V. A. Lalitha
Rameswari
20171025

Evaluation - II

Fault Tolerant Data storage

Given, there are k blocks of data → to be encoded into n blocks ($n > k$)
The condition for fault tolerance is, even if any e of those k blocks are corrupted/modified/erased, we should still be able to retrieve our original k blocks.

It is given in the question, that coding theory ~~ ~~ suggests that this is possible only if $n \geq (k+2e)$ [Something similar to Reed Solomon Encoding].

To prove that using digital signature, we can achieve fault tolerant storage when $(k+e) \leq n$

To prove this, we come up with a scheme based on SHAMIR SECRET SHARING.

Proposed idea: Construct a $(k-1)^{th}$ degree polynomial from the values of k blocks we have as the coefficients.

Let f be a polynomial over a finite field, such that

$$f(x) = c_1 x^{k-1} + c_2 x^{k-2} + \cdots + c_k$$

constant (SECRET)

where c_i is the value of the i^{th} block (we are assuming value to be integer for each block. Even if binary, convert to integer)

Given the polynomial is constructed,
we generate n points
(Choose some x, compute $f(x)$ ⇒ Repeat n times)

This will constitute our n blocks.

But for authentication, we will sign each block also.

<u>Signing a block</u>

Similar to Eval 1, but here, we consider (for each point x_i, y_i) x_i as a message, hash it using the collision resistant hash function. Similarly, hash y_i also, and now do a XOR of these two hashes.

Rest all the steps are similar in terms of signing and verifying

So here, we are actually sending three things for each block (total n) :

x
y
Signature → signature consists of 2 parts (see implementation on ins p)

<u>Verification</u>

For each block, verifier can verify whether the contents of the block are same or have been corrupted after signing.

Why does this work?

Since we are told that errors are e blocks.

Therefore, $(n-e)$ blocks are uncorrupted.

For the verifier to reconstruct the contents in the corrupted e blocks, he will have to get the $(k-1)^{th}$ degree polynomial.

So, he should know atleast k points.

But he knows only $(n-e)$ points which are uncorrupted.

$$\Rightarrow (n-e) \geq k$$

$$\Rightarrow \boxed{(k+e) \leq n}$$

If this condition is satisfied, the verifier can always reconstruct the k blocks through the redundancy as shown above.