Implementability of a Black Box Secret Sharing Scheme

Mathieu DAVID
Supervised by : Pierre KARPMAN

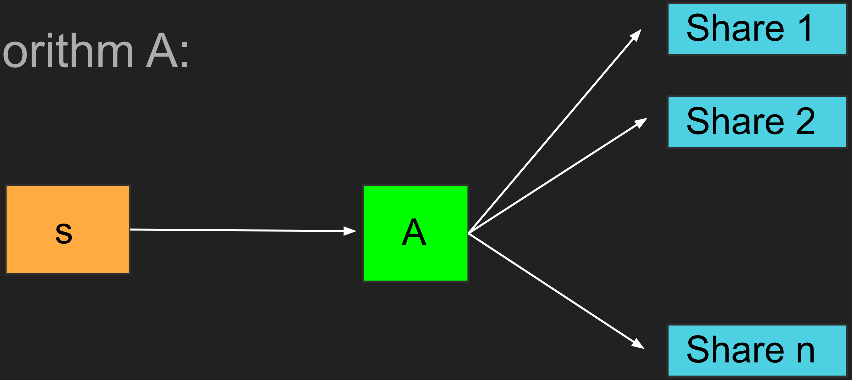
OUTLINE

1. Quick introduction : basic notions related secret sharing
2. Overview of the construction of CRAMER & XING
3. 1st result of the implementation
4. Conclusion/ Quick demo

Introduction

What is a secret sharing scheme ?

A randomized algorithm A:

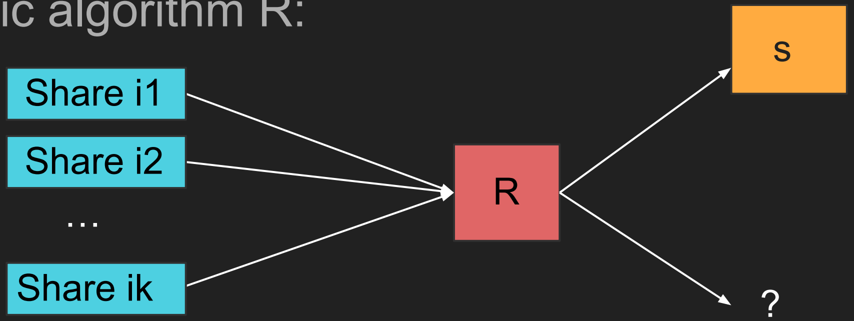


s : the secret

Introduction

What is a secret sharing scheme ?

A deterministic algorithm R:



If the information brought by the shares is sufficient

doesn’t allow us to completely retrieve the secret

s : the secret

Introduction

Access structure in general :

two sets of configurations (and the rest):

Allow us to retrieve the secret	Can bring partial information	Doesn't give any information on the secret

Introduction

Access structure in threshold n-k secret sharing scheme:

Only two sets:

Allow us to retrieve the secret

No combination of shares that bring **partial** information!

Doesn't give any information on the secret

Introduction

Access structure in threshold n-k secret sharing scheme :

Only two sets:

Allow us to retrieve the secret

>= k

Doesn’t give any information on the secret

< k

Construction of Cramer & Xing

The algorithms A and R do algebraic operations. **s** is a value over an **Abelian Group or a Finite field.**

Construction of Cramer & Xing

The algorithms A and R do algebraic operations. **s** is a value over an **Abelian Group or a Finite field.**

most Secret sharing schemes only work over a **specific field** of fixed size

Construction of Cramer & Xing

The algorithms A and R do algebraic operations. **s** is a value over an **Abelian Group or a Finite field.**

most Secret sharing schemes only work over a **specific field** of fixed size

Threshold Black box secret sharing scheme (BBSSS) computing a k-n threshold access structure is one which works on **any abelian Group**

Construction of Cramer & Xing

BBSSS are:

- More **expensive** to build
- More **complex**
- but more **versatile**

1st step towards a secret sharing scheme :

Linear codes are common way to build linear secret sharing scheme and represented by a **“Generator Matrix”.**

$$\begin{bmatrix} s, & g1, & g2, & \dots & , gk\text{-}1 \end{bmatrix} \times \begin{bmatrix} a11 & a12 & \dots & a1n \\ \vdots & & & \\ ak1 & ak2 & \dots & akn \end{bmatrix} = \begin{bmatrix} \text{Share 1} & \text{Share 2} & \cdots & \text{Share n} \end{bmatrix}$$

s : the secret

g1, g2, …, gk-1 : random elements from the Group (random part of A)

