Reversible Data Hiding in Encrypted Image via Secret Sharing Based on GF(2^8)

Group B1
R11921072 謝子涔
R11528026 陳品如

Outline

- Motivation
- Methodology
    - Image Encryption
    - Shamir’s Secret Sharing
    - Data Embedding
    - Data Extraction and Image Recovery
- Result Demo
- Reference

Motivation

Data hiding technique can be used to **embed additional data** into image.

It can be applied in multiple fields including

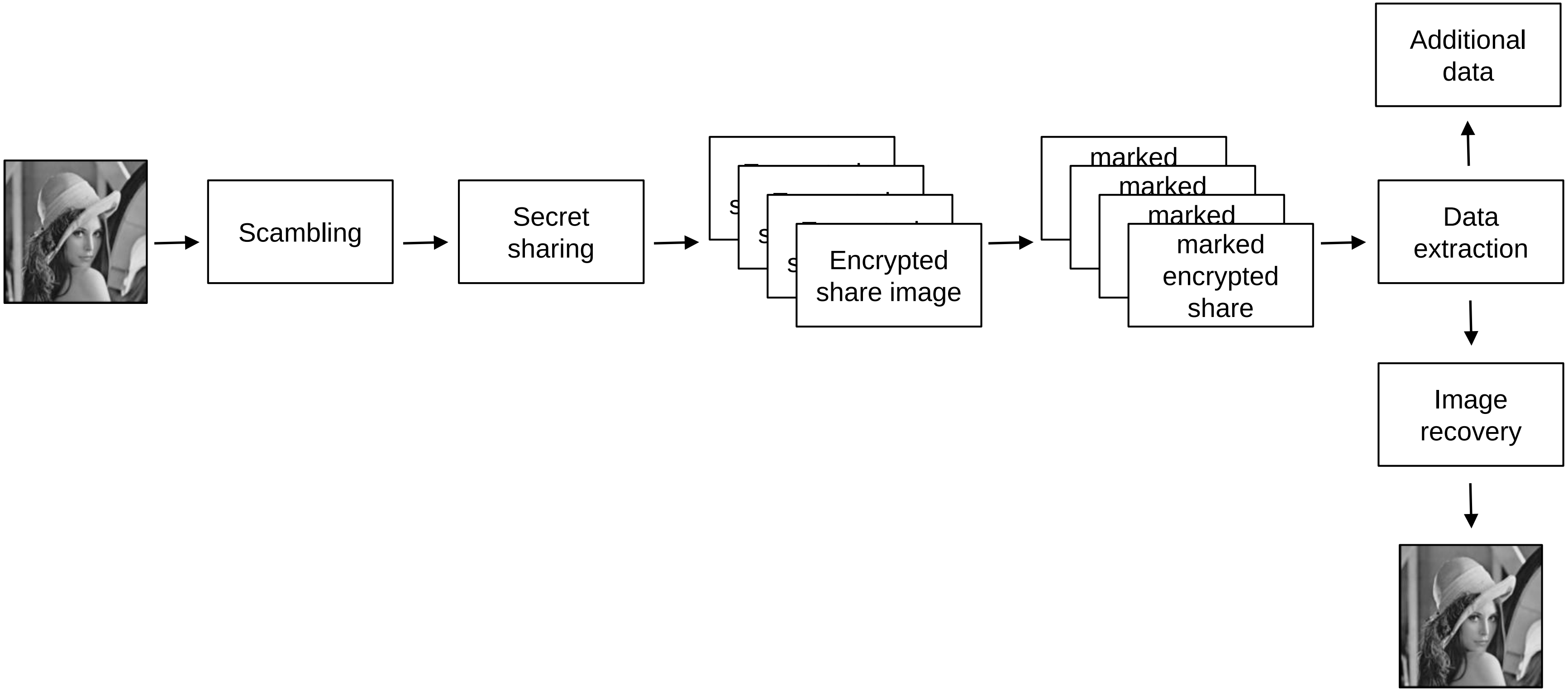
- Military secret transmission
- Medical image privacy protection
- Cloud service

However, during transmission, the image might undergo attacks from the third party. Therefore, **Shamir secret sharing** is applied to assure the protection of secret transmission.

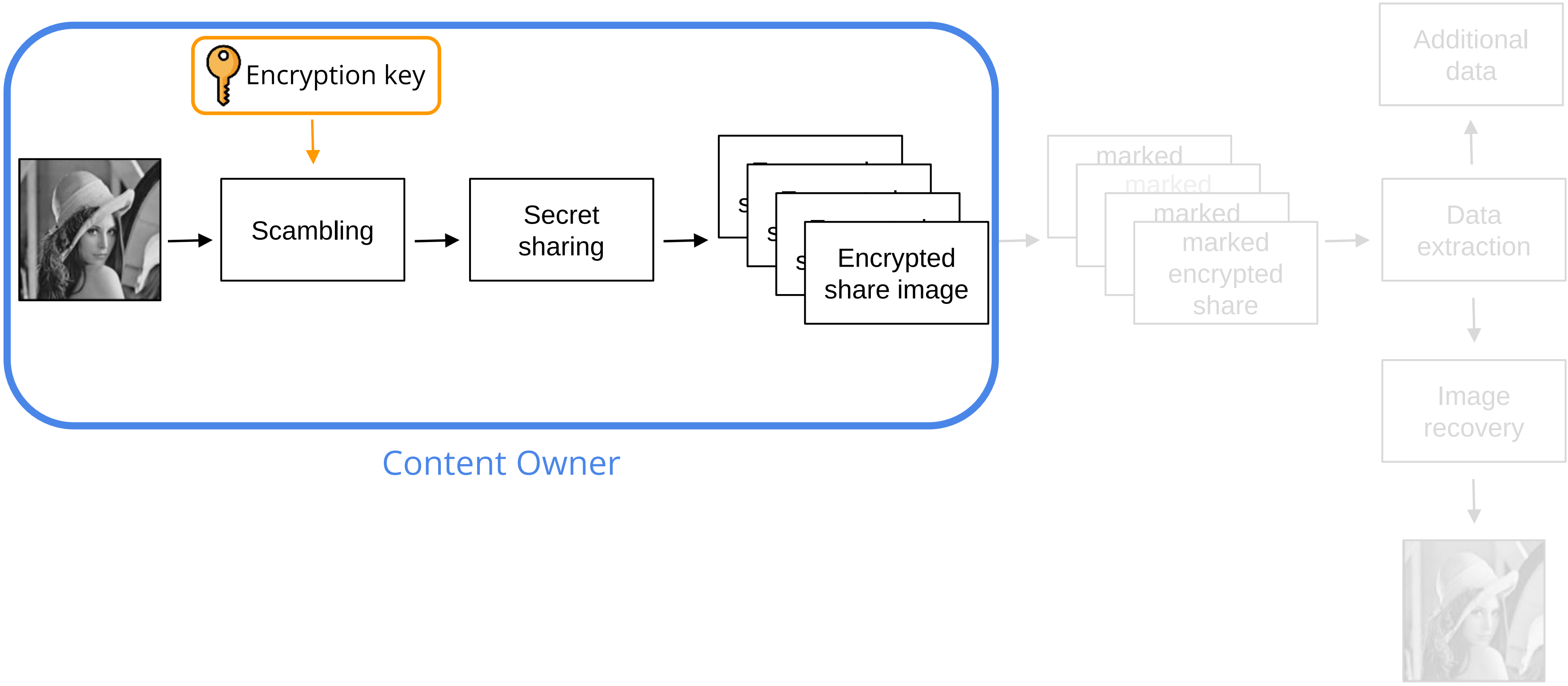
Methodology

1. Image Encryption
   1. Key-based scrambling
   2. ZigZag pattern scrambling
2. Shamir's Secret Sharing
3. Data Embedding
4. Data Extraction and Image Recovery

Overview

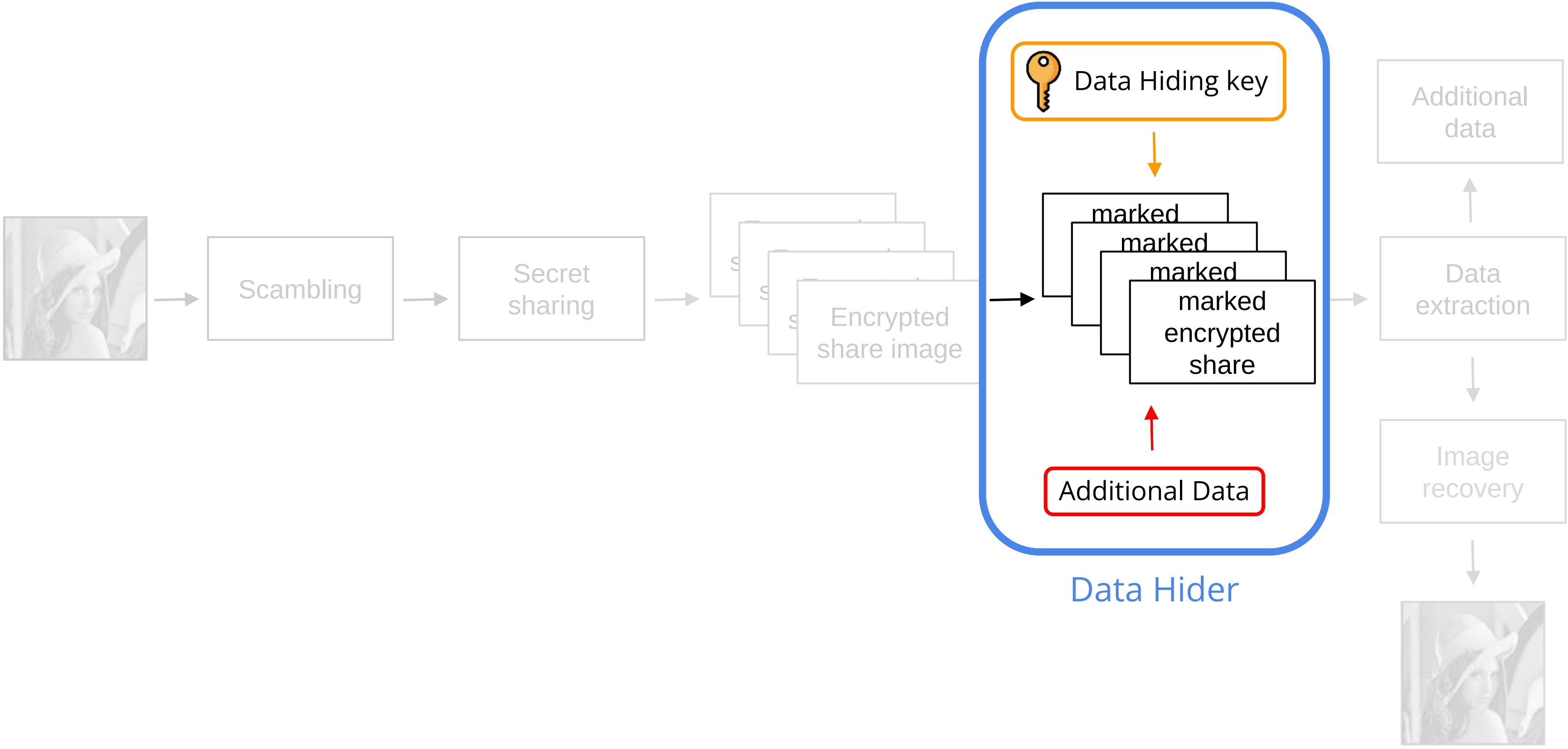

Scambling
Secret sharing
Encrypted share image
marked
marked
marked
marked encrypted share
Data extraction
Additional data
Image recovery

