Secret Sharing Schemes Based on Minimum Bandwidth Regenerating Codes

Masazumi Kurihara(Univ. of Electro-Communications)
Hidenori Kuwakado (Kobe Univ.)

ISITA2012, Honolulu, Hawaii, U.S.A., Oct. 28 - 31, 2012

Outline

1. Introduction

   Distributed storage system, Regenerating Code, and Secrecy

2. (n, k, d) Minimum Bandwidth Regenerating(MBR) Codes

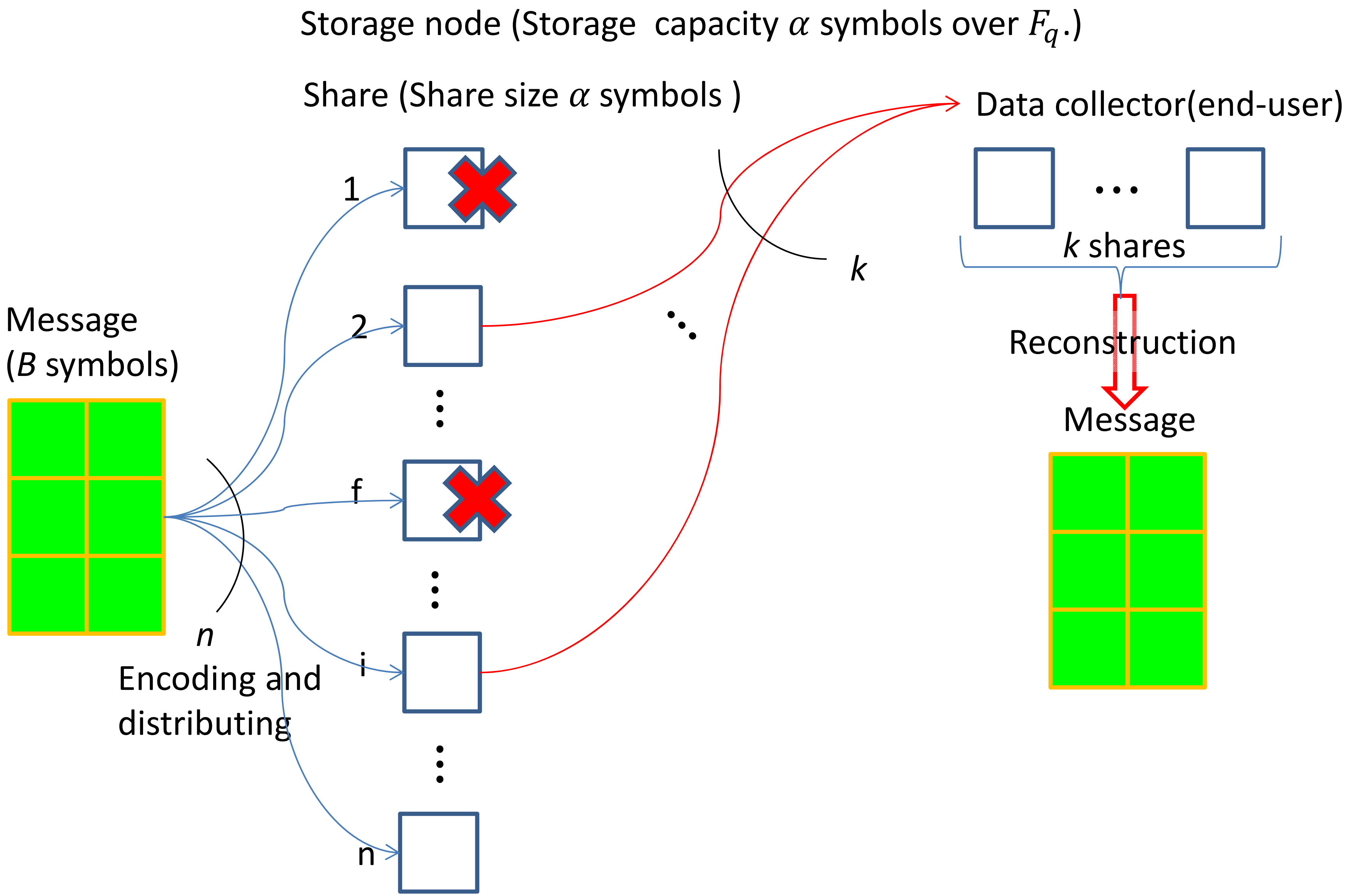
   The (n, k, d) MBR code proposed by Rashmi, Shah and Kumar

