A software implementation of Shamir's secret sharing scheme

RUAN Keda
Advised by Prof. Cunsheng Ding

Introduction

Digital images are widely used in modern society, and protecting images that contain confidential information becomes necessary. Images encryption by using secret sharing scheme is an ideal way for protecting images.
In this project, we implement Shamir's Secret Sharing Scheme over finite field $\mathbb{F}_{2^m} = GF(2^m), 8 \leq m \leq 64$, and build a web application for image and text sharing.

Shamir's Secret Sharing scheme

- Introduced by Adi Shamir (1979)
- Is a (t, n)-Threshold Scheme
- n participants hold shares partitioned from the secret S.
- Recoverability: any t shares can recover the secret S completely.
- Secrecy: any $t - 1$ or less shares cannot recover the secret S.

What is Finite Field?

The finite field (or, Galois field) can be regarded as a set of numbers where arithmetic operations of addition, subtraction, multiplication, and division (multiplicative inverse) can be carried out without error.

Methodology

Secret Reconstruction

To partition the secret S, let $S = a_0$ and we pick random $a_1, \dots, a_{k-1}$ from $\mathbb{F}_{2^m}$ to form $f(x) = S + a_1 x + a_2 x^2 + \cdots + a_{k-1} x^{k-1}$, and shares $(x_i, S_i = f(x_i))$ can be obtained.

$$\begin{bmatrix} 1 & x_1 & \cdots & x_1^{k-1} \\ 1 & x_2 & \cdots & x_2^{k-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_k & \cdots & x_k^{k-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{k-1} \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_k \end{bmatrix}$$

Why Finite Field?

Finite field is a very important concept in computer security, and it is related most of the cryptography methods. Because the calculation in finite field makes no error, this property makes it ideal for cryptography since cryptography requires no error.

Secret Reconstruction

Given (x_i, S_i) to reconstruct the secret S, we only need to solve the previous matrix. But the calculation of matrix inverse in finite field $\mathbb{F}_{2^m}$ is difficult, we use Lagrange Interpolation to sovle:

$$f(x) = \sum_{i \in \mathcal{G}} S_i \prod_{j \in \mathcal{G}, j \neq i} \frac{(x - x_j)}{(x_i - x_j)}$$

And the secret S is $f(x)$ at $x = 0$:

$$S = f(0) = \sum_{i \in \mathcal{G}} S_i \prod_{j \in \mathcal{G}, j \neq i} \frac{-x_j}{(x_i - x_j)}$$