Overview



Encryption key
Scambling
Secret
sharing
Encrypted
share image
marked
encrypted
share
Additional
data
Data
extraction
Image
recovery
Content Owner

Overview



Scambling
Secret sharing
Encrypted share image
Data Hiding key
marked
marked
marked
marked encrypted share
Additional Data
Data Hider
Additional data
Data extraction
Image recovery

Overview

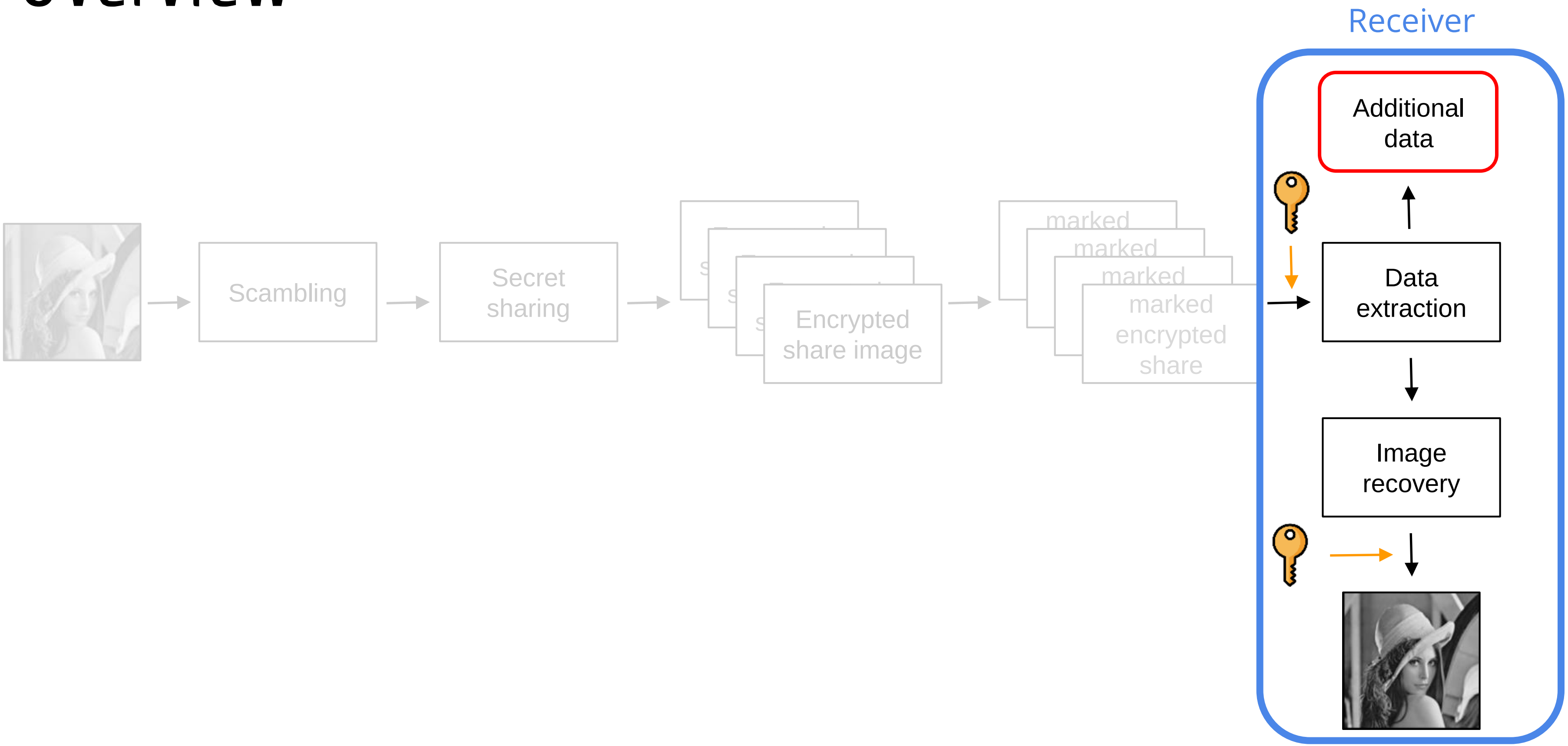

Scambling
Secret sharing
Encrypted share image
marked
marked
marked
marked encrypted share
Receiver
Additional data
Data extraction
Image recovery

Image Encryption

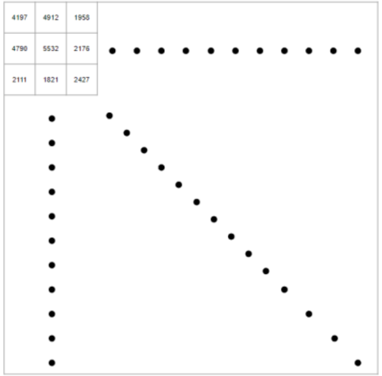
We implement image encryption based on two different strategies

1. Key-based image scrambling
2. ZigZag pattern scrambling

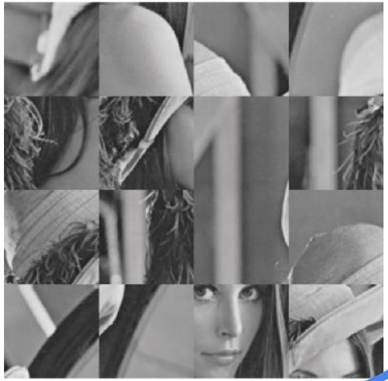
1. Key-based Scrambling

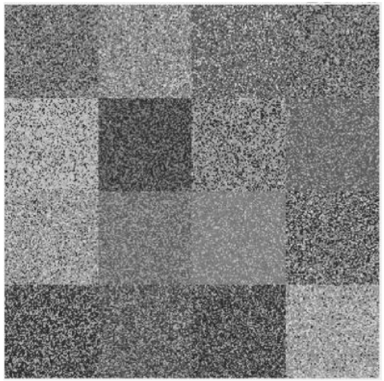
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

