A NEW VERIFIABLE SECRET SHARING SCHEME ON IMAGES

Prepared by:

- Chitradeep Dutta Roy
- Neel Choudhury

Department of Computer Science And Engineering

Institute of Engineering and Management, Kolkata

Secret sharing

A Bank ABC has:

- A vault with password P
- 1 President and 5 Managers
- None of the manager is fully trustworthy

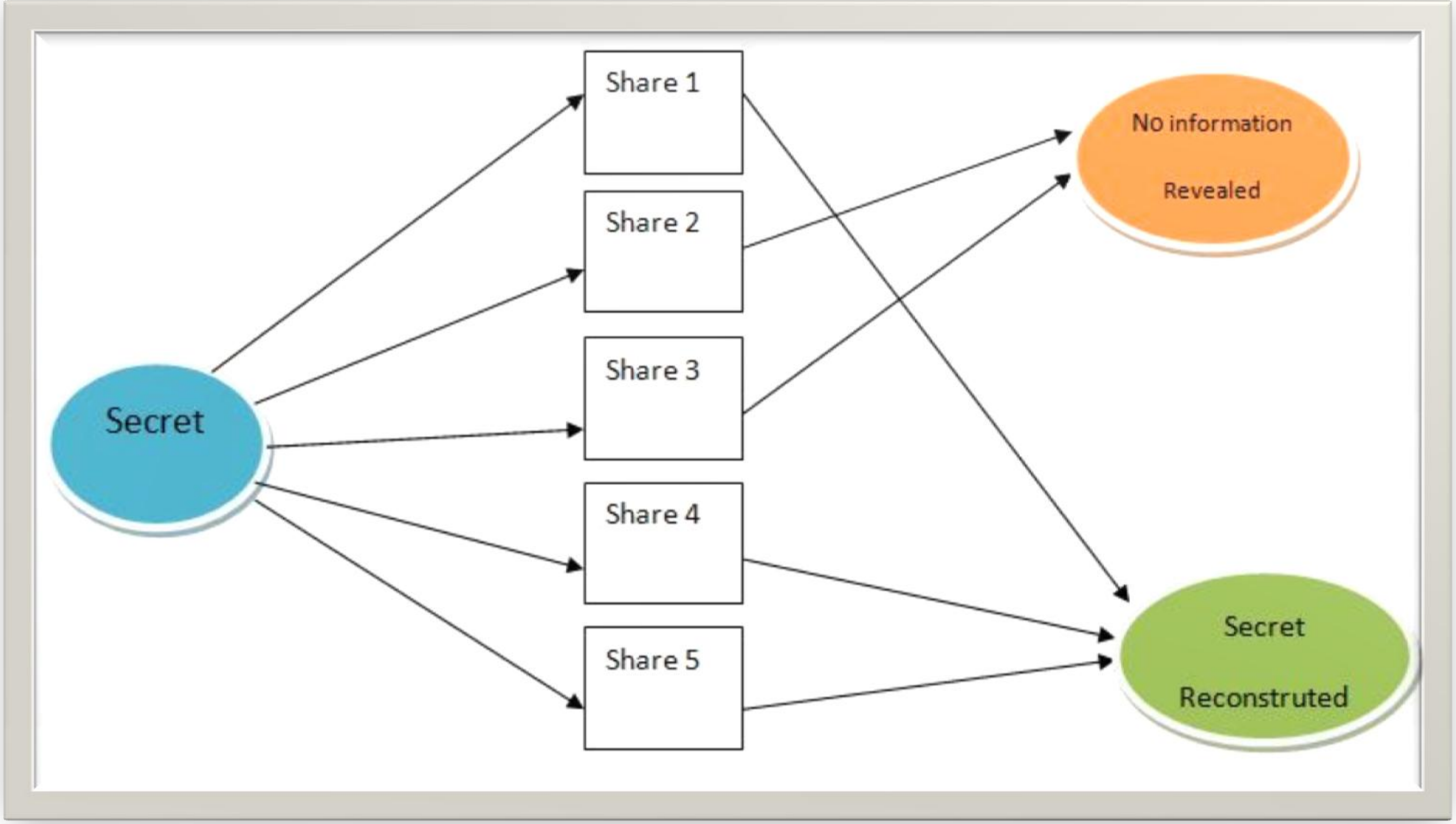
- Scheme has to be developed so that the vault can be opened in the absence of president

Two solutions

1. Providing each manager with a key such that when keys from all the managers are combined the password P will be revealed

2. Providing each manager with a key such that when keys from any 3 of the managers are combined the password P will be revealed

