最小バンドワイド再生成符号を用いたランプ型秘密分散法

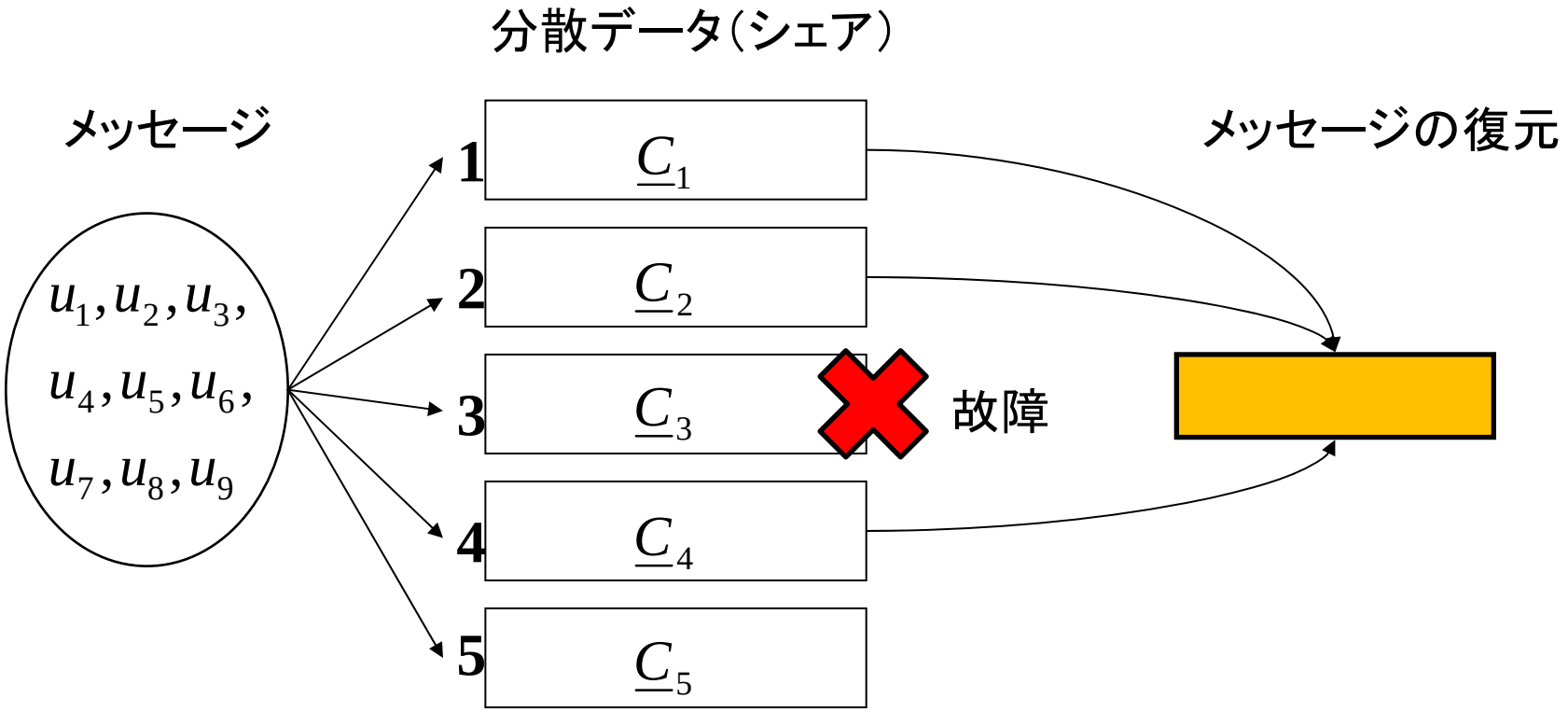
栗原正純　　桑門秀典
（電通大）　　（神戸大）

電子情報通信学会　情報セキュリティ研究会(IEICE ISEC)
大阪電気通信大学　2011/11/14

目次

1. はじめに（研究の背景、関連研究、目的）
2. 再生成符号と秘密分散法の概略
3. (n,k,d) 再生成符号（Rashmi-Shah-Kumar再生成符号）
4. (n,k,d,μ) 秘密分散法とその性能
5. 結論

分散符号化、分散保存、秘密分散



研究の背景

- 分散ストレージシステムの修復問題

  [2] S.Dimakis, P.B.Godfrey, Y.Wu, M.J.Wainwright, K.Ramchandran, "Network Coding for Distributed Storage System," 2010

- メッセージの分散符号化

  1. 復元の機能(メッセージの復元) :データ復元の信頼性
  2. 修復の機能(故障ノード ✖ の修復): システムの維持

     ⇒ 分散データ(シェア)の再生成

  ⇒ **再生成符号**(Regenerating Codes)

- 再生成符号を用いた場合、どのような秘密分散法を実現できるのか？

関連研究と目的

（再生成符号を用いた秘密分散法）

パラメータ (k, d)

1. 最小バンドワイド再生成符号を用いた秘密分散法

   [3] S.Paware, S.E.Rouayheb, K.Ramchandra,
   ”On secure Distributed Data Storage Under Repair Dynamics,” 2010