16	15	3	12
14	10	1	9
6	13	5	2
4	8	11	7

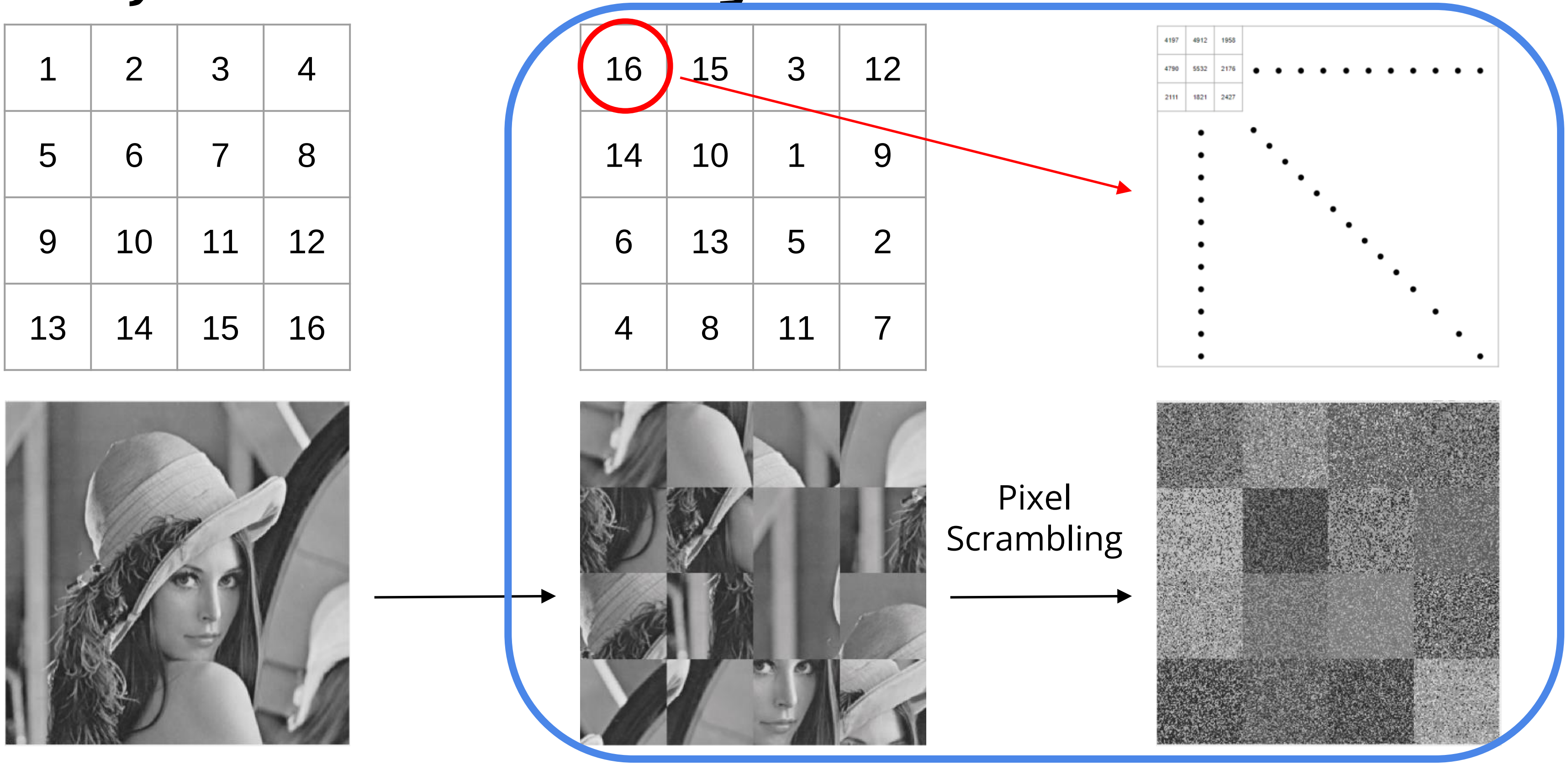


Block Scrambling

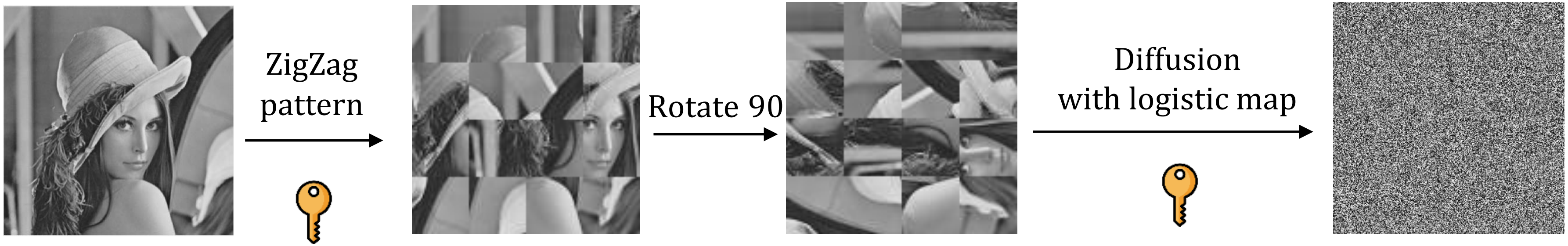




1. Key-based Scrambling



2. ZigZag Pattern Scrambling

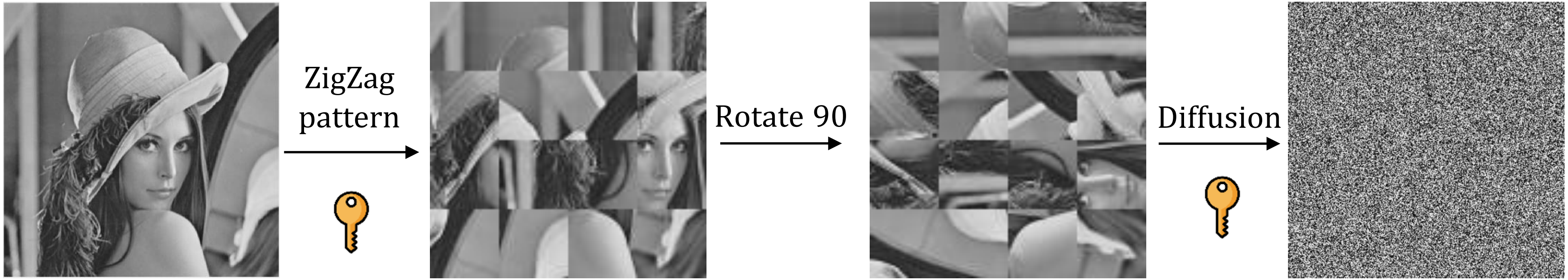


2. ZigZag Pattern Scrambling

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

1	6	10	8
2	3	13	12
5	4	14	15
9	7	11	16

With the reconstruction order, the zigzag index is the first image scrambling key.



2. ZigZag Pattern Scrambling

Diffusion

Logistic map:

M: height, N: width, P: plain image

$$Y_{n+1} = aY_n(1 - Y_n)$$

$$initial\ value{:}\ Y_0 = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} P(i,j)}{M * N * 255}$$

Control parameter: a

$0 < a \leq 4, a \in [3.57, 4]$ is the most chaotic

Iteration number $N_0 = 10000$

Iterate $N_0 + MN$ times, skip the first N_0 elements to get the new sequence S

Use sequence S to calculate the key

$K(i) = mod(floor(S(i) * 10^{14}), 256)$

Bit-wise XOR operation between the key K and the zigzag scramble image to get the encrypted image.

$$E = X \oplus K$$

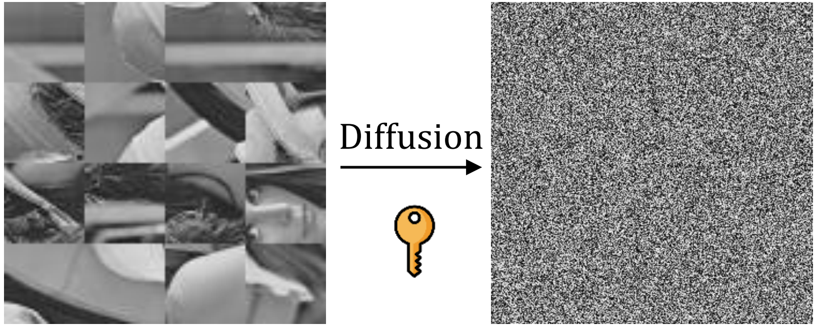
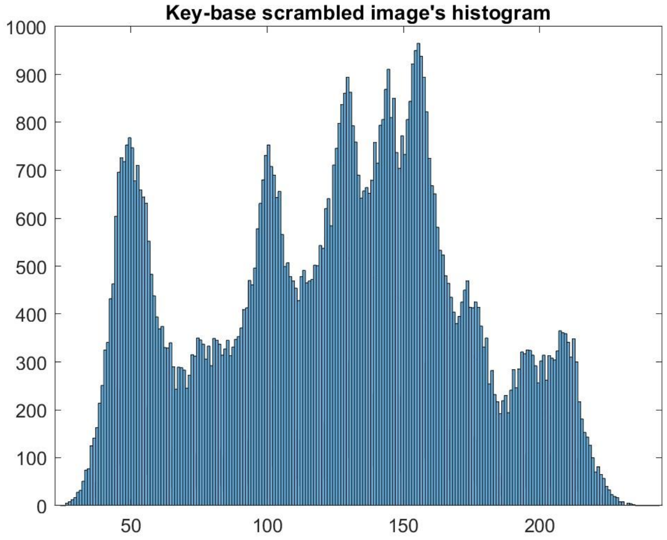
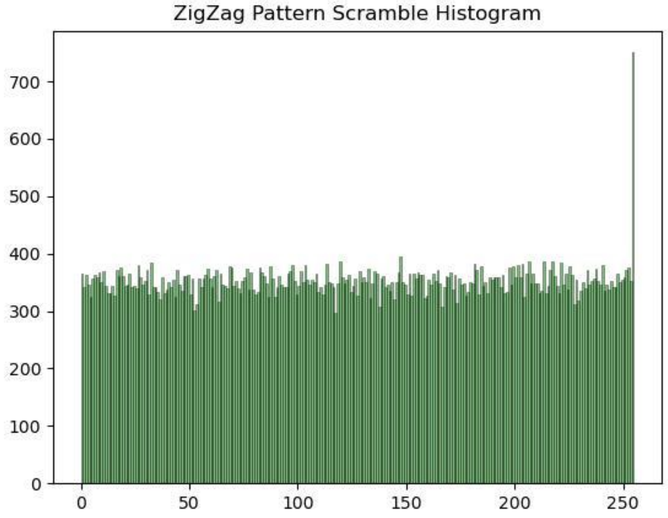


Image Encryption Comparison

Key-based



ZigZag

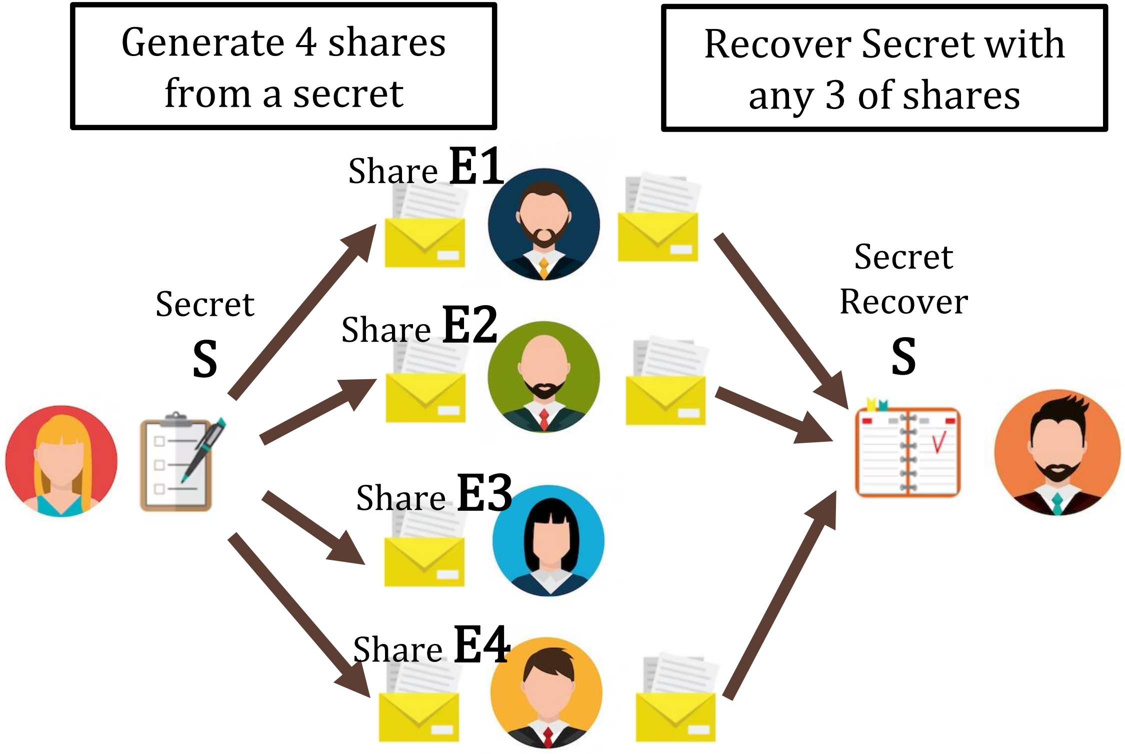


Shamir's Secret Sharing

Shamir's Secret Sharing is an algorithm that allows participants to share ownership of a secret by distributing shares.