3. (n, k, d, m) Secure Regenerating(SR) Codes

   The secure regenerating(SR) code based on the (n, k, d) MBR code

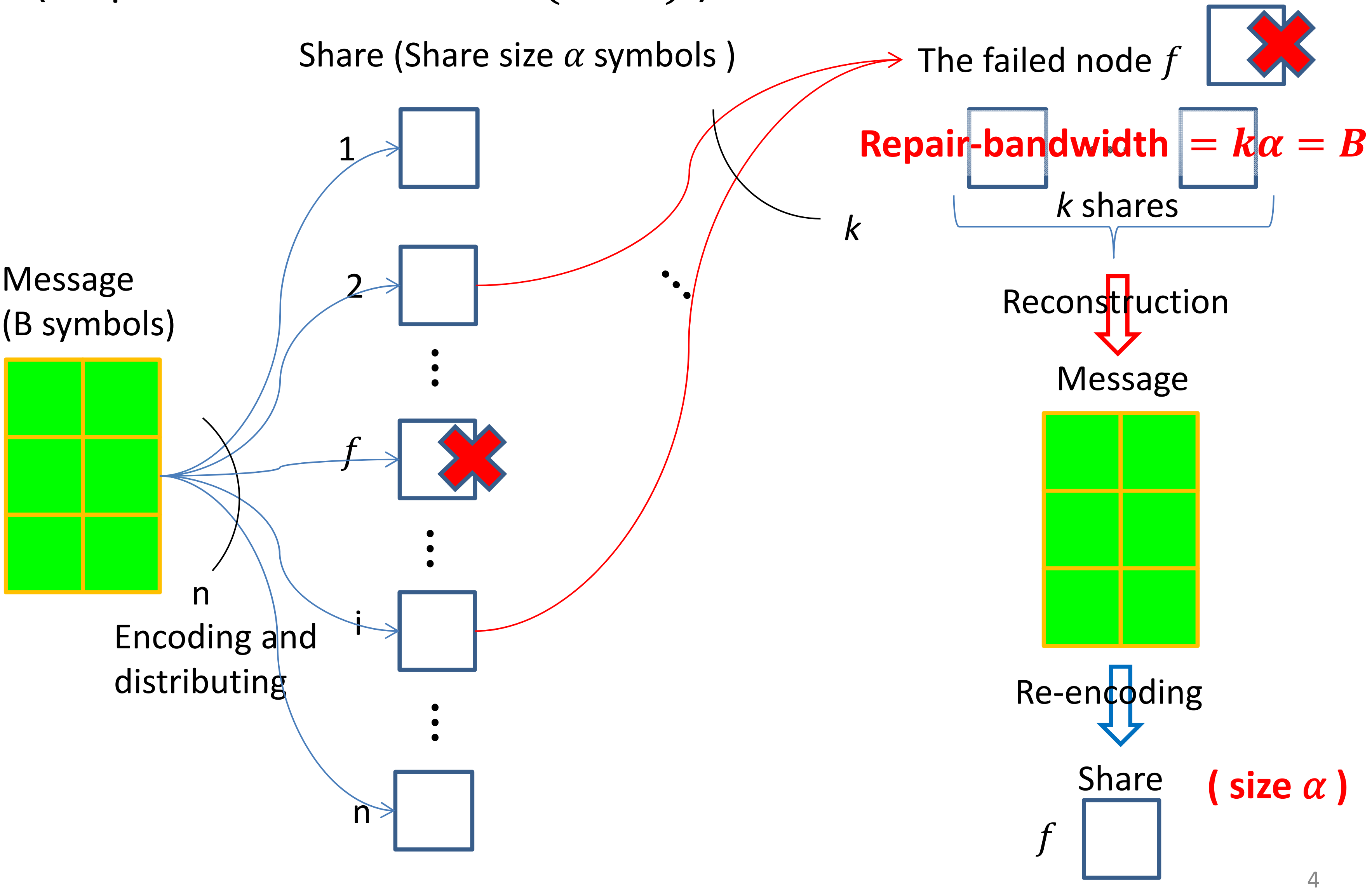
4. Evaluation
5. Conclusions

Distributed Storage System (n, k, α, B)


Storage node (Storage capacity α symbols over F_q.)
Share (Share size α symbols)
Data collector(end-user)
Message
(B symbols)
1
2
f
i
n
n
Encoding and distributing
k
k shares
Reconstruction
Message

Typical repair method using a reconstruction

(Repair-bandwidth $= B\ (\geq \alpha\ )$)



$(n, k, d, \alpha, \beta, B)$ Regenerating Codes

- For the repair problem, Dimakis et al. proposed *a new concept of code called "regenerating code"*.

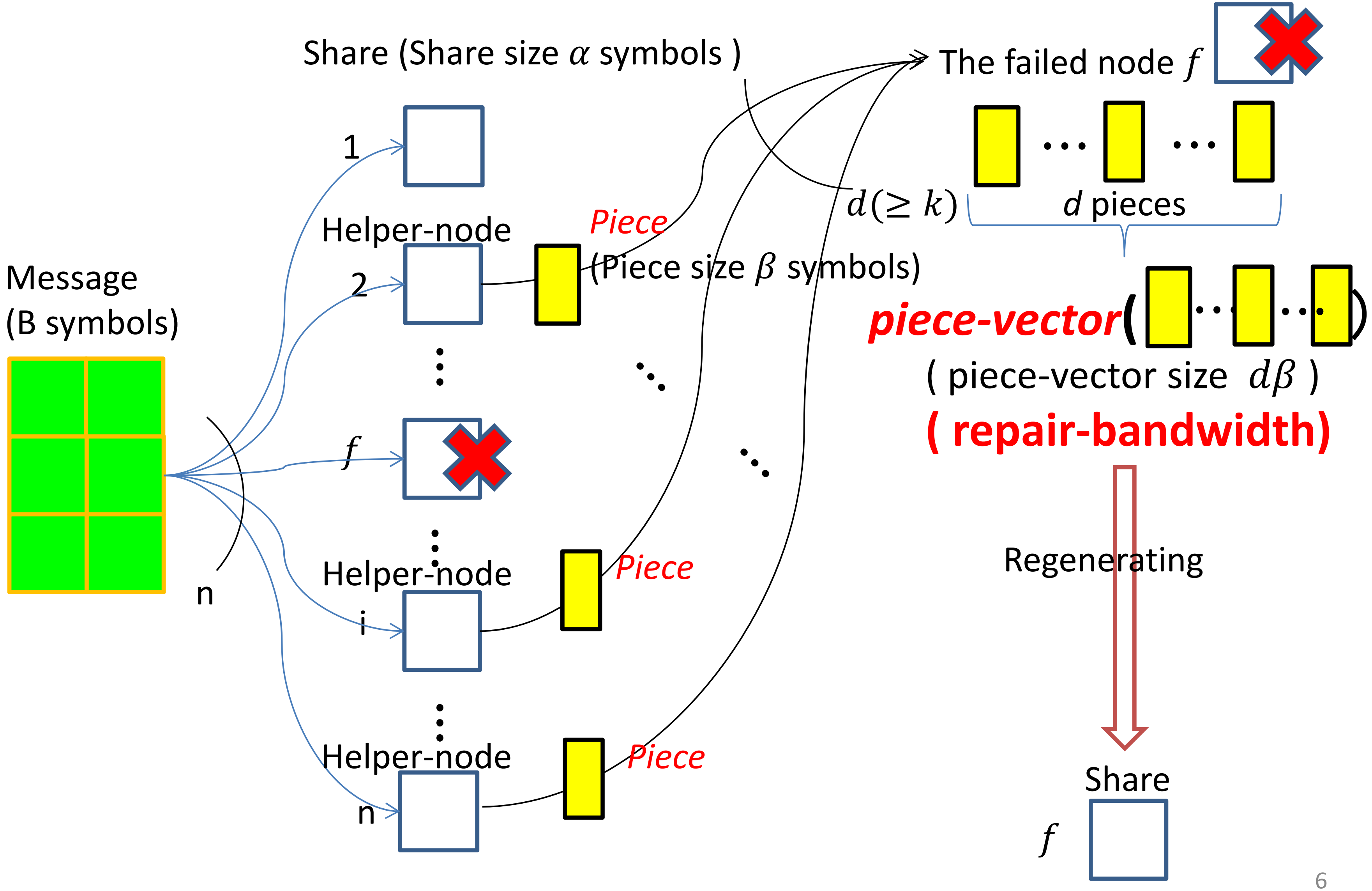
  Dimakis, Godfrey, Wu, Wainwright and Ramchandran[Dimakis, et al., 2010]

- The code is defined by six parameters $(n, k, d, \alpha, \beta, B)$.
- The code have the following two properties:
  - *Reconstruction Property*:
    - An end-user(called data-collector) is permitted to connect to any k active nodes to reconstruct a message.
  - *Regeneration Property*:
    - A failed node is permitted to connect to any $d(\geq k)$ active nodes (called helper-nodes) to repair itself.

- They showed that the regenerating code can reduce the repair-bandwidth.

Method using a regenerating code for repair

(Repair-bandwidth = piece-vector size = $d\beta \leq B$)



$(n, k, d, \alpha, \beta, B)$ Regenerating Codes

- Furthermore, they showed the trade-off between a storage-capacity and a repair-bandwidth.

- In the trade-off, there are two special types of regenerating codes as follows: (for fixed k, d and B)
  - *An Minimum Bandwidth Regenerating(MBR) code*
    - First minimizing β, and then minimizing α.
    - An MBR code satisfies $\beta = \frac{2B}{k(2d-k+1)}, \alpha = d\beta$.
  - *An Minimum Storage Regenerating(MSR) code*
    - The minimization in the reverse order.
    - An MSR code satisfies $\alpha = \frac{B}{k}, \beta = \frac{\alpha}{d-k+1}$.

Secrecy on Distributed storage System

- A regenerating code may be similar to a secret sharing scheme.

- The secret sharing scheme(SSS) produces shares in such a way that a share does not give any information about a secret.
- However, in general, the SSS does not have the regeneration property.

- On the other hand, in the concept of a regenerating code, the regenerating code does not have the secrecy property.

Prior work(related work) for secure MBR codes

- Pawar, Rouayheb and Ramchandran[Pawar, et al., 2011]
  - The first secure regenerating code based on an MBR code.
  - However, the secure regenerating code is confined to the case of $n = d + 1$.