   [6] 栗原, 桑門, “修復可能な分散ストレージシステムにおける
   ランプ型秘密分散法,” 2011

→ $k = d$

2. 最小ストレージ再生成符号を用いた秘密分散法

   [4]栗原、桑門“分散ストレージシステムにおける
   再生成符号と秘密分散について”,2011

   [7] Kuwakado and Kurihara,
   “Computationally-Secure Regenerating Code,” 2011

→ $k,$ $d = 2k - 2$

→ 計算量的に安全な再生成符号（CS-R符号）

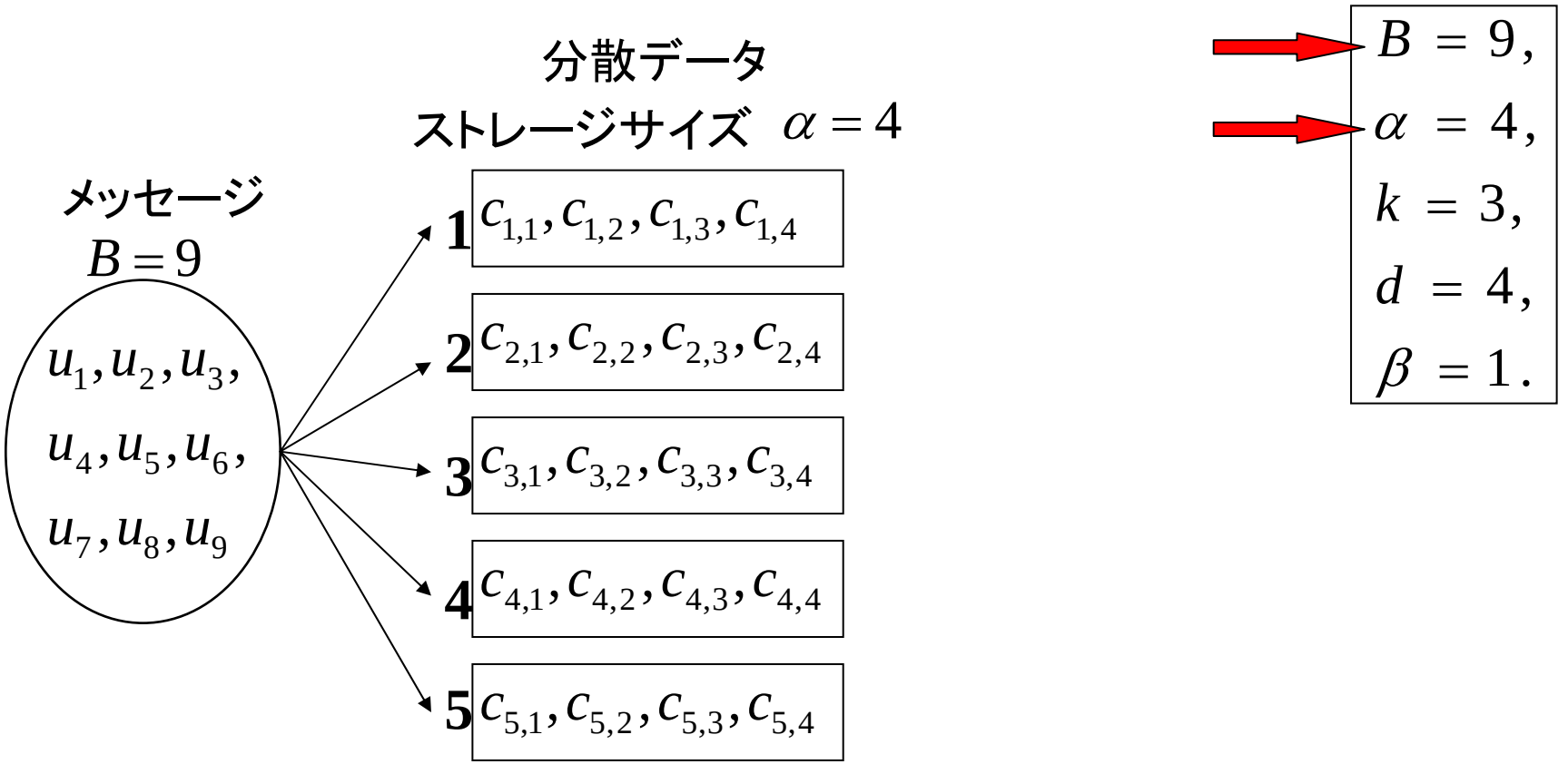
（追加資料あり）

- 本論文の目的

  最小バンドワイド再生成符号を用いた秘密分散法
  （[6]の符号の拡張版の提案）

→ $k \leq d$ 独立に選択

分散ストレージシステム（分散保存）：データ復元の安全性、信頼性の確保



メッセージ

$u_1, u_2, u_3,$

$u_4, u_5, u_6,$

u_7, u_8, u_9

分散符号化

ノード i の分散データ

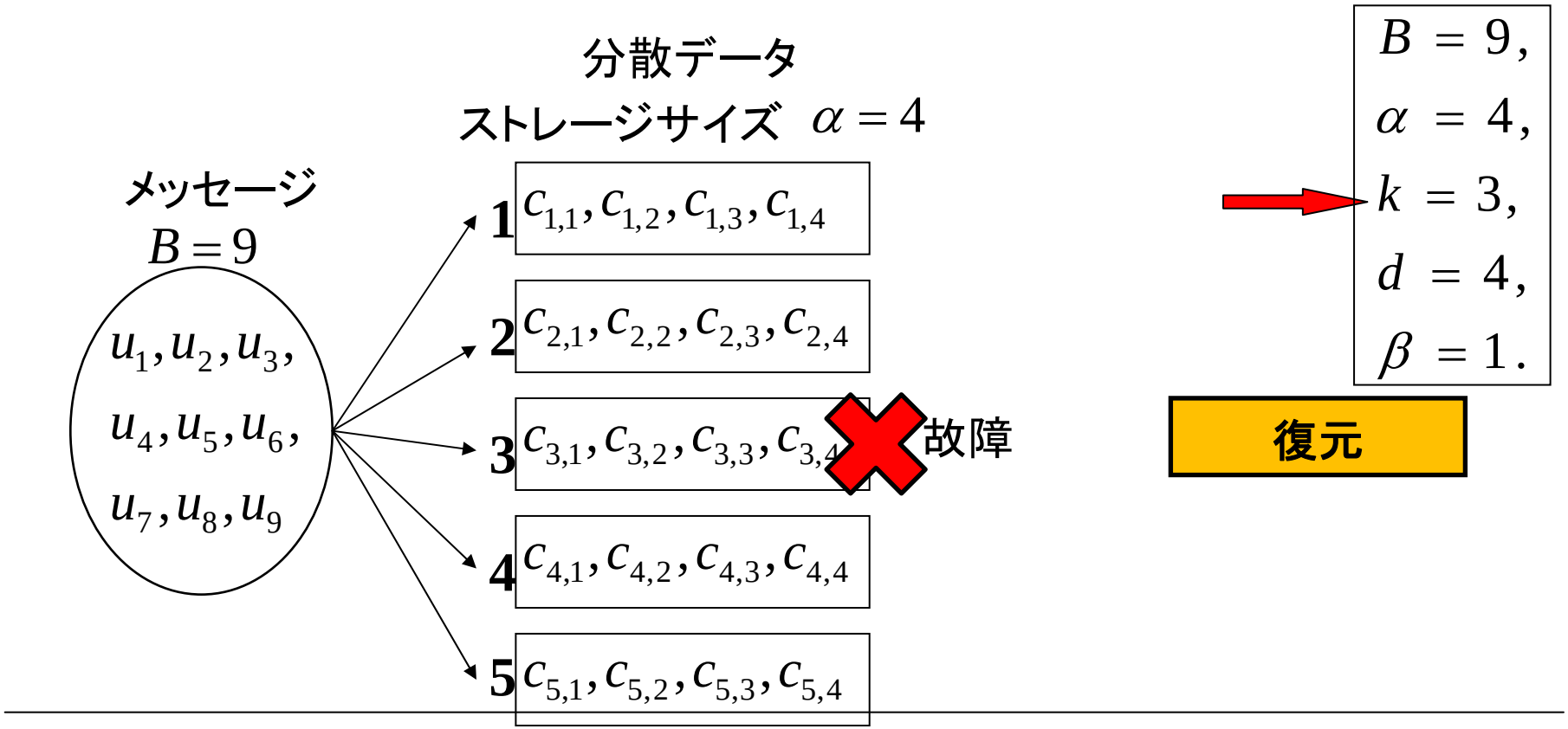
$\underline{C}_i = (c_{i,1}, c_{i,2}, c_{i,3}, c_{i,4})$