(3,5)Threshold Secret Sharing scheme



(k, n) Threshold Secret Sharing scheme

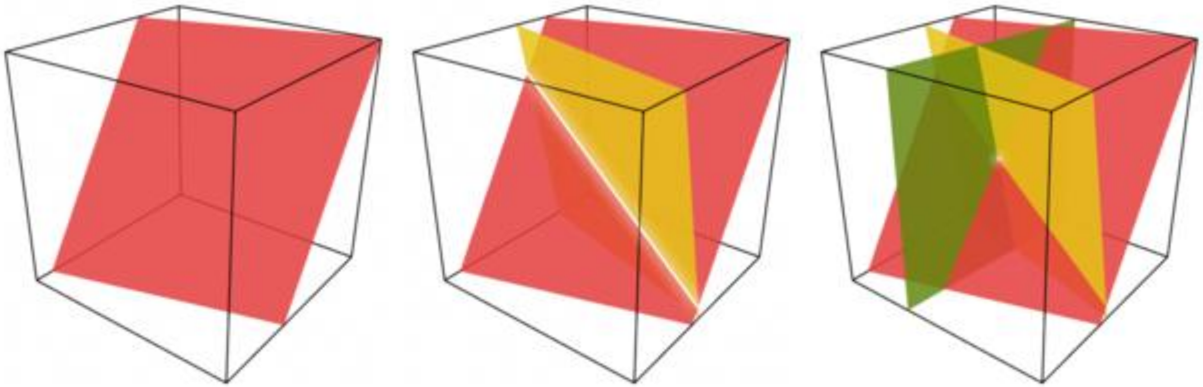
- In a (k, n) threshold scheme a secret s is divided into n shares in such a manner so that:
  - Any k out of the n share can reconstruct the secret S
  - With (k-1) or less shares a absolutely no information about the secret is revealed

- Scheme was devised independently by Adi Shamir and George Blakley in 1979

Underlying Concepts

The Scheme is based on a simple property of linear algebra that:

A System of n linear equations having k variables can be solved by any k out of the n equations



Outline of Secret Distribution

- Secret (s) is composed of data elements ($s_1,s_2,s_3,...s_p$) where p is the length of the secret string.

- Now secret (s) is divided into (p/k) blocks of length k each e.g. B_1 {$s_1,s_2,...s_k$} , B_2{s_{k+1}, $s_{k+2,}$ ... , s_{2k}} ..... and so on.

Outline of Secret Distribution

- A linear system of n equations is formed using k elements from each block as variables.

  $x_1 = a_{11}*s_1 + a_{12}*s_2 + \dots\dots\dots + a_{1k}*s_k$

  $x_2 = a_{21}*s_1 + a_{22}*s_2 + \dots\dots\dots + a_{2k}*s_k$

  $\vdots$

  $x_n = a_{n1}*s_1 + a_{n2}*s_2 + \dots\dots\dots + a_{nk}*s_k$

  where $(a_{i1}, a_{i2}, \dots\dots , a_{ik})$ are random coefficients generated by random number generators taking i as seed.
- x_i is given to ith shareholder.

Outline of Secret Reconstruction

- For reconstruction we use a standard equation solving algorithm to solve the following equations
- The values of s_i s reveal the original secret

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ & \vdots & \vdots & \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_k \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{pmatrix}$$

Verifiable Secret Sharing Scheme

Scenario :

- Dishonest shareholders or participants
- Invalid share submission in an attempt to get an idea about the secret or to prevent the combiner from deciphering it.

Requirement to stop such intrusion :

- A scheme to check the authenticity of each share before reconstructing the secret

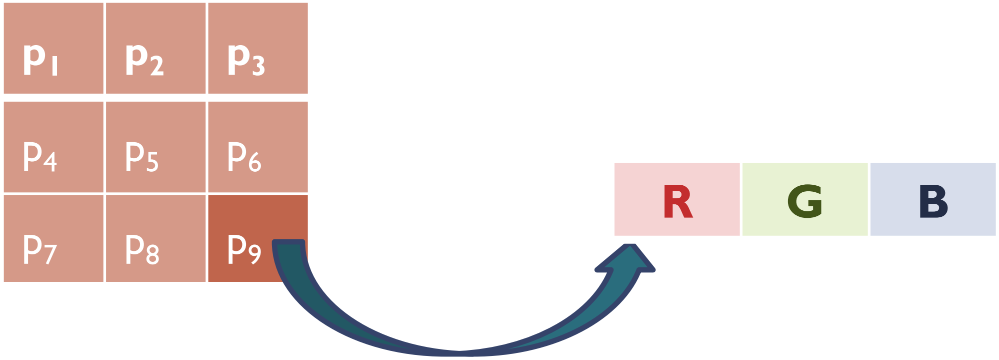
Our Work

- Extended the idea of (k, n) threshold scheme onto 24-bit bitmap images although any popular format(e.g. jpeg, png) can also be used

- Verification scheme is incorporated by embedding a secret code into each share by watermarking

Implementation on Images I

- *Structure of 24-bit bitmap image :*



  - Consists of a header and an array of pixels
  - Each pixel has 3 bytes (3 * 1 byte blocks for R G & B)

- Each such byte is taken as data element (value lying within [0-2^8)) for the secret image.

Implementation on Images 2

- The data elements or each byte is then fed into equation of the type

$$x_i = a_{i1}*s_1 + a_{i2}*s_2 + \ldots\ldots\ldots + a_{ik}*s_k$$

Value of

- s_j lies between the range $[0-2^8)$ as s_j is 1 byte taken from pixel
- a_{ij} s are chosen randomly from a domain of [1-256]

- considering k less than 128 it is easily determined that x_i will always be less than 2^{23} $(2^8 * 2^8 * 2^7)$

