SECRET SHARING

SHARED CONTROL

THRESHOLD SCHEMES - SHAMIR

ATTACKS ON KEY ESTABLISHMENT

ORIGINAL MOTIVATION

Safeguard cryptographic keys from loss, create backup copies.
The more # copies, greater the exposure
Smaller # copies, greater risk of loss
Enhanced reliability w/o increased risk ⇒ secret sharing

SECRET SHARING: BASIC IDEA

Given secret, divide it into pieces called shares and distribute amongst users
Pooled shares of specific subsets of users allow reconstruction of original secret
- cooperation by \( t \) out of \( n \) users
**SHARED CONTROL SCHEMES**

Dual control by modular addition

Secret number $S$, $0 \leq S \leq m-1$ for some $m$

Trusted party generates random $S_1$, $1 \leq S_1 \leq m-1$ and gives $S_1$ and $S - S_1 \mod m$ to A and B, respectively.

A and B are trusted not to collude

Unanimous consent control by modular addition

Divide secret $S$ among $t$ users, all of whom are required to recover $S$

Trusted party generates $t-1$ independent random numbers $S_i$, $0 \leq S_i \leq m-1$, $1 \leq i \leq t-1$

$P_i$ thru $P_{t-1}$ are given $S_i$

$P_t$ given $S_t = S - \sum_{i=1}^{t-1} S_i \mod m$

(modulo $m$ can be replaced with exclusive OR)
**Threshold Schemes**

(t, n) threshold scheme \((t \leq n)\)

Secret shares \(S_i, 1 \leq i \leq n\) computed by trusted party

Any \(t\) or more users who pool their shares may easily recover \(S\). Any group of size \((t-1)\) or less may not.

Perfect threshold scheme: knowing only \((t-1)\) or fewer shares provides no advantage over knowing no shares.

**General Threshold System**

\[
\begin{align*}
S_1 & \quad m \\
S_2 & \quad m \\
\vdots & \\
S_t & \quad m \\
\end{align*}
\]

\[
\sigma = \sigma_S(m)
\]

(Combiner keeps \(S\) hidden)
SHAMIR'S THRESHOLD SCHEME

Based on polynomial interpolation

Fact: A univariate polynomial \( y = f(x) \) of degree \( t-1 \) is uniquely defined by \( t \) points \( (x_i, y_i) \) with distinct \( x_i \) (since these define \( t \) linearly independent equations in \( t \) unknowns)

Setup: Trusted party \( T \) has secret \( S > 0 \)

\( n \) users, \( t \) is threshold

(a) Choose prime \( p > \max(S, n) \)

Set \( q_0 = S \)

(b) \( T \) selects \( t-1 \) random, independent coefficients \( q_1, \ldots, q_{t-1}, 0 \leq q_j \leq p-1 \) defining the random polynomial over \( \mathbb{Z}_p \), \( f(x) = \sum_{j=0}^{t-1} q_j x^j \)

(c) \( T \) computes \( S_i = f(i) \mod p, 1 \leq i \leq n \) (or for any \( n \) distinct points \( i, 1 \leq i \leq p \))

(d) Securely transmits \( S_i \) to user \( P_i \), along with public index \( i \).

Pooling: Any group of \( t \) or more users have \( t \) distinct points \( (x, y) = (i, S_i) \), allowing computation of \( q_j, 0 \leq j \leq t-1 \), \( f(0) = q_0 = S \).
Lagrange Interpolation

(coefficients of an unknown polynomial \( f(x) \) of degree \( t-1 \), defined by points \((x_i, y_i), 1 \leq i \leq t\) are given by:

\[
f(x) = \sum_{i=1}^{t} y_i \prod_{1 \leq j \leq t, j \neq i} \frac{x - x_j}{x_i - x_j}
\]

Since \( f(0) = q_0 = S \), we can write:

\[
S = \sum_{i=1}^{t} c_i y_i , \text{ where } c_j = \prod_{1 \leq j \leq t, j \neq i} \frac{x_j}{x_j - x_i}
\]

Properties of Shamir's Scheme:

Perfect: \( t-1 \) or fewer shares, all values of \( S \) possible

Ideal: Size of one share is size of secret

Extendable for New Users: New shares can be given to new users w/o affecting existing users.

Varying Levels of Control Possible: Provide single user with multiple shares.

No Unproven Assumptions: Does not rely on difficulty of particular problems.
Attacks on Key Establishment Protocols

(i) Man-in-the-middle attack on unauthenticated Diffie Hellman

\[
\begin{align*}
A & \rightarrow g^x & \rightarrow g^{x'} \\
B & \leftarrow g^{y'} & \leftarrow g^y & \leftarrow \text{ (session key } K_A = g^{xy'}) \\
A & \text{ forms session key } & K_A = g^{xy'} \\
B & \text{ forms session key } & K_B = g^{x'y} \\
E, & \text{ can compute both: } (g^{x'y'}) & \text{ and } (g^y)^{x'} \\
\text{ Need to certify public keys!} &
\end{align*}
\]

(ii) Reflection attack

A and B share a symmetric key K and authenticate each other.

\[
\begin{align*}
A & \rightarrow r_A & (1) \\
& \rightarrow E_K(r_A, r_B) & (2) \\
& \rightarrow r_B & (3) \\
\end{align*}
\]

is this secure?

What if E initiates a new protocol?
**Reflection Attack.**

\[
\begin{align*}
A & \quad E \\
\rightarrow & \quad r_A & \leftarrow & \quad (1) \\
& \quad r_A' & \quad (1') \\
\rightarrow & \quad E_k(r_A, r_A') & \leftarrow & \quad (2) \\
& \quad E_k(r_A, r_B = r_A) & \leftarrow & \quad (2') \\
\rightarrow & \quad r_B & \quad (3)
\end{align*}
\]

A believes she has successfully authenticated B.

Use distinct keys \( k \) & \( k' \) for \( A \rightarrow B \) encryption and \( B \rightarrow A \) encryption, or avoid message symmetry by including the identifier of originating party within the encrypted portion of (2).

**Inter-Leaving Attack.**

\( S_A \) denotes signature of party A.

All parties have authentic copies of all others' public keys.

\[
\begin{align*}
A & \quad B \\
\rightarrow & \quad r_A & \quad (1) \\
& \quad r_B, s_B(r_B, r_A, A) & \quad (2) \\
\rightarrow & \quad r_A', s_A(r_A', r_B, B) & \quad (3)
\end{align*}
\]
INTERLEAVING ATTACK (CONT'D.)

E initiates one protocol with B (pretending to be A) and another with A (pretending to be B). Deceives B into believing E is A.

A

\[ \rightarrow r_A \]
\[ r_B, SB(r_B, r_A, A) \leftarrow \]
\[ r_B \leftarrow \]
\[ \rightarrow r_A', SA(r_A', r_B, B) \]

Step 2 for A

E

\[ \rightarrow r_A', SA(r_A', r_B, B) \]
\[ \leftarrow \]
\[ E \text{ uses a message in Step 3} \]

Attack possible due to the symmetry of (2) and (3)

Prevent by binding an identifier to each message indicating a message number, or requiring that the original \( r_A \) take the place of \( r_A' \) in (3)