A secret is split into n shares for n participants, and the secret can be recovered with any t or more shares collected.

For instance, Let (t, n) = (3, 4)
If we lose one of the 4 shares, we can still recover the secret with at least 3 shares.

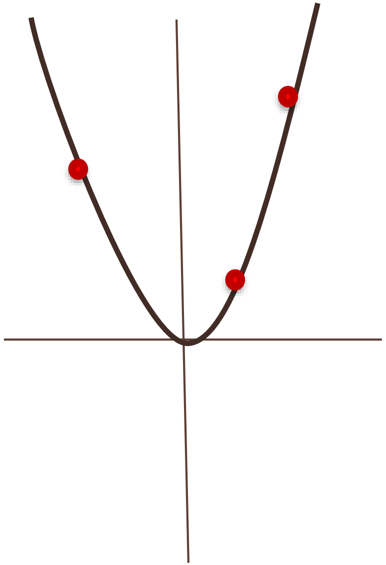


Shamir's Secret Sharing

It is a (t, n) threshold scheme based on polynomial interpolation over finite fields. For a polynomial of degree t-1, we can define this polynomial with t points.

$$f(x) = (s + a_1x + a_2x^2 + \cdots + a_{t-1}x^{t-1})$$

$\boxed{\text{f(x)} = s + a_1x + a_2x^2}$



Let Shamir's secret sharing implement apply over Galois fields GF(2^8), the polynomial f(x) and the irreducible polynomial p(x) are defined as:

$$f(x) = (s \oplus a_1x \oplus a_2x^2 \oplus \cdots \oplus a_{t-1}x^{t-1}) \ mod \ p(x)$$
$$p(x) = x^n + x + 1$$

In GF(2^8), the addition or subtraction operator represents XOR.

Shamir's Secret Sharing

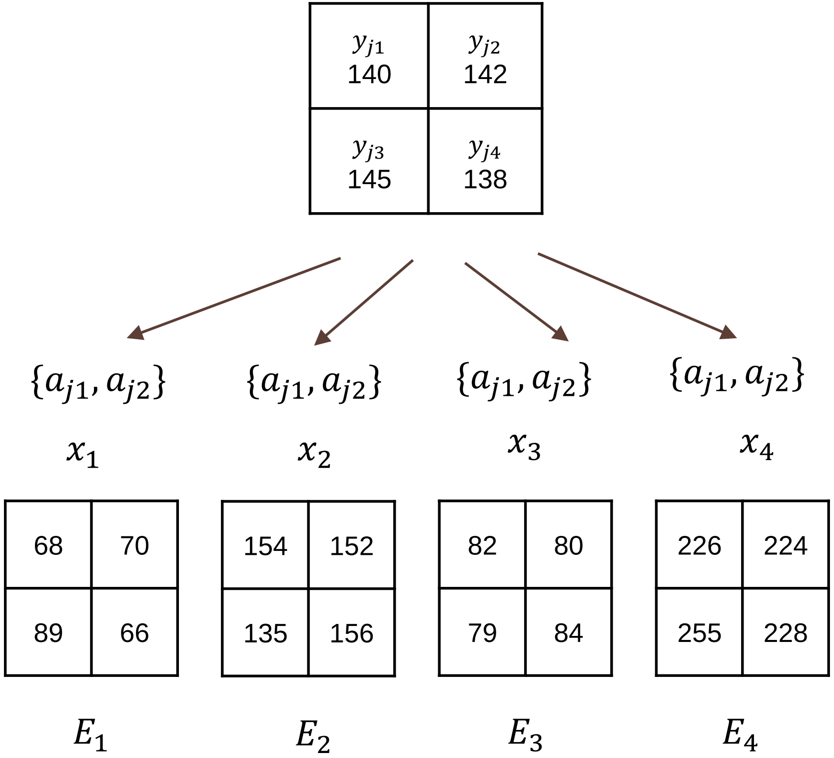
Let the degree of polynomial t-1 = 2, n = 4
Apply GF(2^8) and the irreducible polynomial p(x)

$$f(x) = (s \oplus a_1 x \oplus a_2 x^2) \; mod \; p(x)$$
$$p(x) = x^8 + x^4 + x^3 + x + 1$$

For K blocks, pixels $\{y_{j1}, y_{j2}, y_{j3}, y_{j4}\}$ = s, $1 < j \leq K$
can be transformed into

$\{f_{yj1}(x_i), f_{yj2}(x_i), f_{yj3}(x_i), f_{yj4}(x_i)\}$, $1 < i \leq n$

with $\{a_{j1} \ldots a_{j(t-1)}\}$ and $\{x_{j1} \ldots x_{jn}\}$



Data Embedding

- ❖ Step1. Define Threshold(ε) and Additional data
- ❖ Step2. Blocks are classified into two sets : Es & Ns

  (Es=embedded set, Ns=non-embedded set)

  - ➢ If $\varepsilon=7$

    11>7

  - ➢ Ex. $(68)_2 \oplus (70)_2 = 2$, $(68)_2 \oplus (79)_2 = 11$, $(68)_2 \oplus (66)_2 = 6 \rightarrow$ Ns

    $(70)_2 \oplus (71)_2 = 1$, $(70)_2 \oplus (71)_2 = 1$, $(70)_2 \oplus (69)_2 = 3 \rightarrow$ Es

68	70
79	66

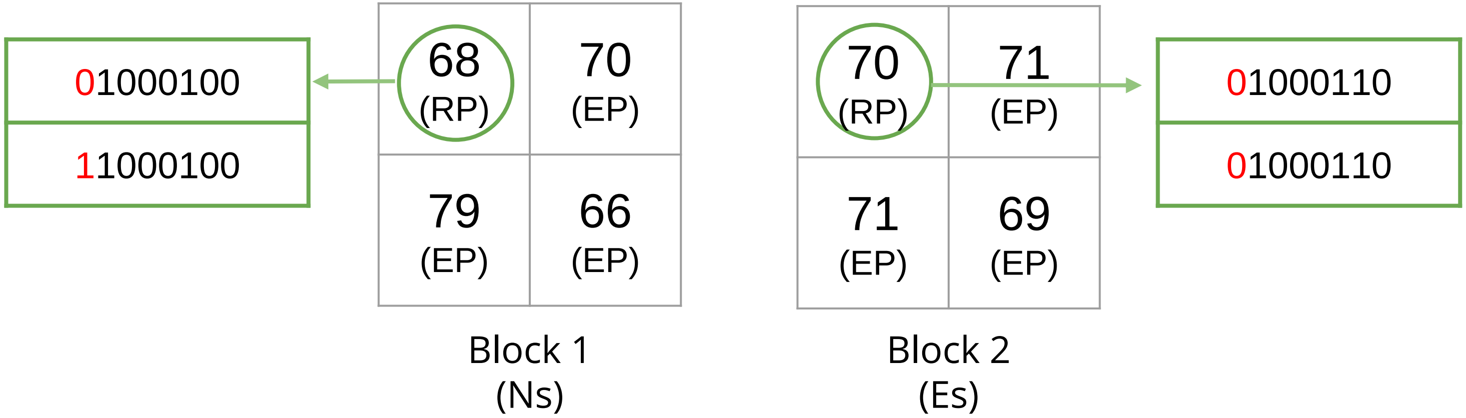
Block 1
(Ns)

70	71
71	69

Block 2
(Es)

Data Embedding

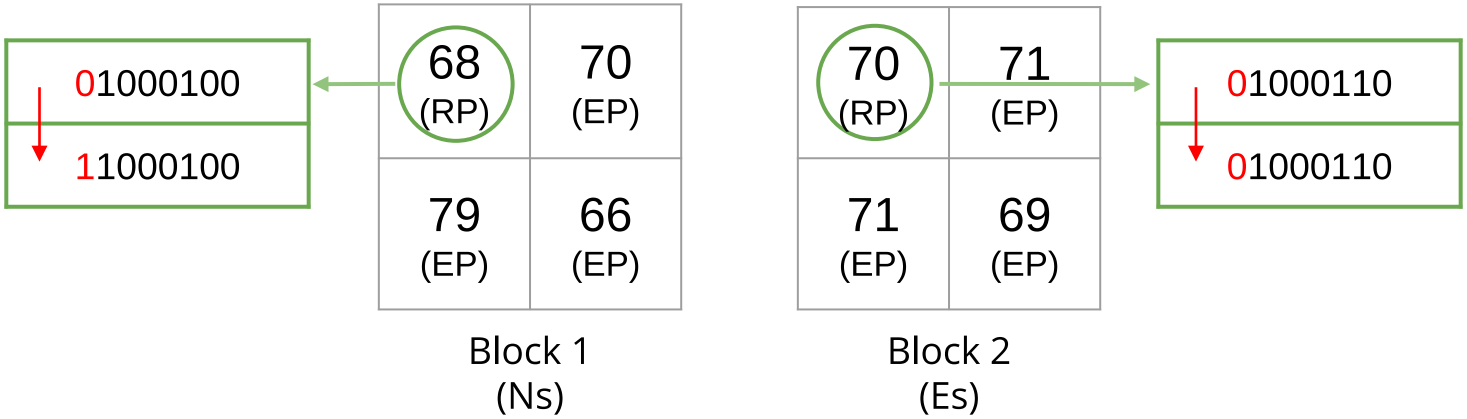
- Step3. Divide all pixels in each blocks into RP & EP
  - RP : Reference pixel in each block
  - EP : Embedded pixels in each block
- Step4. Replace the MSB of RP
  - If Es, then the MSB of RP is ‘0’
  - If Ns, then the MSB of RP is ‘1’



Data Embedding

❖ Step5. Replaced MSB of each RP is combined with the additional data

Ex. Additional data=0101010100011 ⟶ 000101010100011



Data Embedding

- Step6. For EPs in Es
  - According to the ε=7, u1=3, u2=5
  - u1= the result of xor, u2=payload

ε'	D_{jz}	u_1	u_2
0	$D_{jz} = 0$	0	8
1	$D_{jz} = 1$	1	7
3	$D_{jz} \le 3$	2	6
7	$D_{jz} \le 7$	3	5
15	$D_{jz} \le 15$	4	4
31	$D_{jz} \le 31$	5	3
63	$D_{jz} \le 63$	6	2
127	$D_{jz} \le 127$	7	1