有限体

$u_k, c_{j,k} \in GF(q)$

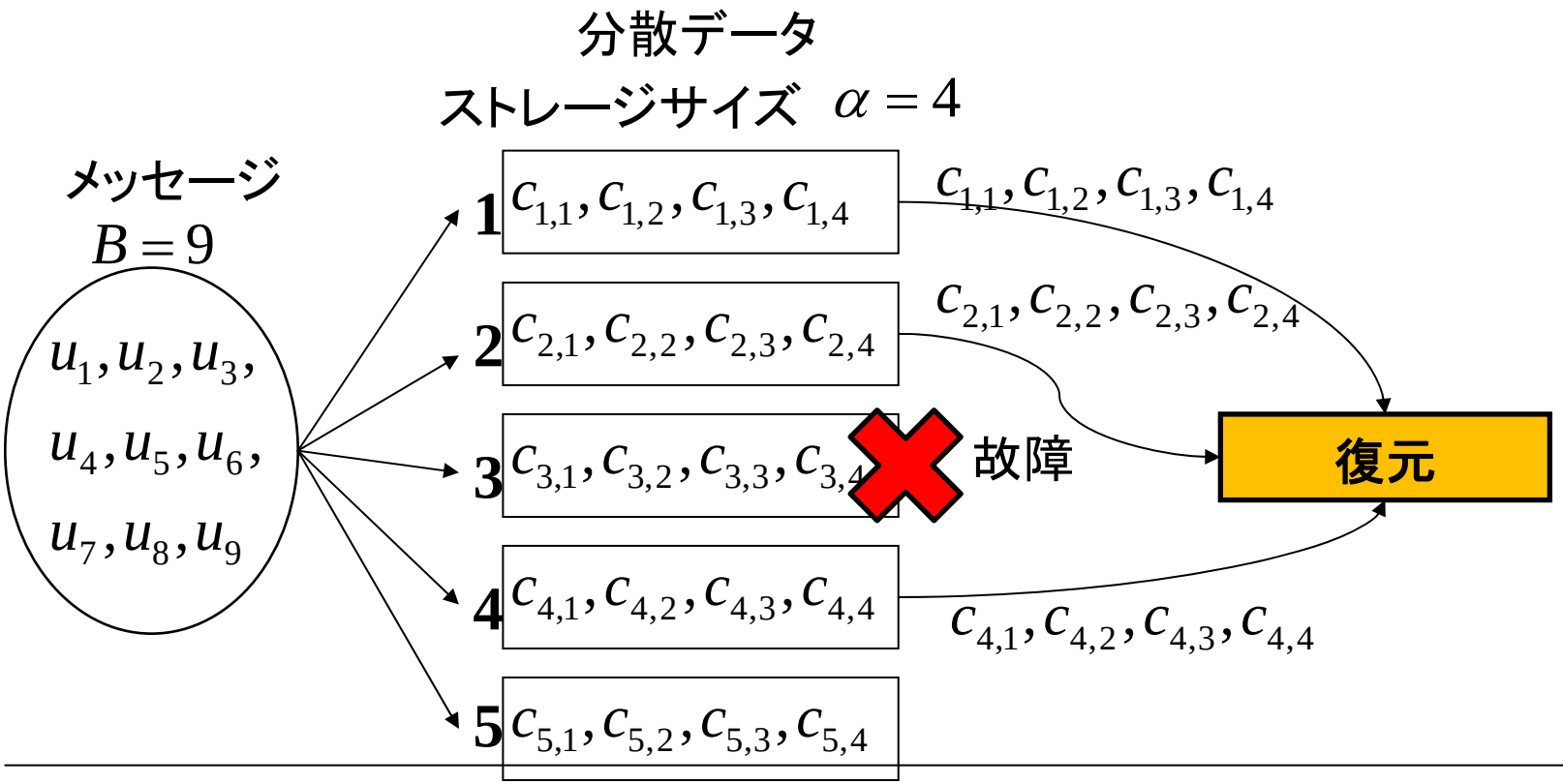
メッセージの復元

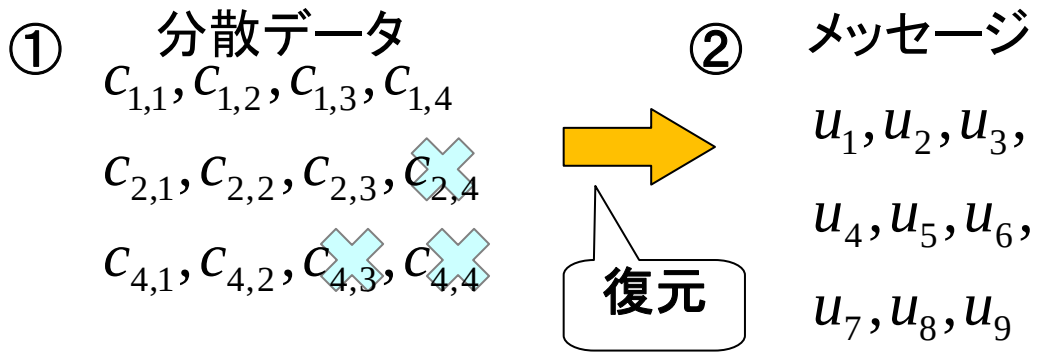
復元のためにアクセスするノード数 $k = 3$



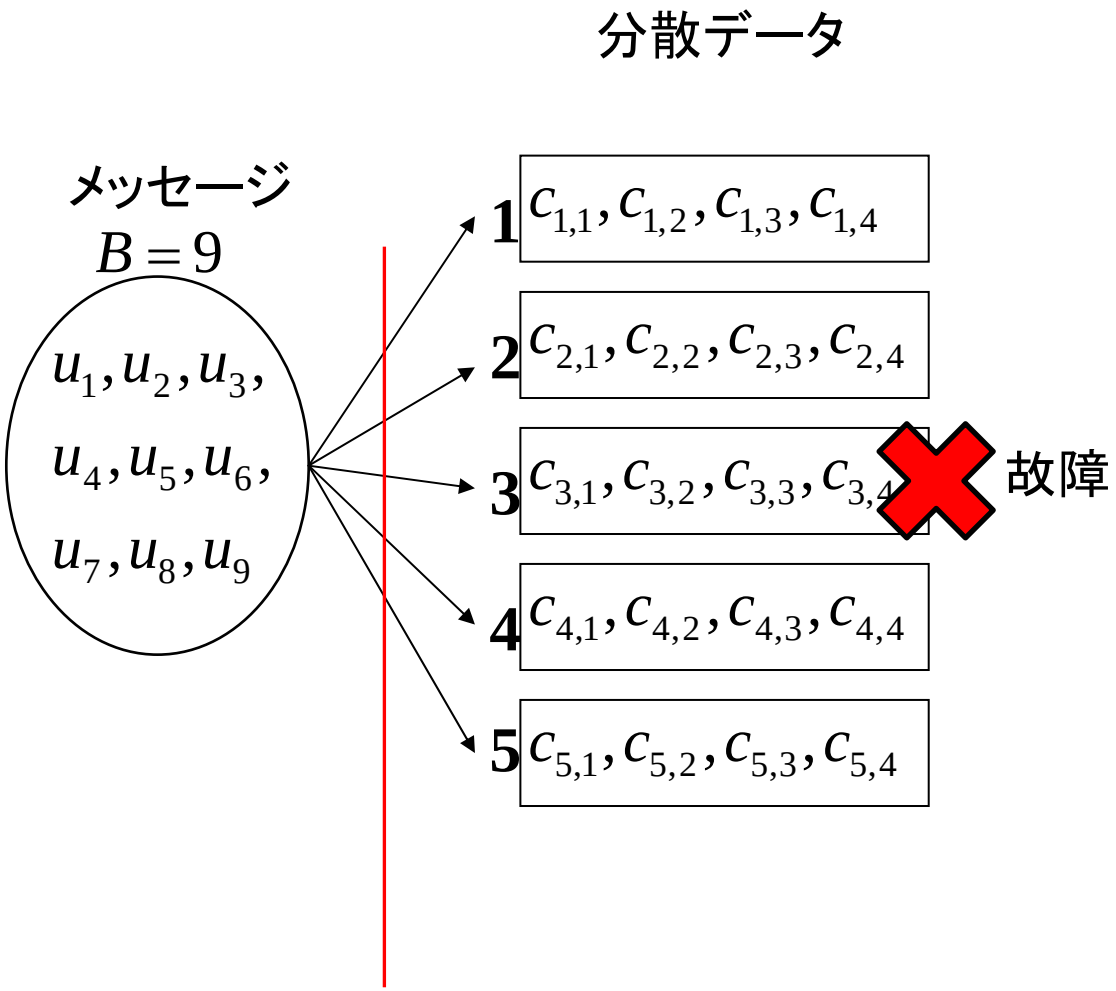
メッセージの復元

復元のためにアクセスするノード数　$k = 3$





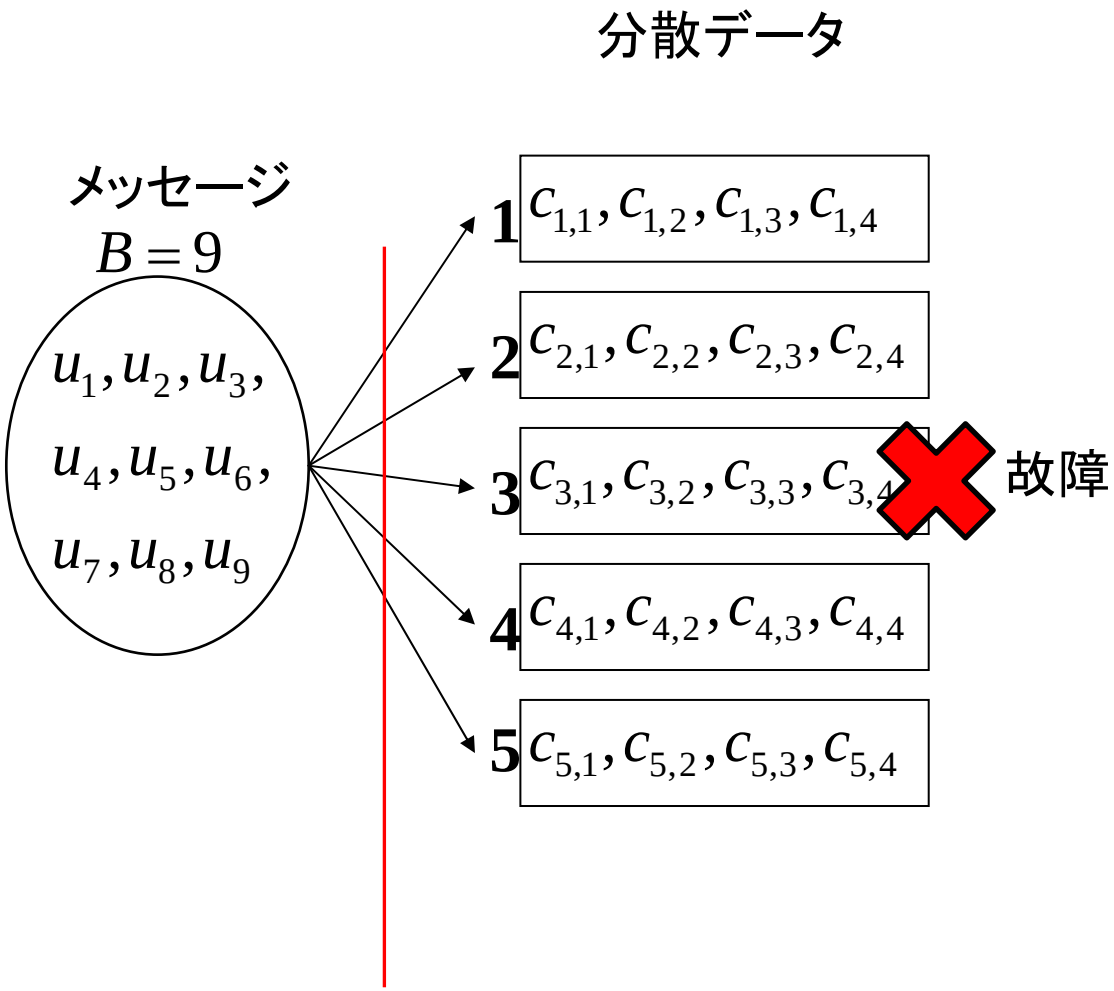
故障ノードの修復問題(システムの維持)



システムの維持のために、故障ノードを効率良く修復したい

「アクセスできない」と仮定する

故障ノードの修復問題(システムの維持)



「アクセスできない」と仮定する

故障ノードの修復の手順

1. 故障ノードを新しいノードに置き換える。

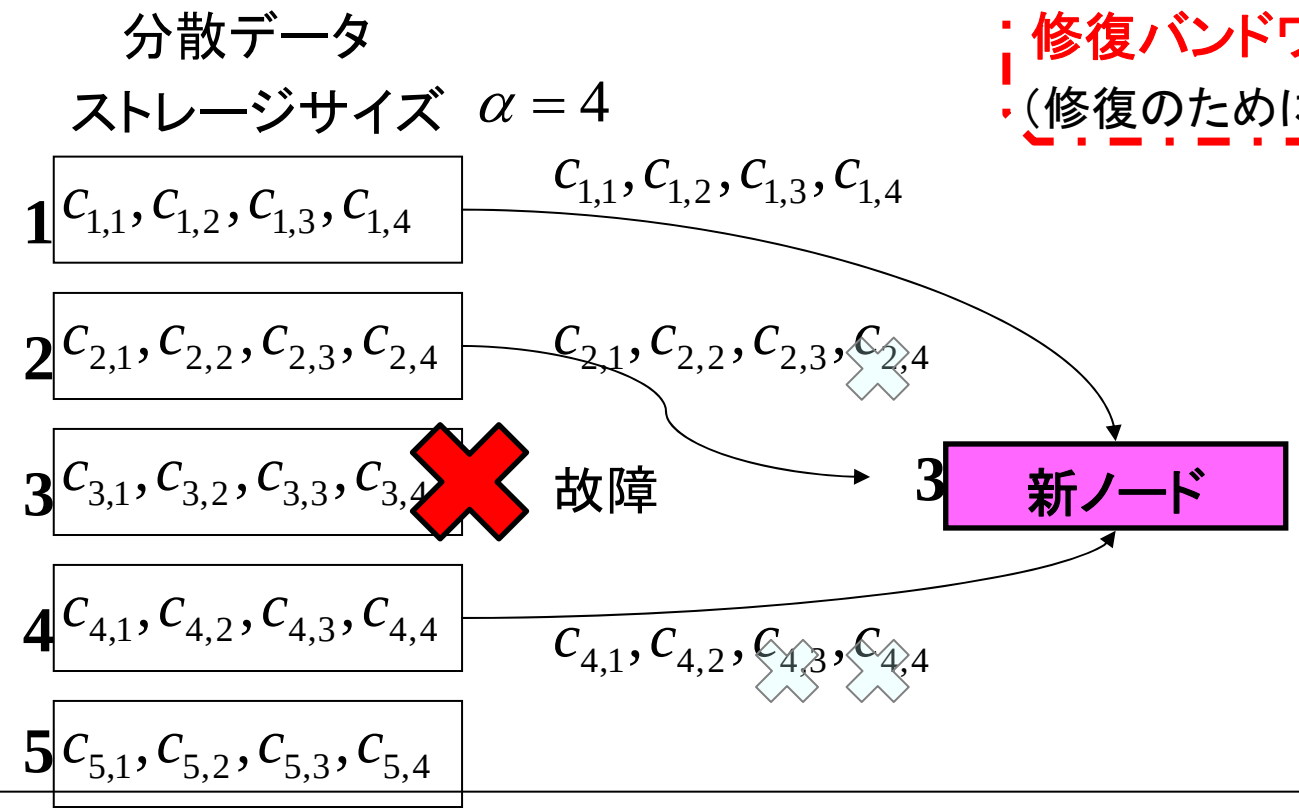
2. そして、故障ノードが保存していた分散データを効率良く複製し、保存したい。

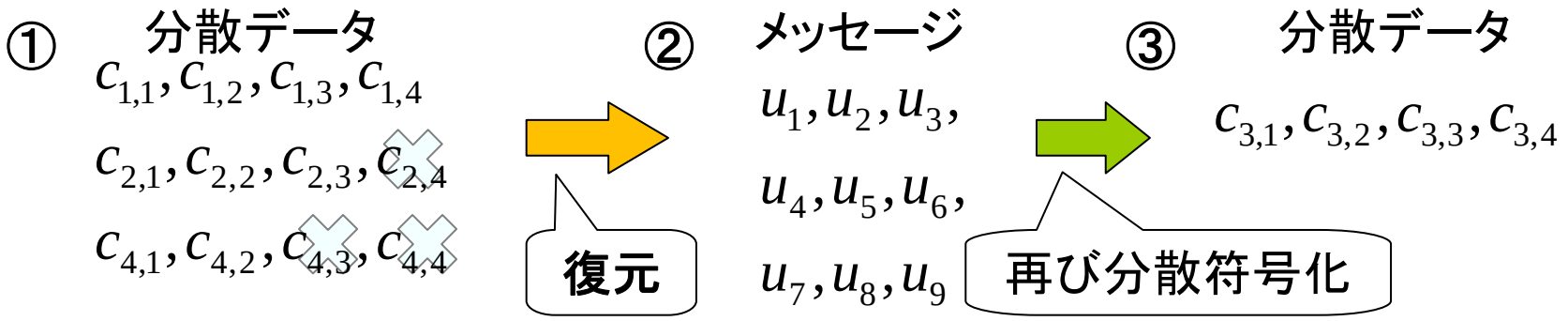
(分散データの再生成)

