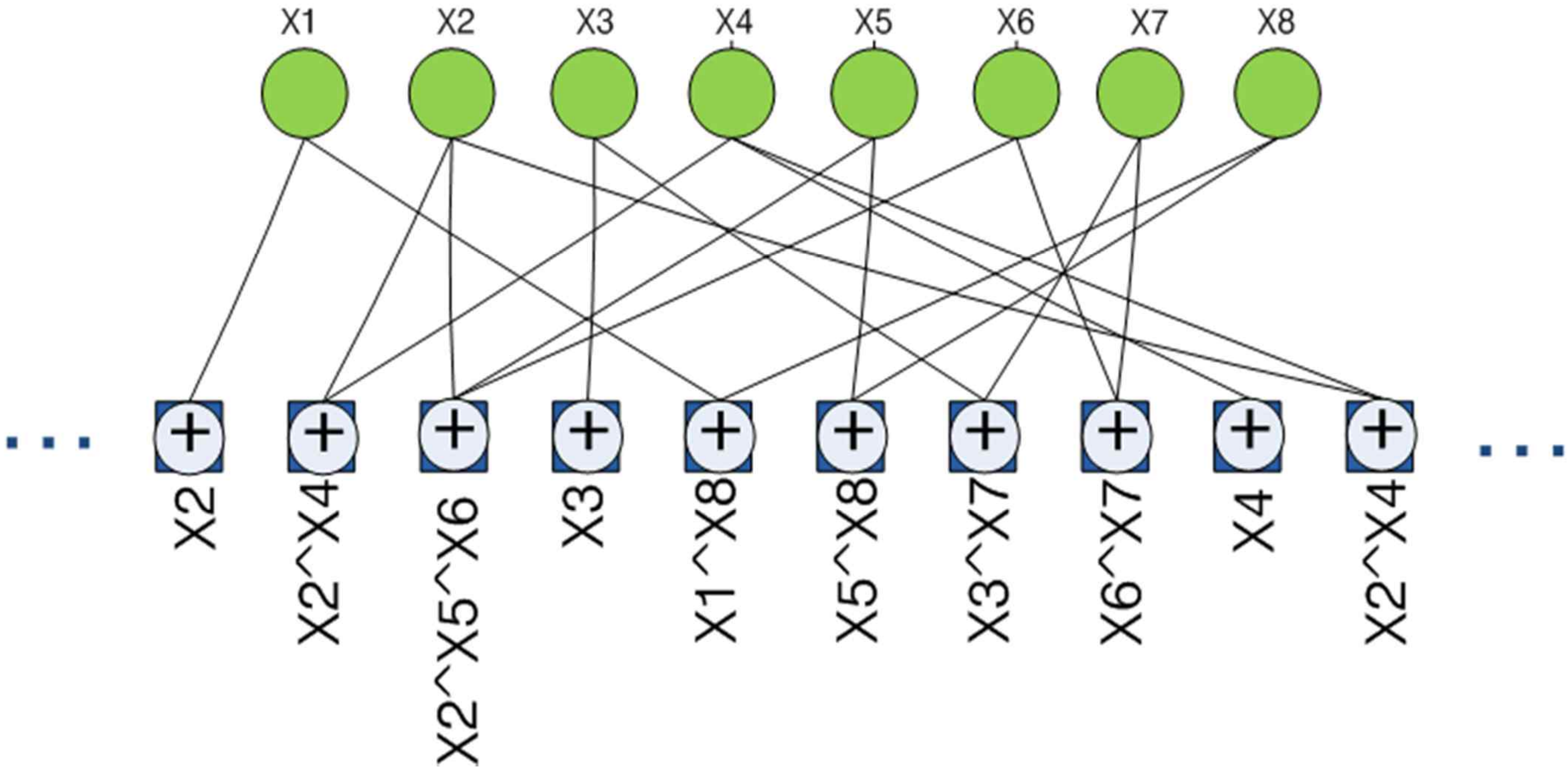
LT 부호의 효율적인 부호화 알고리즘

연세대학교 부호 및 암호 연구실
남미영

Fountain Codes

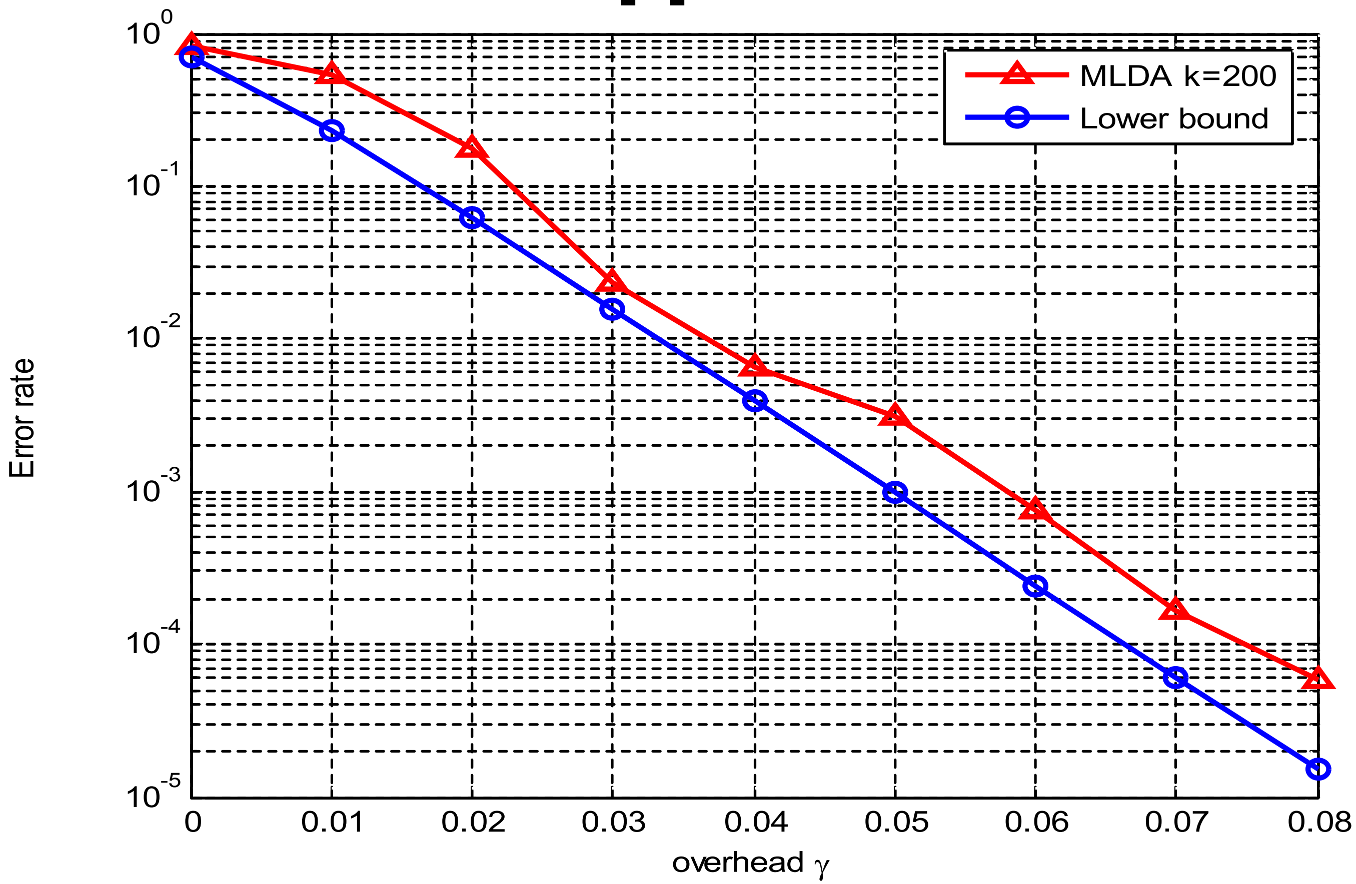


System Model

- **NOTATION**
  - k : number of information symbols
  - n : number of output symbols
  - $\gamma = \mathrm{n/k} - 1$: reception overhead
  - H : binary (nxk)-encoding matrix
  - $|\mathrm{H}|$: weight of H
  - Complexity : number of the edges of the Tanner graph of LT code
- **Binary Erasure Channel**
- **Maximum Likelihood Decoding Algorithm (MLDA)**
  - $Y^T = HX^T$ with encoding symbol vector Y, input symbol vector X
  - Unique solution exists $\xleftrightarrow{\text{iff}}$ $Rank(H) = k$