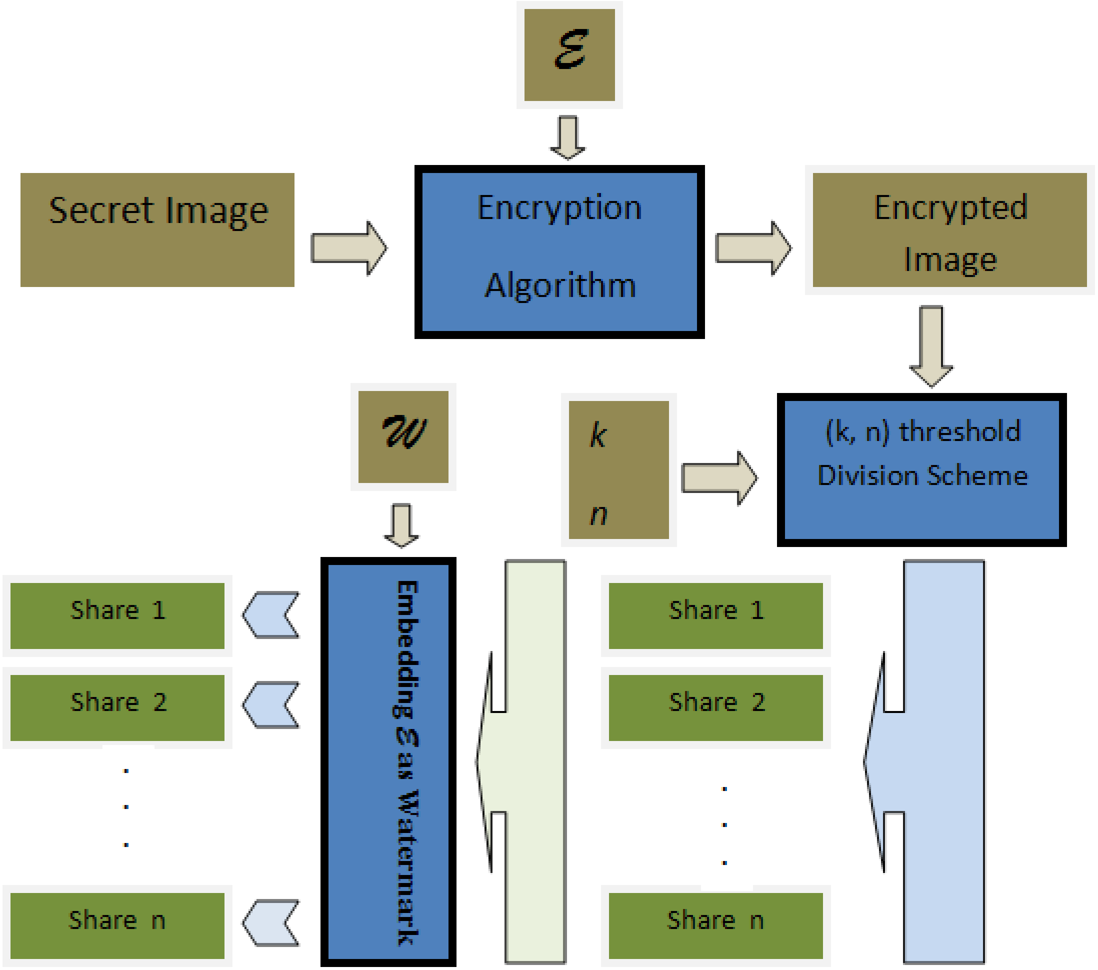
Implementation on Images 3

- x_i is stored in i^{th} image share as a pixel of 24bits where the LSB(1 bit) is intentionally kept free for watermarking

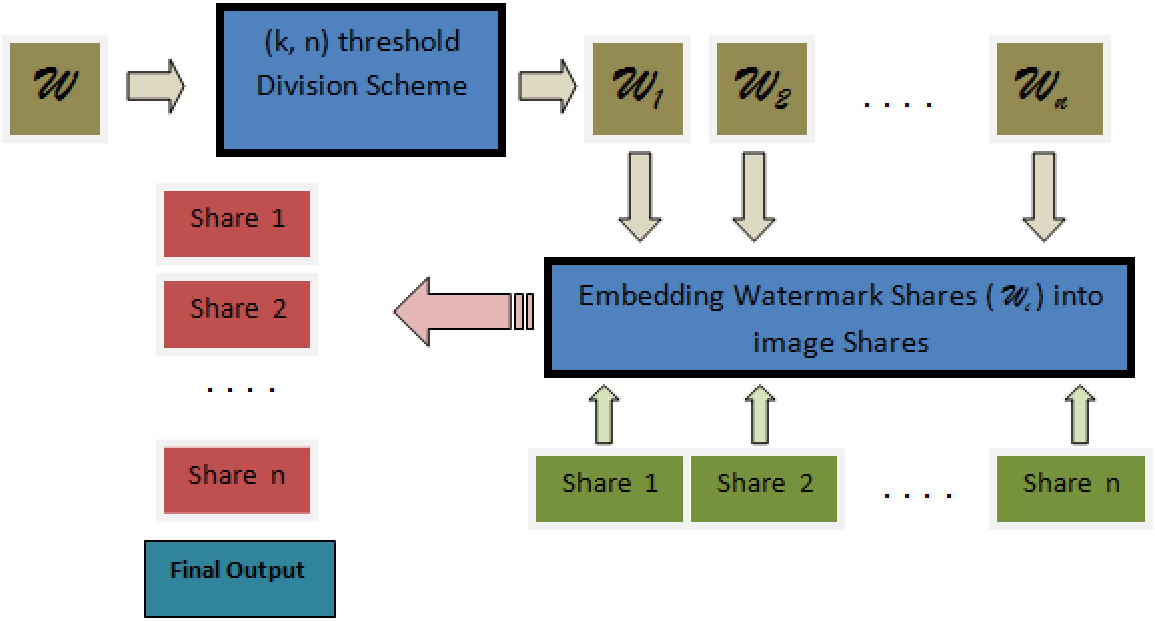
- From k bytes of the secret image 1 pixel of each image share is computed so if k increases the size of each share decreases