- Shah, Rashmi and Kumar[Shah, et al., 2011]
  - An $\{\ell, \ell'\}$ secure Product-Matrix Minimum Bandwidth Regenerating(PM-MBR) code for $n > d$.
  - The code is also based on an MBR code.
  - The parameters n and d are chosen independently.

- Our proposal (n, k, d, m) secure regenerating(SR) code for $n > d$ in this study.
  - Shah et al.'s code and our code are based on the same MBR code.
  - Our code is different from their code.

Secrecy on Regenerating Code

- Let S denote a random variable with a uniform distribution over $F_q^{L_S}$ representing a secret where $L_S \leq B$.
- Let $C_1, \ldots, C_n$ denote random variables representing n shares from the secret S.
- Let $D_1, \ldots, D_n$ denote random variables representing n piece-vectors.

- For a regenerating code, we have to consider the following two secrecy conditions:
  1. *Secrecy for shares*:

     For any m shares $C_{i_1}, \ldots, C_{i_m}$,

     $$H(S|C_{i_1}, \ldots, C_{i_m}) = H(S),$$

     where $m < k$.
  2. *Secrecy for piece-vectors*:

     For any l piece-vectors $D_{i_1}, \ldots, D_{i_l}$,

     $$H(S|D_{i_1}, \ldots, D_{i_l}) = H(S),$$

     where $l < k$.

(n,k,d) MBR codes [Rashmi, et al., 2011](Section 2)

- Rashmi, Shah and Kumar proposed an (n,k,d) MBR code for all values of (n,k,d) where $d \geq k$. [Rashmi, et al., 2011]

- The parameters of the (n,k,d) MBR code satisfy as follows:

$$\alpha = d, \qquad \beta = 1, \qquad B = \frac{1}{2}\, k(2d - k + 1)$$

- Hence, the (n,k,d) MBR code is defined by the three parameters $n, k \text{ and } d$ from the above relations.

A message matrix M of the (n, k, d) MBR code

- The (n, k, d) MBR code with B message symbols is obtained from the following *message matrix M* which is *a $d \times d$ symmetric matrix.*
- The B message symbols are substituted for components of the message matrix M as follows:

$$M = \underbrace{\begin{bmatrix} \overbrace{\begin{matrix} u_{1,1} & \cdots & u_{1,k} \\ \vdots & \ddots & \vdots \\ u_{k,1} & \cdots & u_{k,k} \end{matrix}}^{k} & \overbrace{\begin{matrix} u_{1,k+1} & \cdots & u_{1,d} \\ \vdots & \ddots & \vdots \\ u_{k,k} & \cdots & u_{k,d} \end{matrix}}^{d-k} \\ \begin{matrix} u_{k+1,1} & \cdots & u_{k+1,k} \\ \vdots & \ddots & \vdots \\ u_{d,1} & \cdots & u_{d,k} \end{matrix} & \begin{matrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{matrix} \end{bmatrix}}_{d} \left.\begin{matrix} \left.\vphantom{\begin{matrix} u \\ \vdots \\ u \end{matrix}}\right\} k \\ \left.\vphantom{\begin{matrix} u \\ \vdots \\ u \end{matrix}}\right\} d-k \end{matrix}\right\} d$$

B message symbols

Encoding, Shares and Reconstruction

- For each node $i \in \{1, \ldots, n\}$, *a share* $\underline{c}_i$ is defined as

$$\underline{c}_i = \left[c_{i,1}, \ldots, c_{i,d}\right]^t := M\underline{\phi}_i \in F_q^d$$

where $\underline{\phi}_i = \left[\, 1, x_i, x_i^2, \ldots, x_i^{d-1} \,\right]^t \in F_q^d$ is *a coding vector* associated with node i.

- Hence, n shares $\underline{c}_1, \ldots, \underline{c}_n$ are obtained as follows:

$$\underset{(d\times n)}{[\underline{c}_1, \ldots, \underline{c}_n]} = \underset{(d\times d)}{M} \underset{(d\times n)}{\left[\underline{\phi}_1, \ldots, \underline{\phi}_n\right]}$$

- The message matrix can be reconstructed from any k shares by using the reconstruction method by Rashmi et al.

(n,k,d,m) Secure Regenerating(SR) codes (Section 3)

- An (n,k,d,m) Secure Regenerating(SR) code is based on an (n,k,d) MBR code and have the following properties:

1. The three parameters (n,k,d) are derived from the underlying (n,k,d) MBR code.
2. The new parameter m $(0 \leq m \leq k)$ is a secrecy parameter.
3. The parameter m means the perfect secrecy condition as follows: for any $i_1, \ldots, i_m \in \{1, \ldots, n\}$,
   $H\left(S \middle| C_{i_1}, \ldots, C_{i_m}\right) = H(S)$ and $H\left(S \middle| D_{i_1}, \ldots, D_{i_m}\right) = H(S)$.

Construction of an (n, k, d, m) Secure Regenerating(SR) Code

- To construct an (n, k, d, m) secure regenerating(SR) code, instead of B message symbols, we substitute L_S secret symbols and L_R random symbols for components of the message matrix M.
  - The numbers L_S and L_R are defined by the secrecy parameter m as follows:

    $L_S = \frac{1}{2}(m-k)\big(m-(2d-k+1)\big)$,

    $L_R = \frac{1}{2}m(2d-k+1)$

    and $L_S + L_R = B$.

- The idea of the construction is simple.
- However, we have carefully to choose the components of the message matrix as follows:

A message matrix for the underlying (n,k,d) MBR code

- When $(k = 4, d = 6)$, $B = 18$,

$d = 6$

$k = 4$

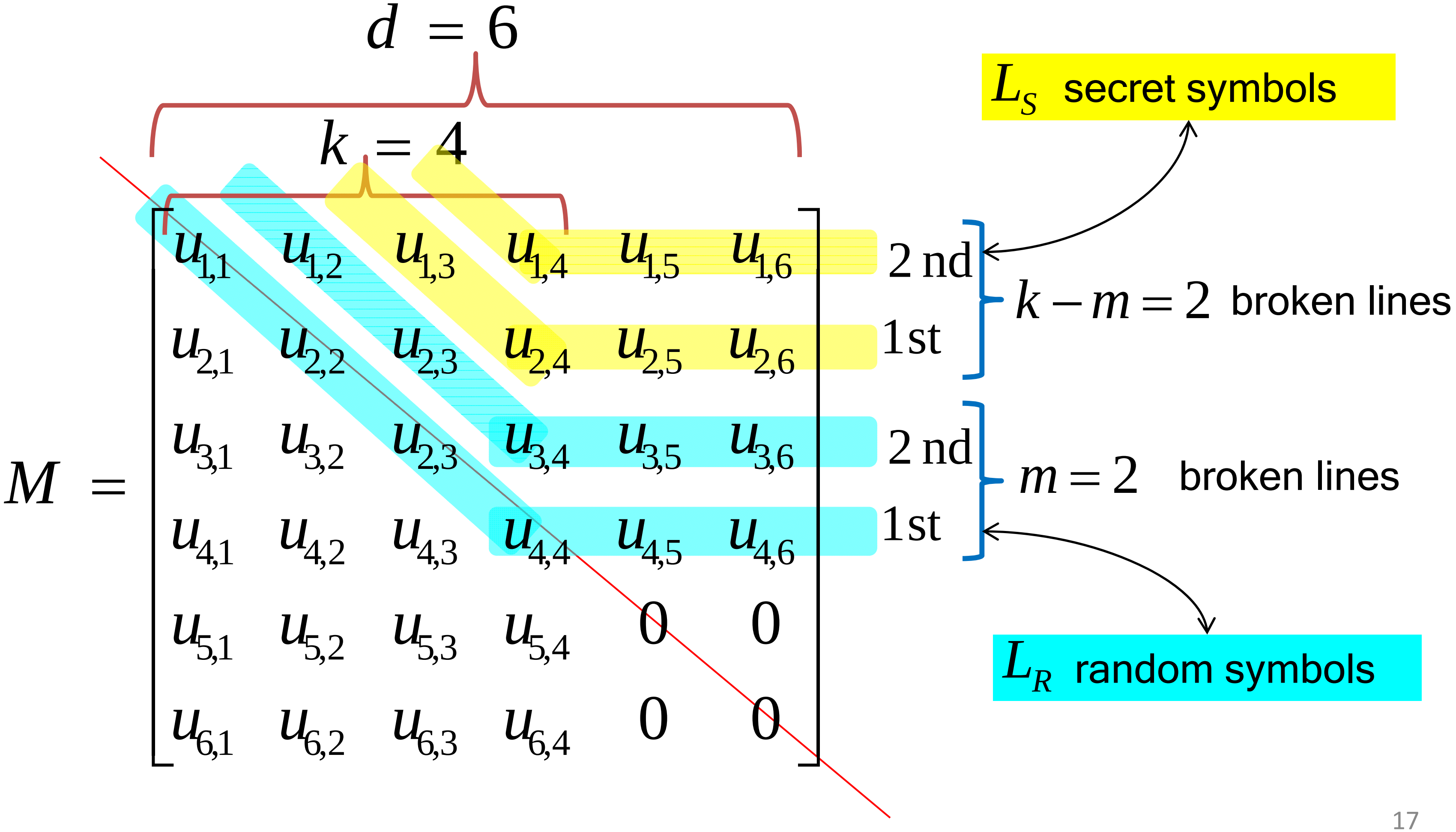
$$M = \left[\begin{array}{cccc|cc} u_{1,1} & u_{1,2} & u_{1,3} & u_{1,4} & u_{1,5} & u_{1,6} \\ u_{2,1} & u_{2,2} & u_{2,3} & u_{2,4} & u_{2,5} & u_{2,6} \\ u_{3,1} & u_{3,2} & u_{2,3} & u_{3,4} & u_{3,5} & u_{3,6} \\ u_{4,1} & u_{4,2} & u_{4,3} & u_{4,4} & u_{4,5} & u_{4,6} \\ \hline u_{5,1} & u_{5,2} & u_{5,3} & u_{5,4} & 0 & 0 \\ u_{6,1} & u_{6,2} & u_{6,3} & u_{6,4} & 0 & 0 \end{array}\right]$$

$k = 4$

B message symbols

A message Matrix
for the (n,k,d,m) secure regenerating(SR) code

- When $(k=4, d=6, m=2)$, $B=18$, $L_R=11$ and L_S =7.



$d=6$

$k=4$

$$M=\begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} & u_{1,4} & u_{1,5} & u_{1,6} \\ u_{2,1} & u_{2,2} & u_{2,3} & u_{2,4} & u_{2,5} & u_{2,6} \\ u_{3,1} & u_{3,2} & u_{2,3} & u_{3,4} & u_{3,5} & u_{3,6} \\ u_{4,1} & u_{4,2} & u_{4,3} & u_{4,4} & u_{4,5} & u_{4,6} \\ u_{5,1} & u_{5,2} & u_{5,3} & u_{5,4} & 0 & 0 \\ u_{6,1} & u_{6,2} & u_{6,3} & u_{6,4} & 0 & 0 \end{bmatrix}$$

2 nd

1st

$k-m=2$ broken lines

2 nd

1st

$m=2$ broken lines

L_S secret symbols

L_R random symbols

- The n shares for the secret S are derived from the encoding method of the underlying (n, k, d) MBR code as follows:

$$[\underline{c}_1, \dots, \underline{c}_n] = M\left[\underline{\phi}_1, \dots, \underline{\phi}_n\right].$$

- We can execute a reconstruction and a regeneration for the (n, k, d, m) secure regenerating(SR) code in the same way as the underlying (n, k, d) MBR code.

Evaluation (shares) (Section 4)

- **Theorem**: For any t shares $C_{i_1}, \ldots, C_{i_t}$ of the (n, k, d, m) secure regenerating(SR) code,
$$H\left(S \middle| C_{i_1}, \ldots, C_{i_t}\right) = g(t)$$
where $H(S) = L_S$, and the function $g(t)$ is defined by
$$g(t) = \begin{cases} L_S, & 0 \le t \le m \\ \frac{1}{2}(t-k)(t-(2d-k+1)), & m+1 \le t \le k-1 \\ 0, & k \le t \le n \end{cases}$$

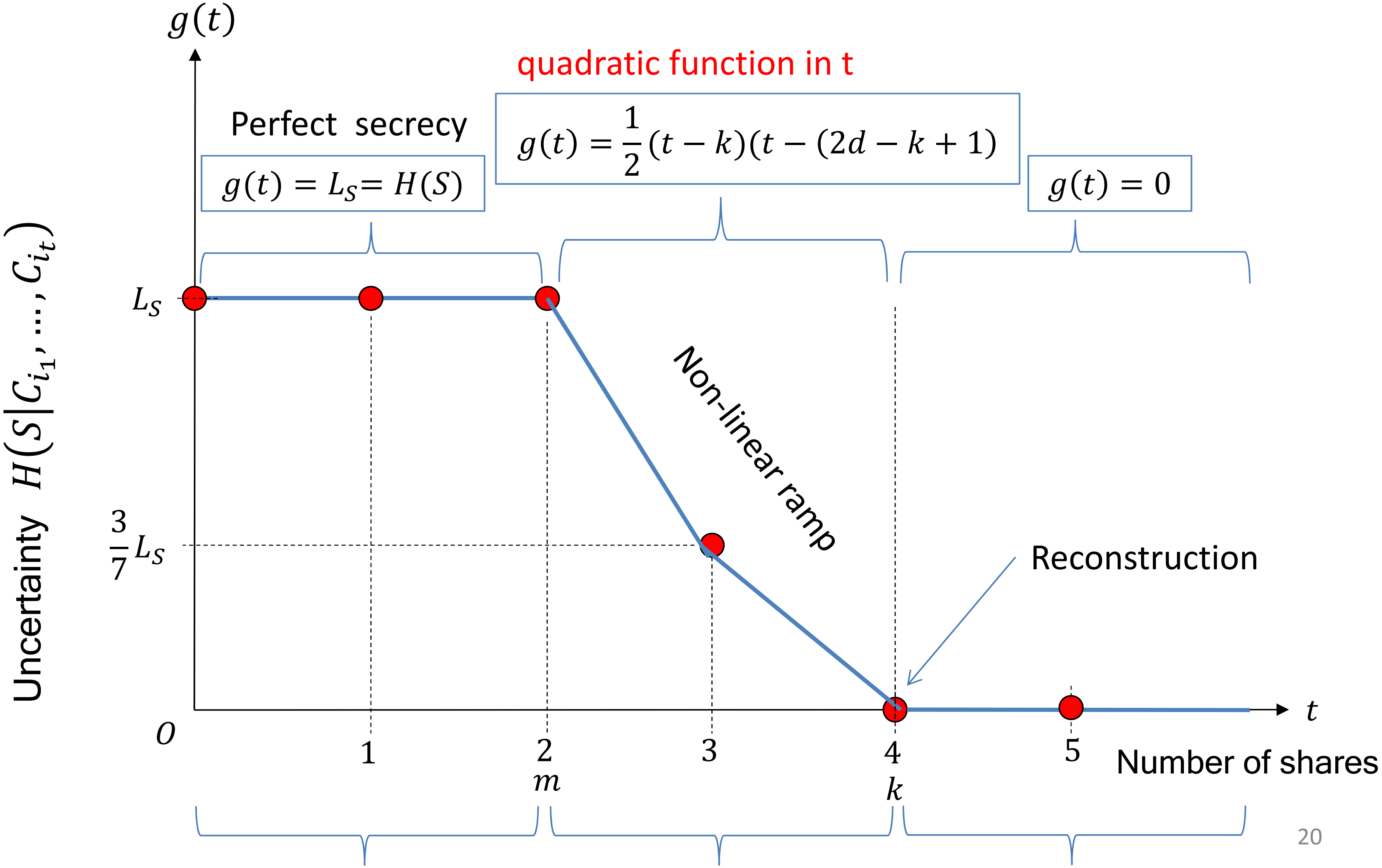
- $g(t)$ is a quadratic polynomial in t in the range $m \le t \le k$.