Ex. Additional data=(00010)(10101)(00011)

70 (EP)	71 (EP)
71 (EP)	69 (EP)

Block 2
(Es)

→

$(70)_2 \oplus (71)_2 = 1$

01000110	00100010
00110101	01100011

$(70)_2 \oplus (71)_2 = 1$ $(70)_2 \oplus (69)_2 = 3$

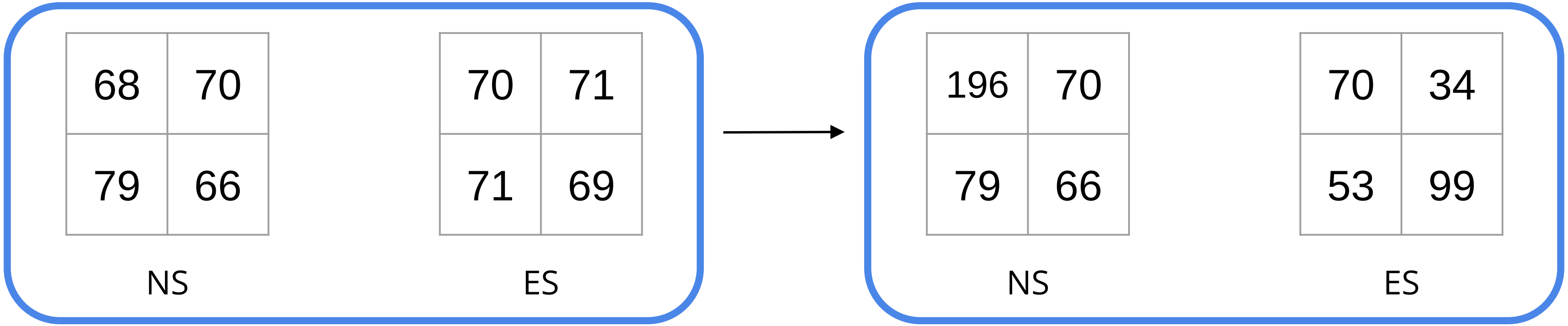
→

70 (RP)	34 (EP)
53 (EP)	99 (EP)

Block 2
(Es)

Data Embedding

❖ Result of pixel values after data embedded

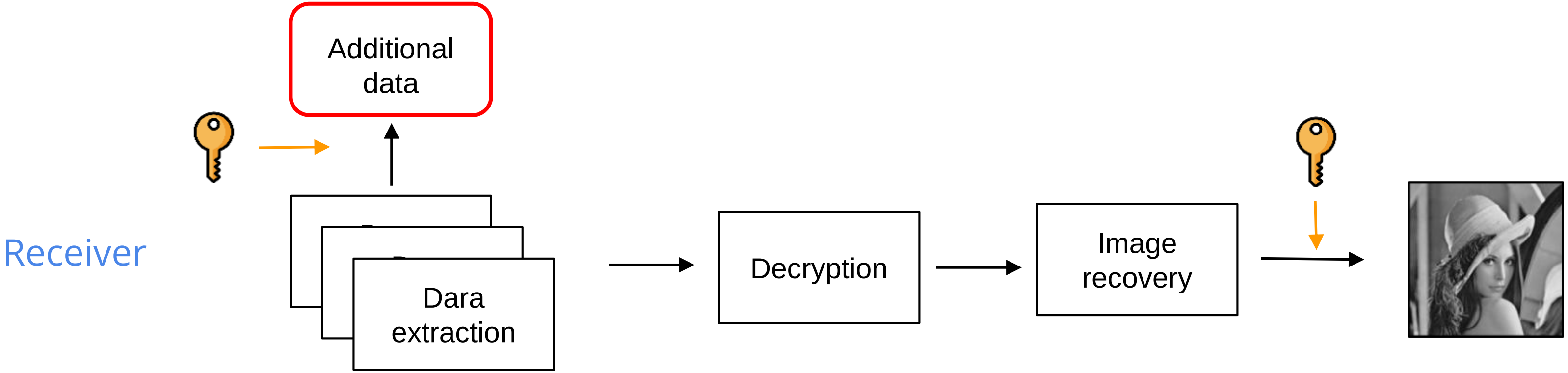


Original pixel values

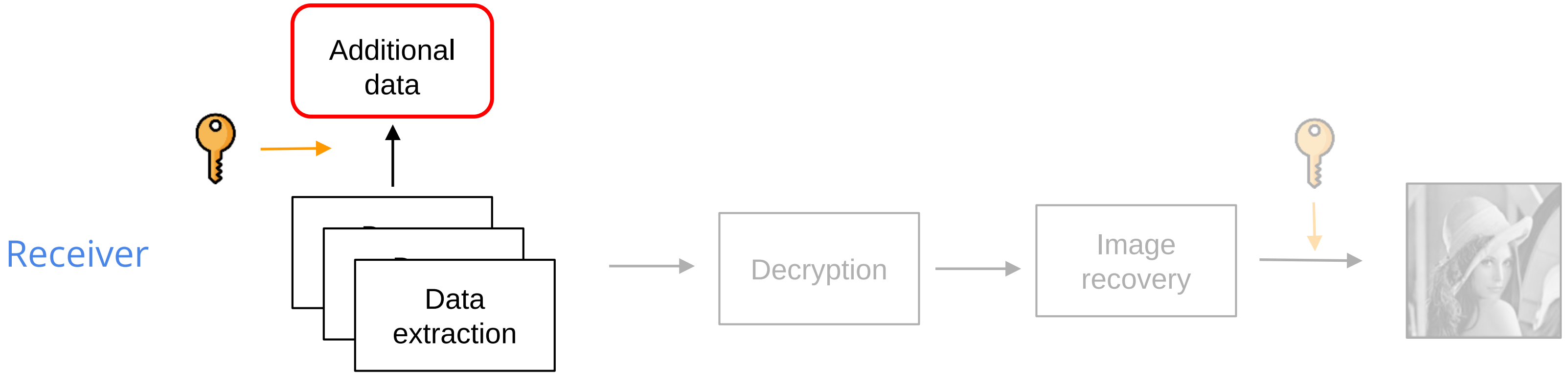
Data embedded
pixel values

Data Extraction and Image Recovery

- Recovery of pixels after secret sharing
- Decryption with the Lagrange method
- Image Recovery
  - key-based
  - zigzag

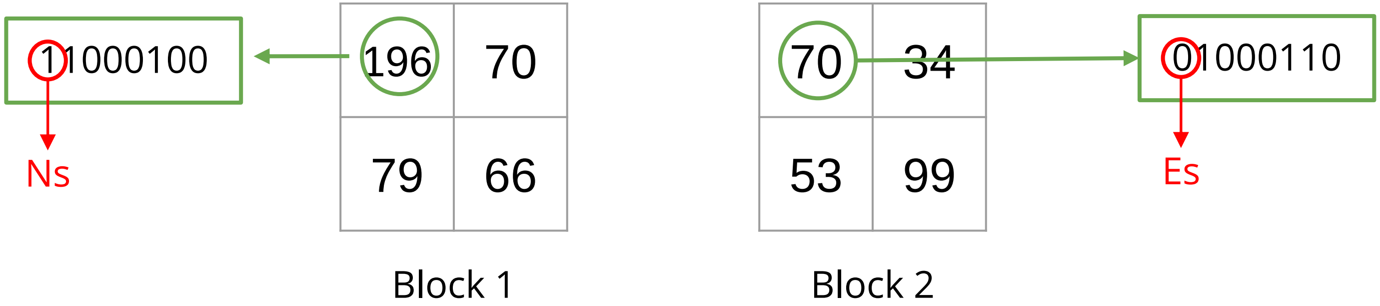


Data extraction



Data extraction

- By Data hiding key, the receiver can extract the additional data
- Step1. Extracted data from Es's EPs
  - According to the MSB of RP, we can know whether the block is ES or NS



Data extraction

- By Data hiding key, the receiver can extract the additional data
- Step1. Extracted data from Es's EPs
  - According to the ε=7, we know that the last five bits represent each part of additional data

70 (RP)	34 (EP)
53 (EP)	99 (EP)

Block 2
(Es)

01000110	00100010
00110101	01100011

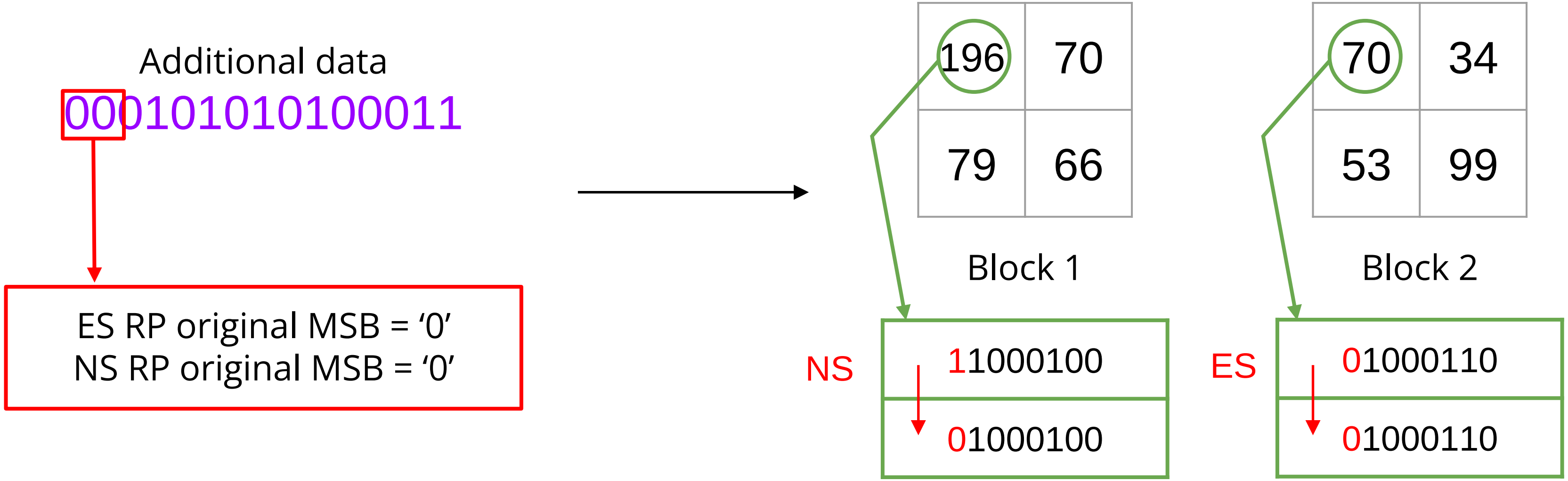
Additional data

(00010)(10101)(00011)

