中山大學 网络空间安全学院

SUN YAT-SEN UNIVERSITY SCHOOL OF CYBER SCIENCE AND TECHNOLOGY

面向图像的秘密共享

学院：网络空间安全学院

专业：网络空间安全

一、算法原理简介

基于 Shamir 秘密共享协议的图像秘密共享方案:

一、基本原理与思想

Shamir 秘密共享协议:假设一个秘密 s，秘密分发者将 s 运用特定算法分成 n 份，然后将 n 份分别分发给 n 个参与者。在重构时，n 个参与者中选取 r 个人来重构这个秘密。任意的 r 个人都可以重构，任意的 r-1 个参与者无法获得秘密的任何信息。

二、将上述基本思想应用到图像领域

2.1 灰度图像处理思路

灰度图像是由一系列像素值组成，可以把每个像素值都视作一个秘密 s，利用 shamir 秘密共享协议的基本原理，用随机生成的多项式生成 n 份子秘密，遍历所有的像素点即可生成相应的 n 份影子图像。在重构阶段，随机选取任意 r 份影子图像，对各个像素点应用拉格朗日插值公式以恢复原像素点，遍历所有的像素点即可恢复原秘密图像。

2.2 具体算法

分发阶段:

1. 读取图像，将图像转换成数组形式，并展平成一维数组，方便后续操作。

2. 对像素值进行预处理:由于 shamir 秘密共享方案中的多项式需要选取一个素数 p，而像素值范围为 0~255，因此我们选择小于 255 的最大素数 251 作为素数 p（选取的素数应小于 255，若大于 255 则会出现秘密值大于 255 的情况，无法以图像形式分发或无法完全恢复图像）。为保证分发和恢复过程中的损失尽量小，我们提前将读取的图像中像素值大于 250 的像素值都修改为 250。

3. 使用一个 exceed 列表存储那些超过 250 的像素值与 250 的差值，作为后续无损恢复的凭证。该凭证可依附于分发图像进行分发，也可以交付给可信第三方进行保管。

4. 严格按照 shamir 秘密共享算法，随机在 p 的有限域内生成多项式系数，接着根据 r 的值确定多项式的次数，计算得出 n 个秘密影子图像的像素值。

5. 将生成的影子图像一维数组 reshape 成原本形状并进行保存分发。

恢复阶段:

1. 为了方便测试，我读取所有分发的影子图像，并且随机选取 r 个影子图像的像素值数据用作恢复图像。

2. 重点：利用拉格朗日插值公式恢复秘密。由于我们的多项式取 251 的模，如果多项式中的系数或常数项（秘密值）出现分数，则需要计算分母的逆元（模 251 条件下），并同时乘分母与其逆元（在模 251 条件下相当于乘 1），将分数化为整数值。

3. 重点：执行完上述步骤后，遍历生成的恢复后的一维数组，若恢复的秘密值等于 250，则查询在分发阶段生成的 exceed 列表，利用该列表的内容恢复其原本的像素值，实现无损恢复。

验证、评估阶段：

逐像素对比原图与恢复图像的像素值，若有不一致的像素值，则输出该像素的位置，并输出原来的像素以及恢复后的像素。若没有不一致的像素值，则输出“两个图像的像素完全一致”。

二、新的探索 【加分项】

上述思路针对的是灰度图像，但其实 RGB 图像也同理。RGB 图像有三个通道，在分发阶段我们可以分别将三个通道视作灰度图像进行处理，再进行分发；恢复阶段可以将三个通道的图像叠加以恢复 RGB 图像。

或者直接将 RGB 图像展平（flatten），逐个像素值进行处理，最后在 reshape 成原来的形式。其余步骤同上述步骤。

综上，彩色图像和灰度图像在本算法中处理方式是相同的。

由于 shamir 秘密共享方案中的多项式需要模素数，在这个过程中可能会产生图像精度损失。本次实验采用 exceed 列表在读取图像时提前存储超过 250 像素的像素值与 250 的差值，以在恢复阶段恢复原本的像素值，实现无损。

基于中国剩余定理（CRT）的图像秘密共享算法（未实现，只是讲一下算法思路）

参考文献：

陈维启,张珍珍,李祯祯,丁海洋,李子臣.基于 C RT 的无损高效门限彩色图像秘密共享信息隐藏算法.计算机系统应用,2022,31(5):269-276. http://www.c-s-a.org.cn/1003-3254/8429.html

本方案基于中国剩余定理 (CRT), 提出了一个 (t, n) 秘密共享方案, 由秘密 s 通过 CRT 相关公式计算得到 y. 再与 n 个递增的模数$m_1, m_2, \cdots, m_n$通过取模运算得到(m_i, y_i)即是共享份, 在实际方案中只需要保存y_i.

2.2.1 参数选取

参数的选择满足以下要求.

(1) $q > s$.

(2) $\gcd(m_1, m_j) = 1, \forall i, j, i \neq j$.

(3) $\gcd(q, m_i) = 1, i = 1, 2, \cdots, n$.

