Lecture 20: Secret Sharing

CS 539 / ECE 526

Sourav Das
Secret Sharing

• Activity in groups of 3

• (2, 1) secret sharing for a bit:
  – A dealer shares a secret bit \( b \)
  – Each party gets a share (2 parties in total)

1. Parties jointly can recover \( b \)
2. Share of a single party reveal no information about \( b \)

• Hint: One party (party 1) will get a random bit \( b_1 \)
Secret Sharing: Protocol

- (2, 1) secret sharing for a bit:
  - A dealer shares a secret bit b
  - Each party gets a share (2 parties in total)

\[
b_1 = b \oplus r \\
b_2 = r
\]
Secret Sharing: Reconstruction

• (2, 1) secret sharing for a bit:

1. Parties jointly can recover b
2. Share of a single party reveal no information about b

\[ b_1 = b \oplus r \]
\[ b_2 = r \]

\[ b_1 \oplus b_2 = b \]
(2, 1) secret sharing for a bit:

1. Parties jointly can recover $b$

2. Share of a single party reveal no information about $b$

\[ \Pr[b_1=0] = \Pr[r=b] = \frac{1}{2} \]
\[ \Pr[b_2=0] = \Pr[r=0] = \frac{1}{2} \]
Secret Sharing

• (n, t) secret sharing:
  – A dealer shares a secret $s$
  – Each party gets a share (n parties in total)
  – Any t shares reconstruct $s$
  – Any t-1 shares reveal no information about $s$

• Tolerate t-1 curious parties and n-t crash faults

Hint 1: Use polynomials of degree t-1
Hint 2: Any t-1 evaluation points does not reveal the entire polynomial
Shamir’s Secret Sharing [Shamir 1979]

• $y = f(x) = s + c_1x + c_2x^2 + c_2x^2 + \ldots + c_{t-1}x^{t-1}$
  
  – $s = f(0)$ is the secret. Other coefficients are random

• Party i’s share is $s_i = f(a_i)$
  
  – $a_1, a_2, a_3, \ldots, a_n$ are distinct public values

• t points fix a degree t-1 polynomial; can reconstruct using Lagrange interpolation
Lagrange Interpolation Formula

Let \((x_1, y_1), \ldots, (x_n, y_n)\) be \(n\) points with different \(x\) coordinates, then

\[
P(x) = \sum_{i=1}^{n} \left( y_i \prod_{j \neq i} \frac{(x - x_j)}{(x_i - x_j)} \right)
\]

is the only polynomial of degree \(\leq n - 1\) that goes through all of them.

\[
X = \{x_1, x_2, \ldots, x_n\}
\]

\[
L_{i,X}(x) = \prod_{j \neq i} \frac{(x - x_j)}{(x_i - x_j)}
\]

1. Degree of \(L_{i,X}(x)\)?
2. Value of \(L_{i,X}(x_i)\)
3. Value of \(L_{i,X}(x_j)\) for \(j \neq i\)
Shamir’s Secret Sharing [Shamir 1979]

• $y = f(x) = s + c_1x + c_2x^2 + c_2x^2 + \ldots + c_{t-1}x^{t-1}$

• Will work with polynomials in a finite field
  – All numbers, and + and * operations are mod $p$
    where $p$ is a pre-chosen prime
  – Secret $s \in \mathbb{Z}_p = \{0, 1, 2, \ldots, p-1\}$
Error Correction Codes

- Encode a message $m$ of $k$ symbols into $n > k$ symbols
- Can decode $m$ despite some missing symbols (erasure) or corrupt symbols (error correction)

- Contrast with secret sharing?
- Some simple codes?
Reed-Solomon Code

• $n = k + d$, i.e., $d$ redundancy

• Can tolerate $d$ erasures or $d/2$ errors

• Encode:
  
  – Chunk msg $m$ as $[m_1, m_2, \ldots, m_k]$ s.t. $m_i \in \mathbb{Z}_p$
  
  – Find a degree $k-1$ polynomial $f(x)$ s.t. $f(a_i) = m_i \ \forall \ i \leq k$
  
  – Compute $f(a_i)$ for $\forall \ k+1 \leq i \leq n$
  
  – Encoded msg = $[ f(a_1), f(a_2), \ldots, f(a_n) ]$
Reed-Solomon Code

• Decode with erasure: Lagrange interpolation!

• Decode with error correction
  – Given $b_1, b_2, \ldots, b_n$ where $b_i = f(a_i)$ except $d/2$ points
  – Let $e(x)$ be an “error locating polynomial”, i.e.,
    $$e(a_i) = 0 \iff b_i \neq f(a_i)$$
    • $e(x)$ has $\leq d/2$ distinct roots, hence degree $\leq d/2$
    • We have $e(a_i) f(a_i) = e(a_i) b_i$
  – Can solve the above system equations!
Reed-Solomon Code

• e(x) has $\leq d/2$ distinct roots, hence degree $\leq d/2$

• Solve system equations $e(a_i) f(a_i) = e(a_i) b_i$

  – How many unknowns?
  • All coefficients of e() and f(), so $d/2 + k$

  – How many equations?
  • n equations but $d/2$ of them are same ($0 = 0$)
  • At least $n - d/2 = k + d - d/2 = k + d/2$