ε'	D_{jz}	u_1	u_2
0	$D_{jz} = 0$	0	8
1	$D_{jz} = 1$	1	7
3	$D_{jz} \le 3$	2	6
7	$D_{jz} \le 7$	3	5
15	$D_{jz} \le 15$	4	4
31	$D_{jz} \le 31$	5	3
63	$D_{jz} \le 63$	6	2
127	$D_{jz} \le 127$	7	1

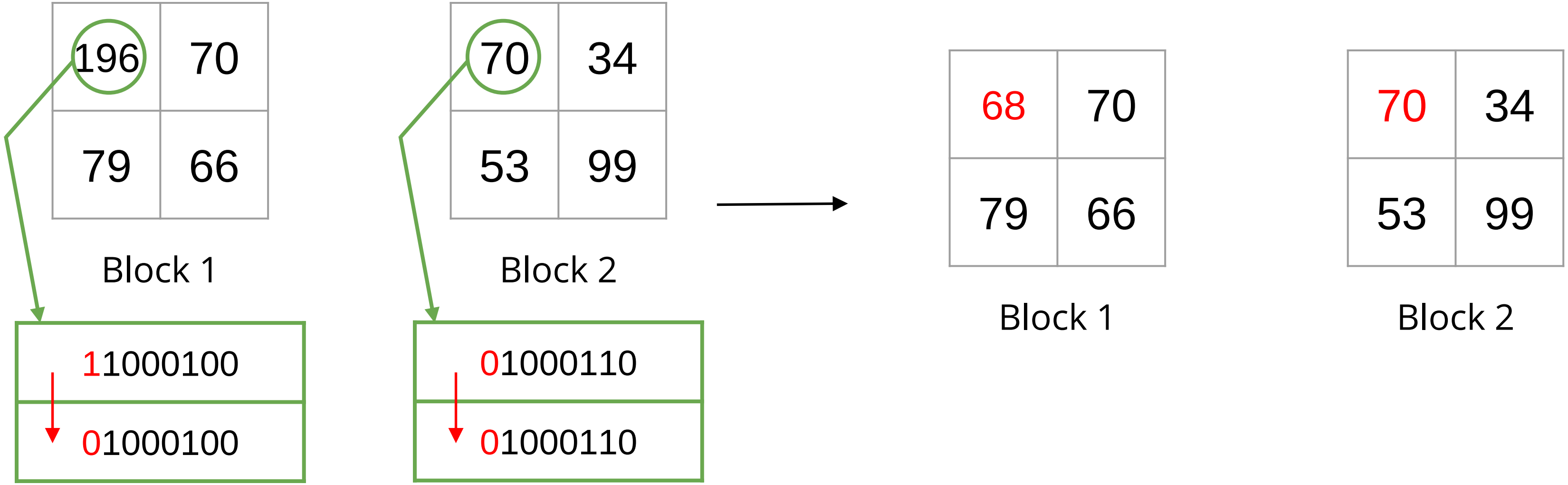
Data extraction

- By Data hiding key, the receiver can extract the additional data
- Step2. the first and second bits in additional data represent the MSB of the RP for Es & Ns



Data extraction

- By Data hiding key, the receiver can extract the additional data
- Step2. the first and second bits in additional data represent the MSB of the RP for Es & Ns



Data extraction

❖ Step3. According to the ε=7, we know that the first three bits of EPs represent the XOR result with RP

70	34
53	99

Block 2
(Es)

→

01000110	00100010
00110101	01100011

→

$(00000001 \oplus 01000110)_{10} = 71$
$(00000001 \oplus 01000110)_{10} = 71$
$(00000011 \oplus 01000110)_{10} = 69$

Data extraction

❖ Data extraction and pixel recovery

Additional data

$\boxed{00}0101010100011$

block 1’s first pixel’s MSB = ‘0’
block 2’s first pixel’s MSB = ‘0’

Block 1

196	70
79	66

Block 2

70	34
53	99

Data embedded pixel values

Block 1

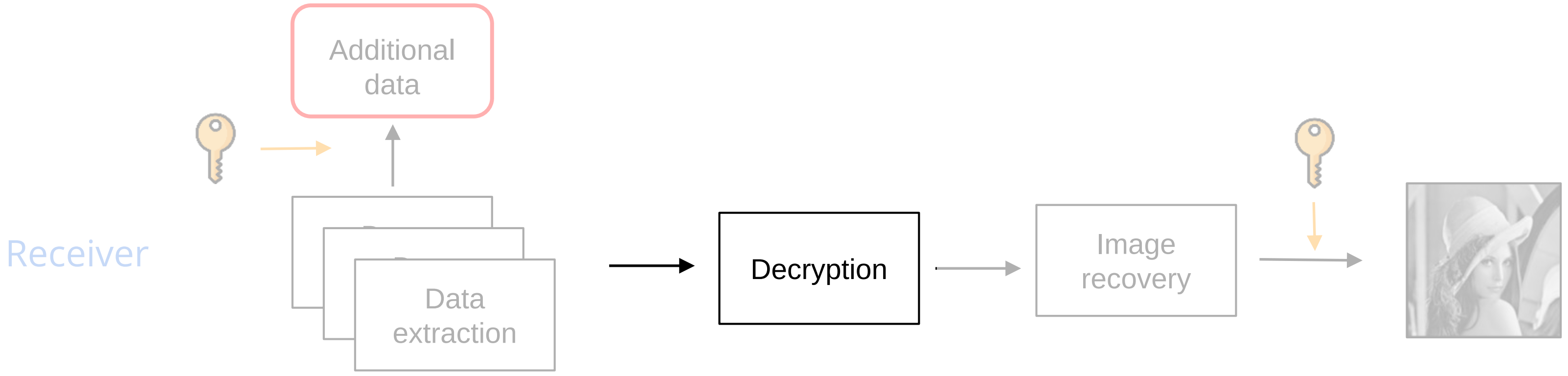
68	70
79	66

Block 2

70	71
71	69

Recovery pixel values

Decryption with the Lagrange method



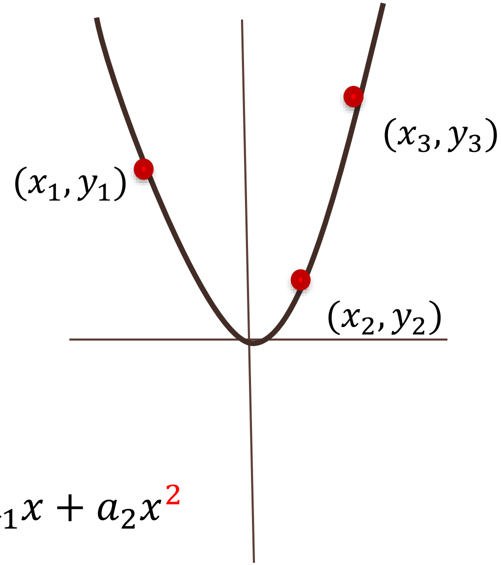
Decryption with the Lagrange method

$$\boxed{\mathrm{f(x)} = s + a_1 x + a_2 x^2}$$

According to Shamir's secret sharing,
when any t or more shares the known x are collected,
the coefficients a of f(x) and the secret message s can be
reconstructed by using a Lagrange interpolation method.

$$f(x) = \sum_{q=1}^{t} \left(f(x_q) \prod_{\substack{1<w\leq t \\ w\neq q}} \frac{x - x_w}{x_q - x_w} \right)$$

$$s = f(0) = \sum_{q=1}^{t} \left(f(x_q) \prod_{\substack{1<w\leq t \\ w\neq q}} \frac{- x_w}{x_q - x_w} \right)$$



$$f(x) = f(x_1)\frac{x - x_2}{x_1 - x_2}\frac{x - x_3}{x_1 - x_3} + f(x_2)\frac{x - x_1}{x_2 - x_1}\frac{x - x_3}{x_2 - x_3} + f(x_3)\frac{x - x_1}{x_1 - x_3}\frac{x - x_2}{x_2 - x_3} = s + a_1 x + a_2 x^2$$

Decryption with the Lagrange method

As our scheme, the degree of polynomial t-1 = 2, n = 4

With GF(2^8) and the irreducible polynomial p(x)

$$f(x) = (s \oplus a_1x \oplus a_2x^2) \; mod \; p(x)$$

$$p(x) = x^8 + x^4 + x^3 + x + 1$$

We only need any 3 shares to recover the secret

$$f(x) = \sum_{q=1}^{t} \left(f(x_q) \prod_{\substack{1<w\le t \\ w\neq q}} \frac{x - x_w}{x_q - x_w} \right) mod \; p$$

$$s = f(0) = \sum_{q=1}^{t} \left(f(x_q) \prod_{\substack{1<w\le t \\ w\neq q}} \frac{-x_w}{x_q - x_w} \right)$$

