Q Design a fault tolerant storage system that requires $(k+e) \leq n < (k+2e)$ blocks where k is the no. of data blocks and e is the allowed no. of block corruptions.

A. Let p be a prime, s.t. $|\mathbb{F}_p| > n$ where $\mathbb{F}_p$ is the finite field of integers modulo p and $p > 2^b$.

Encoding

For a message $m = m_0 m_1 \ldots m_{k-1}$ of length k-blocks, where each $m_i \in \mathbb{F}_p$, consider the polynomial in this field given by –

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

We know that a polynomial of degree $k-1$ is uniquely defined by k evaluations. Consider $k+e \leq n < k+2e$ lets evaluate the above polynomial at n points.

$$C(m) = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & a_0 & a_0^2 & \cdots & a_0^{k-1} \\ 1 & a_1 & a_1^2 & \cdots & a_1^{k-1} \\ & & \vdots & & \\ 1 & a_{n-1} & a_{n-1}^2 & \cdots & a_{n-1}^{k-1} \end{bmatrix} \begin{bmatrix} m_0 \\ m_1 \\ \vdots \\ m_{k-1} \end{bmatrix}$$

Now using the digital signature developed in the previous evaluation we can sign each of $c_0, c_1, \ldots, c_n$. Note that the size of this signature is independent of n, k and e and only depends on the group chosen for the signing algorithm.

We can thus appropriately choose the group size so that the total bits do not exceed or equal the total bits when using $k+2e$ blocks.

Thus we have a scheme that requires $k+e \le n < k+2e$ blocks each of size $b' = b + O(1)$ bits.

Decoding and Recovery

Let I be the set of blocks that were corrupted s.t. $|I| \le e$. For each block, verify whether this block was corrupted by verifying the sign. For all $i \notin I$ verify will return true. Now considering that a PPTM adversary is responsible for this corruption, for all $i \in I$ verify will return true with negl() probability. Thus with high probability we recognize the blocks that have been corrupted.

Since at most e blocks are corrupted there are $n-e \ge k$ non-corrupted blocks remaing. Each block is an evaluation of the message polynomial, which is of deg $k-1$. Thus we can do polynomial interpolation, for ex - using Gaussian Elimination to recover the coefficients and hence the message.

Gaussian Elimination -

We want to invert the initial linear transform, i.e.

$$m = \begin{bmatrix} 1 & a_0 & a_0^2 \cdots & a_0^{k-1} \\ & & \vdots & \\ 1 & a_{n-1} & a_{n-1}^2 \cdots & a_{n-1}^{n-1} \end{bmatrix}^{-1} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix}$$

To do so we can use Gaussian elimination on the finite field $\mathbb{F}_p$. There the addition and multiplication are done modulo p. Division is considered to be multiplication by the modular inverse. A rough algorithm -

A : Augmented Matrix, i.e. for $Cx = b$ $A = [C|b]$

```
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.
```