Secret sharing schemes

Cryptology (1)

Martin Stanek

2025

KI FMFI UK Bratislava

Secret sharing schemes – introduction

- secret sharing schemes
  - distribute a secret (a key) among some group of participants (users, servers)
  - rules – what group can reconstruct the secret
  - share – secret piece of information owned by an individual participant
- a scheme consists of two algorithms/protocols:
  - producing and distributing the shares (usually a trusted dealer is used)
  - reconstructing the shared secret
- motivation
  - Can you trust a single authority (admin or server)?
  - basis for other constructions – threshold cryptography, distributing computation among group of trusted servers, multi-party secure computation, voting, …

- n participants $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$
- shared secret s
- shares: $P_i \leftarrow s_i$
- access structure $\mathcal{A} \subseteq 2^{\mathcal{P}}$ (power set): $A \subseteq \mathcal{P}$ can reconstruct $s \Leftrightarrow A \in \mathcal{A}$
  - usually a monotone access structure: $\forall A, B \subseteq \mathcal{P} : \; A \subseteq B \;\&\; A \in \mathcal{A} \implies B \in \mathcal{A}$
  - (t, n) threshold access structure, for $1 \leq t \leq n$: $\{A \mid A \subseteq \mathcal{P} \;\&\; |A| \geq t\}$

Simple examples

- $(1, n)$ threshold
  - distribute the secret as individual shares: $s_i = s$
- (n, n) threshold – 1st attempt
  - let $s \in \{0, 1\}^l$
  - divide s into n shares $s_1, \ldots, s_n$ of length $\approx l/n$ bits
  - reconstruction: $s = s_1 \parallel \ldots \parallel s_n$
  - $n - 1$ participants reconstruct a large part of s, approx. $l(n-1)/n$ bits

- $(1, n)$ threshold
  - distribute the secret as individual shares: $s_i = s$
- (n, n) threshold – 1st attempt
  - let $s \in \{0,1\}^l$
  - divide s into n shares $s_1, \ldots, s_n$ of length $\approx l/n$ bits
  - reconstruction: $s = s_1 \parallel \ldots \parallel s_n$
  - $n - 1$ participants reconstruct a large part of s, approx. $l(n-1)/n$ bits
- (n, n) threshold
  - let $s \in \{0,1\}^l$
  - let $s_i \in_R \{0,1\}^l$ for $i = 1, \ldots, n-1$, and $s_n = s \oplus s_1 \oplus \ldots \oplus s_{n-1}$
  - reconstruction: $s = s_1 \oplus \ldots \oplus s_n$
  - security: any $n - 1$ (or less) participants learn nothing about s
  - *perfect* scheme

Shamir's secret sharing scheme

- idea: t points uniquely determine some polynomial of degree $t-1$
- finite field $\mathbb{Z}_p$, for a prime $p > n$
- shared secret $s \in \mathbb{Z}_p$
  - let us assume $s \in_R \mathbb{Z}_p$

Shares

- choose a random polynomial $f(x) = s + a_1 x + \ldots + a_{t-1} x^{t-1}$, where $a_i \in_R \mathbb{Z}_p$ for $i = 1, \ldots, t-1$
- share for P_i: (i, s_i), where $s_i = f(i)$
- notice that $f(0) = s$

- idea: t points uniquely determine some polynomial of degree $t-1$
- finite field $\mathbb{Z}_p$, for a prime $p > n$
- shared secret $s \in \mathbb{Z}_p$
  - let us assume $s \in_R \mathbb{Z}_p$

Shares

- choose a random polynomial $f(x) = s + a_1 x + \ldots + a_{t-1} x^{t-1}$, where $a_i \in_R \mathbb{Z}_p$ for $i = 1, \ldots, t-1$
- share for P_i: (i, s_i), where $s_i = f(i)$
- notice that $f(0) = s$

Reconstruction

- t participants $P_1, \ldots, P_t$ (WLOG)
- Lagrange interpolation in $\mathbb{Z}_p$ using (i, s_i) for $i = 1, \ldots, t$:

$$f(x) = \sum_{i=1}^{t} \underbrace{f(i)}_{s_i} \cdot \prod_{\substack{1 \le j \le t \\ j \ne i}} \frac{x - j}{i - j}$$

