پیاده سازی رمزنگاری تصویر توسط تابع فرا آشوب ۷ بعدی و ماتریس پاسکال

نویسنده : سید محمد رضا هدایی
استاد راهنما : دکتر ابراهیم زارعی

دانشگاه اصفهان پردیس خوانسار – ۱۴۰۲/۱۱/۰۱

Available at https://github.com/smrhodaee/image_encryption

فهرست مطالب

فهرست مطالب

مقدمه [چرا به رمزنگاری تصویر نیاز داریم؟]

- افزایش حملات سایبری مخرب
- محافظت از تصاویر پزشکی و نظامی
- بهترین نوع میان روش های محافظت از تصاویر
  - Image WaterMarking
  - Image Steganography
  - **Image Encryption**
- تبدیل تصویر اولیه به تصویر ناخوانا

مقدمه [چرا از تابع فرا آشوب ۷ بعدی استفاده می کنیم؟]

- رفع محدودیت های توابع با ابعاد کمتر
- غیرخطی بودن
- تصادفی بودن
- غیر قابل پیش بینی بودن
- ساخت دنباله کلید با فضای کلید بزرگ
- افزایش سطح امنیت با فرا آشوب بودن

مقدمه [چرا الگوریتم جدید پیاده سازی کردیم؟]

- فضای کلید کوچک در الگوریتم های قبلی (حملات بروت فروس)
- ناکارایی در حملات نویز نمک فلفلی
- امنیت پایین در حملات دیفرانسیلی و آماری
- حذف کردن هم بستگی میان پیکسل های همجوار
- افزایش تصادفی بودن و غیرقابل پیش بینی بودن
- کاهش زمان پردازش

پیش نیاز ها [ماتریس متقارن پاسکال] [۱]

- **ماتریس مربعی $P(n)$ را با ابعاد $n \times n$ در نظر می گیریم**

$$P_{i,j} = \binom{i+j}{i} = \frac{(i+j)!}{i!j!}, 0 \leq i, j < n. \quad (۱)$$

$$p_{i,j} = p_{i-1,j} + p_{i,j-1} \quad (۲)$$

پیش نیاز ها [ماتریس متقارن پاسکال] [۲]

$$P(1)=\begin{bmatrix}1\end{bmatrix}, P(2)=\begin{bmatrix}1 & 1\\ 1 & 2\end{bmatrix}, P(3)=\begin{bmatrix}1 & 1 & 1\\ 1 & 2 & 3\\ 1 & 3 & 6\end{bmatrix}, P(4)=\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & 2 & 3 & 4\\ 1 & 3 & 6 & 10\\ 1 & 4 & 10 & 20\end{bmatrix} \quad (۳)$$

- **ماتریس پاسکال معکوس با ابعاد ۴*۴**

$$P^{-1}(4)=\begin{bmatrix}4 & -6 & 4 & -1\\ -6 & 14 & -11 & 3\\ 4 & -11 & 10 & -3\\ -1 & 3 & -3 & 1\end{bmatrix} \quad (۴)$$

پیش نیاز ها [سیستم فرا آشوب ۷ بعدی]

$$
\begin{aligned}
\dot{x_1} &= -ax_1 + ax_5 - bx_5x_6x_7 \\
\dot{x_2} &= -cx_2 + dx_6 + x_1x_6x_7 \\
\dot{x_3} &= -ax_3 + ax_5 - gx_1x_2x_7 \\
\dot{x_4} &= -ax_4 + ex_1 + x_1x_2x_3 \\
\dot{x_5} &= -ax_5 + ex_7 - x_2x_3x_4 \\
\dot{x_6} &= -ex_6 + ex_5 + x_3x_4x_5 \\
\dot{x_7} &= -bx_7 + fx_2 - hx_4x_5x
\end{aligned}
\qquad (۵)
$$

$$a = 15,\ b = 5,\ c = 0.5,\ d = 25,\ e = 10,\ f = 4,\ g = 0.1,\ h = 1.5$$

الگوریتم پیشنهادی [الگوریتم رمزنگاری] [۱]

۱. وارد کردن تصویر خاکستری به عنوان G

۲. تبدیل G به بردار V

۳. محاسبه مقادیر اولیه کلید سیستم فرا آشوب

$$x_1 = \frac{\sum_{i=1}^{MN} V(i) + MN}{2^2 3 + MN} \quad (۶)$$

$$x_i = \mathrm{mod}(10^7 x_{i-1}), i = 2, 3, 4, 5, 6, 7 \quad (۷)$$

۴. ساختن دنباله S با پیمایش سیستم فرا آشوب ۷ بعدی و انتخاب دنباله

$$(x_1, x_2, x_7)$$

الگوریتم پیشنهادی [الگوریتم رمزنگاری] [۲]

۵. مرتب کردن S به صورت صعودی و برگرداندن موقعیت پیکسل های مرتب شده در بردار SS

۶. محاسبه بردار جایگشت داده شده B

$$\mathrm{B} = V(SS)$$

۷. بردار B را به ماتریس D با اندازه MN تغییر شکل دادن.