自明な方法（復元を利用した修復）

修復のためにアクセスするノード数 $d = 3$

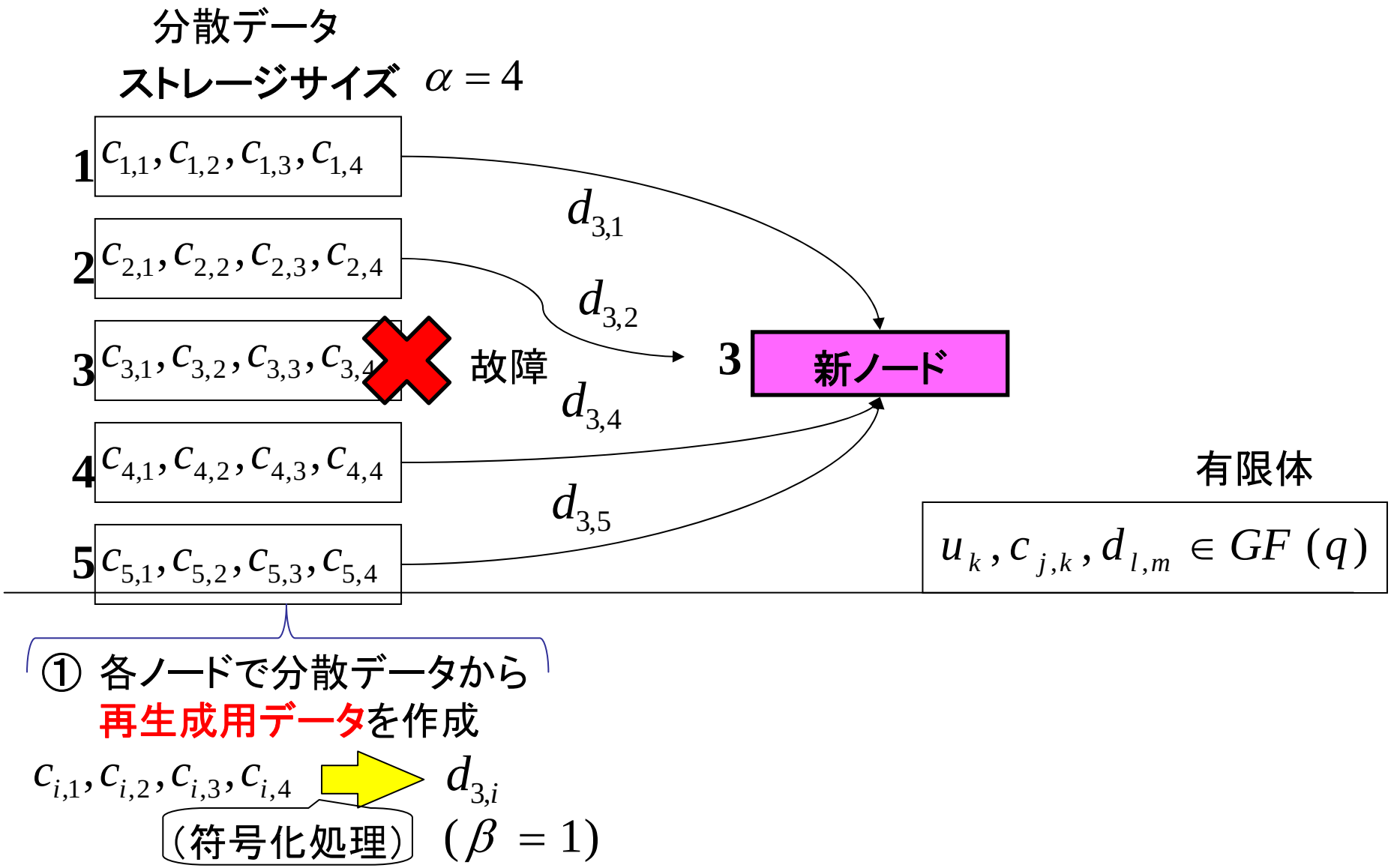
修復バンドワイド　9
（修復のために集めたデータサイズ）





自明でない方法（再生成符号を利用した修復）

修復のためにアクセスするノード数 $d = 4$



① 各ノードで分散データから
再生成用データを作成

$c_{i,1}, c_{i,2}, c_{i,3}, c_{i,4}$ ⇒ $d_{3,i}$

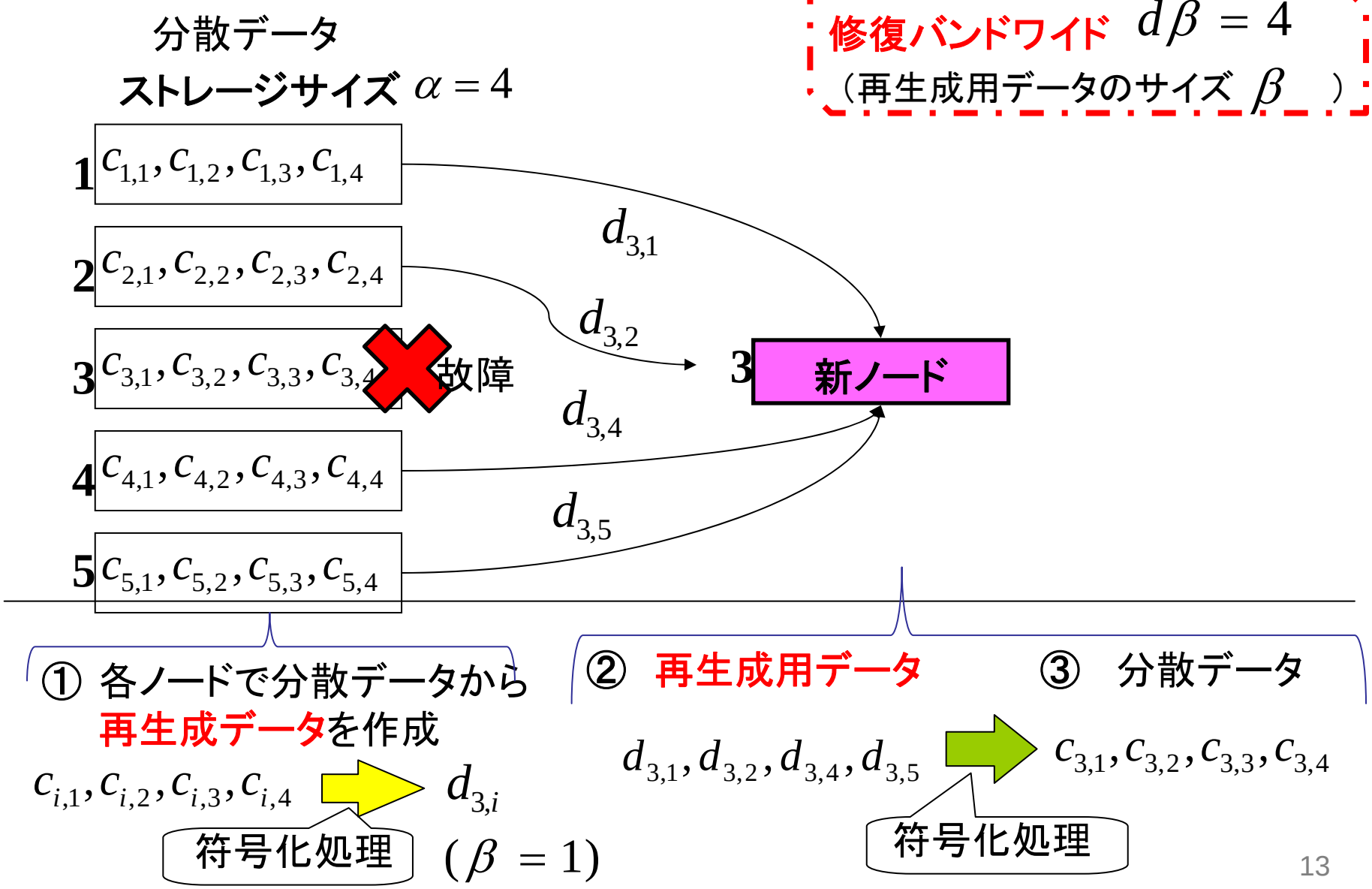
（符号化処理） $(\beta = 1)$

自明でない方法（再生成符号を利用した修復）

修復のためにアクセスするノード数 $d = 4$

修復バンドワイド $d\beta = 4$

（再生成用データのサイズ β ）



13

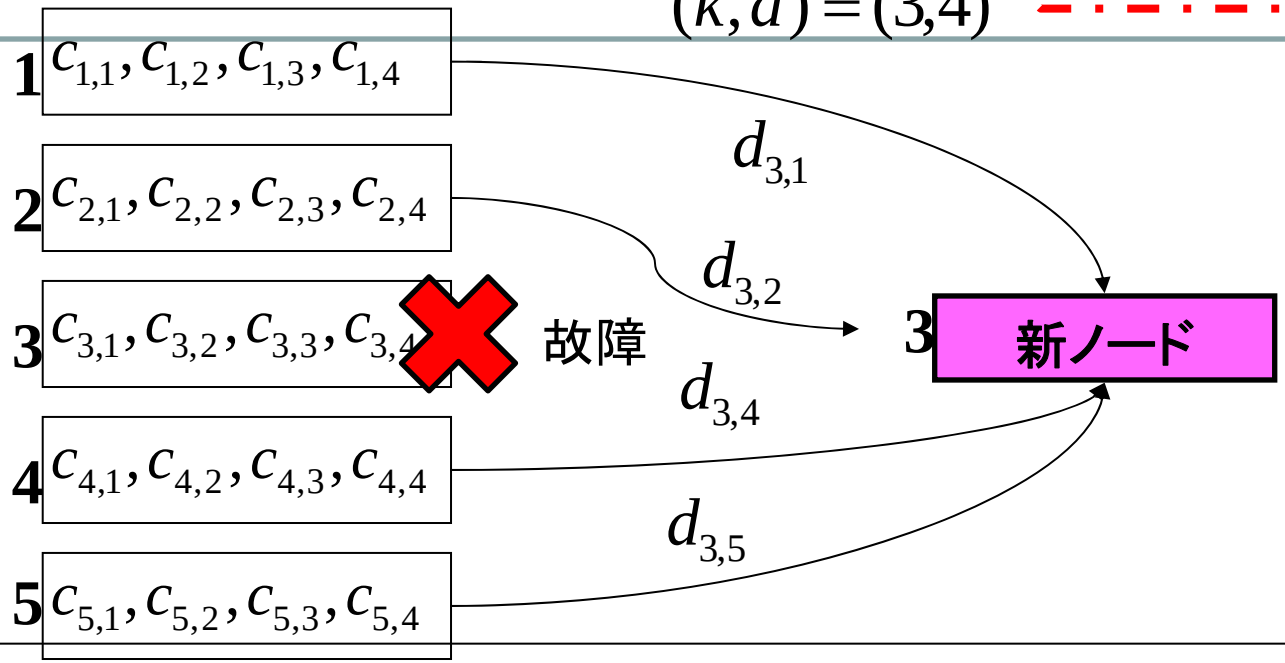
自明でない方法（再生成符号を利用した修復）

分散データ
ストレージサイズ $\alpha = 4$

トレード・オフ

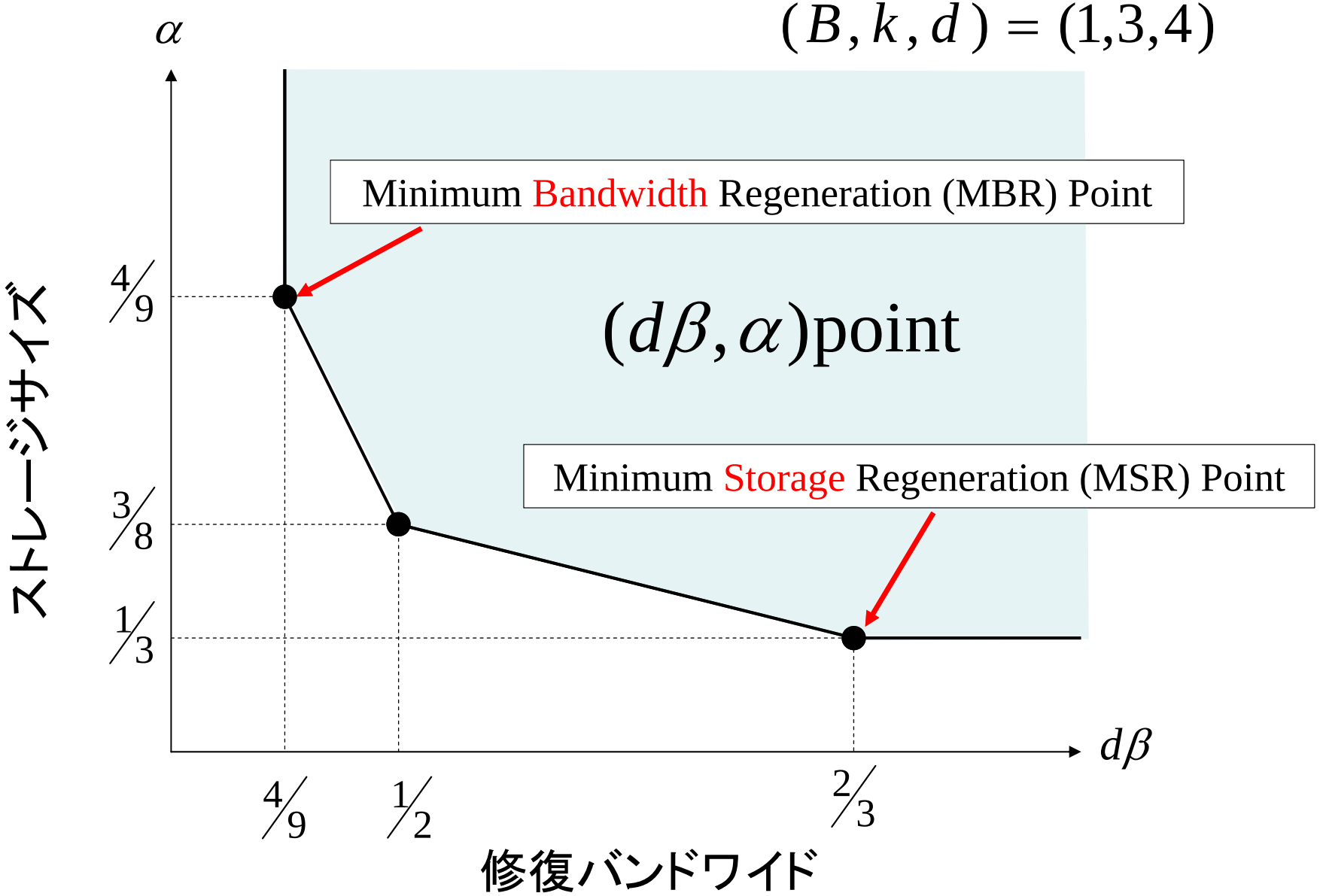
$(k,d) = (3,4)$

修復バンドワイド $d\beta = 4$
（再生成用データのサイズ β ）



修復バンドワイド $d\beta$ とストレージサイズ α のトレード・オフ[1]