- compute $s = f(0)$

Shamir's secret sharing scheme – security

- consider group of $t-1$ participants (WLOG $P_1, \ldots, P_{t-1}$)
- the shared secret can be anything:
  - combine the shares and point $(0, s')$ for an arbitrary $s' \in \mathbb{Z}_p$
  - t points $\Rightarrow$ unique polynomial f'
  - f' is consistent with shares of $P_1, \ldots, P_{t-1}$
- $P_1, \ldots, P_{t-1}$ are in the same position as someone without any share
  - probability of finding s is $1/p$ (guessing)
- perfect secret sharing scheme

Linear equations perspective

- unknown polynomial f (its coefficients)
- a share (i, s_i) forms a linear equation: $s_i = a_0 + a_1 i + \ldots + a_{t-1} i^{t-1}$
- t cooperating participants – the system of t equations with t variables
  - square Vandermonde matrix with distinct elements (i.e., a non-zero determinant)
  - the system has a unique solution
- $t - 1$ cooperating participants – the system of $t - 1$ equations with t variables
  - add an additional equation: $s' = a_0$
  - square Vandermonde matrix with distinct elements (because any $i \neq 0$)
  - the system has a unique solution for any s' … perfect scheme

Remarks

- reconstruction is just a linear combination of shares (for $S \subseteq \{1, ..., n\}$, $|S| = t$):

$$f(0) = \sum_{i \in S} s_i \cdot \lambda_i, \quad \text{where } \lambda_i = \prod_{j \in S \setminus \{i\}} \frac{-j}{i-j}$$

- any points $(x_i, f(x_i))$ for distinct non-zero $x_1, ..., x_n$ can be used as shares
- homomorphic property with respect to addition:
  - two (t, n) threshold schemes defined by polynomials f and g
  - adding shares: $(i, f(i)), (i, g(i)) \mapsto (i, f(i) + g(i))$
  - polynomial (the shared secret is the addition of shared secrets $a_0 + a_0'$):

$$f(x) + g(x) = \sum_{i=0}^{t-1} a_i x^i + \sum_{i=0}^{t-1} a_i' x^i = \sum_{i=0}^{t-1} (a_i + a_i') x^i$$

Remarks (2)

- efficiency
  - polynomial time
  - long s can be divided into shorter pieces and shared by independent schemes (or encrypt s and share the encryption key)
- trusted dealer – generates the polynomial and distributes the shares
- one-time scheme?
  - secret revealed after reconstruction vs. black-box reconstruction
- cheating in reconstruction:
  - for example – $P_1, \ldots, P_t$ try to reconstruct s
  - P_1 cheats and reveals an incorrect share $(1, s_1')$
  - the participants compute: $s' = s + s_1'\lambda_1 - s_1\lambda_1$
    … and P_1 can easily compute s from s'

Information rate

- the size of share(s) vs. the size of the shared secret
- notation
  - S – set of secrets
  - $K(P_i)$ – set of all possible shares for P_i
  - random variables
- information rate for P_i: $\rho_i = H(S)/H(K(P_i))$
- information rate of the scheme: $\rho = \min_i \rho_i$
- uniform probability case: $\rho = \min_i \lg|S| \,/\, \lg|K(P_i)|$

Information rate (2)

- information rate for Shamir's scheme: $\rho = 1$
- perfect secret sharing scheme $\Rightarrow$ $\rho \leq 1$
  - let us assume that $\rho > 1 \Rightarrow \forall i : \rho_i > 1$
  - for all i: $\lg|S| \,/\, \lg|K(P_i)| > 1 \quad \Rightarrow \quad |S| > |K(P_i)|$
  - there exists $A \subseteq \mathcal{P}$: $P_i \notin A$, $A \notin \mathcal{A}$, and $A \cup \{P_i\} \in \mathcal{A}$
  - take all shares from participants in A and all candidate shares from $K(P_i)$
  - compute all possible values of the shared secret … less than $|S|$
  - the scheme cannot be perfect (we can exclude some "impossible" secrets)
- a perfect secret sharing scheme with $\rho = 1$ is called ideal