$$So\ \ sharesize \propto \frac{1}{k}$$

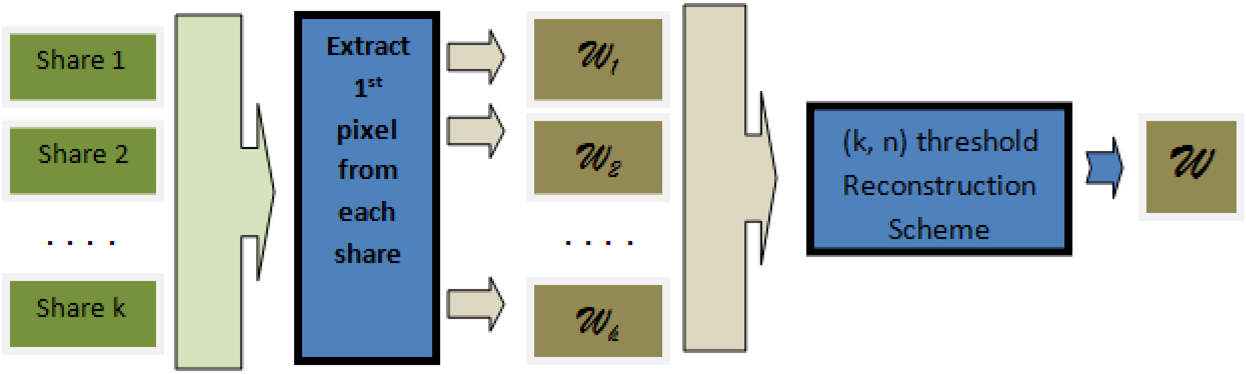
Methodology : Distribution I



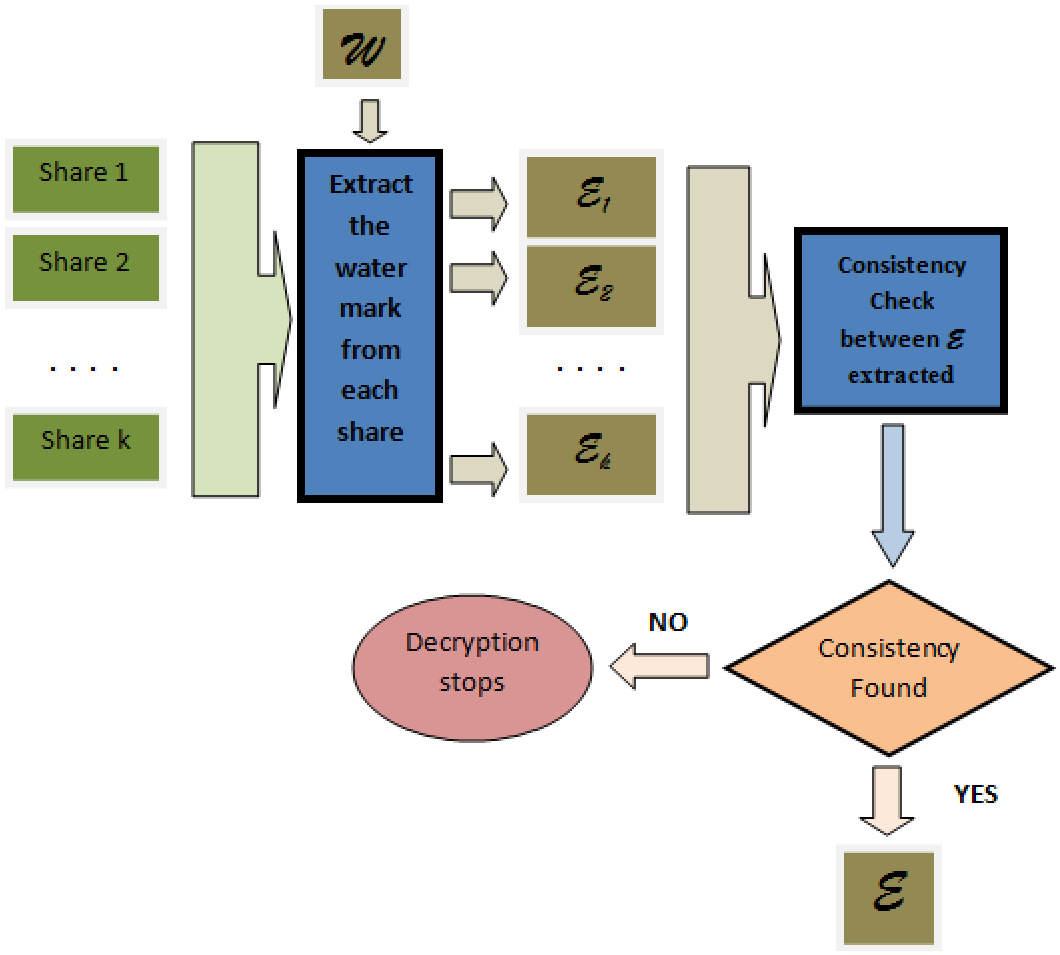
Methodology : Distribution 2



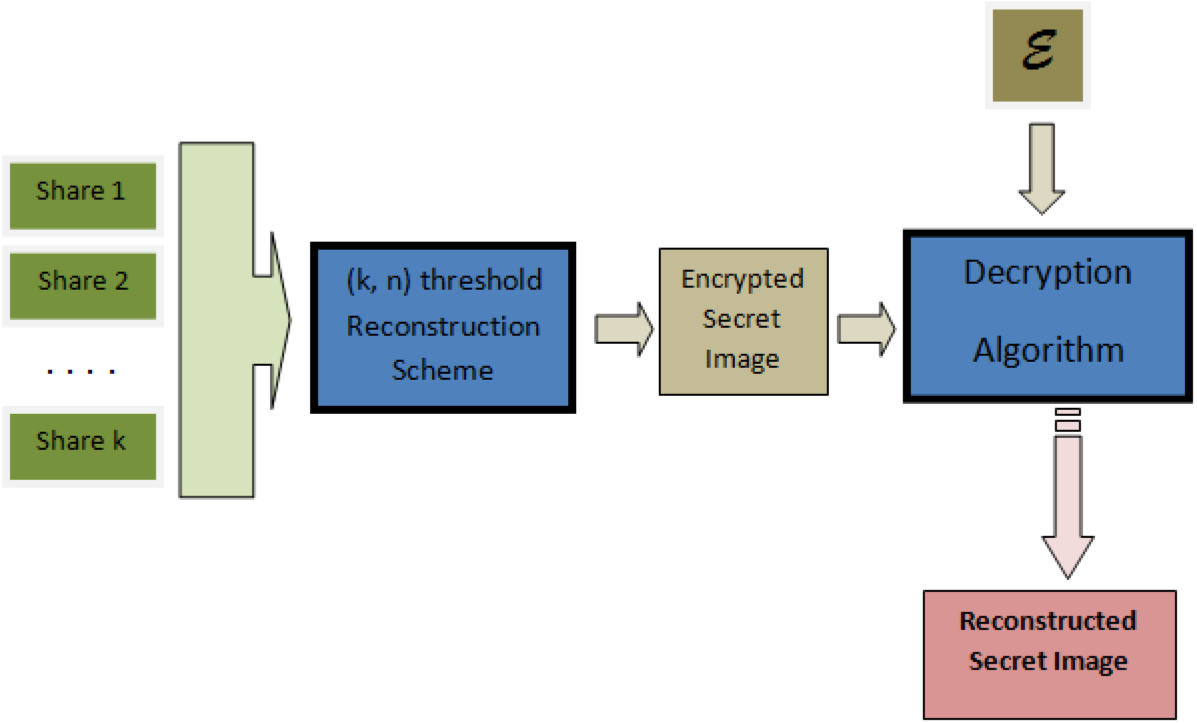
