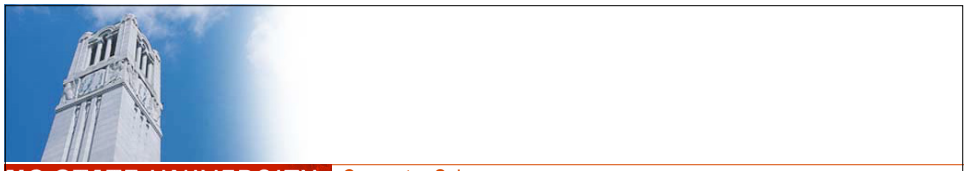


NC STATE UNIVERSITY Computer Science

CSC 774 Advanced Network Security

Topic 2.4 Rabin's Information Dispersal Algorithm

Slides by Sangwon Hyun

Motivation

- IDA was developed to provide safe and reliable transmission of information in distributed systems.
- Inefficiency of retransmission of lost packets
  - In multicast transmission, different receivers lose different sets of packets.
  - Re-request and retransmission increases delays.
- Forward error correction technique might be desirable in distributed systems.

Basic Idea of IDA

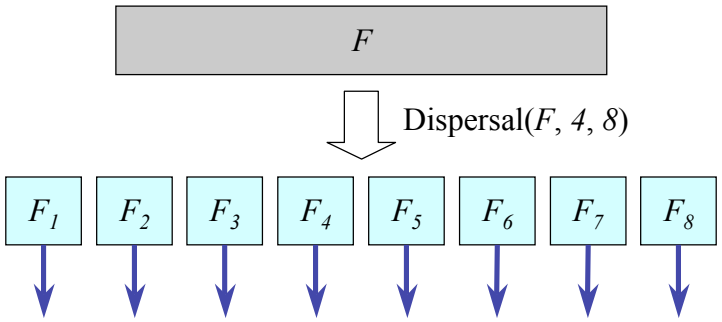
Dispersal(F, m, n)

- Let F be a data of size N in byte ($|F|$=N).
- m should be less than or equal to n ($m \leq n$).

- Dispersal(F, m, n):
  - splitting the data F with some amount of redundancy resulting in n pieces F_i ($1 \leq i \leq n$).
  - $|F_i|=|F|/m$
    - Thus, the size of F, N, should be a multiple of m.

Dispersal(F, m, n) – Example 1

- $|F|$=32 bytes, m=4, n=8



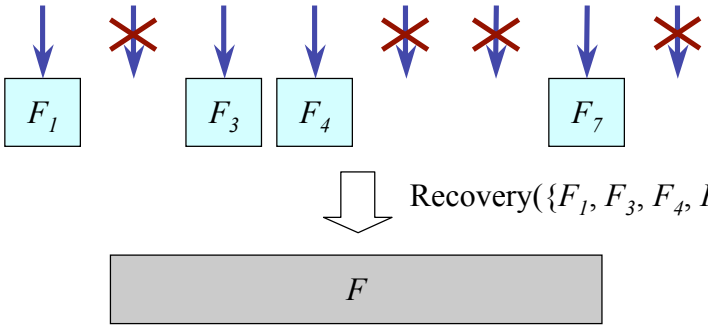
  - $|F_i| = 32/4 = 8$ bytes $(1 \le i \le n)$

Recovery($\{F_{i_j} | (1 \le j \le m), (1 \le i_j \le n)\}$, m, n)

- Recovery($\{F_{i_j} | (1 \le j \le m), (1 \le i_j \le n)\}$, m, n):
  - reconstructing the original data F from any m pieces among n pieces (F_i $(1 \le i \le n)$)

Recovery($\{Fi_j \mid (1 \le j \le m), (1 \le i_j \le n)\}$, m, n) – Example 2

- $|F|$=32 bytes, m=4, n=8, $|F_i|$=8 bytes $(1 \le i \le 8)$
- Let us assume that the following 4(=m) pieces are received.



Detailed Operations

Dispersal(F, m, n)

- $F = b_1, b_2, \ldots, b_N$
  - $|F|=N$, and b_i represents each byte in F ($0 \le b_i \le 255$).
  - All computations should be done in GF(2^8).
    - GF(2^8) is closed under addition and multiplication.
    - Every nonzero element in GF(2^8) has a multiplicative inverse.
- $F = (b_1, \ldots, b_m), (b_{m+1}, \ldots, b_{2m}), \ldots, (b_{N-m+1}, \ldots, b_N)$
  - $\mathbf{S_i} = (b_{(i-1)m+1}, \ldots, b_{im})$ ($1 \le i \le N/m$)
- The matrix, $\mathbf{M}$ ($m \times N/m$), is constructed as follows:
  - $\mathbf{M} = [\, \mathbf{S_1}\ \mathbf{S_2}\ \ldots\ \mathbf{S_{N/m}} \,]$

Dispersal(F, m, n)

- The matrix, $\mathbf{A}$ ($n \times m$), is constructed as follows:

$$A = \begin{bmatrix} a_1 \\ a_2 \\ \ldots \\ a_n \end{bmatrix}$$

  - $\mathbf{a_i} = (a_{i1}, \ldots, a_{im})$ ($1 \le i \le n$)
    - chosen such that every subset of m different vectors are linearly independent.

Dispersal(F, m, n)

- The following *Vandermonde matrix* satisfies the property required for **A**.