Verifiable secret sharing

- secret sharing that allows participants to verify the correctness of their shares

- Feldman's scheme ≈ Shamir's scheme + commitments of coefficients
  - (t, n) threshold access structure

- $f(x) = s + a_1 x + \ldots + a_{t-1} x^{t-1}$ over $\mathbb{Z}_p$
  - let g be a generator of a subgroup $G \subseteq \left(\mathbb{Z}_p^*, \cdot\right)$ of prime order q $\quad (q \mid p-1)$
  - the dealer creates (public) commitments $c_i = g^{a_i}$, for $i = 0, \ldots, t-1$
  - P_i can verify the share (i, s_i):

$$c_0 \cdot c_1^{i} \cdot c_2^{i^2} \cdot \ldots \cdot c_{t-1}^{i^{t-1}} = \prod_{j=0}^{t-1} g^{a_j \cdot i^j} = g^{f(i)} = g^{s_i}$$

- a problem: secrecy of s depends on dlog problem (not perfect anymore)
  - improved schemes exist

Applications

Threshold cryptography

- threshold cryptography: sharing a secret key, such that
  1. any group of size t or more can perform a cryptographic operation, and
  2. any group of size $t - 1$ or less cannot perform the operation
- adversary can compromise up to $t - 1$ parties
- cryptographic operation: signing, decrypting, etc.
  - signing:
    1st property means *robustness* (DoS prevention)
    2nd property means *unforgeability*
- key distribution:
  - trusted dealer or distributed key generation (DKG)

Schnorr signature with threshold signing

Schnorr signature scheme:

- group G of prime order p, generator g
- private/secret key $\mathsf{sk} = x \in_R \mathbb{Z}_p$
- public key $\mathsf{pk} = y = g^x$
- $\mathsf{Sig}_{\mathsf{sk}}(m) = (R, s) \in G \times \mathbb{Z}_p$
  - $R = g^k$ for $k \in_R \mathbb{Z}_p$
  - $c = H(R \parallel y \parallel m)$
  - $s = k + xc$
- $\mathsf{Vrf}_{\mathsf{pk}}(m, (R, s))$: $g^s \stackrel{?}{=} R \cdot y^c$, where
  - $c = H(R \parallel y \parallel m)$

- threshold Schnorr signatures
  - redundancy (if someone is unavailable)
  - not a single person should be authorized to sign
- some desired properties:
  - result is a regular signature
  - private key is **not** revealed in the process
- signature aggregator (SA)
  - some participant or independent subject
  - can prevent signature creation but does not learn anything about the private key
  - simplifies the presentation

Threshold Schnorr signatures – simple approach

- Stinson, Strobl (2001)
- private key x is shared in a secret sharing scheme (trusted dealer):
  - $f(z) = x + \sum_{j=1}^{t-1} a_j z^j$, public key $y = g^x$
  - P_i gets his share $x_i = f(i)$, for $i = 1, \ldots, n$, together with the public key y
- $P_1, \ldots, P_t$ want to sign m:
  1. $P_i \to$ SA: $R_i = g^{k_i}$, where $k_i \in_R \mathbb{Z}_p$
  2. SA $\to P_i$: R, m, where $R = \prod_{j=1}^{t} R_j$
  3. $P_i \to$ SA: $s_i = k_i + x_i \cdot c \cdot \lambda_i$, where $c = H(R \parallel y \parallel m)$, and λ_i is the Lagrange coefficient
  4. SA computes $s = \sum_{i=1}^{t} s_i$, and outputs the signature (R, s)
- correctness: $g^s = g^{\sum_i s_i} = \prod_i g^{s_i} = \prod_i R_i \cdot g^{x_i c \lambda_i} = R \cdot \left(\prod_i g^{x_i \lambda_i}\right)^c = R \cdot (g^x)^c = R \cdot y^c$