(4) $N = \prod_{i=1}^{t} m_i > q \prod_{i=1}^{t-1} m_{n-j+1}$.

2.2.2 秘密共享与恢复

参数 A 经由随机选取, 范围是$0 \leqslant A \leqslant \lfloor N/q \rfloor - 1$. 根据公式$y = s + Aq$, 求得用于计算共享份的参数 y, 该参数满足$y < q + Aq = (A+1)q \leqslant \lfloor N/q \rfloor \cdot q \leqslant N$. 由$y_i \equiv y \bmod m_i (i = 1, 2, \cdots, n)$. (m_i, y_i)是共享份额, 因为模数公开, 所以将其中的y_i作为子共享, 分发给用户. 达到门限值的 t 个用户$i_1, i_2, \cdots, i_t$提供自己的子共享, 根据$\{(m_{i_j}, y_{i_j}) | i = 1, 2, \cdots, t\}$建立方程组.

$$\begin{cases} y_{i_1} \equiv y \bmod m_{i_1} \\ y_{i_2} \equiv y \bmod m_{i_2} \\ \quad \vdots \\ y_{i_t} \equiv y \bmod m_{i_t} \end{cases} \tag{1}$$

可以求得:

$$y \equiv y' \bmod N' \tag{2}$$

其中, $y' \equiv \sum_{j=1}^{t} y_{i_j} M_{i_j} M_{i_j}^{-1}$, $M_i = M/m_i$. 同时$M_i M_i^{-1} \equiv 1 (\bmod m_i)$. M_i^{-1}是M_i的逆元. 本文使用基于费马小定理的逆元求解法. 只需进行模幂运算就可以快速求解, 即$M_i^{-1} \equiv M_i^{m_i - 2} (\bmod m_i)$.

最后根据$N' \equiv \sum_{j=1}^{t} m_{i_j} \geqslant N$, 得到$y \equiv y' \bmod N'$, 由$y \bmod q$ 解得秘密 s.

由于 CRT 秘密共享算法在彩色图像应用实现上会出现像素溢出问题, 将超过 255 像素值的共享数据根据上文分别存成倍数图像与余数图像。

标黄的思路也可以应用于我本次实现的基于 shamir 秘密共享的图像共享方案。也可以实现无损。

三、实验结果分析

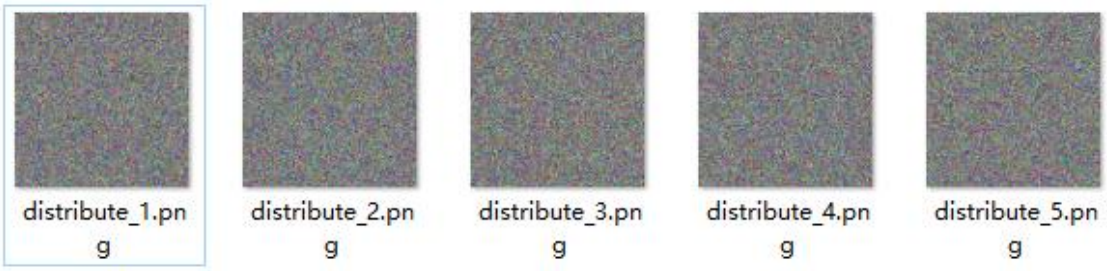
3.1 实验结果截图简要分析。

3.2 新的探索算法结果分析。

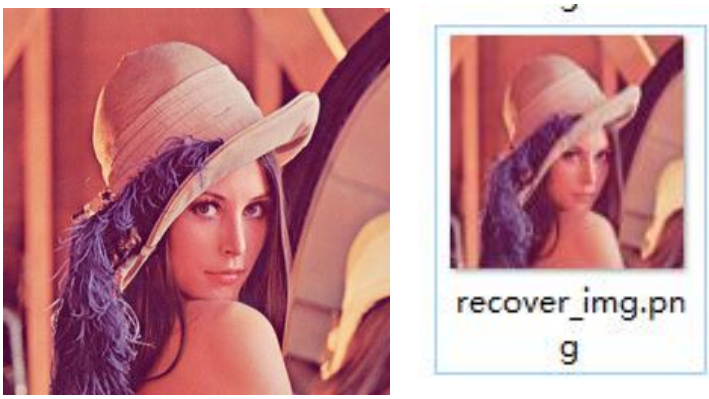
两个合二为一进行分析（最终实现的效果）：

原始图像：

分发的五个影子图像：



恢复图像：



终端运行的结果：

```
PS C:\Users\Lenovo\Desktop\数据安全与隐私保护\image-secret-sharing> & C:/Users/Lenovo
/.conda/envs/pytorch/python.exe c:/Users/Lenovo/Desktop/数据安全与隐私保护/image-secr
et-sharing/main.py
100%|                                                | 196608/196608 [00:44<00:00, 4453.10it/s]
两个图像的像素完全一致。
```

验证部分的输出为“两个图像的像素完全一致”，可以看到完全恢复了原有的图像。测试的图像为 256×256×3，故总共要处理的像素值为 196608。

整个项目的文件夹：