Ad Hoc Network Security

References:

How to share a secret [Shamir 1979]

\((K, N)\) threshold scheme

- Secret \(D\) is represented by \(N\) pieces \(D_1, \ldots, D_N\)
  - \(D\) is easily computable from any \(K\) or more pieces
  - \(D\) cannot be determined with knowledge of \(K-1\) or fewer pieces

- Tradeoff between reliability and security
  - Reliability: \(D\) can be recovered even if \(N-K\) pieces are destroyed
  - Security: foe can acquire \(K-1\) pieces and still cannot uncover \(D\)

- Tradeoff between safety and convenience
  - Example—A company’s checks must be (digitally) signed by three executives
(K, N) scheme by polynomial interpolation

Given K points in 2D, \((x_1, y_1), \ldots, (x_K, y_K)\), with distinct \(x_i\)'s,

- there is one and only one polynomial \(q(x)\) of degree \(K-1\) such that \(q(x_i) = y_i\) for all \(i\).

Let the secret \(D\) be a number.

- Randomly select a \(K-1\) degree polynomial
  \[
  q(x) = a_0 + a_1x + \ldots + a_{k-1}x^{K-1}
  \]
  where \(a_0 = D\)

- Compute \(N\) values of \(q(x)\)
  \[
  D_1 = q(1), \ldots, D_i = q(i), \ldots, D_N = q(N)
  \]
Scheme by polynomial interpolation (cont.)

- Given any subset of K of the \((i, D_i)\) pairs, the coefficients of \(q(x)\) can be found by interpolation\(^1\) (or solving a set of K linear equations with K unknowns)

  - The secret D is \(q(0)\)

- Knowledge of just K-1 of the \((i, D_i)\) pairs provides no information about D
Explanation of the previous claim

- Consider the special case of a finite field $GF(p)$ where $p$ is a prime number (larger than $D$, larger than $N$)
  - The coefficients, $a_1, \ldots, a_{K-1}$, are randomly chosen from a uniform distribution over the integers in $[0, p)$
  - $D_1, \ldots, D_N$ are computed modulo $p$

- Suppose $K-1$ of the $(i, D_i)$ pairs are revealed to a foe. For each value $D'$ in $[0, p)$, the foe can construct one and only one polynomial $q'(x)$ of degree $K-1$.
  - These $p$ possible polynomials are equally likely. So there is nothing the foe can deduce about the real value of $D$. 
Useful properties

- Size of each piece $D_i$ is not larger than size of secret $D$.
- When $K$ is kept fixed, $D_i$ pieces can be dynamically added or deleted.
- Individual $D_i$ pieces can be changed without changing the secret $D$.
  - Such changes enhance security over the long term.
  - How?
    - Use a new polynomial with the same $a_0$ value ($D$).
- VIPs can be given more than one $D_i$ pieces.
Application to mobile ad hoc networks
[Kong et al. 2001]
Distributed Certificate Authority

- A mobile ad hoc network has no infrastructure support

- **Requirement**: No single node in the network knows the *system secret*, i.e., the key for signing digital certificates
  - N nodes hold *secret shares* of the system secret (signing key)

- K nodes with secret shares in a one-hop locality jointly sign new certificates
  - *Certificates* enable data confidentiality, authenticity, and integrity
Distributed Certificate Authority (cont.)

- Design Challenges for mobile ad hoc networks
  - Security breach of some nodes over a large time window is likely
  - Mobility - anywhere service
  - Network dynamics - nodes join, leave, fail
  - Scalability

- Certificates signed by K nodes in a neighborhood
  - Tolerant of up to K-1 collaborative intruders, N-K failures
Two possible intrusion models

A node's private key will not be exposed for a certain period of time (even after capture by an intruder)

- Each node updates its certificate periodically for a new public key (and private key)
  
- Intruder cannot get the correct private key in time to respond to a challenge

A node's private key may be exposed but its ID, $v_i$, is not forgeable by an intruder or any intruder's attempt to pretend to be $v_i$ can be detected.
RSA

- The system certification authority (CA)'s RSA key pair is \( \{SK, PK\} \) where SK is the signing key (system secret).

- For message \( M \), large primes, \( p \) and \( q \), and \( n=pq \), the signed message is

\[ M^{SK} \mod n \]
Notation

- Each node $v_i$
  - may hold a secret share, $P_{v_i}$, 
  - maintains an individual RSA key pair $(sk_i, pk_i)$ and
  - holds a certificate $\langle v_i, pk_i, T_{\text{sign}}, T_{\text{expire}} \rangle$

where $T_{\text{sign}}$ is signing time and $T_{\text{expire}}$ is certificate expiration time

$$T_{\text{expire}} \leq T_{\text{sign}} + T_{\text{renew}}$$
Certificate issuing

- At network initialization, nodes can obtain their certificates from a trusted central manager.