Image Partition Flow Chart

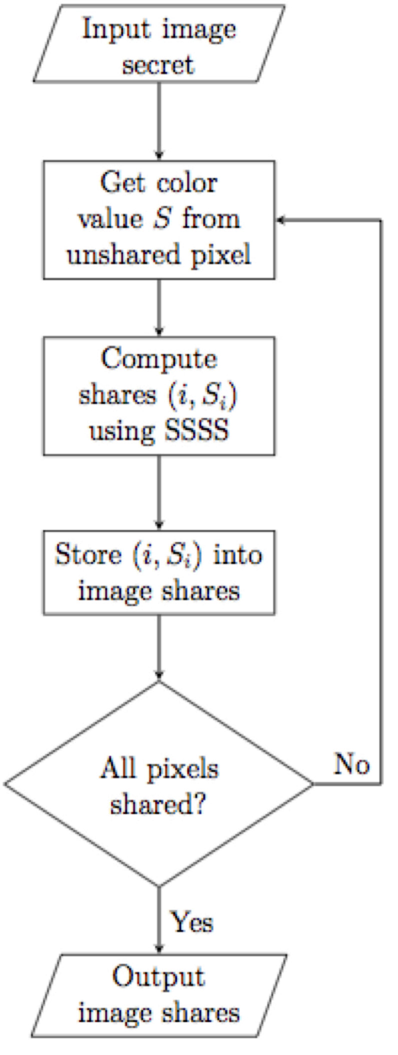
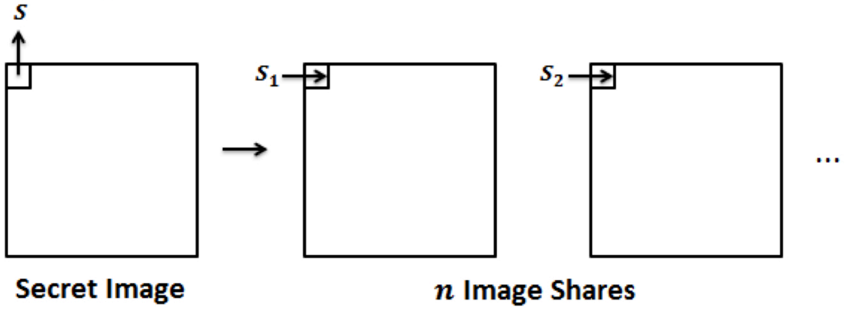
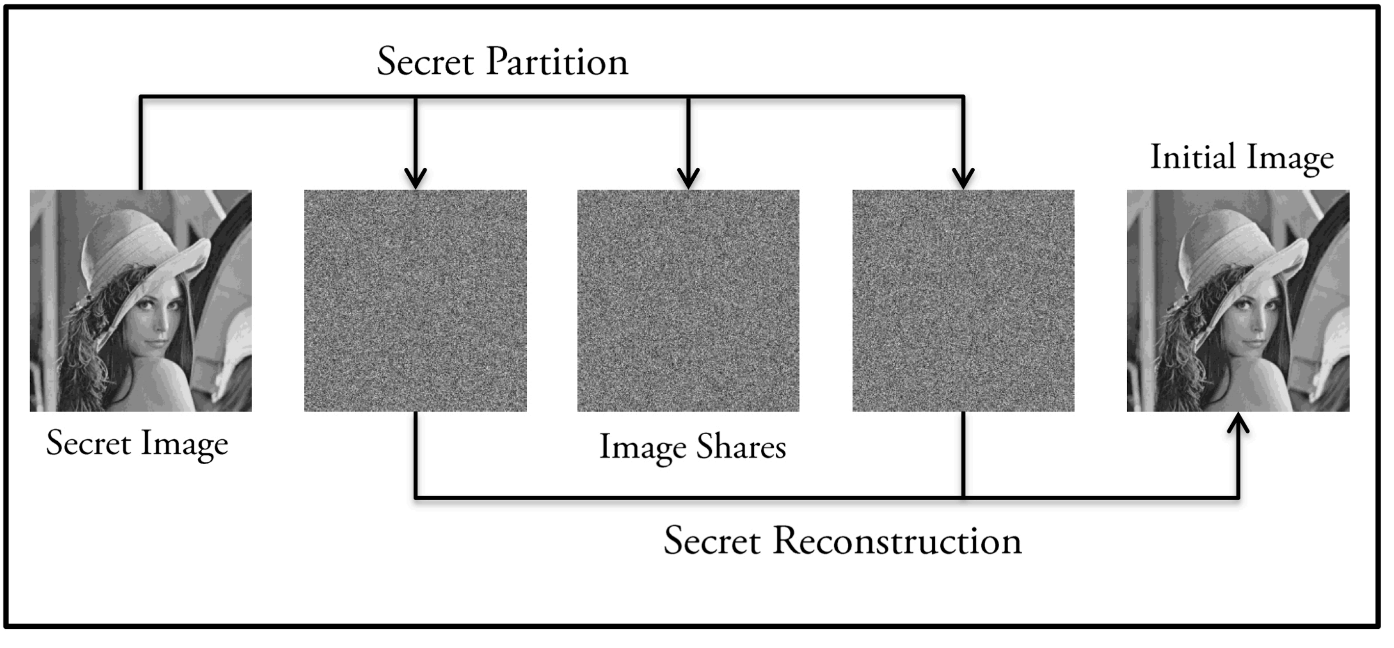