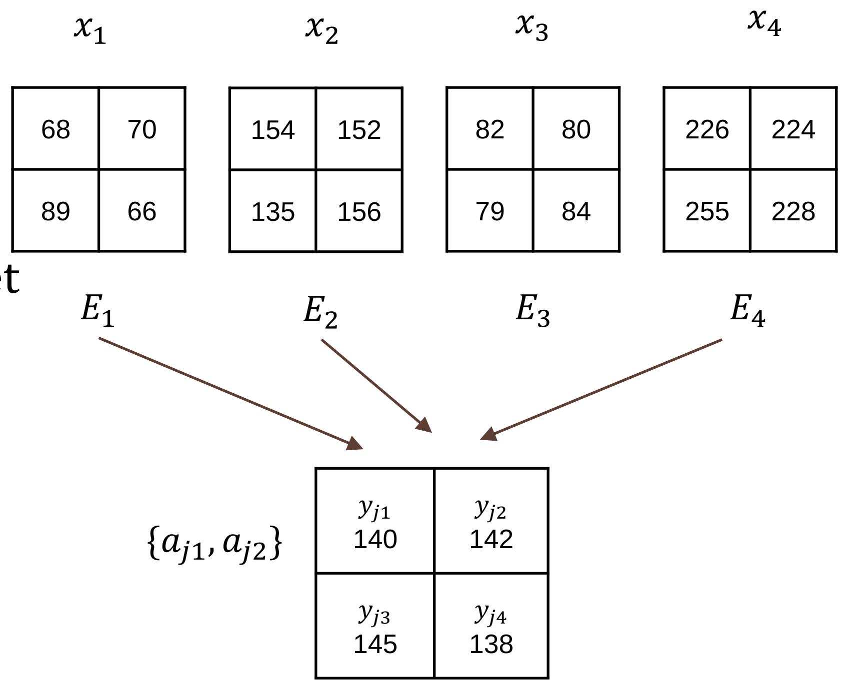


Image Recovery

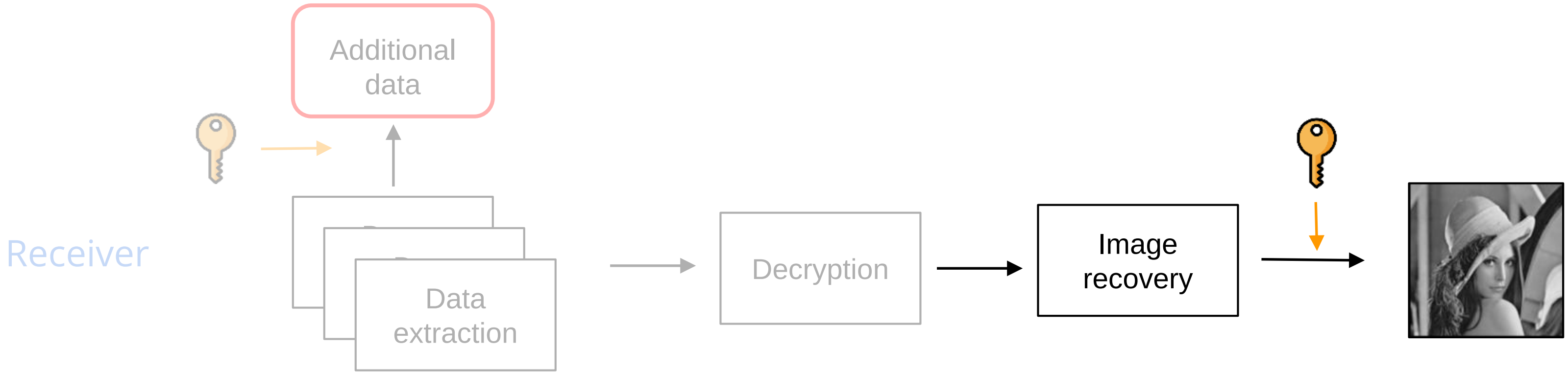
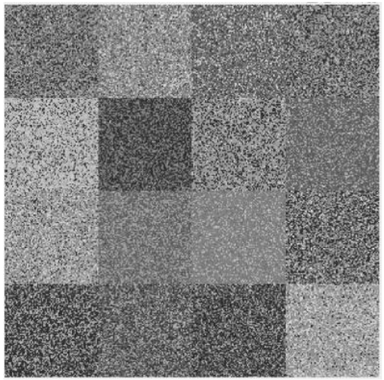


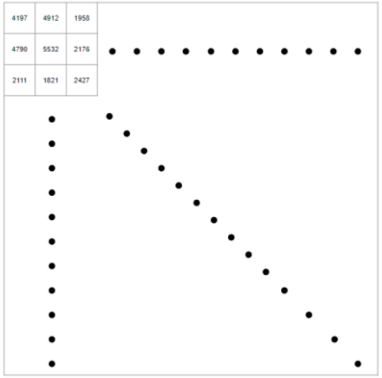
Image Recovery: key-based

Encryption key

Block Scramble Order

Pixel Scramble Order





Pixel Scramble Order

Image Recovery: key-based

Encryption key

Block Scramble Order

Pixel Scramble Order

16	15	3	12
14	10	1	9
6	13	5	2
4	8	11	7

Block Scramble Order

Image Recovery: zigzag

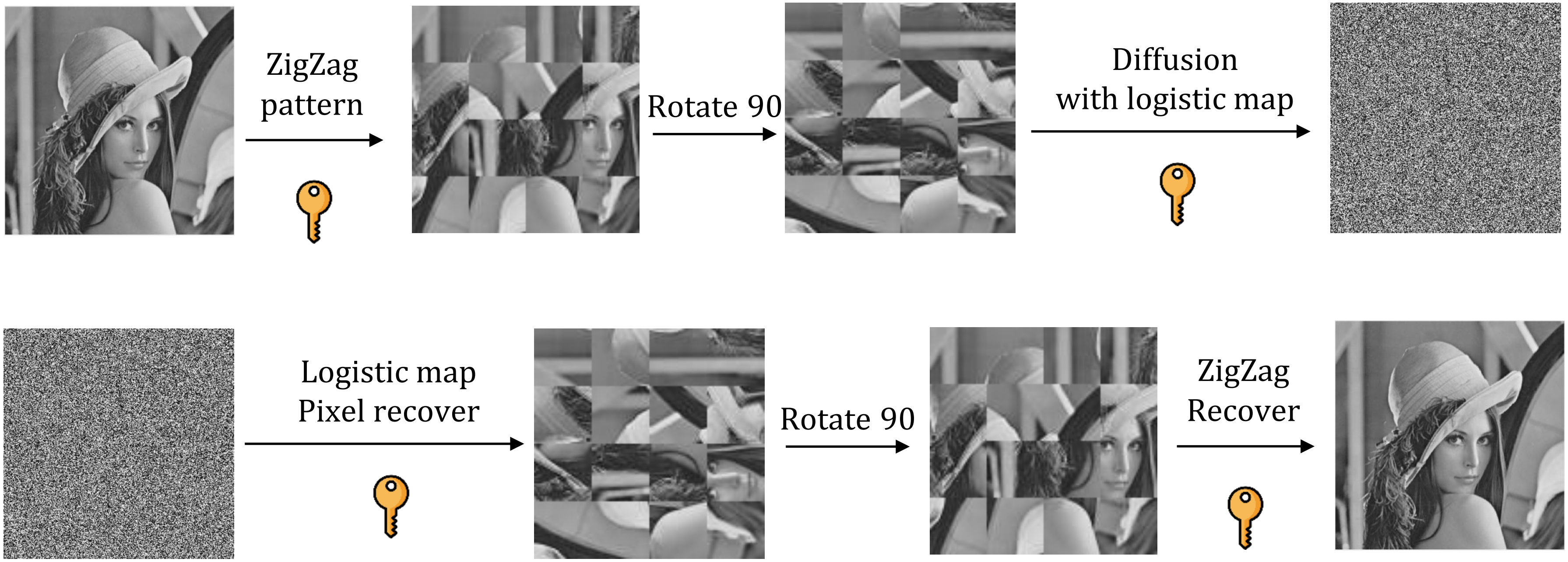
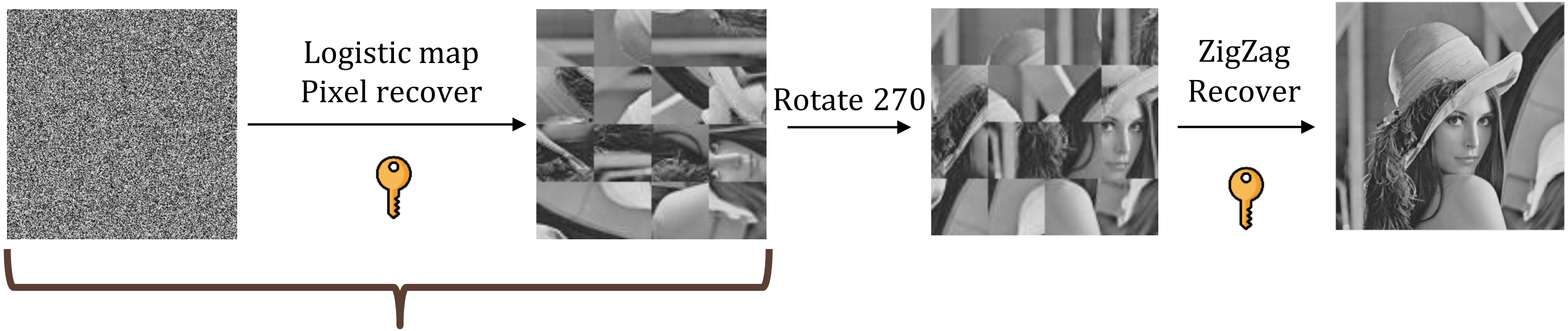


Image Recovery: zigzag



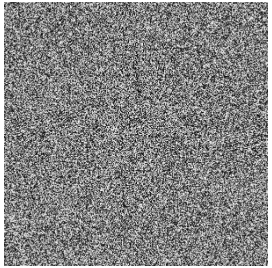
In image scrambling process, XOR operation between the key K and the zigzag scramble image to get the encrypted image.