Performance of LT Codes

- **MLDA vs. Lower Bound[6] with k=200**

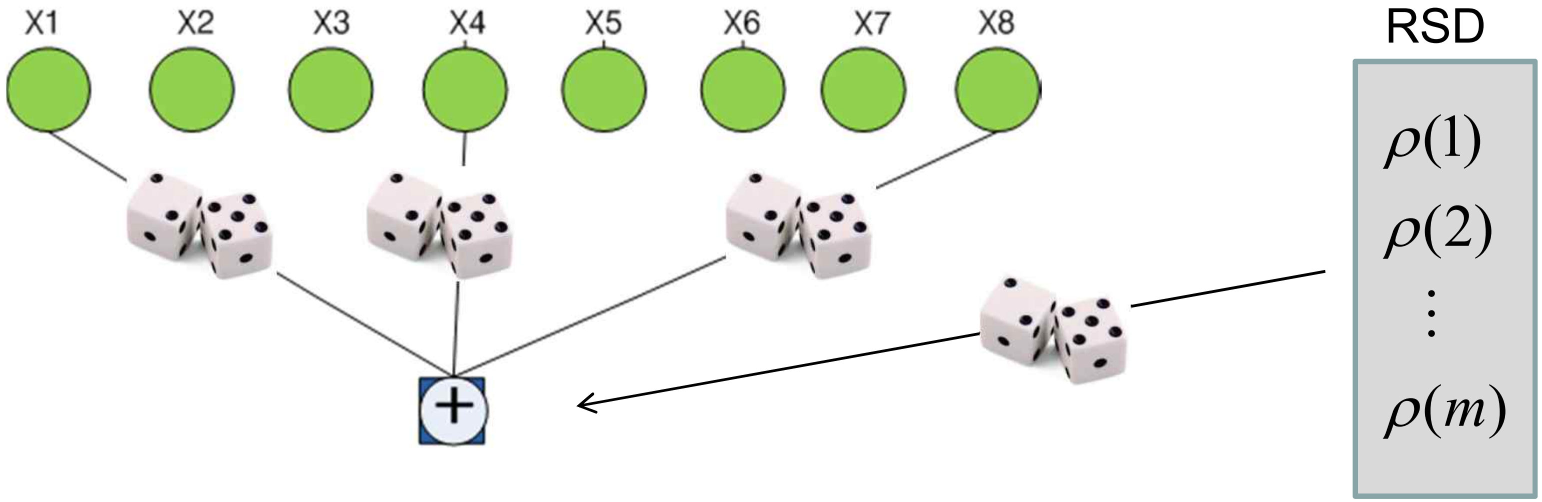


[6] K.-M. Lee, H. Radha, B.-J. Kim and H.-Y. Song, "Kovalenko's Full-Rank Limit and Overheads as Lower Bounds of Error-Performances of LDPC and LT Codes Over Binary Erasure Channels," International Symposium on Information Theory and its Applications, 2008.