Methodology : Reconstruction 1



Methodology : Reconstruction 2



Methodology : Reconstruction 3

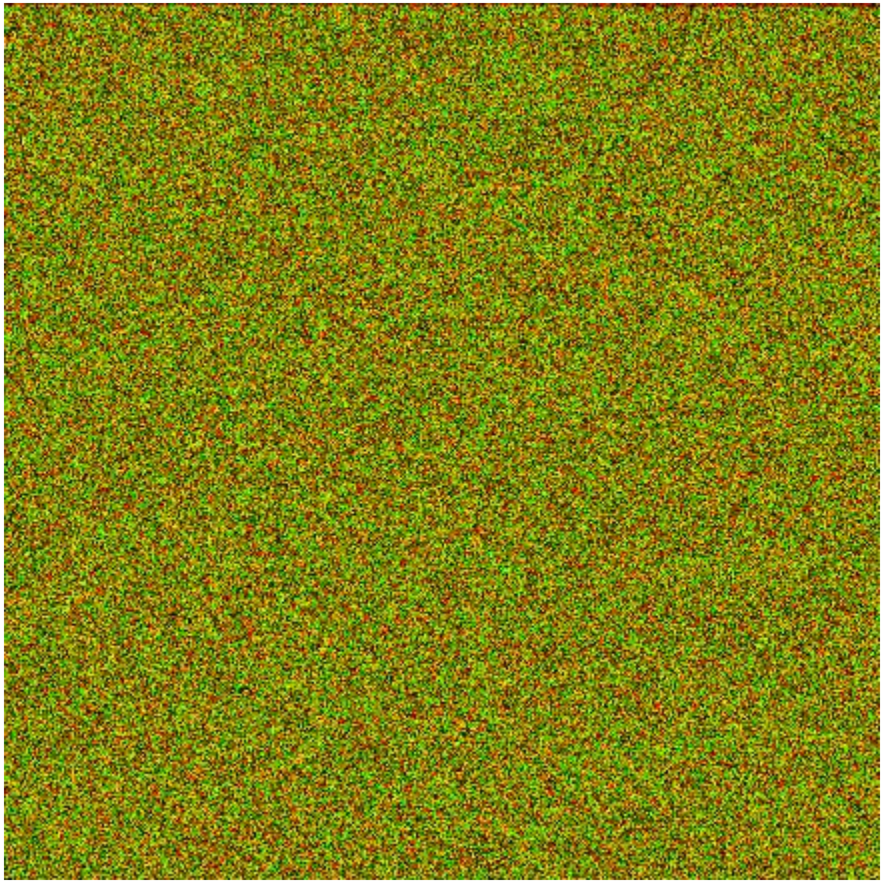


Experimental Data

- Original Image(450 X425)

