CRYPTOGRAPHY LECTURE, 2022

Computer Science and Algorithmics, PWr

Mirosław Kutyłowski

Secret Sharing

Problem: data may fall into hands of an adversary, keeping everything in one place is risky

Application Example: Certification Authority, keeping the main master key

Goals: security, safety, resilience to faults

Secret Sharing Idea

data D $\longrightarrow$ n shares of D

Properties

i. each share is for a different person/location/device...

ii. the shares make it possible to recover the original D

iii. which subset of shares is necessary for recovery should be flexible

Access structure

a family I of subsets such that:

- if a set of shares $R \in I$, then reconstruction of D is possible
- if $R \notin I$, then no information on D should be possible to recover

Remark:

I should be an *ideal* in the language of set theory:

if $R \in I$ and $R \subset R'$, then $R' \in I$

Example access structures

- n-of-n: all shares
- threshold k: at least k different shares
- something more complicated: e.g. at least 2 shares from subset A and at least 2 shares from subset B

n of n secret sharing

Data to be shared: d

Shares:

- for $i = 1, \dots, n-1$: share s_i chosen at random
- $s_n := d \oplus s_1 \oplus \cdots \oplus s_{n-1}$

Recovery: $d := s_1 \oplus \cdots \oplus s_n$

Secrecy: question is "what is conditional probability for D given $n-1$ shares

Fact: given $n-1$ shares and a value $\Delta,$ there is exactly 1 share for which recovery results in Δ

Threshold scheme k-of-n : Shamir's Secret Sharing

Input: d from a finite field F

Constructing shares:

i. choose a random polynomial P of degree $k-1$ such that $P(0) = d$

ii. the ith share is $P(i)$

Reconstruction of d:

given the value of P in k different points, apply Lagrangian interpolation for $P(0)$

$$P(z) = \lambda_1(z) \cdot P(a_1) + \lambda_2(z) \cdot P(a_2) + \cdots + \lambda_k(z) \cdot P(a_k)$$

where Lagrangian coefficient $\lambda_i(z)$ equals

$$\prod_{j \neq i} \frac{z - a_j}{a_i - a_j}$$

for reconstruction of $P(0)$ we put $z = 0$ and therefore:

$$P(z) = \lambda_1 \cdot P(a_1) + \lambda_2 \cdot P(a_2) + \cdots + \lambda_k \cdot P(a_k)$$

where

$$\lambda_i = \prod_{j \neq i} \frac{a_j}{a_j - a_i}$$

Shamir's Secret Sharing

correctness:

the reconstructed polynomial P' has the same values at $a_1, \ldots, a_k$ as P

$P - P'$ has degree at most $k-1$ and has k zero points $\Rightarrow P' - P = 0$

security:

given $k-1$ shares: each value at 0 corresponds to exactly one polynomial

Threshold PKE

goal: enable decryption of a ciphertext if at least k out of n secret keys are available

applications: some secret keys lost...

Threshold PKE based on ElGamal

Key generation:

- master secret key x, public key $X = g^x$
- shares of x: use Shamir Secret Sharing, the ith share is sk_i

Encryption

m encrypted as $(c_2, c_1) := (X^t \cdot m, g^t)$ for random t

Partial Decryption

calculate $m_i := c_1^{\mathrm{sk}_i}$

Threshold PKE based on ElGamal

Key generation:

- master secret key x, public key $X = g^x$
- shares of x: use Shamir Secret Sharing, the ith share is sk_i

Encryption

m encrypted as $(c_2, c_1) := (X^t \cdot m, g^t)$ for random t

Partial Decryption

calculate $m_i := c_1^{\mathrm{sk}_i}$

Message recovery (via Lagrangian interpolation in the exponent)

it is enough to calculate X^t

take: $\prod_{i=1}^{k} m_{j_i}^{\lambda_{j_i}(0)}$ for Lagrangian multipliers λ_i

note: $\prod_{i=1}^{k} m_{j_i}^{\lambda_{j_i}} = \prod_{i=1}^{k} (g^t)^{\lambda_{j_i}\mathrm{sk}_i} = (g^t)^{\sum \lambda_{j_i}\mathrm{sk}_i} = (g^t)^x = X^t$

Access structures based on monotonic circuits

Boolean circuits with gates AND, OR, THRESHOLD

- a circuit with connections defining acyclic directed graph with a single sink
- leaves – shares of the users
- put one on a leaf iff its share is available
- set of shares should eneble recovery iff output of the circuit is 1

Access structures based on monotonic circuits

some examples:

Access structures based on monotonic circuits -inductive construction

- top-down starting from the root
- each node labeled by a share
- AND with k inputs: split share with k-of-k scheme
- OR - duplicate shares
- THRESHOLD$_k$ on n inputs: split shares with k-of-n scheme