Properties on linear codes

Properties of a code influence the access structure :

For threshold secret sharing scheme => need **Maximum Distance Separable (MDS)** code

For BBSSS, another interesting property : **the expansion factor**

To go further: Monotone Span Program

Very close definition to linear code except we added the way of encoding and decoding the secret:

To go further: Monotone Span Program

Very close definition to linear code except we added the way of encoding and decoding the secret:

- A target vector which will describe how to encode the secret

An important **condition** : the target vector is **spanned** by the submatrix of M of support s

To go further: Monotone Span Program

Very close definition to linear code except we added the way of encoding and decoding the secret:

- A target vector which will describe how to encode the secret

An important **condition** : the target vector is **spanned** by the submatrix of M of support s

- A surjective function that will **group the rows** of the matrix M for each share

An interesting family of codes : Reed-Solomon codes

Reed Solomon codes are **MDS** codes, here is an example of secret sharing scheme with Reed Solomon codes :

s , g1 , g2 , … , gk-1

X

$$\begin{pmatrix} \alpha_1^0 & \dots & \alpha_n^0 \\ \alpha_1^1 & \dots & \alpha_n^1 \\ \vdots & \vdots & \vdots \\ \alpha_1^{k-1} & \dots & \alpha_n^{k-1} \end{pmatrix}$$

=

Share 1 Share 2 … Share n

alpha_1, alpha_2, … alpha_n are pairwise distinct elements over the instantiation field

Generator matrix of the code

An interesting family of codes : Reed-Solomon codes

Reed Solomon codes are **MDS** codes, here is an example of secret sharing scheme with Reed Solomon codes :

$$
\boxed{s , g1 , g2 , \quad \dots \quad , gk\text{-}1} \;\times\; \begin{pmatrix} \alpha_1^0 & \dots & \alpha_n^0 \\ \alpha_1^1 & \dots & \alpha_n^1 \\ \vdots & \vdots & \vdots \\ \alpha_1^{k-1} & \dots & \alpha_n^{k-1} \end{pmatrix} = \boxed{\text{Share 1}} \; \boxed{\text{Share 2}} \; \cdots \; \boxed{\text{Share n}}
$$

Limitations : the size of the Field should be greater than n !

Construction of Cramer & Xing

We want: threshold BBSSS with a small expansion factor

Construction of Cramer & Xing

We want: threshold BBSSS with a small expansion factor

To do so, we need a **secret sharing scheme** which the **matrix over Z** compute the access structure **modulo every prime number p**

Construction of Cramer & Xing

We want: threshold BBSSS with a small expansion factor

- one generator matrix which work for any p <= n

Construction of Cramer & Xing

We want: threshold BBSSS with a small expansion factor

- one generator matrix which work for any $p <= n$
- one generator matrix for any $p > n$ => Vandermonde matrix !

Construction of Cramer & Xing: the first generator matrix

For a desired configuration n and k:

- We compute m such that **m >= log(n)**

Construction of Cramer & Xing: the first generator matrix

For a desired configuration n and k:

- We compute m such that **m >= log(n)**

For **each prime number** p <= n:

- we generate a Reed Solomon code over the finite field of size p^m

Construction of Cramer & Xing: the first generator matrix

2

RS code over the finite field of size 2^m

Construction of Cramer & Xing: the first generator matrix

2

RS code over the finite field of size 2^m

3

RS code over the finite field of size 3^m

Construction of Cramer & Xing: the first generator matrix

2

RS code over the finite field of size 2^m

3

RS code over the finite field of size 3^m

p <= n

RS code over the finite field of size p^m

(Matrices of size k x n)

Construction of Cramer & Xing: the first generator matrix

For a desired configuration n and k:

- We compute m such that **m >= log(n)**

For **each prime number** p <= n:

- we generate a Reed Solomon code over the finite field of size p^m

We **write each matrix** (over Fpm) on the finite field of size p

Construction of Cramer & Xing: the first generator matrix

example : taking a random simple RS code over F_2³

$$\begin{pmatrix} 1 & 1 & 1 \\ y^2+y+1 & y^2+1 & y+1 \end{pmatrix}$$

Construction of Cramer & Xing: the first generator matrix

example : obtaining the RS code over F_2

I4 : companion matrix of 1

companion matrix of alpha_1…

$$\left(\begin{array}{ccc|ccc|ccc}
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
\hline
1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1
\end{array}\right)$$

