cardinality $t$.
- The dealing protocol, where $D$ distributes the shares among $P_1, \ldots, P_n$.
- The recovery protocol, where a privileged set computes $v$. 
Shamir’s threshold secret sharing scheme

- Let \( v \in \mathbb{F} \) for some (finite) field \( \mathbb{F} \).
  - In practice, \( \mathbb{F} = \mathbb{Z}_p \) for some suitable prime \( p \).
- Shamir’s \((n, t)\)-scheme is for \( n \) parties, where \( \wp \) is the \( t \)-threshold structure and \( n < |\mathbb{F}| \).
- Dealing:
  - The dealer randomly chooses values \( a_1, \ldots, a_{t-1} \in \mathbb{F} \).
  - He defines the polynomial
    \[
    q(x) = v + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1}.
    \]
  - The dealer securely sends to each \( P_i \) his share \( s_i = q(i) \).
- Recovering \( v \):
  - The parties \( P_{i_1}, \ldots, P_{i_t} \) together know that
    - \( q(i_1) = s_i, \ldots, q(i_t) = s_t \);
    - The degree of \( q \) is at most \( t - 1 \).
  - This information is sufficient to recover the coefficients of \( q \).
Theorem. Let $x_1, y_1, \ldots, x_t, y_t \in \mathbb{F}$, such that the values $x_1, \ldots, x_t$ are all different. Then there exists exactly one polynomial $q$ of degree at most $t - 1$, such that $q(x_i) = y_i$ for all $i \in \{1, \ldots, t\}$.

Proof. This polynomial $q$ is (Lagrange interpolation formula)

$$q(x) = \sum_{j=1}^{t} y_j \prod_{k \neq j} \frac{x - x_k}{x_j - x_k}.$$ 

It’s degree is $\leq t - 1$ and it satisfies $q(x_i) = y_i$ for all $i$.

There cannot be more than one: if $q'(x_i) = y_i$ for all $i \in \{1, \ldots, t\}$ and $\deg q' \leq t - 1$, then $(q - q')$ is a polynomial of degree at most $t - 1$ with at least $t$ roots $(x_1, \ldots, x_t)$. Hence $q - q' = 0$. $\Box$
Shamir’s scheme: simpler recovery

- The parties $P_{i_1}, \ldots, P_{i_t}$ are not interested in the entire polynomial, but just the secret value $v = q(0)$.
- According to Lagrange interpolation formula

$$v = \sum_{j=1}^{t} s_{ij} \prod_{k \neq j} \frac{i_k}{i_k - i_j}.$$ 

- In particular, note that $v$ is computed as a linear combination of the shares $s_{ij}$ with public coefficients.
Security of Shamir’s scheme

- Suppose that we are given shares $s_{i_1}, \ldots, s_{i_{t-1}}$.
- Then for each possible value of $v$, there exists exactly one polynomial $q$ of degree at most $t$, such that

\[
q(0) = v, \quad q(i_1) = s_{i_1}, \ldots \quad q(i_{t-1}) = s_{i_{t-1}}.
\]

- Hence all values of $v$ are possible. Moreover, they are equally possible.
  - There is the same number of suitable polynomials for each value of $v$.
- Similarly, if we have even less shares then all values of $v$ are equally possible.
Let two secrets be shared:

- the shares of $v$ are $s_1, \ldots, s_n$;
- the shares of $v'$ are $s'_1, \ldots, s'_n$.

Let $a, b \in \mathbb{F}$. How can the parties $P_1, \ldots, P_n$ obtain shares for the value $av + bv'$?
Verifiable secret sharing

- If some party $P_i$ is malicious, then it can input a wrong share to the recovery protocol.
- The recovered secret $v$ will then be incorrect.
- Also, a malicious dealer may give inconsistent shares to the parties $P_i$.
- In **verifiable secret sharing** the parties commit to the shares they have received.
Verifiable secret sharing

- If some party $P_i$ is malicious, then it can input a wrong share to the recovery protocol.
- The recovered secret $v$ will then be incorrect.
- Also, a malicious dealer may give inconsistent shares to the parties $P_i$.
- In verifiable secret sharing the parties commit to the shares they have received.
- A malicious party $P_i$ may also send $s_{it}$ to one party, but $s'_{it}$ to some other party.
- In multi-party protocols with malicious participants, a broadcast channel is often needed.
  - We thus assume the existence of a broadcast channel.
- It can be implemented using point-to-point channels and the Byzantine agreement.
Feldman’s scheme