$(B, k, d) = (1,3,4)$



再生成符号(復元と再生成)

- ストレージサイズと修復バンドワイドのトレード・オフ

1. 修復バンドワイドを最小
   →最小バンドワイド再生成(MBR)符号
   （Minimum Bandwidth Regenerating (MBR) Code）
2. ストレージサイズを最小
   →最小ストレージ再生成(MSR)符号
   （Minimum Storage Regenerating(MSR) Code）

- Rashmi-Shah-Kumar再生成符号

[3] K.V.Rashmi, N.B.Shah, P.V.Kumar,
"Optimal Exact-Regenerating Codes for Distributed Storage at the MSR point and MBR point via a Product-Matrix Construction," 2010

一般的な符号の構成法を示す。

次に、再生成符号を用いた秘密分散法（情報理論的な安全性）

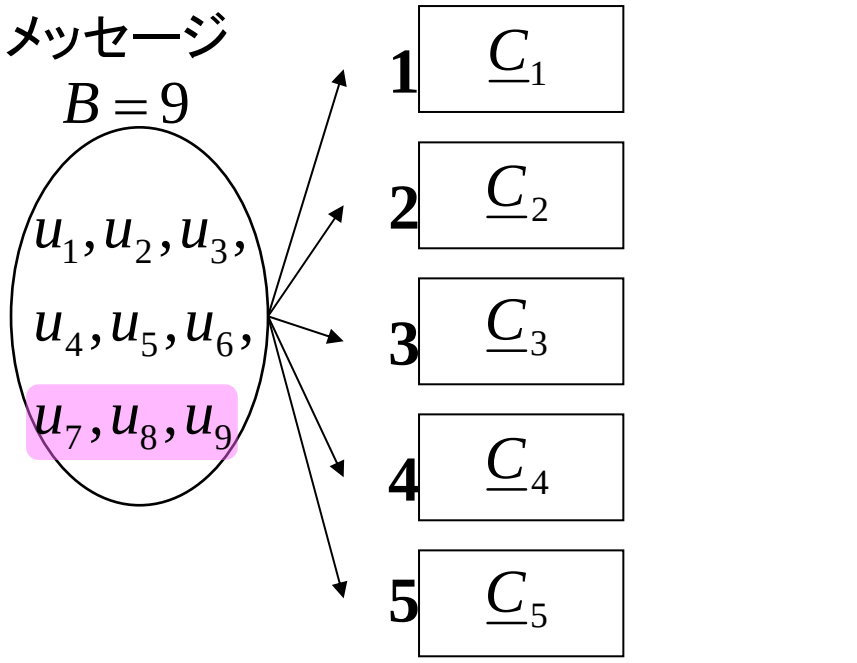
$$(k, d, \mu) = (3,4,1)$$

秘密情報: $\underline{S} = (u_1, u_2, \cdots, u_6) : 6$

乱数: $\underline{R} = (u_7, u_8, u_9) : 3$

分散データ:

$$\underline{C}_i = (c_{i,1}, c_{i,2}, c_{i,3}, c_{i,4})$$



分散データ(再生成符号を用いた秘密分散法)

$(k, d, \mu) = (3,4,1)$

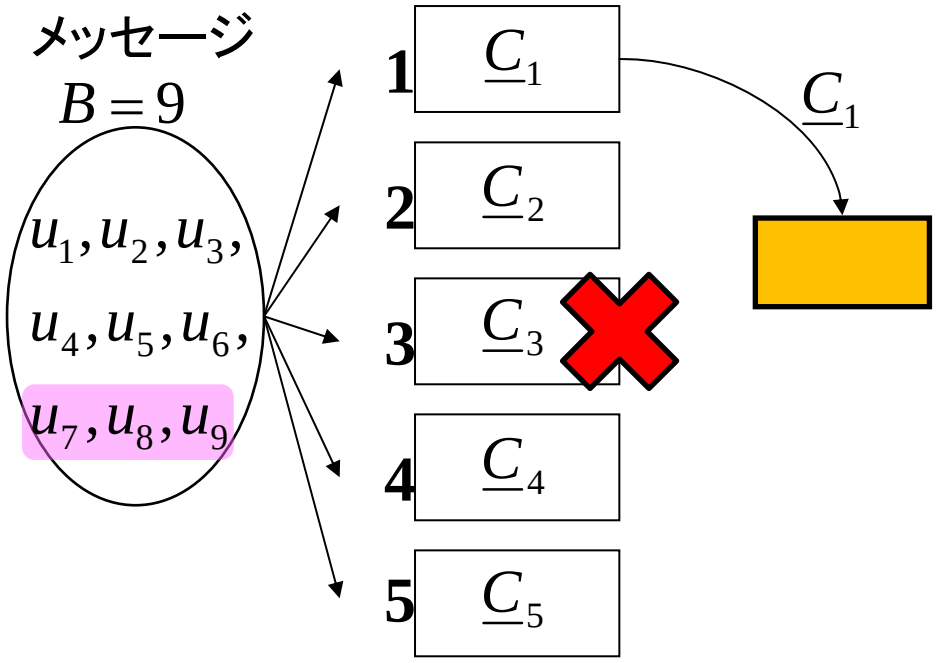
秘密情報: $\underline{S} = (u_1, u_2, \cdots, u_6) : 6$

乱数: $\underline{R} = (u_7, u_8, u_9) : 3$

集めた分散データの個数

分散データ:

$\underline{C}_i = (c_{i,1}, c_{i,2}, c_{i,3}, c_{i,4})$



#1 $H(\underline{S} \mid \underline{C}_i) = H(\underline{S})$ $(\mu = 1)$

#2 $H(\underline{S} \mid \underline{C}_i \underline{C}_j) = \frac{1}{3} H(\underline{S})$

#3 $H(\underline{S} \mid \underline{C}_i \underline{C}_j \underline{C}_m) = 0$ $(k = 3)$

再生成用データ(再生成符号を用いた秘密分散法)

$(k, d, \mu) = (3,4,1)$

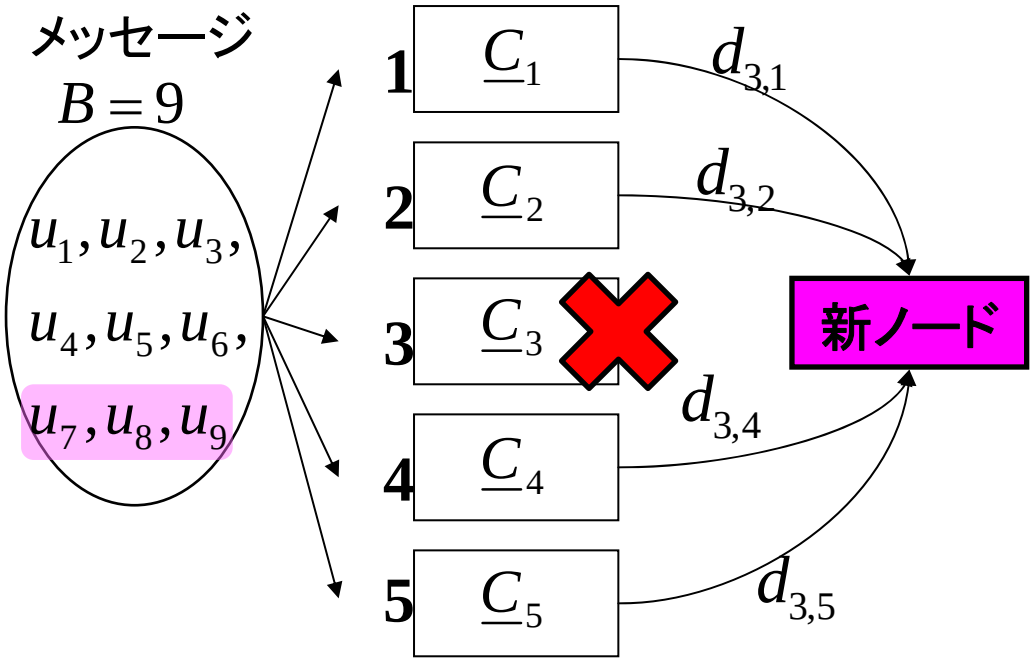
秘密情報 : $\underline{S} = (u_1, u_2, \cdots, u_6) : 6$

乱数 : $\underline{R} = (u_7, u_8, u_9) : 3$

$$H(\underline{S} \mid \underline{D}_i) = H(\underline{S})$$

分散データ:

$\underline{C}_i = (c_{i,1}, c_{i,2}, c_{i,3}, c_{i,4})$



再生成用データ

$\underline{D}_3 = (d_{3,1}, d_{3,2}, d_{3,4}, d_{3,5})$

に対し、

$$H(\underline{S} \mid \underline{D}_3) = H(\underline{S})$$

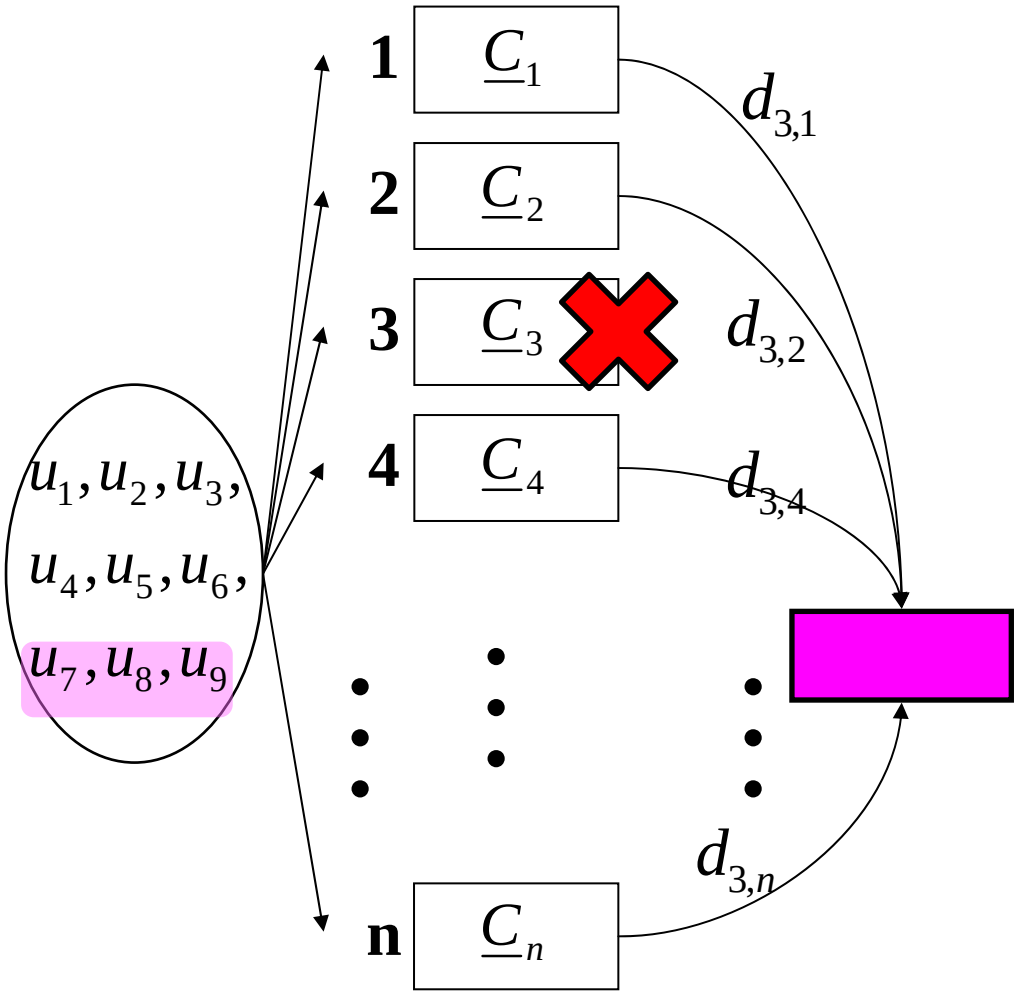
が成り立つ。

再生成符号を用いると、秘密情報の情報がまったくもれることなく、分散データを再生成できる。

さらに、

再生成用データ（再生成符号を用いた秘密分散法）

$(k, d, \mu) = (3,4,1)$



$$H(\underline{S} \mid \underline{D}_i) = H(\underline{S})$$

$$\underline{D}_3 = (d_{3,1}, d_{3,2}, d_{3,4}, d_{3,5})$$

（ならば）

$$H(\underline{S} \mid d_{3,1} d_{3,2} \cdots d_{3,n}) = H(\underline{S})$$

4個より多くの再生成用データを集めたとしても、秘密情報の情報は、まったくもれない。

自明な方法である復元を利用した修復の場合

$(k, d, \mu) = (3,4,1)$

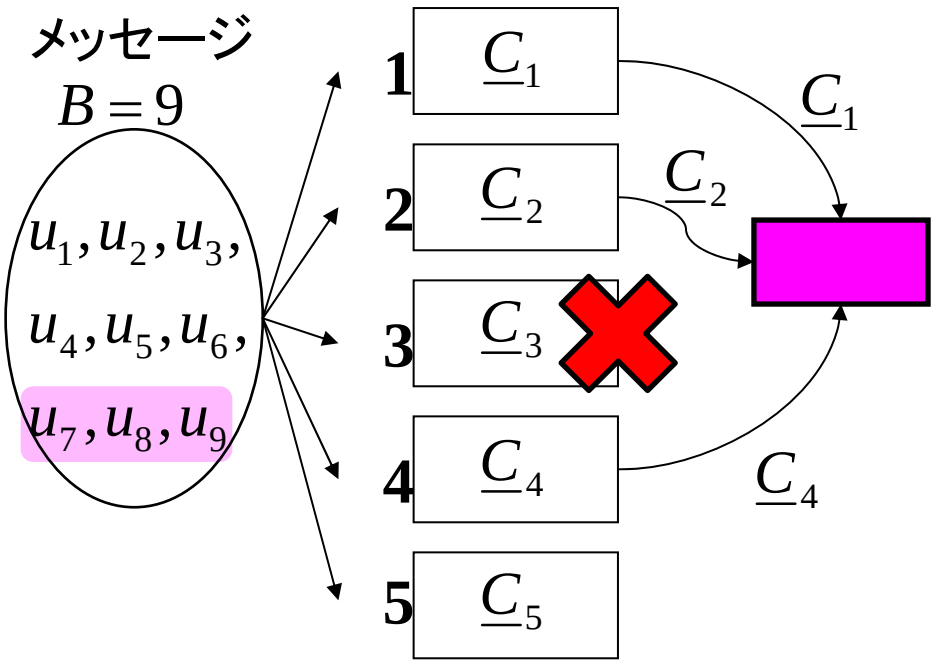
秘密情報 : $\underline{S} = (u_1, u_2, \cdots, u_6) : 6$

乱数 : $\underline{R} = (u_7, u_8, u_9) : 3$

$$H(\underline{S} \mid \underline{C}_i \underline{C}_j \underline{C}_m) = 0$$

分散データ:

$\underline{C}_i = (c_{i,1}, c_{i,2}, c_{i,3}, c_{i,4})$



① 分散データ $\underline{C}_1, \underline{C}_2, \underline{C}_4$

復元

② メッセージ $u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8, u_9$

再び分散符号化

③ ノード3の分散データ

$C_3 = (c_{3,1}, c_{3,2}, c_{3,3}, c_{3,4})$

(n,k,d) 最小バンドワイド再生成(MBR)符号の設定[3]

Minimum Bandwidth Regenerating (MBR) Code

- **パラメータ** (n,k,d,α,β,B)