- In particular,
  - $H\left(S \middle| C_{i_1}, \ldots, C_{i_m}\right) = L_S$, $(t = m)$: Perfect secrecy
  - $H\left(S \middle| C_{i_1}, \ldots, C_{i_k}\right) = 0$, $(t = k)$: Reconstruction

- The reason using the function $g(t)$ is that we are interested not only in a perfect secrecy, but also in a ramp type's secrecy.

$H(S|C_{i_1},\ldots,C_{i_t}) = g(t)$ versus t

When $(k = 4, d = 6, m = 2)$,



Evaluation(piece-vectors)

- Similarly, we have the following theorem for piece-vectors.

- **Theorem**: For any t piece-vectors $D_{i_1}, \dots, D_{i_t}$ of (n, k, d, m) secure regenerating(SR) code,

$$H\left(S \middle| D_{i_1}, \dots, D_{i_m}\right) = g(t).$$

- In particular,
  - $H\left(S \middle| D_{i_1}, \dots, D_{i_m}\right) = L_S, \ \ (t = m)$: Perfect secrecy
  - $H\left(S \middle| D_{i_1}, \dots, D_{i_k}\right) = 0, \ \ (t = k)$: Reconstruction

Conclusions(Section 5)

1. We have proposed a construction of an (n, k, d, m) secure regenerating(SR) code based on an (n, k, d) MBR code.

2. We have showed the secrecy ability of the (n, k, d, m) secure regenerating(SR) code is as follows:
   1. $H\left(S \middle| C_{i_1}, \dots, C_{i_t}\right) = g(t)$ for any t shares.
   2. $H\left(S \middle| D_{i_1}, \dots, D_{i_t}\right) = g(t)$ for any t piece-vectors.
3. We have explained that the (n, k, d, m) secure regenerating(SR) code is a (non-linear) ramp scheme.

4. The (n, k, d, m) secure regenerating(SR) code achieves the upper bound of the secrecy capacity $C_S(n, k, d, \alpha = d, \beta = 1; l = m, m)$.

Additional Slides

Distributed Storage System (n, k, α, B)

- There are n *storage nodes* in a network.
- *The storage capacity* of each node is α symbols over a finite field F_q .

- *Encoding and Distribution*:
- *A message* consisting of B *message symbols* is encoded to n *shares* in such a way that the message can be reconstructed from any k shares, and the n shares are stored across n storage nodes.
- The *share-size* equals to the storage capacity.

- In the system, the message can be reconstructed from active nodes even if several nodes fail.

Repairing a failed node

- On the other hand, we have to repair the failed node to maintain the system, that is, the failed node have to regenerate the share of itself.
- In a typical repair method, the failed node can regenerate the share by using a reconstruction.
- However, the reconstruction spends the network traffic because the message-size B is greater than the share-size α .
- The amount of downloaded data for repair is called *the repair-bandwidth*.
- In the case of a reconstruction, the repair-bandwidth is B, which is the message-size.

$(n, k, d, \alpha, \beta, B)$ Regenerating Codes

- They showed that the regenerating code can reduce the repair-bandwidth.

- The data-size of downloaded data(called *piece*) from each helper-node is β symbols. Consequently, the repair band-width is $d\beta$.
- The vector consisting of d pieces is called *a piece-vector*.

Secrecy on Regenerating Code

- Let S denote a random variable with a uniform distribution over $F_q^{L_S}$ representing a secret where $L_S \le B$.

- Let $C_1, \dots, C_n$ denote random variables representing n shares from the secret S.
- The reconstruction can be represented as follows:

  for any k shares $C_{i_1}, \dots, C_{i_k}$, $H(S|C_{i_1}, \dots, C_{i_k}) = 0$.

- Let $D_1, \dots, D_n$ denote random variables representing n piece-vectors.
- The regeneration can be represented as follows:

  for a failed node f, $H(C_f|D_f) = 0$.

- From the regeneration property, we have $H(S|C_f) \ge H(S|D_f)$.

Regeneration for the (n, k, d) MBR code

- Two pages.

Regeneration for the (n, k, d) MBR code

- Suppose that a node f fails and d helper-nodes $h_1, \dots, h_d$are active.
- Each helper node h computes a piece for the failed node as follows: $(\beta = 1)$

$$d_{f,h} = \underline{c}_h^t \, \underline{\phi}_f \quad \in F_q$$

where $h \in \{h_1, \dots, h_d\}$, and send it to the failed node.

- As a result, the failed node obtains the piece-vector as follows: $(d\beta = d)$

$$\underline{d}_f = \left[d_{f,h_1}, \dots, d_{f,h_d} \right]^t \in F_q^d$$

- Note that the repair-bandwidth equals to the size of piece-vector.

Regeneration for the (n, k, d) MBR code

- The failed node can regenerate the share $\underline{c}_f$ from the piece-vector $\underline{d}_f$ as follows:

$$\underline{c}_f = \left(\left[\underline{\phi}_{h_1}, \ldots, \underline{\phi}_{h_d}\right]^t \right)^{-1} \underline{d}_f$$

where the $d \times d$ matrix $\left[\underline{\phi}_{h_1}, \ldots, \underline{\phi}_{h_d}\right]$ is nonsingular

(i.e.,$H(C_f|D_f) = 0$.)

- Form the above relation between $\underline{d}_f$ and $\underline{c}_f$, the piece-vector $\underline{d}_f$ is also determined from the share $\underline{c}_f$ (i.e., $H(D_f|C_f) = 0$).

- Hence, for the (n, k, d) MBR code,

"$H\left(S\middle|C_{i_1}, \ldots, C_{i_m}\right) = H(S)$" is equivalent to "$H(S|D_{i_1}, \ldots, D_{i_m}) = H(S)$".

The difference between Shah et al.' code and our code (four pages)

- When $\ell = m$ and $\ell' = 0$, their code and our code have the same secrecy ability.