- Later on, when a node joins the network, a node wishes to renew its certificate, or recovers from a crash, it will obtain a certificate from K nodes with secret shares within a one-hop neighborhood.
  - out of bound physical proofs - human perceptions, biometrics, etc.
Certificate revocation

- **Implicit** — A node’s certificate expires if it is not renewed prior to its expiration time

- **Explicit** — If \( v_x \)'s certificate is considered compromised, a \( SK \)-signed counter-certificate with time stamp \( T^C_{\text{sign}} \) is flooded over the network
  - Neighbor nodes can safely exchange their local certificate revocation list (CRL) cache
  - With implicit certificate revocation, each node only needs to maintain counter certificates signed within the past \( T_{\text{renew}} \) time.
Protocol design using Shamir's scheme

- At system initialization, a trusted secret share dealer obtains the RSA signing key \(<d, n>\) and randomly selects a secret polynomial \(f(x)\) with degree \(K-1\)

\[
f(x) = d + f_1x + ... + f_{K-1}x^{K-1}
\]

- Each node \(v_i\) (\(i=1, 2, ..., N\)) holds a secret share

\[
P_{v_i} = f(v_i) \mod n
\]
Interpolation polynomial in the Lagrange form

Given K points in 2D, \((v_1, f(v_1)), \ldots, (v_K, f(v_K))\), with distinct \(v_i\)'s,

\[
L(x) = \sum_{j=1}^{K} f(v_j) \ell_{v_j}(x) = \sum_{j=1}^{K} P_{v_j} \ell_{v_j}(x)
\]

where

\[
\ell_{v_j}(x) = \prod_{i=1, i \neq j}^{K} \frac{(x - v_i)}{(v_j - v_i)}
\]

Since \(\ell_{v_j}(v_j) = 1\) and \(\ell_{v_j}(v_i) = 0\), we have

\[
L(v_j) = f(v_j) \text{ for } 1 \leq j \leq K
\]

Thus \(L(x) = f(x)\)
The system secret $d$

- Given $K$ secret share holders in a neighborhood

\[ d = f(0) = L(0) \quad \leftarrow \text{system secret} \]

\[ \equiv \sum_{j=1}^{K} ( P_{v_j} \ell_{v_j}(0) \mod n ) \equiv \sum_{j=1}^{K} SK_j \mod n \]

where $SK_j$ is computed by secret share holder $j$ using $P_{v_j}$ and $\{v_1, \ldots, v_K\}$

- The system secret $d$ as well as secret shares $\{SK_i\}$ should not be revealed!
A multi-signature

- Let $M$ be the new public key certificate of a node to be signed.
- Each of the $K$ secret share holders provides a partially signed certificate, for $i = 1, \ldots, K$, without revealing its private $SK_i$,
  
  $$M^{SK_i} \mod n$$

- Having collected $K$ partial certificates, the requesting node can obtain the fully signed certificate with some more work.
Partially signed certificates

Figure 3. Localized Certification Service
K-bounded coalition offsetting

The product of the K partial certificates is

\[ M^{SK_1} \cdot M^{SK_2} \cdot ... \cdot M^{SK_K} = M^{SK_1 + SK_2 + ... + SK_K} \]

The system secret is however

\[ d = (\sum_{i=1}^{K} SK_i) \mod n \]

Thus

\[ \sum_{i=1}^{K} SK_i = t \cdot n + d \]

for some integer t

The signed certificate should be \( M^d \)
K-bounded coalition offsetting (cont.)

- From modular arithmetic, $SK_i$ is a value from 0 to $n-1$
  - Thus $t$ satisfies $0 \leq t \leq K$
  - Try each possible value of $t$. Decrypt using the system’s public key and find the correct $t$ value with help of the original message $M$

- Complexity is the sum of $O(1)$ exponentiation, $O(K)$ modular multiplications, and $O(K)$ RSA public operations
  - RSA public key operation is relatively fast
Becoming a secret share holder

- Each certificate-holding node $v_x$ can also obtain a secret share $P_{v_x}$, which it uses to derive $SK_x$ to become a secret share holder.

- $K$ secret share holders can compute the secret share of $v_x$ by Lagrange interpolation:

$$P_{v_x} = f(v_x) = \sum_{j=1}^{K} P_{v_j} \ell_{v_j}(v_x) \equiv \sum_{j=1}^{K} SS_{x,j} \pmod{n}$$

- **Problem**—Since $\{v_i\}$ of the $K$ secret share holders are publicly known, $v_x$ can derive $P_{v_j}$ from $SS_{x,j}$ and $\ell_{v_i}(v_x)$. 
Becoming a secret share holder (2)

- Shuffling scheme—A random nonce is exchanged between any two members of the K share holders (the one with lower id creates the nonce)
  - The node with larger id treats it as a positive number
  - The other node treats it as a negative number.
  - Each share holder, \( v_i \), has K-1 such nonces. It sums the nonces and \( SS_{x,j} \). The sum is sent to \( v_x \)

- It is easy to show that \( v_x \) gets the same value
Becoming a secret share holder (3)

- Each nonce is encrypted with the individual public key of the intended receiver

- The requester forwards encrypted nonces

- Still K-out-of-N secure if there are at least two uncompromised nodes in the K nodes
Conclusions

- Application of Shamir’s idea to mobile ad hoc networks
- Most results borrowed from crypto literature
- Authors claim that prior work assumes a fixed number of secret share holders, not applicable to large networks with dynamic node membership
  - In this paper, existing secret shares and the current signing key are not affected by membership changes

(We have omitted many details in the paper.)
The End