n: ストレージノードの個数（分散データの個数）

k: 復元のためにアクセスするノード数

d: 修復のためにアクセスするノード数

$k \le d$ 独立に選択

α: 分散データのサイズ

β: 再生成用データのサイズ　$\beta = 1$

B: メッセージのサイズ

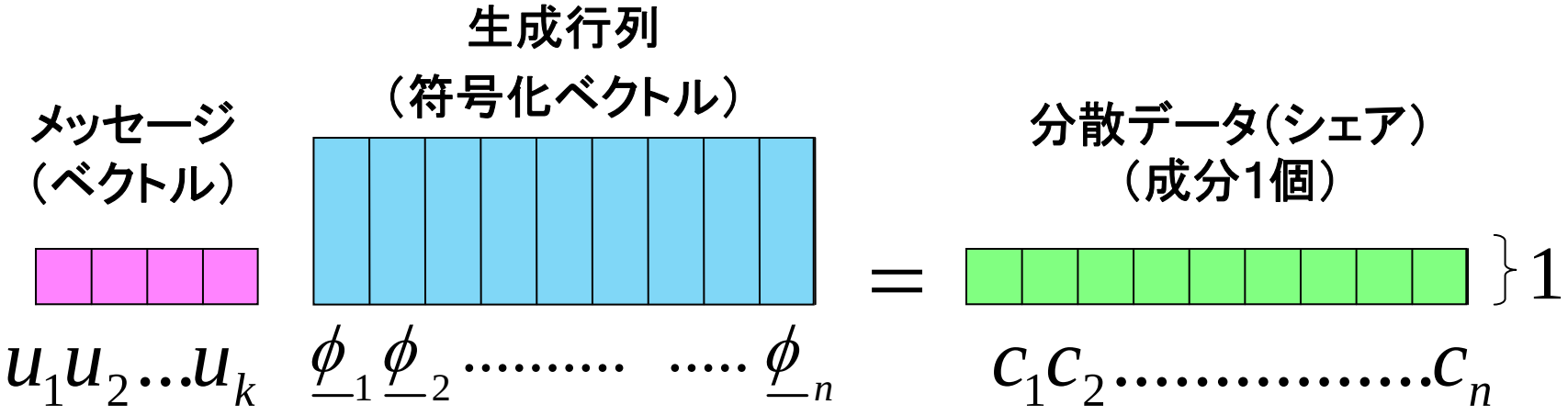
- **パラメータの設定**

$$\alpha = d,$$
$$B = k(2d-k+1)/2,$$
$$\beta = 1,$$
$$n > |GF(q)|$$

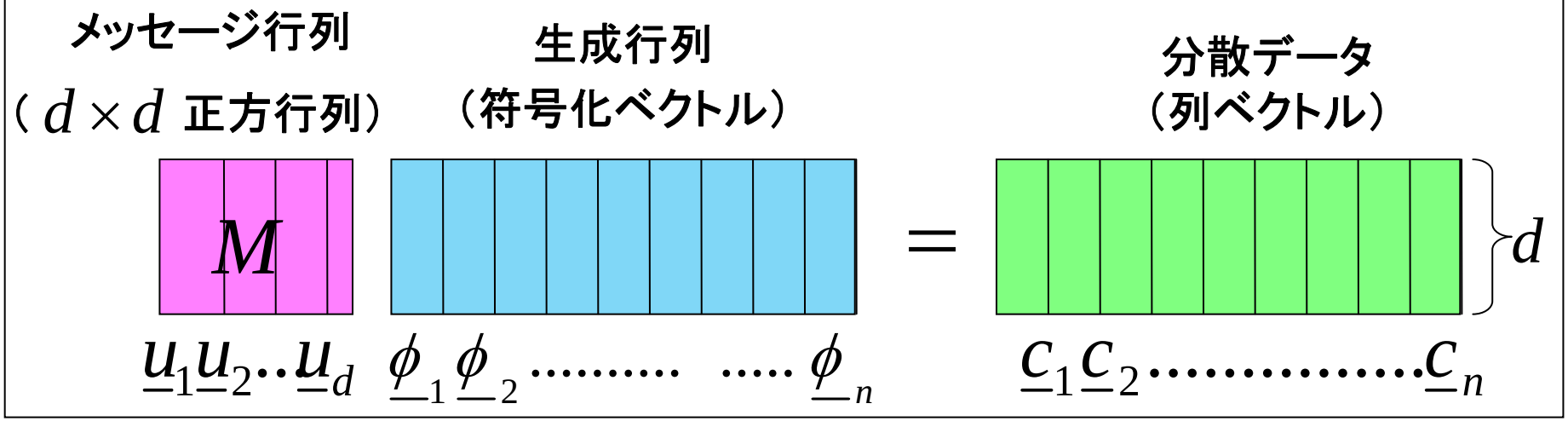
（ k,d によって定まる）

MBR 符号の符号化の様子

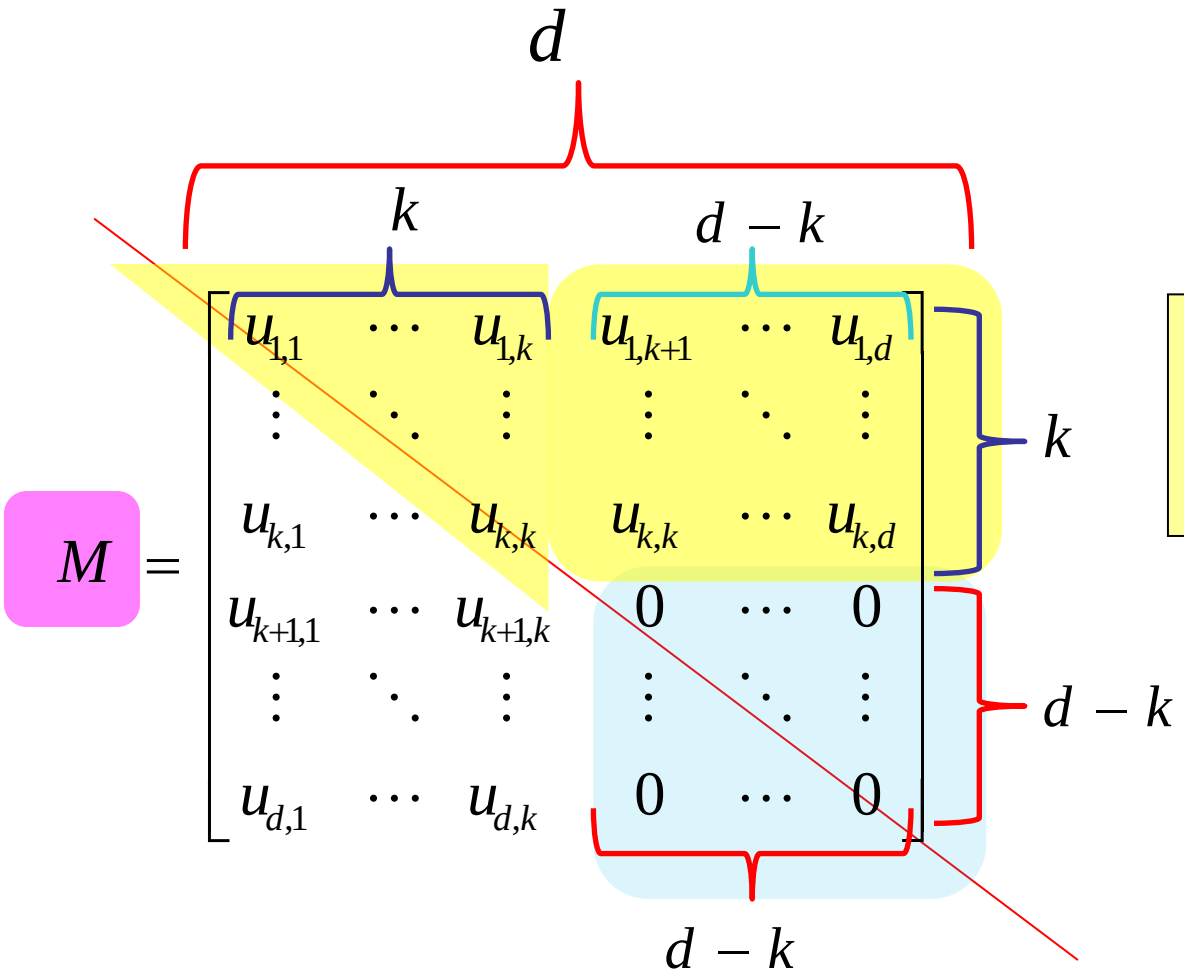
MDS 符号の場合



MBR 符号の場合



(n, k, d) MBR符号のメッセージ行列(対称行列) [3]



$k \leq d$

独立に選択

メッセージサイズ

$$B = k(2d - k + 1)/2$$

（追加資料あり）

(n,k,d) MBR符号を用いた (n,k,d,μ) ランプ型秘密分散法

1. MBR符号の設定：パラメータ　(n,k,d)
2. 秘密分散法の設定：パラメータ　μ　$(\mu < k)$

$\mu \Rightarrow$

- 秘密情報シンボル列 $\underline{S} \in GF(q)^{L_S}$, $L_S = H(\underline{S})$
- 乱数シンボル列 $\underline{R} \in GF(q)^{L_R}$,

安全性

任意の μ 個の分散データ $\underline{C}_1, \cdots, \underline{C}_\mu$ に対し、

$$H(\underline{S} \mid \underline{C}_1 \cdots \underline{C}_\mu) = H(\underline{S})$$

任意の k 個の分散データ $\underline{C}_1, \cdots, \underline{C}_k$ に対し、

$$H(\underline{S} \mid \underline{C}_1 \cdots \underline{C}_k) = 0$$

(n,k,d,μ) ランプ型秘密分散法の設定

メッセージ行列に 秘密情報 と 乱数 のシンボルを以下のように設定する

$$(k,d,\mu)=(4,6,2)$$

$d=6$

$k=4$

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

秘密情報シンボル

2

1

乱数シンボル

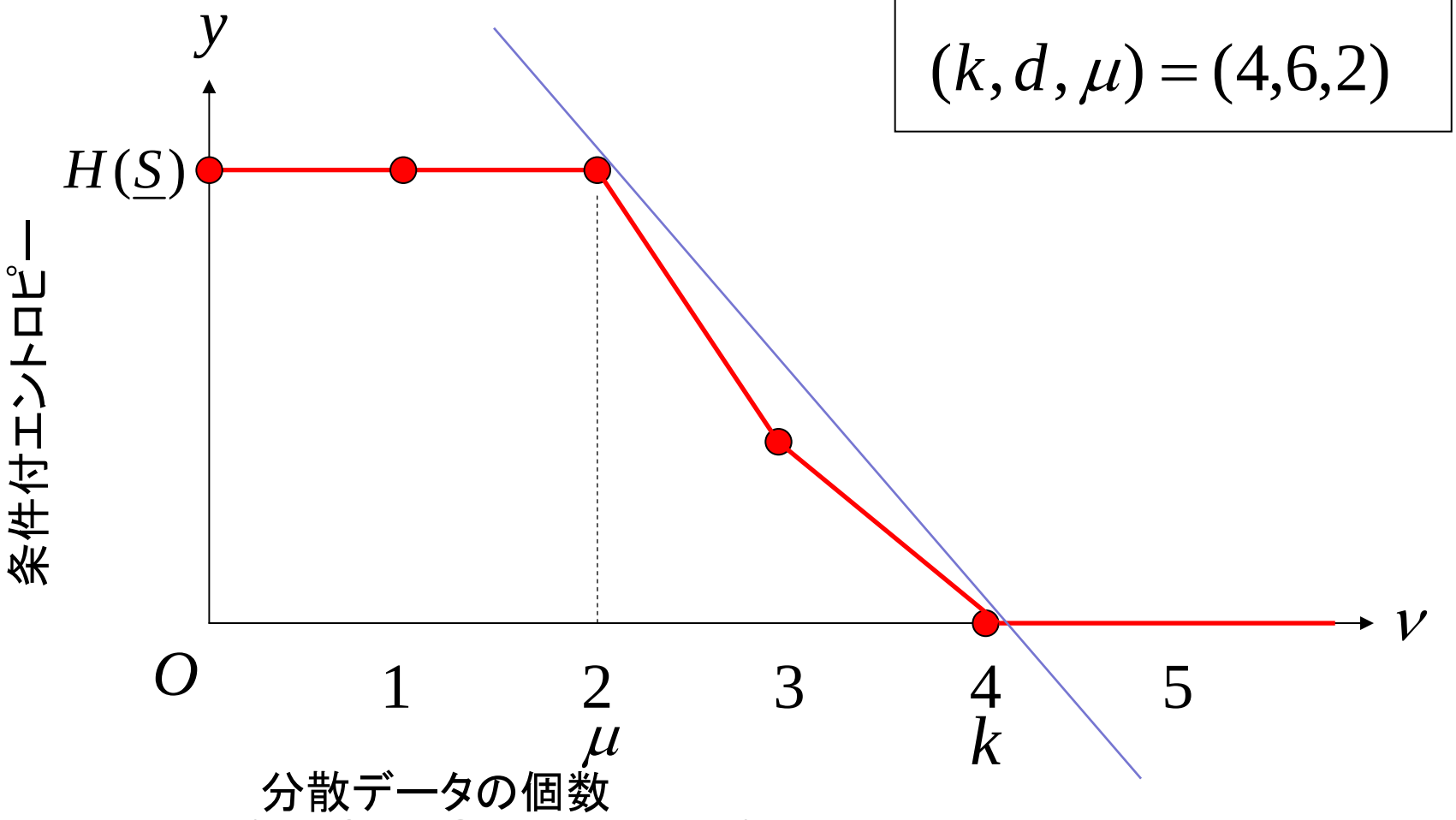
$(\mu=2)$

(n,k,d,μ) ランプ型秘密分散法の性能

分散データが与えられた下での秘密情報の条件付エントロピー

$$y = H(\underline{S} \mid \underline{C}_1 \cdots \underline{C}_v)$$

$(k,d,\mu)=(4,6,2)$



(n,k,d,μ) ランプ型秘密分散法の性能(1/2)

[定理 9] 任意の ν 個の **分散データ** $\underline{C}_1,\cdots,\underline{C}_\nu$ に対し、
以下が成り立つ:

$$H(\underline{S}|\underline{C}_1\ \cdots\ \underline{C}_\nu) = \begin{cases} H(\underline{S}), & (0 \le \nu \le \mu) \\ \dfrac{(\nu-k)(\nu-2d+k-1)}{2L_S}H(\underline{S}), & (\mu+1 \le \nu \le k-1) \\ 0, & (k \le \nu) \end{cases}$$

[定理 11] 任意の ν 個の **再生成用データ** $\underline{D}_1,\cdots,\underline{D}_\nu$ に対し、
以下が成り立つ:

$$H(\underline{S}|\underline{D}_1\ \cdots\ \underline{D}_\nu) = \begin{cases} H(\underline{S}), & (0 \le \nu \le \mu) \\ \dfrac{(\nu-k)(\nu-2d+k-1)}{2L_S}H(\underline{S}), & (\mu+1 \le \nu \le k-1) \\ 0, & (k \le \nu) \end{cases}$$

（追加資料あり）

(n,k,d,μ) ランプ型秘密分散法の性能(2/2)

[定理 12] n 個の再生成用データ $d_{f,1}, d_{f,2}, \cdots, d_{f,n}$ に対し、

以下が成り立つ:

$$\mu \geq 1 \Rightarrow H(\underline{S} \mid \underline{D}_f) = H(\underline{S})$$

$$\underline{D}_f = (d_{f,i}, d_{f,j}, d_{f,m}, d_{f,p})$$

（ならば）

$$H(\underline{S} \mid d_{f,1}, d_{f,2}, \cdots, d_{f,n}) = H(\underline{S})$$

結論

- 再生成符号とそれを利用した秘密分散法について説明をした。
- (n,k,d) MBR符号を用いた (n,k,d,μ) ランプ型秘密分散法の構成法とその性能を示した。

構成： メッセージ行列の設定（[6]の拡張）

復元のためにアクセスするノード数 k

修復のためにアクセスするノード数 d

$$k \le d$$

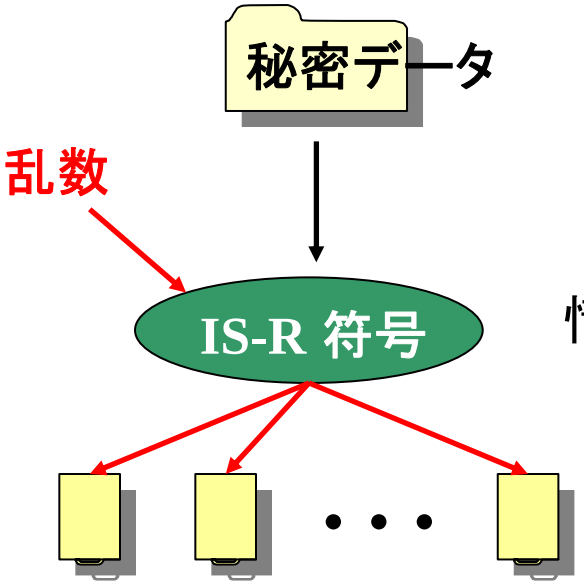
性能（安全性） （Secrecy Capacity を達成する最適な符号）

$$H(\underline{S}|\underline{C}_1 \cdots \underline{C}_\nu) = \begin{cases} H(\underline{S}), & (0 \le \nu \le \mu) \\ \dfrac{(\nu-k)(\nu-2d+k-1)}{2L_S} H(\underline{S}), & (\mu+1 \le \nu \le k-1) \\ 0, & (k \le \nu) \end{cases}$$