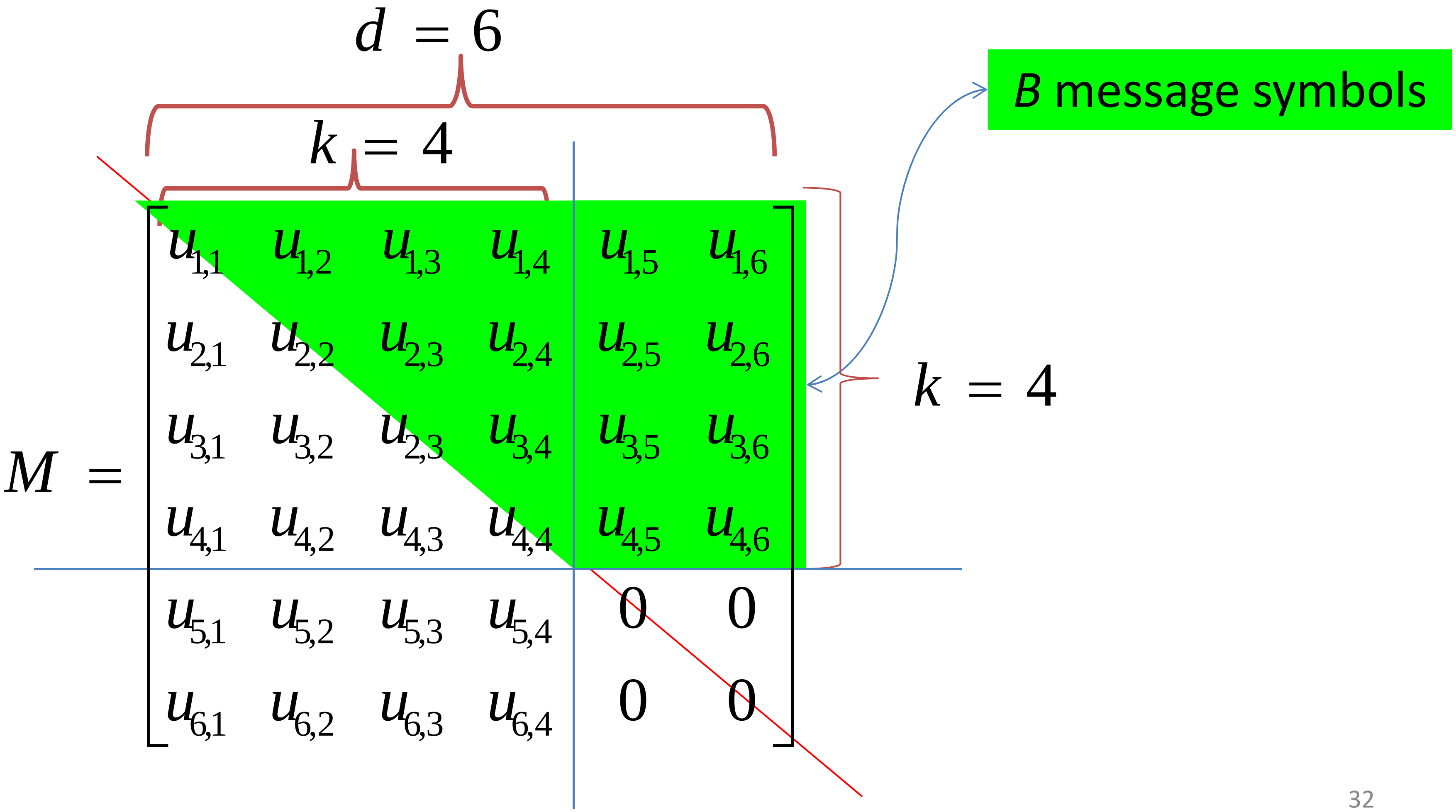
  $(\{\ell = m, \ell' = 0\}$-PM-secure-MBR code

  (n, k, d, m) secure regenerating code

- Their code and our code differ in the position of random symbols and that of secret symbols in a message matrix M as follows:

Message Matrix M for the underlying (n, k, d) MBR code

- When $(k = 4, d = 6)$, $B = 18$,



$d = 6$

$k = 4$

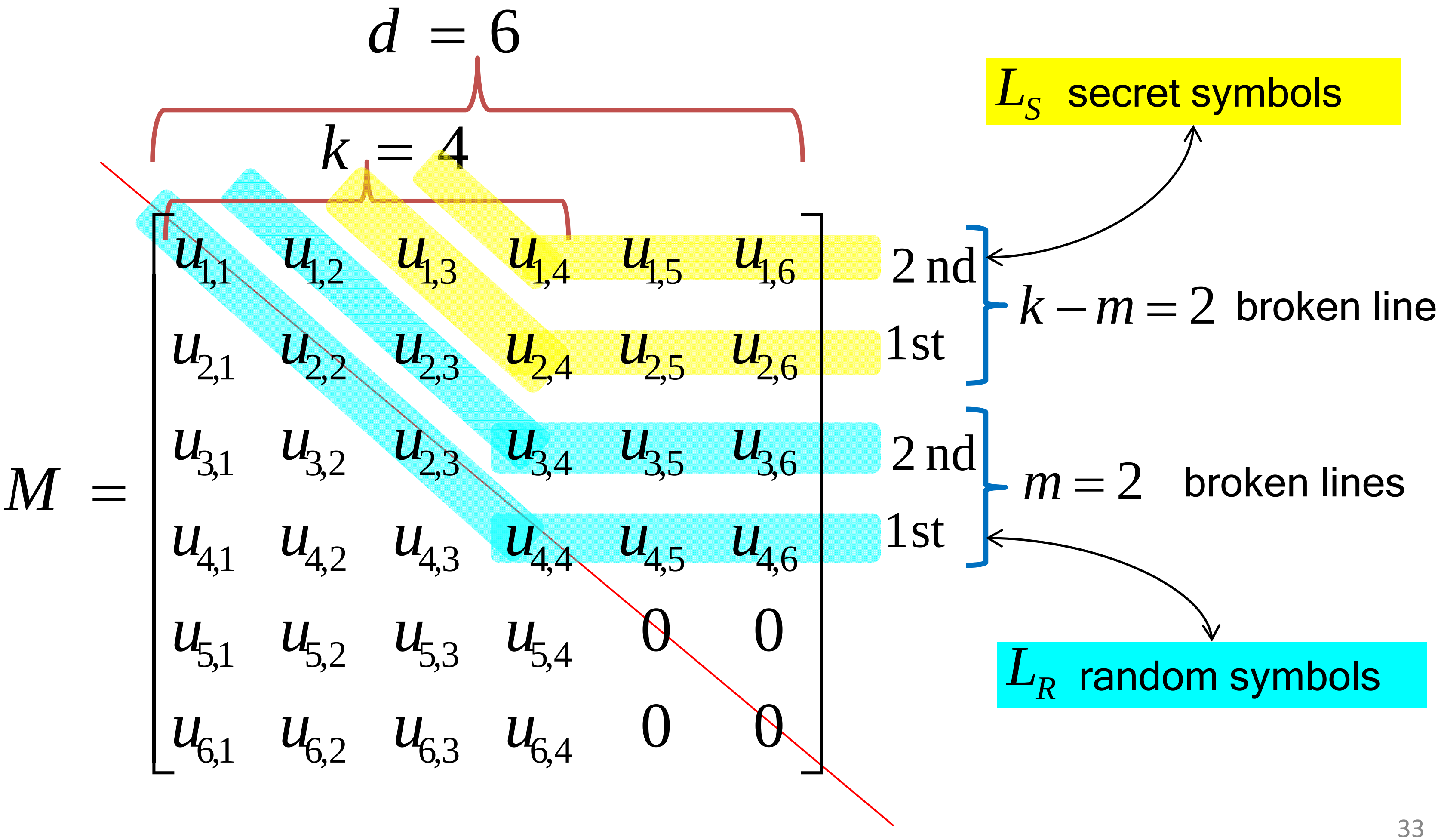
$$M = \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} & u_{1,4} & u_{1,5} & u_{1,6} \\ u_{2,1} & u_{2,2} & u_{2,3} & u_{2,4} & u_{2,5} & u_{2,6} \\ u_{3,1} & u_{3,2} & u_{2,3} & u_{3,4} & u_{3,5} & u_{3,6} \\ u_{4,1} & u_{4,2} & u_{4,3} & u_{4,4} & u_{4,5} & u_{4,6} \\ u_{5,1} & u_{5,2} & u_{5,3} & u_{5,4} & 0 & 0 \\ u_{6,1} & u_{6,2} & u_{6,3} & u_{6,4} & 0 & 0 \end{bmatrix}$$