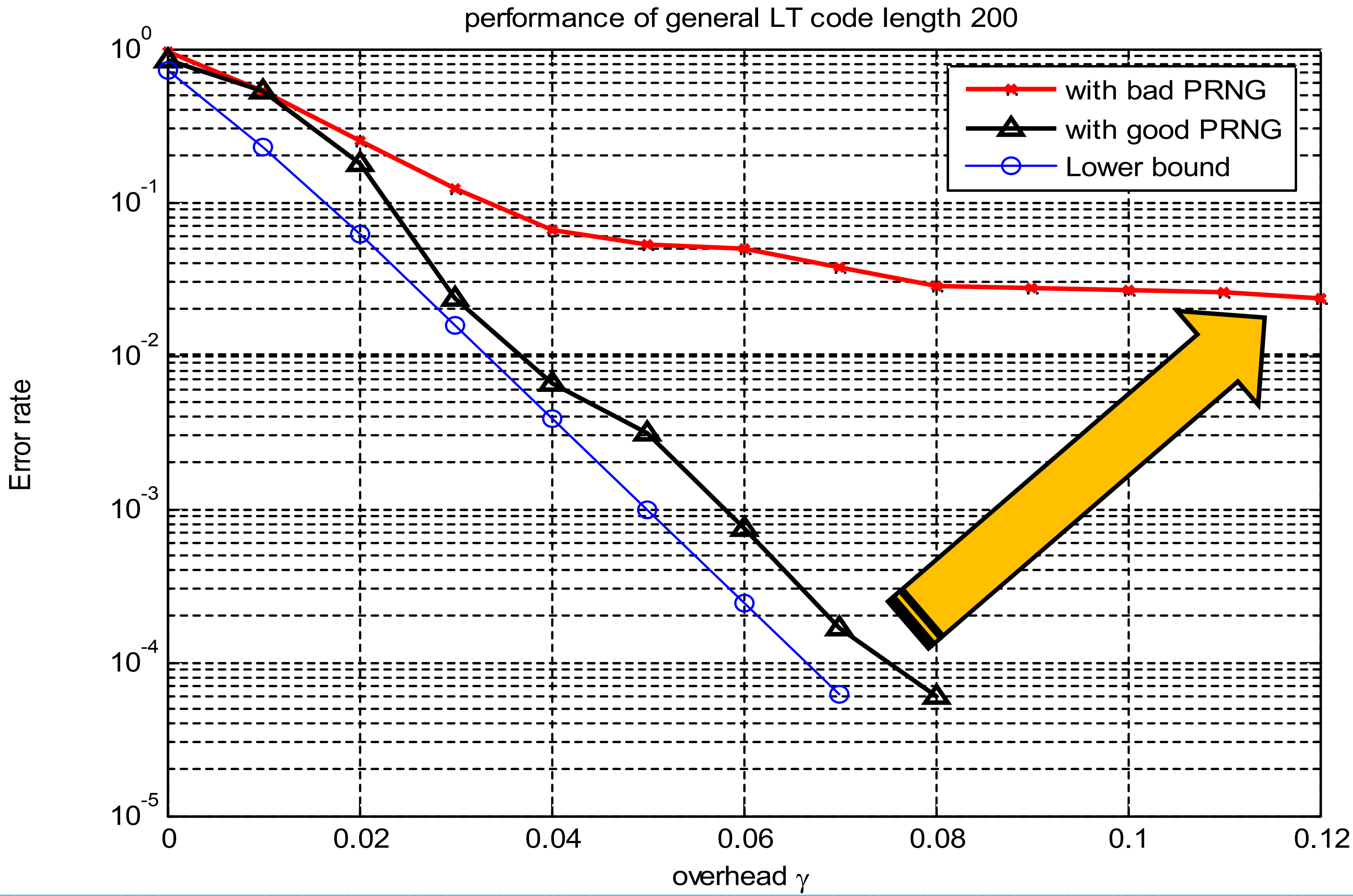
Encoding of LT Codes

Algorithm 1 A general LT encoding algorithm

repeat
    choose a degree d from degree distribution $\rho(d)$.
    choose uniformly at random d input symbol blocks $m_{i_1}, ..., m_{i_d}$.
    send $m_{i_1} \oplus m_{i_2} \oplus \cdots \oplus m_{i_d}$.
until enough output symbols are received.



Non-uniform Column Weight Distribution



Negative Influence of Non-uniform Distribution of Column Weight

- **WHY?**
  - Null column effect – sometimes some input symbols never be chosen
    - ✓ These are never recovered

The case of encoding k=200 information symbols using BAD PRNG

overheads	Null columns / Frame	overheads	Null columns / Frame
0.00	0.0796460177	0.04	0.0466507177
0.01	0.0821256039	0.05	0.0439461883
0.02	0.0714285714	0.06	0.0351380423
0.03	0.0501138952	0.07	0.0368344274

Using Permutations

Algorithm 1 An LT encoding algorithm using permutations

$c = recv()$
$c_0 = f(c)$
$s_0 = P_k(c_0)S_k$
$W = RSD(P_n(c)S_n)$
repeat
  send $\bigoplus_{j=1}^{W[i]} x_{s_t[j+index\%k]}$
  $index = index + W[i]$
  if $index > k$ **then**
    $t = index/k$
    $c_t = f(c_{t-1})$
    $s_t = P_k(c_t)s_{t-1}$
  end if
until enough output symbols are received

Permute by decimation

- **Decimation**
  - Given a sequence s_t and any integer $d \geq 1$, a d th decimation of s_t is any sequence r_t obtained by taking every d th term of original sequence

$$r_t = s_{td+i} \qquad t \geq 0$$

- **We can generate some permutations without transmission of any seed value**

- **Select the number d appropriately to avoid short period of permutation patterns**

The selection of d

- $\gcd(d, k) = 1$

- **Define the order of d as**
  - $d^{ord(d)} \equiv 1 \pmod{k}$

- **We want to have the order of d is not too small to get various combinations of information symbols**

- **We choose the d which has the largest order among all the coprime number of k**