Construction of Cramer & Xing: the first generator matrix

2	3		p <= n
Code over the finite field of size 2	**Code over the finite field of size 3**	→	**Code over the finite field of size p**

(Matrices of size km x nm)

Construction of Cramer & Xing: the first generator matrix

For a desired configuration n and k:

- We compute m such that **m >= log(n)**

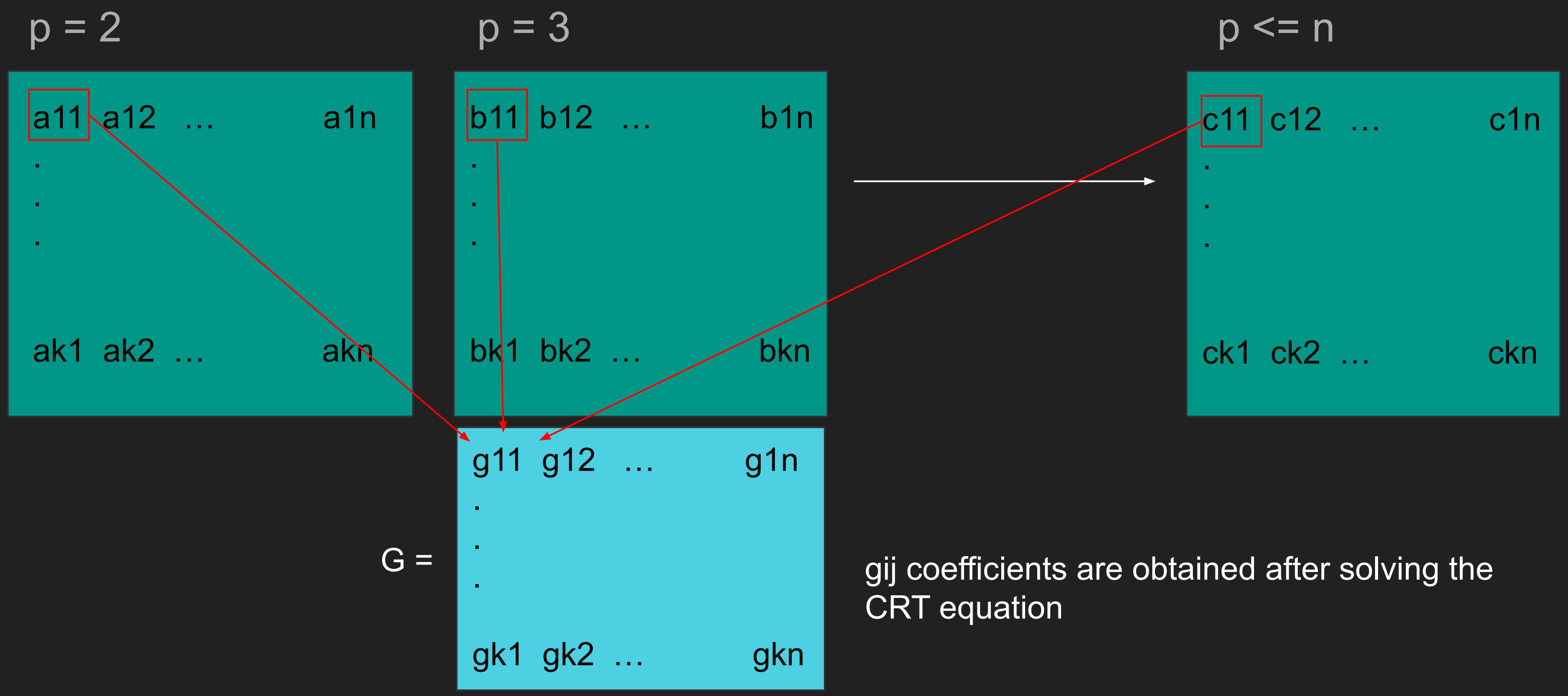
For **each prime number** p <= n:

- we generate a Reed Solomon code over the finite field of size p^m

We **write each matrix** (over Fpm) on the finite field of size p

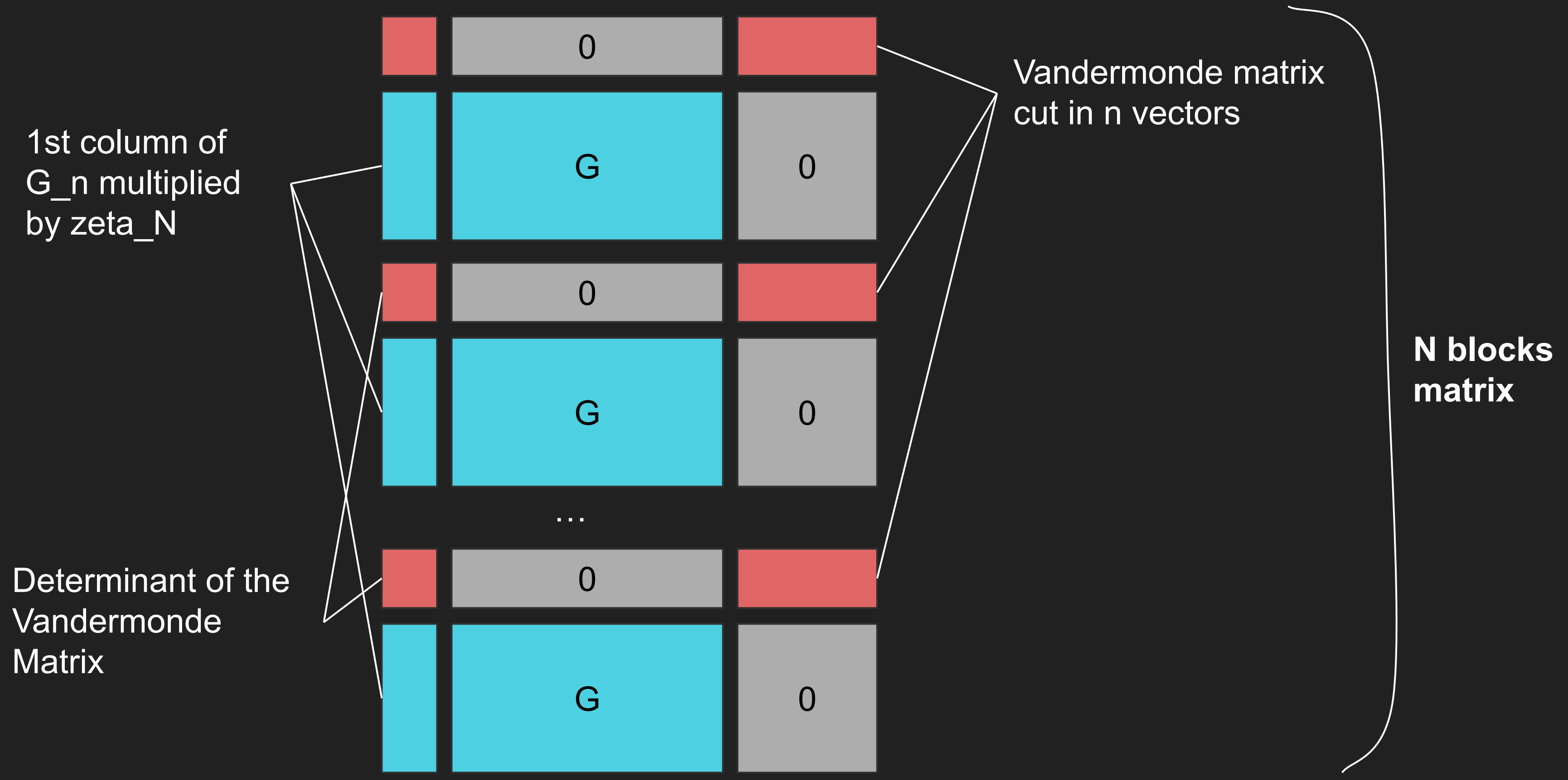
We apply **Chinese remainder theorem** to each coefficients to obtain the desired matrix

Construction of Cramer & Xing: the first generator matrix



gij coefficients are obtained after solving the CRT equation

Construction of Cramer & Xing: the full matrix



Instantiation of the BBSSS

we use **SageMath**

We instantiate the matrix with

n = 6

k = 3

m = 3 (m >= log(n))

Instantiation of the BBSSS

Implementation of all the steps defined before :

- instantiation of each RS code
- writing over the sub field
- vandermonde matrix

- gluing each matrix

Limitations ?

- in our instantiation: zeta_n > 10^1800

$$\rho_N = \prod_{S \subset [n], |S|=t} \left(\prod_{A \in \mathcal{M}_t(N_S), \det(A) \neq 0} \det(A) \right)$$

Limitations ?

