Sarthak Singhal
(20171091)

PQR-2

7th April 2020

Question

To store k blocks of data/information (say each block is of b bits) in a fault-tolerant way, you may encode the k blocks into n blocks (using some error-correction code) such that if any e of the n blocks are corrupted, it is still possible to retrieve the original k blocks of information. Specifically (for large enough b), coding theory suggests that this is possible if and only if $n \geq (k + 2e)$. However, show that using digital signatures, it is possible to achieve the above fault-tolerant storage even when $(k + e) \leq n < (k + 2e)$, assuming a PPTM-adversary and a negligible probability of error is permitted.

Answer

Shamir secret sharing

Shamir's Secret Sharing is a form of secret sharing, where a secret is divided into parts, giving each participant its own unique part. To reconstruct the original secret, a minimum number of parts is required. In the threshold scheme this number is less than the total number of parts. Otherwise all participants are needed to reconstruct the original secret.

Fault tolerant storage

- This can be solved using a variant of shamir secret sharing.
- We take our data and divide it into k blocks.

- Then we construct a polynomial of degree (k-1) using these k blocks as coefficients over a finite field F where F = Z_p^*.
- Given that we have encoded these k blocks into n blocks out of which e are corrupted, we have n-e non corrupted blocks.
- We know that we can reconstruct a polynomial of degree n if we have at least n+1 points lying on it.
- Thus using the above principle, we can reconstruct the polynomial if and only if n-e>=k (as degree of our constructed polynomial is k-1)
- This relation gives us that $n \geq k + e$

Thus using digital signatures, we can achieve fault tolerant storage with

$$(k + e) < n \leq (k + 2e)$$

Algorithm

- Let the given data be D
- Let D = $d_0 \,||\, d_1 \,||\, \ldots \,||\, d_{k-1}$
- Let our polynomial be $P(x) = d_0 + d_1x + d_2x^2 + \ldots + d_{k-1}x^{k-1}$
- Let the field be Z_p^* over which we construct P(x)
- Now let us take n points on this polynomial as (x1,y1), (x2,y2) ... (xn,yn)
- To identify which of these points are corrupted, we encode k blocks of D into n blocks where each block is of the form

  $$Bx = \{i, P(i) mod p, sign(P(i) mod p)\}$$

- Since i and sign(M) are small in comparison to M, size of our block is roughly equal to the size of original block (that is b bits)
- Next we identify the corrupted blocks, by verifying M against its signature
- Since we have e corrupted points, and n>=k+e, we have at least k non corrupted points using which we can reconstruct the polynomial
- We reconstruct the polynomial using Vandermonde matrix

Vandermonde matrix

- Given at least k points of a (k-1) degree polynomial, we can find the polynomial which will pass through all the given k points
- Let the k points be (x1,y1), (x2,y2) ... (xk,yk)
- We construct a kXk matrix M s.t. M[i][j] = x_i^j
- Given this M (known as vandermonde matrix), we can find coefficients of P(x)
- Let A be a column vector s.t. A[i] = a_i where a_i is the coefficient of x^i in P(x)
- Let B be a column vector s.t. B[i] = y_i where y_i = P(x_i)
- Using these M, A, B we have $A = M^{-1} Y$