追加資料(additional slides)

1. 計算量的に安全な再生成符号
2. 再生成符号の復元と修復（再生成）
3. 複数の再生成用データに対する性能

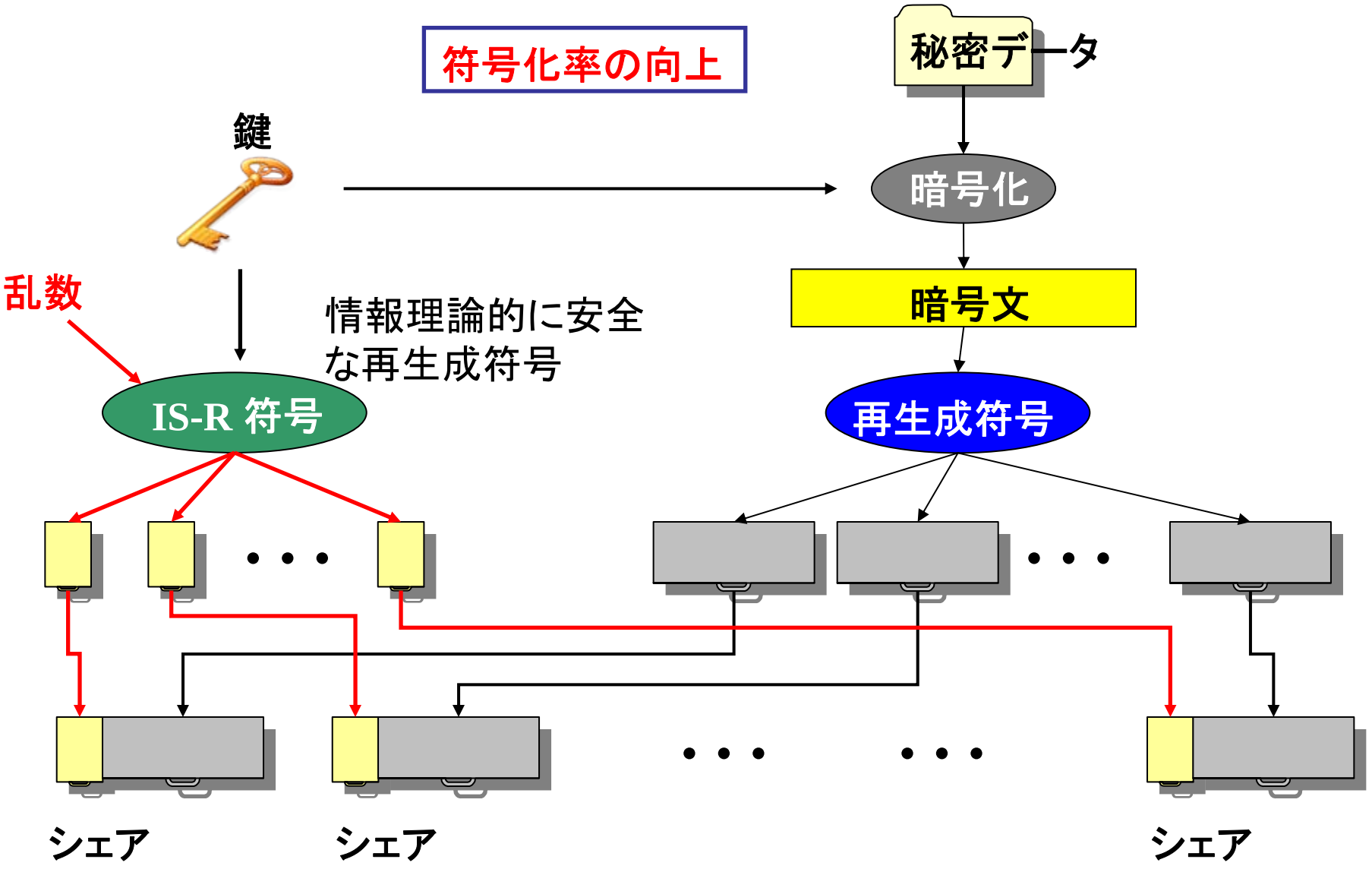
計算量的に安全な再生成符号(CS-R符号)[7]



情報理論的に安全な再生成符号(IS-R 符号)

Credit: [7]Kuwakado and Kurihara , “Computationally-Secure Regenerating code,” CSS2011

計算量的に安全な再生成符号(CS-R符号)[7]



Credit: [7]Kuwakado and Kurihara , “Computationally-Secure Regenerating code,” CSS2011

符号化ベクトル と 分散データ（MBR符号[3]）

1. ノード $i \in \{1,2,\cdots,n\}$ の符号化ベクトル

$$\phi_i = (1, x_i, x_i^2, x_i^3, \cdots, x_i^{d-1})^t \in GF(q)^d$$

ただし、$x_i \in GF(q), \quad x_i \neq x_i \quad if \quad i \neq j.$

2. ノード i の分散データ

$$\underline{C}_i = (c_{i,1}, c_{i,2}, c_{i,3}, \cdots, c_{i,d})^t = M \phi_i \in GF(q)^d$$

メッセージの復元（MBR符号[3]）

1. 任意の k 個のノードにアクセスし、 k 個の分散データを集めることで復元できる。

2. しかし、Rashmiら[3]のオリジナルの手法では、サイズ kd のデータを集める方法である。

$$kd \geq \frac{k(2d-k+1)}{2} = B \text{ (メッセージサイズ)}$$

（ $k = 1$ のとき、そのときのみ等号が成り立つ。）

3. 本研究の結果から、サイズ B のデータを集めるだけで復元可能である。

故障ノードの修復(再生成)（MBR符号[3]）

故障ノード $f \in \{1,2,\cdots,n\}$ の修復：

故障していない任意の d 個のノード $h_1, h_2, \cdots, h_d$ にアクセスし、再生成用データ

$$d_{f,hp} = \underline{C}_{hp}^{t} \phi_f \in GF(q), \quad 1 \le p \le d$$

をダウンロードし、再生成用データのベクトル

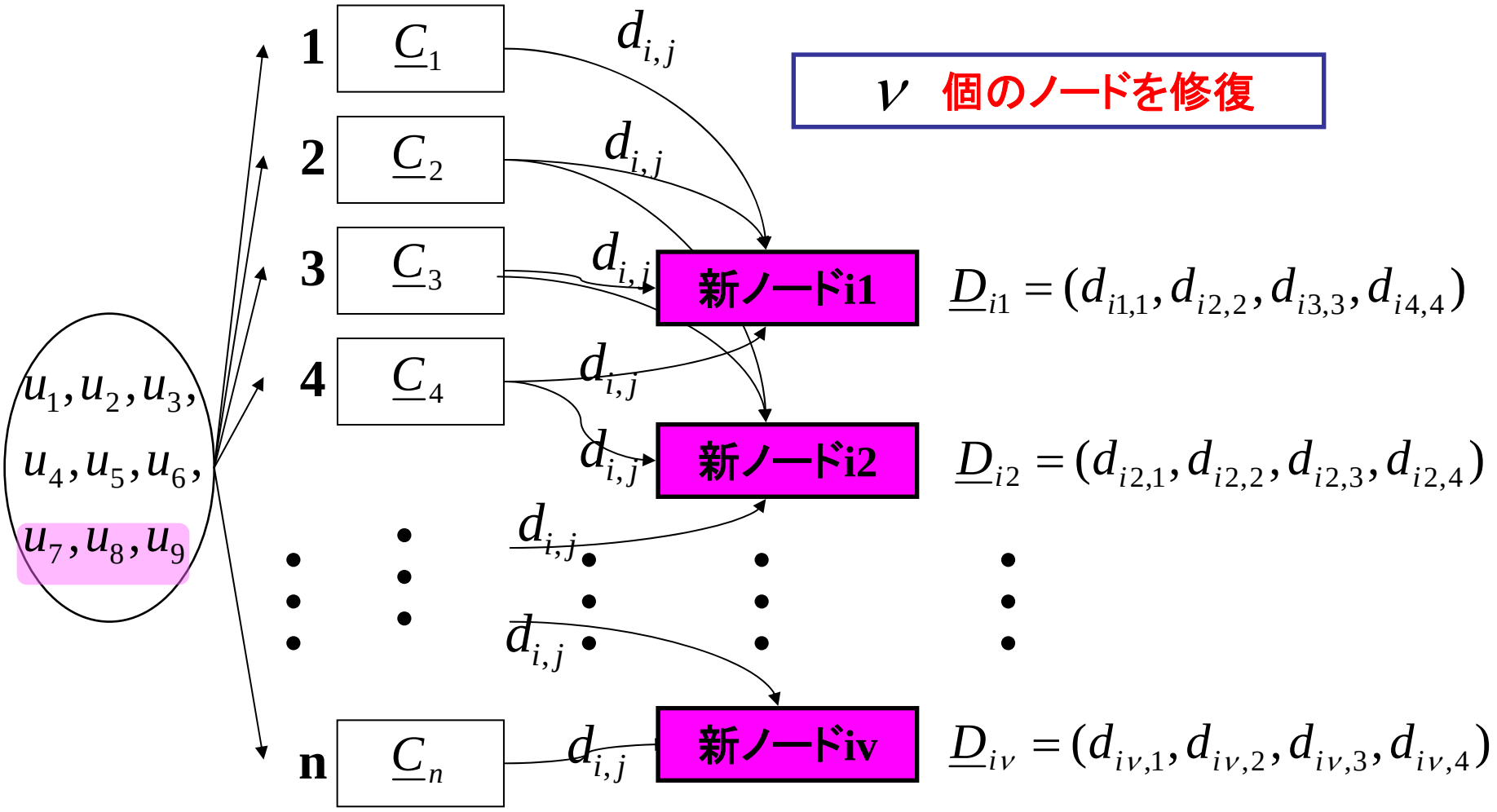
$$\underline{D}_f = (d_{f,h1}, d_{f,h2}, \cdots, d_{f,hd})^t \in GF(q)^d$$

を得る。そして、$\underline{D}_f$ から分散データ $\underline{C}_f$ を一意に再生成できる：

$$\underline{C}_f = \left[(\phi_{h1}, \phi_{h2}, \cdots, \phi_{hd})^t\right]^{-1} \underline{D}_f$$

任意の　ν 個の再生成用データ $\underline{D}_{i1}, \underline{D}_{i2}, \cdots, \underline{D}_{i\nu}$ が与えられた下での秘密情報の条件付エントロピー

$$H(\underline{S} \mid \underline{D}_{i1}\ \underline{D}_{i2}\ \cdots\ \underline{D}_{i\nu})$$



以上