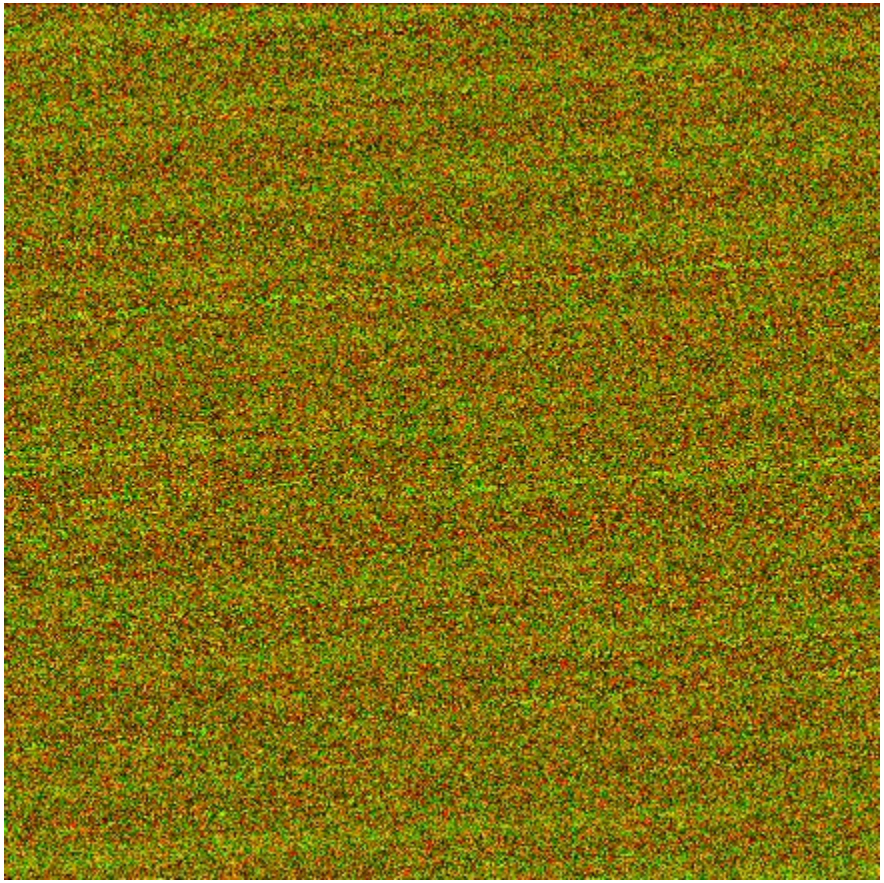
Shares Generated

- Share 1(438 X 438)



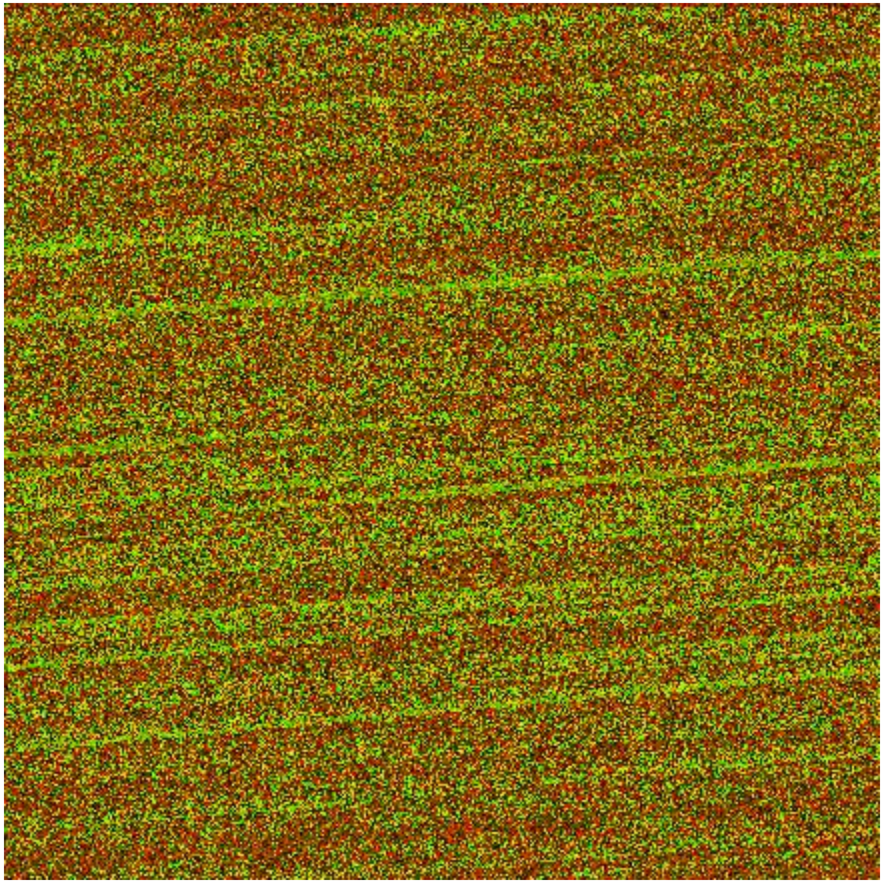
Shares Generated

- Share 2(438 X 438)