Image Partition Procedure



Example For Image Sharing

Choose $n = 3, k = 2$



Technical Challenge

Problem – Heavy Computational Cost

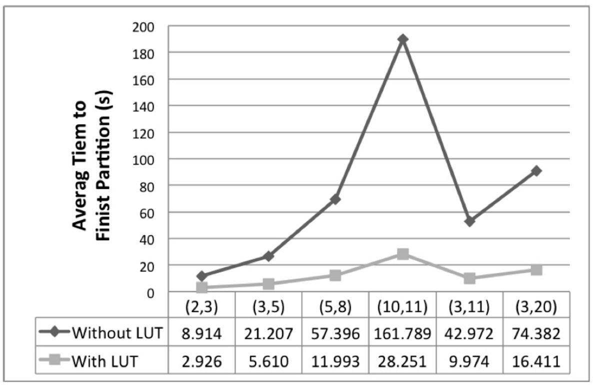
Secret partition: $O(nk)$ finite field arithmetic operations

Secret reconstruction: $O(k^2)$ finite field arithmetic operations

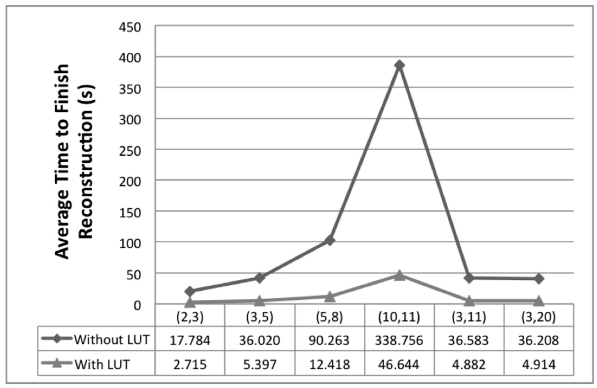
Solution – Lookup Table Method

The results for finite field multiplication and multiplicative inverse are stored in advance, the it only take $O(1)$ time to obtain the result. The heavy computation cost are eliminated.

Evaluation



	(2,3)	(3,5)	(5,8)	(10,11)	(3,11)	(3,20)
Without LUT	8.914	21.207	57.396	161.789	42.972	74.382
With LUT	2.926	5.610	11.993	28.251	9.974	16.411



	(2,3)	(3,5)	(5,8)	(10,11)	(3,11)	(3,20)
Without LUT	17.784	36.020	90.263	338.756	36.583	36.208
With LUT	2.715	5.397	12.418	46.644	4.882	4.914

From the result, we can find that the Lookup Table method increase the speed for image sharing dramatically.

Example For Text Sharing

Choose $n = 3, k = 2, m = 20$

text_test.txt: This is secret!
text_test_share_1.txt: fd91f083d0b2d7c6cc12f92467827d0 dfae5cafc1260c7d5a51623e0054c00 cbd0b83b851b3
text_test_share_2.txt: fb2cb1071865a4ad98b1f24e5f04411 bfc9b959824c8dfabe52c4d900a0e01 9d5170ea0a30c
text_test_share_3.txt: 06b80184a0d775fb54d00b68388655 16014e5f4436af287e253a68400f300 150d1c8a58f29e
merged_file.txt: This is secret!

Application Interface

Image Partition (Before)

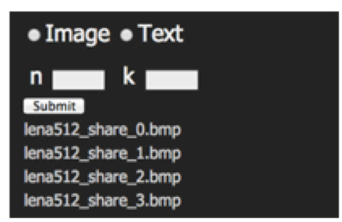


Image Partition (After)

Image Reconstruction (Before)

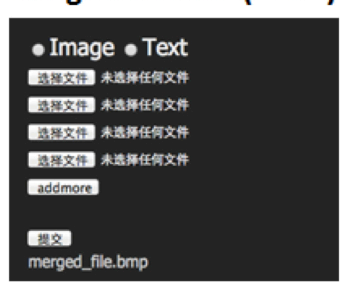


Image Reconstruction (After)