- the scheme is secure in sequential setting
- concurrent (parallel) insecurity / parallel composition
  - $t-1$ malicious parties (including the SA)
  - single honest participant (let it be P_1)
  - attacking "group" can participate in multiple signing sessions simultaneously
- P_1 will sign $\left(R^{(1)}, m^{(1)}\right), \left(R^{(2)}, m^{(2)}\right), \ldots, \left(R^{(l)}, m^{(l)}\right)$, i.e., P_1 produces l values $s_1^{(j)} = k_1^{(j)} + x_1 \cdot c^{(j)} \cdot \lambda_1$, where $c^{(j)} = H\left(R^{(j)} \parallel y \parallel m^{(j)}\right)$
- assume, that we can find $\left(R^{(j)}, m^{(j)}\right)_{j=1}^{l}$ and (R^*, m^*), such that $\sum_{j=1}^{l} c^{(j)} = c^* = H(R^* \parallel y \parallel m^*)$
- compute $s_1^* = \sum_{j=1}^{l} s_1^{(j)} = \sum_{j=1}^{l} k_1^{(j)} + x_1 \cdot \lambda_1 \cdot \sum_{j=1}^{l} c^{(j)} = k^* + x_1 \cdot \lambda_1 \cdot c^*$
  - the attacking group can calculate P_1's contribution, and finish signing of m^*

Remarks

- ROS problem: Random inhomogeneities in an Overdetermined Solvable system
  - allows to find required $\left(R^{(j)}, m^{(j)}\right)_{j=1}^{l}$ and (R^*, m^*)
  - Wagner (2002): subexponential time
  - Benhamouda et al. (2020): polynomial time for $l > \lg p$
- there are schemes that address this problem
  - Sparkle+, FROST/2/3, etc.

ElGamal encryption scheme:

- group G of prime order p, generator g
- private/secret key $\mathsf{sk} = x \in_R \mathbb{Z}_p$
- public key $\mathsf{pk} = y = g^x$
- $\mathrm{Enc}_{\mathsf{pk}}(m)$: $(r, s) = \left(g^k, m \oplus H\left(y^k\right)\right)$
  - $k \in_R \mathbb{Z}_p$
  - message space $\{0,1\}^l$
  - $H : G \to \{0,1\}^l$
- $\mathrm{Dec}_{\mathsf{sk}}(r, s)$: $m = s \oplus H(r^x)$

- public key y is known
- x is distributed in a threshold scheme: $f(z) = x + \sum_{i=1}^{t-1} a_i z^i$
- P_i gets a share $x_i = f(i)$
- a client C wants to decrypt a ciphertext (r, s):
  - assume $P_1, \ldots, P_t$ will assist
  - $P_i \to C$: $d_i = r^{x_i}$
  - C computes:

$$H\left(\prod_{i=1}^{t} d_i^{\lambda_i}\right) \oplus s = H\left(\prod_{i=1}^{t} r^{x_i \lambda_i}\right) \oplus s$$

$$= H\left(r^{\sum_{i=1}^{t} x_i \lambda_i}\right) \oplus s = H(r^x) \oplus s = m$$

Remarks

- non-interactive, $P_1, \ldots, P_t$ do not need to communicate with each other
- we can publish "per party public keys": $y_i = g^{x_i}$
  - … and verify the validity of partial decryptions
  - otherwise incorrect decryption caused by a malicious party
  - P_i *proves* the equality of discrete logarithms: $\mathrm{dlog}_r\, d_i = \mathrm{dlog}_g\, y_i$, *without disclosing* the discrete log itself (x_i), and preferably do it *non-interactively*
- it is OK for static security
  - adversary corrupts a static set of at most $t-1$ parties ≈ adversary knows the secret keys from the beginning
- adaptive security: adversary can adaptively corrupt up to $t-1$ parties
  - any moment in the computation a party can be corrupted
  - more involved schemes were proposed for this setting

Exercises

1. *Discuss a modification of Shamir's scheme, where the polynomial $f(x)$ must be of degree $t-1$, i.e., $a_{t-1} \in_R \mathbb{Z}_p \setminus \{0\}$. Is the scheme perfect? Explain.*
2. *Design a perfect secret sharing scheme for participants $\{A, B, C, D\}$ with the following access structure:*
   a) *"at least two participants, but not A together with B"*
   b) *"at least two participants, but not A together with B or C"*
3. *Try to simplify the threshold ElGamal encryption scheme when we are interested in (n, n)-threshold scheme only.*