$$E = X \oplus K$$

Therefore, with the key K, we can recover the scrambling image with

$$X = E \oplus K$$

Image Recovery: zigzag



Logistic map
Pixel recover

Rotate 270

ZigZag
Recover

1	6	10	8
2	3	13	12
5	4	14	15
9	7	11	16

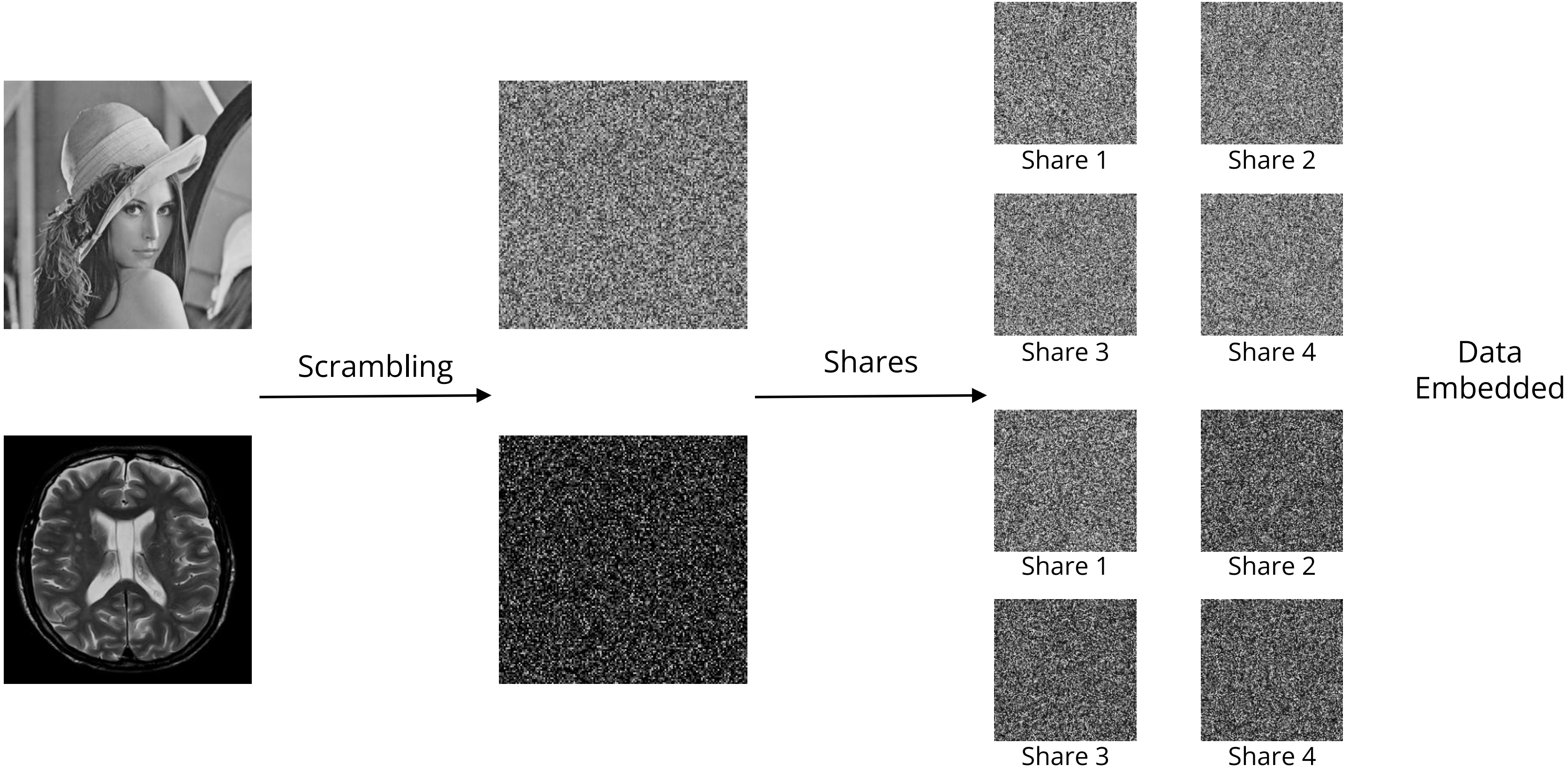
Index Sort

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

With the key of zigzag pattern, we can sort the pattern to get the sorted order of image.

Therefore, with the order key, we can rearrange each block to the original position.

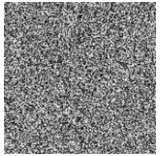
Result Demo

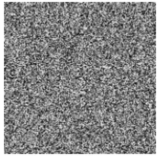


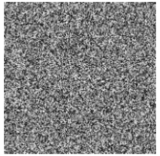
Scrambling
Shares
Share 1
Share 2
Share 3
Share 4
Data Embedded
Share 1
Share 2
Share 3
Share 4

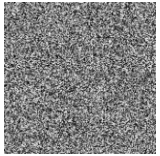
Result Demo

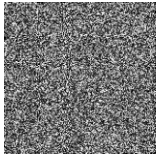
Data Extraction

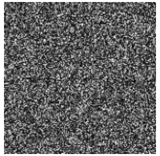

Share 1

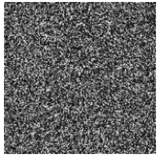

Share 2

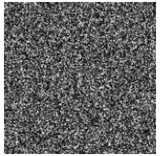

Share 3


Share 4

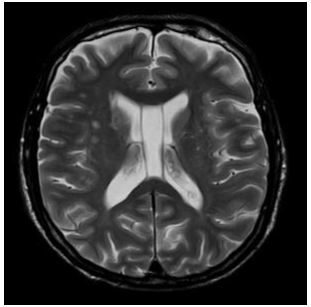

Share 1


Share 2


Share 3


Share 4

Decryption



Result Demo SSIM

Key based scrambling

Image \ Compare	Decrypted image	scrambled_image	share 1	share 2	share 3	share 4
Lena	1	0.0633	0.0417	0.0409	0.0547	0.0432
MRI_Brain	1	0.046	0.0257	0.0344	0.0333	0.0358

ZigZag scrambling

Image \ Compare	Decrypted image	scrambled_image	share 1	share 2	share 3	share 4
Lena	1	0.0336	0.0334	0.0348	0.0327	0.0347
MRI_Brain	1	0.0259	0.0263	0.0264	0.0250	0.0242

Result Demo PSNR

Key based scrambling

Compare / Image	Decrypted image	scrambled_image	share 1	share 2	share 3	share 4
Lena	Inf	11.5604	10.3183	10.3299	10.3948	10.3223
MRI_Brain	Inf	10.8464	7.1114	8.1534	8.1534	8.1699

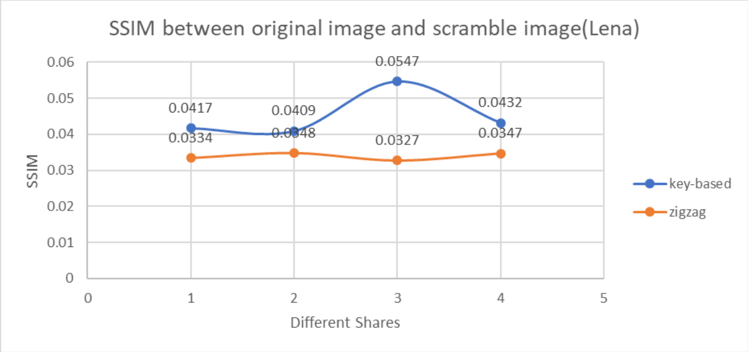
ZigZag scrambling

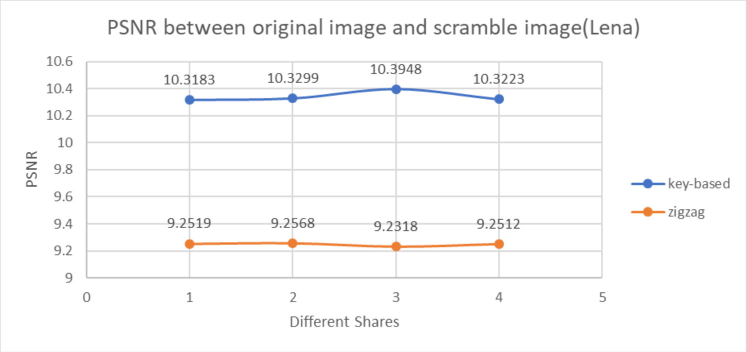
Compare / Image	Decrypted image	scrambled_image	share 1	share 2	share 3	share 4
Lena	Inf	9.2393	9.2519	9.2568	9.2318	9.2512
MRI_Brain	Inf	10.8464	6.4724	6.4725	6.4609	6.4618

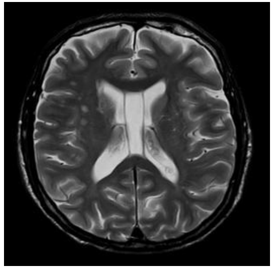
Result Demo

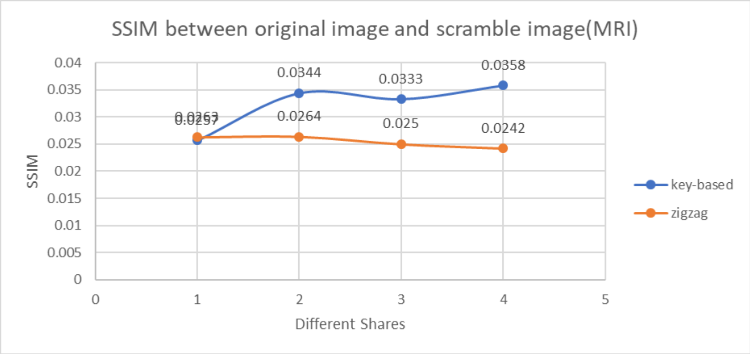
SSIM

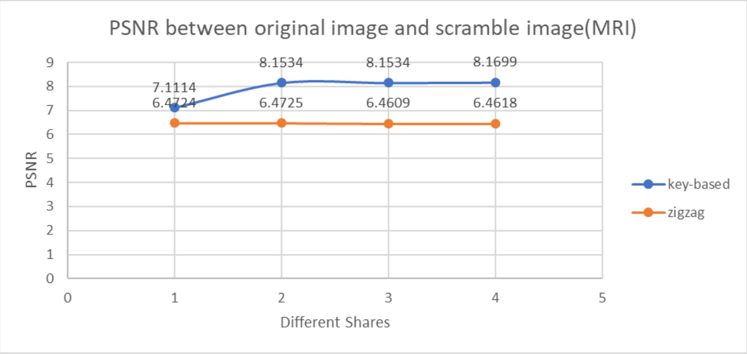
PSNR











Reference

[1] C. Qin, C. Jiang, Q. Mo, H. Yao and C. -C. Chang, "Reversible Data Hiding in Encrypted Image via Secret Sharing Based on GF(p) and GF(2^8)," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 32, no. 4, pp. 1928-1941, April 2022

[2] S. T. Kamal, K. M. Hosny, T. M. Elgindy, M. M. Darwish and M. M. Fouda, "A New Image Encryption Algorithm for Grey and Color Medical Images," in IEEE Access, vol. 9, pp. 37855-37865, 2021

[3] https://medium.com/taipei-ethereum-meetup/%E7%A7%81%E9%91%B0%E5%88%86%E5%89%B2-shamirs-secret-sharing-7a70c8abf664

[4] http://rportal.lib.ntnu.edu.tw/handle/20.500.12235/98585

[5]https://www.tcrc.edu.tw/TANET2013/paper/M12-681-1.pdf