- in our instantiation: zeta_n > 10^1800
- possible to bound it ?

$$\rho_N = \prod_{S \subset [n], |S| = t} \left(\prod_{A \in \mathcal{M}_t(N_S), \det(A) \neq 0} \det(A) \right)$$

Limitations ?

- In our instantiation: zeta_n > 10^1800
- possible to bound it ?
- zeta_n is a product of determinant of submatrix of **G :**

- Use the **Hadamard's inequality** to bound the determinant of each submatrix

$$\rho_N = \prod_{S \subset [n], |S|=t} \left(\prod_{A \in \mathcal{M}_t(N_S), \det(A) \neq 0} \det(A) \right)$$

Limitations ?

- In our instantiation: zeta_n > 10^1800
- possible to bound it ?
- zeta_n is a product of determinant of submatrix of **G :**

- Use the **Hadamard's inequality** to bound the determinant of each submatrix

- Bound the number of submatrices that have their determinant != 0

$$\rho_N = \prod_{S \subset [n], |S| = t} \left(\prod_{A \in \mathcal{M}_t(N_S), \det(A) \neq 0} \det(A) \right)$$

Limitations ?

At the end, our bound is:

$$(((\sqrt{m(k-1)}\kappa)^{m(k-1)})^{\binom{mk-1}{m(k-1)}})^{\binom{n}{k-1}}$$

approximately equals to 10^4700 => **Not Tight !**

Encoding function:

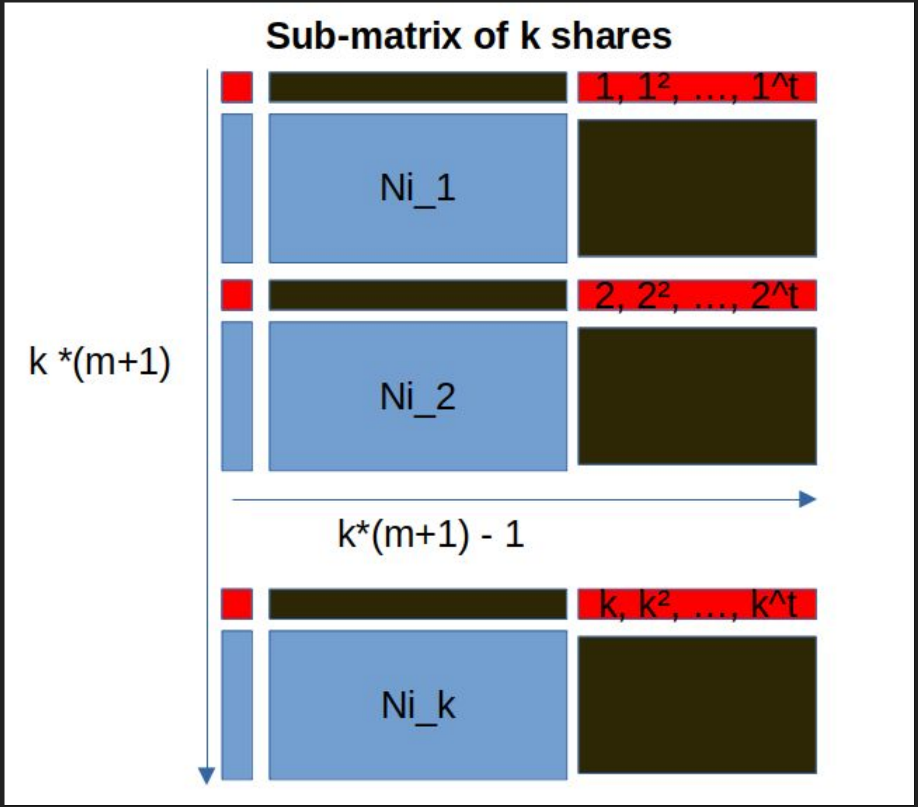
s , g_2, g_3,   … , g_l   X   Final Matrix   =   Coord1   Coord 2   …   Coord l’

where l = k * m + k -1

where l’ = n * (m+1)

Decoding function:

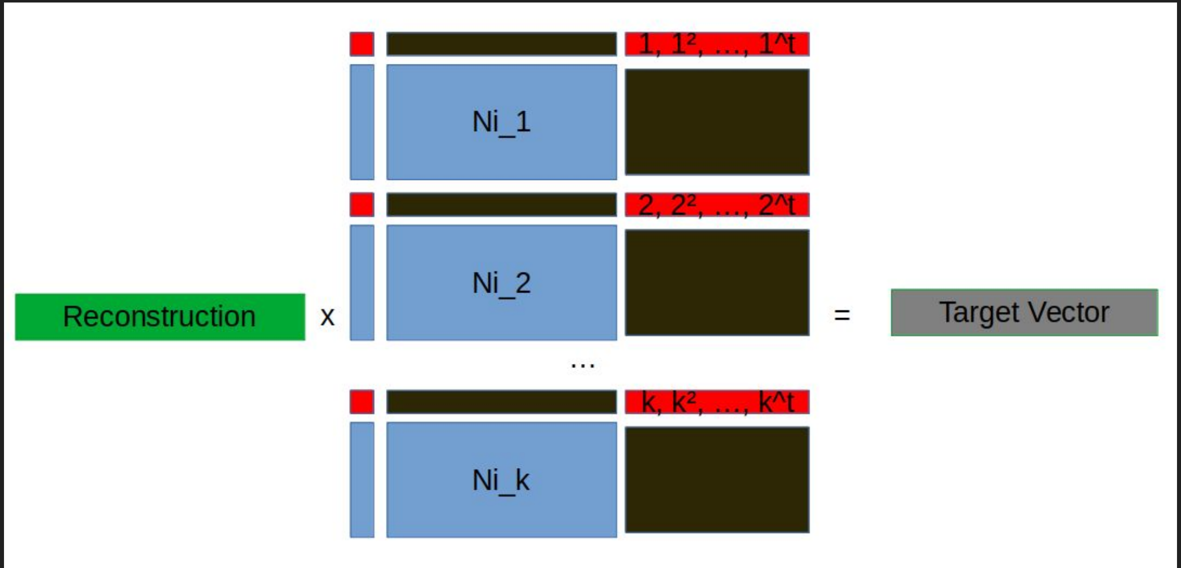
1 - we keep the corresponding submatrix



Decoding function:

1 - we keep the corresponding submatrix

2 - we solve the following linear system



Decoding function:

1 - we keep the corresponding submatrix

2 - we solve the following linear system

3 - compute the dot product **Solution * Shares**

Reconstruction ·

Share i_1,
Share i_2,
…,
Share i_k

= s

Conclusion

Theoretical construction, some limitations…

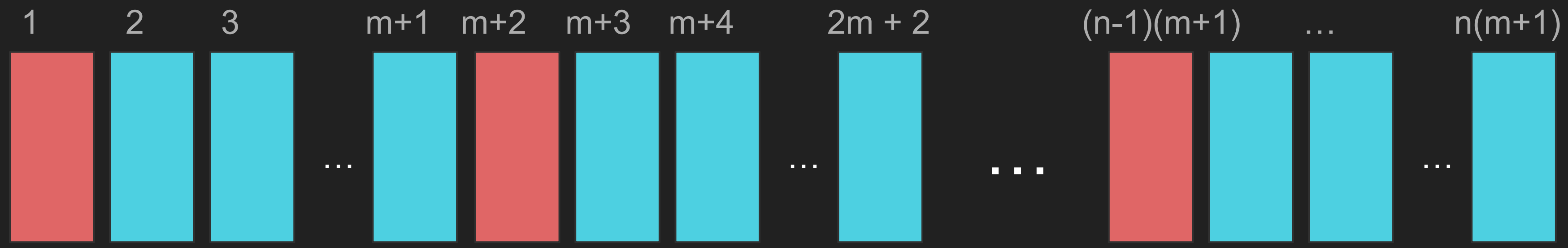
Limitations are only involved in the creation of the BBSSS !

References

- Wikipedia contributors. (2023, May 22). Hadamard's inequality. In *Wikipedia, The Free Encyclopedia*. Retrieved 13:10, June 8, 2023, from https://en.wikipedia.org/w/index.php?title=Hadamard%27s_inequality&oldid=1156416002
- Cramer and Xing, 2019] Ronald Cramer and Chaoping Xing. Blackbox secret sharing revisited: A coding-theoretic approach with application to expansionless near-threshold schemes. Cryptology ePrint Archive, Paper 2019/1134, 2019. https://eprint.iacr.org/2019/1134
- Massey, J.L. (1999). Minimal Codewords and Secret Sharing.

Appendix: towards a deeper understanding of the construction

Structures of the shares obtained by the MSP



The multiplication with the glued matrix returns n x (m+1) shares…

,