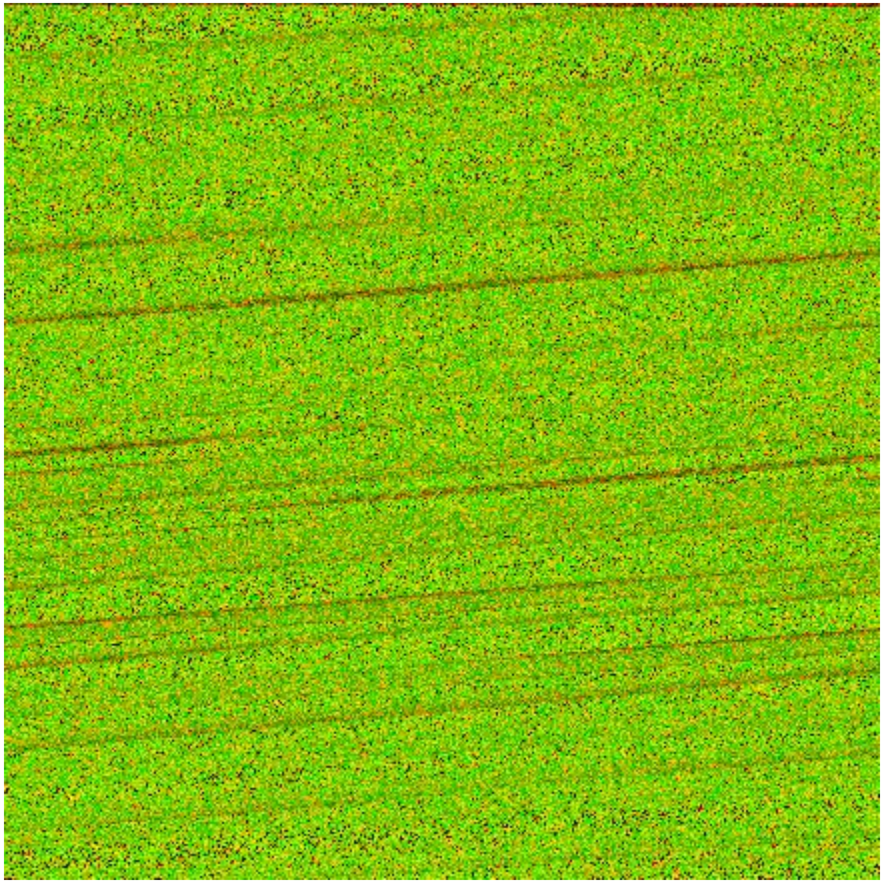
Shares Generated

- Share 3(438 X 438)



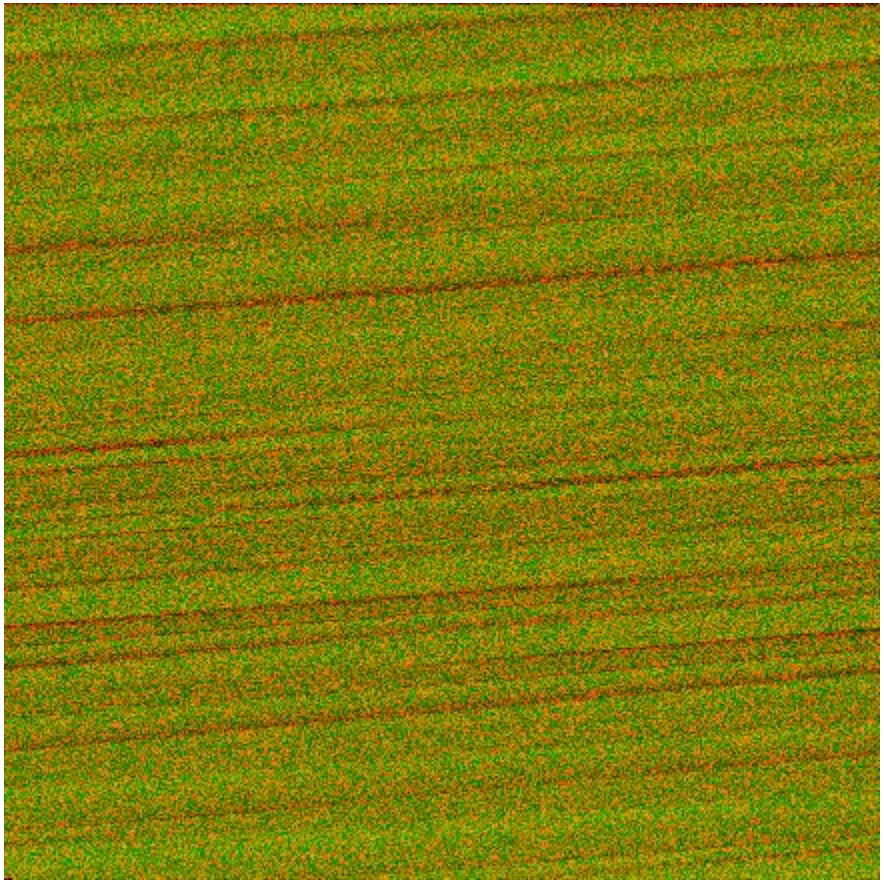
Shares Generated

- Share 4(438 X 438)