$k = 4$

B message symbols

Our code
(the (n, k, d, m) secure regenerating code)

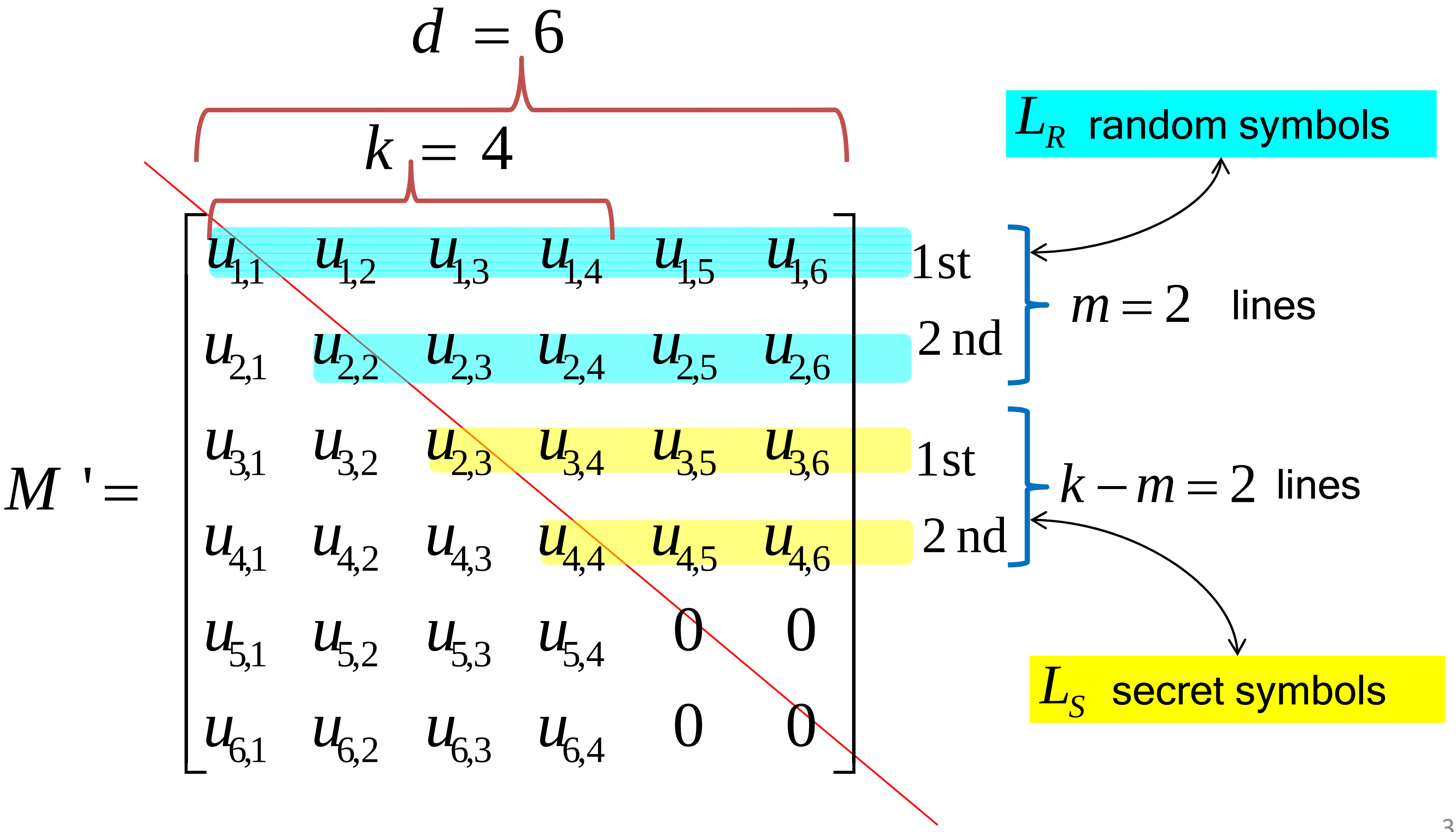
- When $(k = 4, d = 6, m = 2)$, $B = 18$, $L_R = 11$ and L_S =7.



$$M = \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} & u_{1,4} & u_{1,5} & u_{1,6} \\ u_{2,1} & u_{2,2} & u_{2,3} & u_{2,4} & u_{2,5} & u_{2,6} \\ u_{3,1} & u_{3,2} & u_{2,3} & u_{3,4} & u_{3,5} & u_{3,6} \\ u_{4,1} & u_{4,2} & u_{4,3} & u_{4,4} & u_{4,5} & u_{4,6} \\ u_{5,1} & u_{5,2} & u_{5,3} & u_{5,4} & 0 & 0 \\ u_{6,1} & u_{6,2} & u_{6,3} & u_{6,4} & 0 & 0 \end{bmatrix}$$

Shah et al.'s secure MBR code[Shah, et al., 2012]
(the $\{\ell = m, \ell' = 0\}$-PM-secure-MBR code)

- When $(k = 4, d = 6, m = 2)$, $B = 18$, $L_R = 11$ and $L_S = 7$.



$$d = 6 \qquad k = 4$$

$$M' = \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} & u_{1,4} & u_{1,5} & u_{1,6} \\ u_{2,1} & u_{2,2} & u_{2,3} & u_{2,4} & u_{2,5} & u_{2,6} \\ u_{3,1} & u_{3,2} & u_{2,3} & u_{3,4} & u_{3,5} & u_{3,6} \\ u_{4,1} & u_{4,2} & u_{4,3} & u_{4,4} & u_{4,5} & u_{4,6} \\ u_{5,1} & u_{5,2} & u_{5,3} & u_{5,4} & 0 & 0 \\ u_{6,1} & u_{6,2} & u_{6,3} & u_{6,4} & 0 & 0 \end{bmatrix}$$