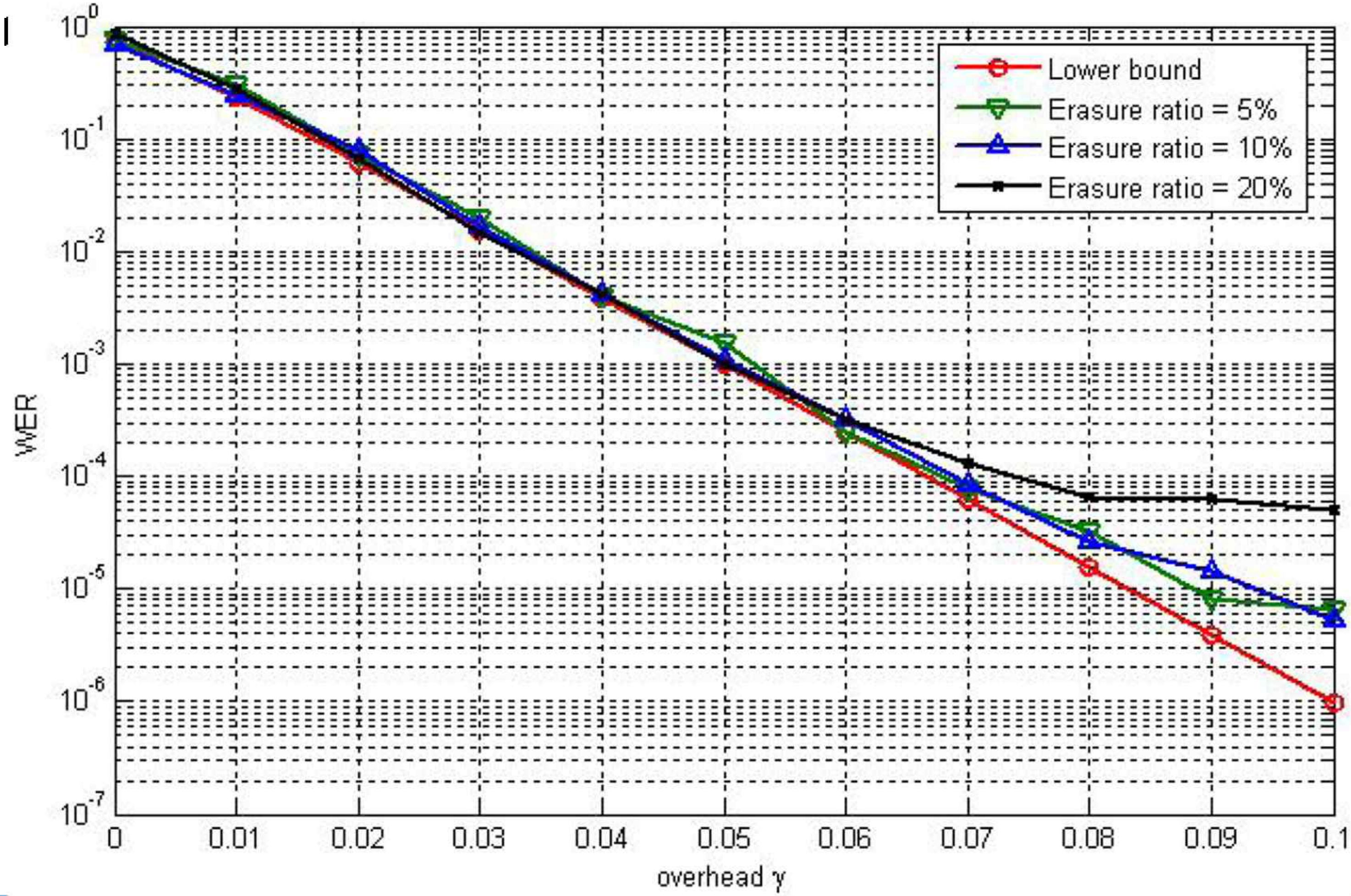
The selection of d

- The orders of every coprimes when the vector size k=200
- The largest order is 20 while the smallest one is 2
- We'd better choose the larger one than the smaller one
- Small order makes a few patterns repeated
- Large order makes the vector have more various patterns

d	$ord(d)$	d	$ord(d)$	d	$ord(d)$
3	20	7	4	9	10
11	10	13	20	17	20
19	10	21	10	23	20
27	20	29	10	31	10
33	20	37	20	39	10
41	5	43	4	47	20
49	2	51	2	53	20
57	4	59	10	61	10
63	20	67	20	69	10
71	10	73	20	77	20
79	10	81	5	83	20
87	20	89	10	91	10
93	4	97	20	99	2
101	2	103	20	107	4
109	10	111	10	113	20
117	20	119	10	121	5
123	20	127	20	129	10
131	10	133	20	137	20
139	10	141	10	143	4
147	20	149	2	151	2
153	20	157	4	159	10
161	5	163	20	167	20
169	10	171	10	173	20
177	20	179	10	181	10
183	20	187	20	189	10
191	10	193	4	197	20
199	2				

Simulation Result (2)

- **k=200, d=17, ord(d)=20, using MLDA, with various erasure**



THANK YOU!

{my.nam, hysong} @ yonsei.ac.kr