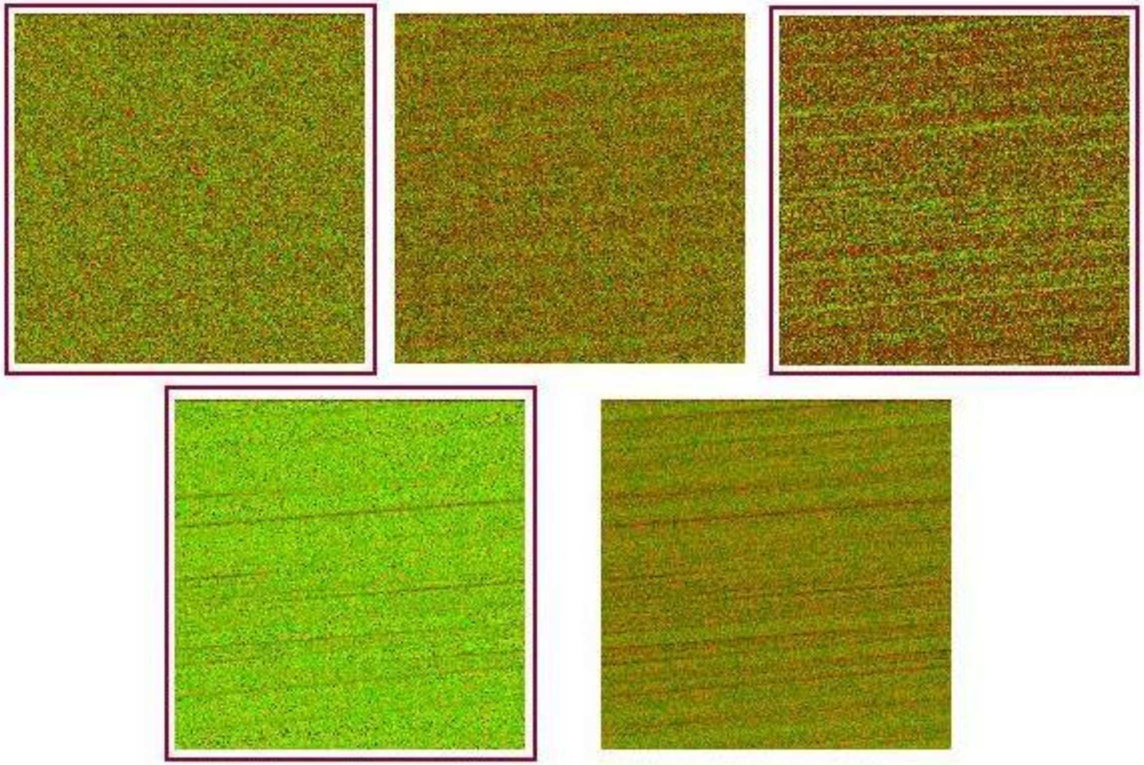


Shares Generated

- Share 5(438 X 438)



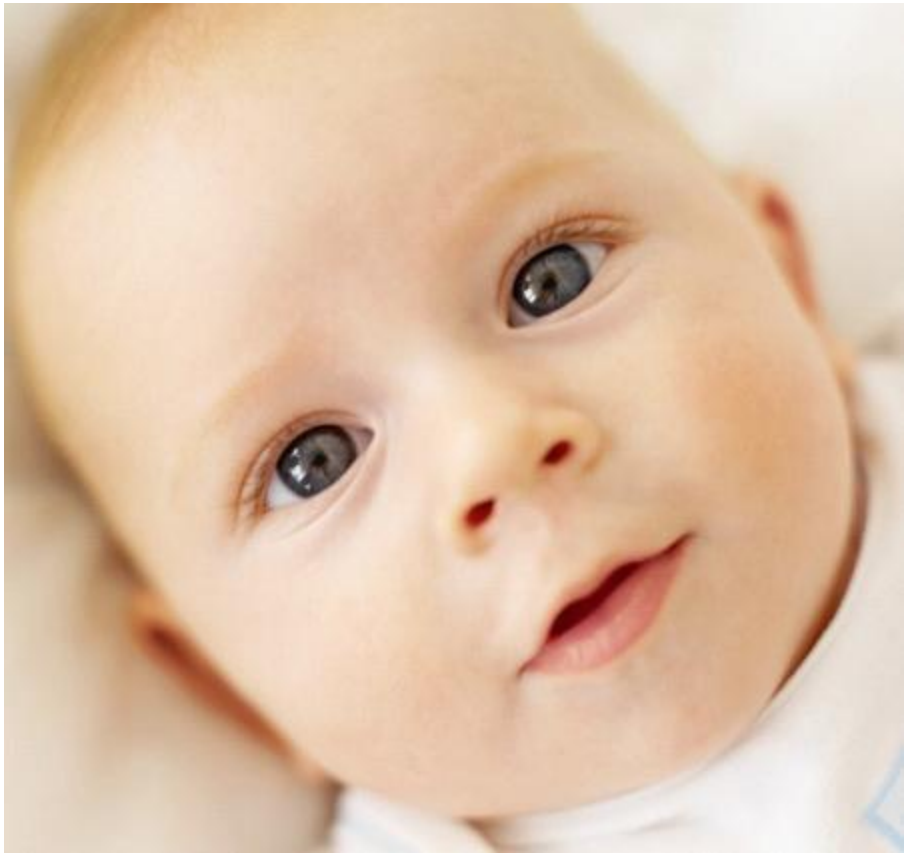
5 Image Shares obtained



Reconstructed Image

- The reconstructed image from share 1,3 and 4.

Resolution: 450X425(same as the original image)



Applications

- The scheme provided will have opportunities in the following fields

- Military organizations
- High security Research Labs
- Financial firms
- E-voting

Thank You

- Acknowledgement :
  - Professor Atal Chowdhury
  - Department of Computer Science, Jadavpur University