- Let $\mathbb{F} = \mathbb{Z}_p$. Let $G$ be a group with hard discrete log., such that $|G|$ is divisible by $p$. Let $g \in G$ have order $p$.
- Let $D$ use Shamir’s scheme to share $v$. When $D$ has constructed the polynomial $q(x) = v + \sum_{i=1}^{t-1} a_i x^i$, he (authentically) broadcasts

$$y_0 = g^v, \quad y_1 = g^{a_1}, \quad \ldots, \quad y_{t-1} = g^{a_{t-1}}$$

in addition to sending the shares to the parties $P_i$.
- Whenever a party sees a share $s_j$ he checks its consistency:

$$g^{s_j} \overset{?}{=} \prod_{i=0}^{t-1} y_j^{i_j}.$$

Exercise. What does the consistency check do?
Security of Feldman’s scheme

- Nobody can cheat — the “commitments” $y_0, \ldots, y_{t-1}$ fix the polynomial $q$.
  - Everybody can check whether $q(i)$ equals a given value.
- Something about the secret can be leaked, because $y_0 = g^v$ does not fully hide $v$.
  - Use only the hard-core bits of discrete logarithm to store the “real” secret in $v$.
- This makes the shares larger.
Recall Pedersen’s commitment scheme:

- Let $h \in G$ be another element of order $p$, such that nobody knows $\log_g h$.
- To commit $m \in \mathbb{Z}_p$, the committer randomly generates $r \in \mathbb{Z}_p$ and sends $g^mh^r$ to the verifier.
- To open the commitment, send $(m, r)$ to the verifier.
- The commitment is unconditionally hiding, because $g^mh^r$ is a random element of $\langle g \rangle$.
- The commitment is computationally binding, because the ability to open a commitment in two different ways allows to compute $\log_g h$.

In Pedersen’s VSS, the dealer commits to the coefficients of the polynomial $q$. 
Pedersen’s scheme

■ Dealing protocol

◆ \( D \) randomly chooses \( a_1, \ldots, a_{t-1}, a'_0, \ldots, a'_{t-1} \in \mathbb{Z}_p \). Also defines \( a_0 = v \).

◆ Define \( q(x) = \sum_{i=0}^{t-1} a_i x^i \) and \( q'(x) = \sum_{i=0}^{t-1} a'_i x^i \).

◆ The share \((s_i, s'_i)\) of \( P_i \) is \((q(i), q'(i))\).

◆ \( D \) broadcasts \( y_i = g^{a_i} h^{a'_i} \) for \( i \in \{0, \ldots, t-1\} \).

■ Verification: when somebody sees a share \((s_i, s'_i)\), he verifies

\[
g^{s_i} h^{s'_i} \overset{?}= \prod_{i=0}^{t-1} y_i^{j_i}
\]
The broadcast value $y_0$ hides $v$ unconditionally.

Ability to change a share (or the pair $(v, a'_0)$) implies the knowledge of $\log_q h$.

Having less than $t$ shares allows one to freely choose the secret $v$. Then there exists an $a'_0$ that is consistent with $y_0$.

**Exercise.** How to construct linear combinations of shared secrets when using Feldman’s or Pedersen’s secret sharing scheme? I.e. how do the dealer’s commitments change?
Threshold encryption

- Public-key encryption system.
- The public key is a single value.
- The secret key is distributed among several authorities.
- To decrypt a ciphertext $c$:
  - Each authority computes $D(sk_i, c)$ and broadcasts it.
  - If at least $t$ authorities have broadcast the share of the decrypted ciphertext, the plaintext can be reconstructed from them.
ElGamal encryption scheme

Let $G, g, p$ be as before.

- Secret key — $\alpha \in \mathbb{Z}_p$. Public key — $\chi := g^\alpha$.
- Plaintext space: $G$. Ciphertext space: $G \times G$.
- To encrypt a plaintext $m \in G$:
  - randomly generate $r \in \mathbb{Z}_p$;
  - output $(g^r, m \cdot \chi^r)$.
- To decrypt a ciphertext $(c_1, c_2)$:
  - output $c_2 \cdot c_1^{-\alpha}$.
- Note, that after the decryption, the value $c_1^\alpha = \chi^r$ is not sensitive any more.
Threshold scheme

- Use ElGamal scheme. Distribute the secret key $\alpha$ among the $n$ authorities $P_1, \ldots, P_n$ using Shamir’s $(n, t)$-scheme.
  - Let the shares be $s_1, \ldots, s_n$.
  - Recall that for each $Q = \{i_1, \ldots, i_t\}$ there exist coefficients $\gamma_{i_1}^Q, \ldots, \gamma_{i_t}^Q \in \mathbb{Z}_p$, depending only on $Q$, such that
    $$\alpha = \sum_{j=1}^t \gamma_{i_j}^Q s_{i_j}.$$  

- Decryption:
  - given $(c_1, c_2)$, the authority $P_i$ broadcasts $d_i = c_1^{s_i}$.
  - given $d_{i_1}, \ldots, d_{i_t}$, where $\{i_1, \ldots, i_t\} = Q$, we find
    $$c_1^\alpha = \prod_{j=1}^t d_{i_j}^{\gamma_{i_j}^Q}$$

    and the plaintext is $m = c_2 \cdot (c_1^\alpha)^{-1}$.

Exercise. How could we use Feldman’s scheme for verifiability?