Evaluation 2

20171076

Q. Design a fault-tolerant storage using digital signatures that requires the message to be split into $k+e \leq n \leq k+2e$ data blocks, where k is the minimum no. of data blocks needed and e is the allowed no. of block corruptions.

——— x ———

A. In order to ensure fault-tolerant storage, we can define a k-1 degree polynomial defined over a sufficiently large field.

Consider a finite field F_p (integers modulo p)

s.t. p is a prime

$p > 2^b$

$|F_p| > n$

Encoding the message

Split the message into k blocks, $m = m_0 m_1 m_2 \dots m_{k-1}$

s.t. $m_i \in F_p \ \forall i$

We know that a polynomial of degree k-1 can uniquely be defined by 'k' evaluations. Consider polynomial in field given:

$$M(x) = m_0 + m_1 x + m_2 x^2 + \dots + m_{k-1} x^{k-1}$$

Now we defined n such that $k+e \leq n \leq k+2e$

Evaluating $M(x)$ at n points $a_{0 \dots n-1}$ $(M(a_0), M(a_1), \dots, M(a_{n-1}))$, we can then digitally sign each of the results (say c_i).

We now have a scheme that encodes a message in $k+e \leq n \leq k+2e$ blocks.

$$C(m) = \underbrace{\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{n-1} \end{bmatrix}}_{\text{Digitally sign}} = \begin{bmatrix} 1 & a_0 & a_0^2 & \dots & a_0^{k-1} \\ 1 & a_1 & a_1^2 & \dots & a_1^{k-1} \\ 1 & a_2 & a_2^2 & \dots & a_2^{k-1} \\ \vdots & & & & \\ 1 & a_{n-1} & a_{n-1}^2 & \dots & a_{n-1}^{k-1} \end{bmatrix} \begin{bmatrix} m_0 \\ m_1 \\ m_2 \\ \vdots \\ m_k \end{bmatrix}$$

Verifying Tamper Detection with $n \leq k+2e$

A PPTM adversary can tamper with message blocks, with negligible error in detection permitted. Let $\mathbb{T}$ be the set of tampered blocks, with $|\mathbb{T}| \leq e$ (by problem definition). As all blocks are digitally signed, we can use verify to check if a block has been tampered or not. For all $t \in T$ verify(t) will return true (not tampered) with negl() probability (as we have a PPTM adversary).

Recovering from k blocks

As $|T| \leq e$, $n-e \geq k$ untampered blocks remain. Each block c_i is an evaluation of the unique message polynomial (of degree $k-1$), allowing us to recover the message by using polynomial interpolation to recover the coefficients.

We use Gaussian Elimination for this

The goal is to invert the original transform:

$$m = \begin{bmatrix} 1 & a_0 & a_0^2 & \cdots & a_0^{k-1} \\ & & \vdots & & \\ 1 & a_{n-1} & a_{n-1}^2 & \cdots & a_{n-1}^{k-1} \end{bmatrix}^{-1} \begin{bmatrix} c_0 \\ \vdots \\ c_{n-1} \end{bmatrix}$$

As Gaussian elimination is done on finite field F_p, with operations done modulo p, we consider division to be multiplication with the modular inverse.

Consider a rough algorithm for such. $A = [c \| b]$ for $cx = b$.

```
for c = 0 to k
    r = row with A[r][c] ≠ 0
    swap rows r and c
```

```
r = c
for i = 0 to k
    if i ≠ r
        w = - A[i][c] / A[r][c]
        for j = 0 to k+1
            A[i][j] += w . A[r][j]
for i = 0 to k
    m[i] = A[r][k] / A[r][i]
return m
```