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{m-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{m-1} \\ 1 & x_3 & x_3^2 & \dots & x_3^{m-1} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_{n-1} & x_{n-1}^2 & \dots & x_{n-1}^{m-1} \\ 1 & x_n & x_n^2 & \dots & x_n^{m-1} \end{bmatrix}$$

  - $m \le n$, and all x_i's are nonzero elements in GF(2^8) and pairwise different.
  - Any m different rows are linearly independent, so any matrix composed of a set of any m different rows is invertible.

Dispersal(F, m, n)

- n pieces, F_i ($1 \le i \le n$), are computed as follows:

$$A \cdot M = \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_n \end{bmatrix} \begin{bmatrix} S_1 & S_2 & \dots & S_{N/m} \end{bmatrix}$$

$$= \begin{bmatrix} a_1 \cdot S_1 & a_1 \cdot S_2 & \dots & a_1 \cdot S_{N/m} \\ a_2 \cdot S_1 & a_2 \cdot S_2 & \dots & a_2 \cdot S_{N/m} \\ \dots & \dots & \dots & \dots \\ a_n \cdot S_1 & a_n \cdot S_2 & \dots & a_n \cdot S_{N/m} \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ \dots \\ F_n \end{bmatrix}$$

  - $\mathbf{a_i} \cdot \mathbf{S_k} = (a_{i1}b_{(k-1)m+1} + \dots + a_{im}b_{km})$

Dispersal(F, m, n) – Example 3

- $|F|$=32 bytes, m=4, n=8
  - $F = b_1, b_2, \ldots, b_{32}$
  - $F = (b_1, \ldots, b_4), (b_5, \ldots, b_8), \ldots, (b_{29}, \ldots, b_{32})$
  - **M** (4×8)

$$M = [S_1 \quad S_2 \quad \ldots \quad S_8] = \begin{bmatrix} b_1 & b_5 & \ldots & b_{29} \\ b_2 & b_6 & \ldots & b_{30} \\ b_3 & b_7 & \ldots & b_{31} \\ b_4 & b_8 & \ldots & b_{32} \end{bmatrix}$$

Dispersal(F, m, n) – Example 3

- **A** (8×4)

$$A = \begin{bmatrix} a_1 \\ a_2 \\ \ldots \\ a_8 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ \ldots & \ldots & \ldots & \ldots \\ 1 & x_8 & x_8^2 & x_8^3 \end{bmatrix}$$

Dispersal(F, m, n) – Example 3

- F_i $(1 \le i \le 8)$ are computed as follows:

$$A \cdot M = \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_8 \end{bmatrix} \begin{bmatrix} S_1 & S_2 & \dots & S_8 \end{bmatrix}$$

$$= \begin{bmatrix} a_1 \cdot S_1 & a_1 \cdot S_2 & \dots & a_1 \cdot S_8 \\ a_2 \cdot S_1 & a_2 \cdot S_2 & \dots & a_2 \cdot S_8 \\ \dots & \dots & \dots & \dots \\ a_8 \cdot S_1 & a_8 \cdot S_2 & \dots & a_8 \cdot S_8 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ \dots \\ F_8 \end{bmatrix}$$

Recovery($\{F_{i_j} | (1 \le j \le m), (1 \le i_j \le n)\}$, m, n)

- Given m pieces F_{i_j} ($(1 \le j \le m), (1 \le i_j \le n)$),

$$\begin{bmatrix} F_{i_1} \\ F_{i_2} \\ \dots \\ F_{i_m} \end{bmatrix} = \begin{bmatrix} a_{i_1} \\ a_{i_2} \\ \dots \\ a_{i_m} \end{bmatrix} \cdot M = A' \cdot M$$

- **M** can be recovered from the given m pieces F_{i_j} ($(1 \le j \le m), (1 \le i_j \le n)$) because A' is invertible.

$$\begin{bmatrix} a_{i_1} \\ a_{i_2} \\ \dots \\ a_{i_m} \end{bmatrix}^{-1} \begin{bmatrix} F_{i_1} \\ F_{i_2} \\ \dots \\ F_{i_m} \end{bmatrix} = M$$

Recovery($\{F_{i_j} | (1 \le j \le m), (1 \le i_j \le n)\}, m, n$) – Example 4

- $|F|$=32 bytes, m=4, n=8
- In example 3, F_i ($1 \le i \le 8$) pieces of 8 bytes are resulted.
- Assume that $\{F_1, F_3, F_4, F_7\}$ are received among them.

$$\begin{bmatrix} F_1 \\ F_3 \\ F_4 \\ F_7 \end{bmatrix} = \begin{bmatrix} a_1 \cdot S_1 & a_1 \cdot S_2 & \dots & a_1 \cdot S_8 \\ a_3 \cdot S_1 & a_3 \cdot S_2 & \dots & a_3 \cdot S_8 \\ a_4 \cdot S_1 & a_4 \cdot S_2 & \dots & a_4 \cdot S_8 \\ a_7 \cdot S_1 & a_7 \cdot S_2 & \dots & a_7 \cdot S_8 \end{bmatrix} = \begin{bmatrix} a_1 \\ a_3 \\ a_4 \\ a_7 \end{bmatrix} \cdot M$$

Recovery($\{F_{i_j} | (1 \le j \le m), (1 \le i_j \le n)\}, m, n$) – Example 4

- The original data M can be recovered by the following computation:

$$\begin{bmatrix} a_1 \\ a_3 \\ a_4 \\ a_7 \end{bmatrix}^{-1} \begin{bmatrix} F_1 \\ F_3 \\ F_4 \\ F_7 \end{bmatrix} = M$$