$m = 2$ lines

$k - m = 2$ lines

L_R random symbols

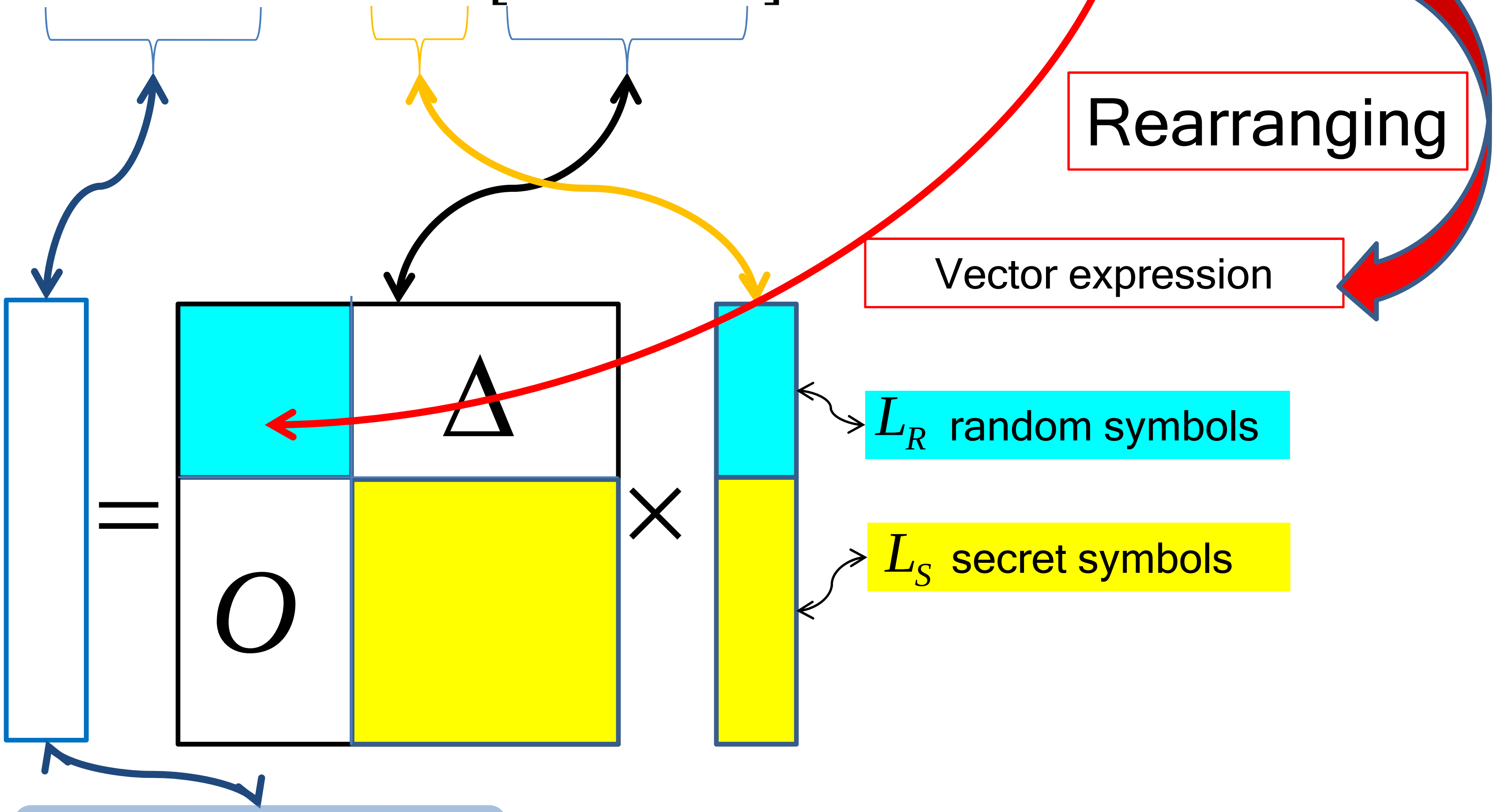
L_S secret symbols

Proof(two pages)

- The idea of construction of the (n, k, d, m) secure regenerating code is simple.
- However, many pages are expended to proof the secrecy of the (n, k, d, m) secure regenerating code.

- It is a key point of the proof that the $L_R \times L_R$ submatrix is nonsingular.

- $[\underline{c}_1, \dots, \underline{c}_m] = M \left[\underline{\phi}_1, \dots, \underline{\phi}_m\right]$



Matrix expression

Rearranging

Vector expression

$=$ Δ O $\times$

L_R random symbols

L_s secret symbols

- The L_s components of m shares are linearly independent.

Secrecy capacity and its upper bound

- Four pages

Secrecy capacity and its upper bound

- The secrecy capacity is defined to be the maximum amount of data that can be stored in the distributed storage system such that the reconstruction property and two the conditions are simultaneously satisfied for all possible data-collectors and eavesdroppers, that is,

$$C_S(n,k,d,\alpha,\beta;l,m) \quad = \quad \sup \quad H(S)$$

$$H\left(S\middle|C_{i_1},\dots,C_{i_k}\right) = 0$$

$$H\left(S\middle|C_{i_1},\dots,C_{i_m}\right) = H(S)$$

$$H(S|D_{j_1},\dots,D_{j_l}) = H(S)$$

- Furthermore, we have the following upper bound of $C_S(n,k,d,\alpha,\beta;l,m)$:

$$C_S(n,k,d,\alpha,\beta;l,m) \leq \sum_{j=\max\{l,m\}+1}^{k} \{(d-j+1)\beta,\alpha\}$$

- Both the secrecy capacity and the upper bound are the refined versions of that proposed by Pawar et al.[Pawar, et al.,2011].

For an MBR code, we can assume that $l = m$ without loss of generality

- In particular, for an MBR code, when a regenerating function is bijective, the following two propositions are true because $H(D_f|C_f) = 0$ and $H(C_f|D_f) = 0$.

- $H\left(S\middle|C_{i_1}, \ldots, C_{i_m}\right) = H(S)$ implies $H(S|D_{i_1}, \ldots, D_i) = H(S)$.
- $H(S|D_{j_1}, \ldots, D_{j_l}) = H(S)$ implies $H\left(S\middle|C_{j_1}, \ldots, C_{j_l}\right) = H(S)$.

- Hence, we can assume that $l = m$ without loss of generality for an MBR code.
- Consequently, $H\left(S\middle|C_{i_1}, \ldots, C_{i_m}\right) = H(S)$ is equivalent to $H(S|D_{i_1}, \ldots, D_{i_m}) = H(S)$.

Secrecy capacity and its upper bound for an (n, k, d, m) secure regenerating code

- For an (n, k, d, m) secure regenerating (secure MBR) code, that is, $l = m$, we have the following simplified expressions:

- The secrecy capacity :

$$C_S(n, k, d, \alpha = d, \beta = 1; l = m, m) \quad = \quad \sup_{\substack{H(S|C_{i_1}, \dots, C_{i_k}) = 0 \\ H(S|C_{i_1}, \dots, C_{i_m}) = H(S) \\ H(S|D_{j_1}, \dots, D_{j_m}) = H(S)}} H(S)$$

- The upper bound of the secrecy capacity:

$$C_S(n, k, d, \alpha = d, \beta = 1; l = m, m) \le \sum_{j=m+1}^{k} (d - j + 1) = L_S$$

- Both the secrecy capacity and the upper bound are identical to that of Pawar et al.[Pawar, et al.,2011].

Evaluation(upper bound)

- Finally, for the parameters of an (n, k, d, m) secure regenerating code, the upper bound of the secrecy capacity is simplifies to

$$C_S(n, k, d, \alpha = d, \beta = 1; l = m, m) \leq \sum_{j=m+1}^{k} (d - j + 1) = L_S$$

- Hence, the (n, k, d, m) secure regenerating code achieves the upper bound of the secrecy capacity because of $H(S) = L_S$.