۸. ماتریس D را به زیرماتریس های مرتبه ۴ تقسیم کردن

الگوریتم پیشنهادی [الگوریتم رمزنگاری] [۳]

۹. ماتریس E را با ضرب هر زیرماتریس در ماتریس پاسکال مرتبه ۴ بدست آوردن به شکل زیر

$$\begin{bmatrix} E_{i,j} & E_{i,j+1} & E_{i,j+2} & E_{i,j+3} \\ E_{i+1,j} & E_{i+1,j+1} & E_{i+1,j+2} & E_{i+1,j+3} \\ E_{i+2,j} & E_{i+2,j+1} & E_{i+2,j+2} & E_{i+2,j+3} \\ E_{i+3,j} & E_{i+3,j+1} & E_{i+3,j+2} & E_{i+3,j+3} \end{bmatrix} \quad (۸)$$

$$= \begin{bmatrix} D_{i,j} & D_{i,j+1} & D_{i,j+2} & D_{i,j+3} \\ D_{i+1,j} & D_{i+1,j+1} & D_{i+1,j+2} & D_{i+1,j+3} \\ D_{i+2,j} & D_{i+2,j+1} & D_{i+2,j+2} & D_{i+2,j+3} \\ D_{i+3,j} & D_{i+3,j+1} & D_{i+3,j+2} & D_{i+3,j+3} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 6 & 10 \\ 1 & 4 & 10 & 20 \end{bmatrix} \mod 256$$

الگوریتم پیشنهادی [الگوریتم رمزنگاری] [۴]

۱۰. با استفاده از دو مرحله از فرآیند رمزگذاری، تصویر رمزگذاری شده E را بدست آوردن. به منظور دستیابی به نتایج رمزگذاری بهتر، تنها دو دور فرآیند درهمسازی و انتشار برای اصلاح موقعیت ها و حذف همبستگی بین پیکسل های همسایه در تصویر رمزگذاری شده کافی است.

الگوریتم پیشنهادی [الگوریتم رمزگشایی] [۱]

۱. تصویر رمز شده E به زیر ماتریس های مرتبه ۴ تقسیم می شود و سپس با ضرب هر زیر ماتریس در معکوس ماتریس پاسکال از معادله زیر برای زیر ماتریس های تصویر استفاده می شود.

$$\begin{bmatrix} D_{i,j} & D_{i,j+1} & D_{i,j+2} & D_{i,j+3} \\ D_{i+1,j} & D_{i+1,j+1} & D_{i+1,j+2} & D_{i+1,j+3} \\ D_{i+2,j} & D_{i+2,j+1} & D_{i+2,j+2} & D_{i+2,j+3} \\ D_{i+3,j} & D_{i+3,j+1} & D_{i+3,j+2} & D_{i+3,j+3} \end{bmatrix}$$

$$= \begin{bmatrix} E_{i,j} & E_{i,j+1} & E_{i,j+2} & E_{i,j+3} \\ E_{i+1,j} & E_{i+1,j+1} & E_{i+1,j+2} & E_{i+1,j+3} \\ E_{i+2,j} & E_{i+2,j+1} & E_{i+2,j+2} & E_{i+2,j+3} \\ E_{i+3,j} & E_{i+3,j+1} & E_{i+3,j+2} & E_{i+3,j+3} \end{bmatrix} \begin{bmatrix} 4 & -6 & 4 & -1 \\ -6 & 14 & -11 & 3 \\ 4 & -11 & 10 & -3 \\ -1 & 3 & -3 & 1 \end{bmatrix} \mathrm{mod} 256 \quad (۹)$$

۲. تصویر D به دست آمده از فاز قبلی به بردار U تبدیل می شود.

الگوریتم پیشنهادی [الگوریتم رمزگشایی] [۲]

۳. محاسبه زیر از بردار S ایجاد شده در مرحله رمزگذاری برای بازگرداندن پیکسل ها به موقعیت اصلی خود استفاده می کند:

$$O(S_k) = U_k; k = 1 : MN \quad (۱۰)$$

۴. بردار O را به ماتریس تبدیل کنید تا به تصویر رمزگشایی شده G' برسید

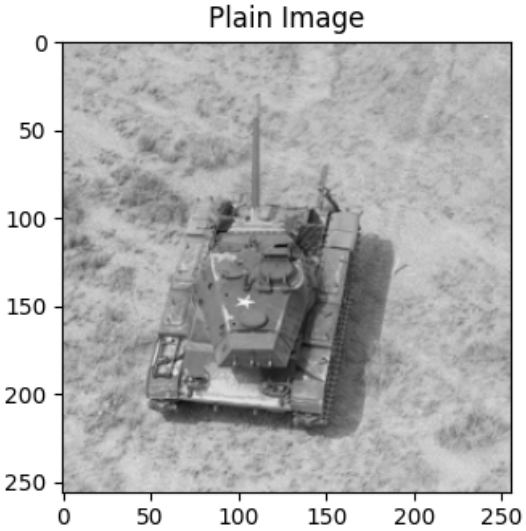
۵. برای دستیابی به تصویر رمزگشایی شده، دو بار فرآیند رمزگشایی مورد نیاز است

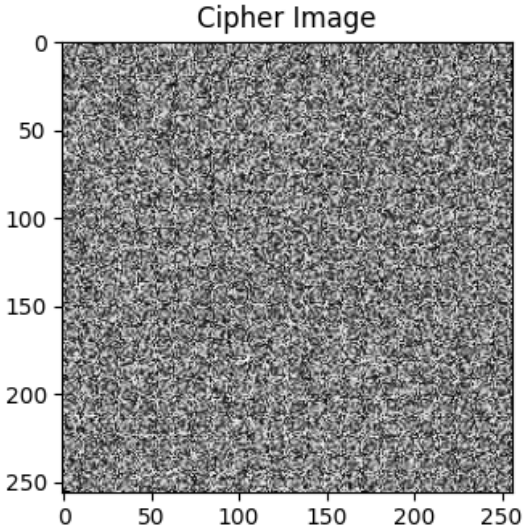
نتایج شبیه سازی [پلت فرم آزمایش]

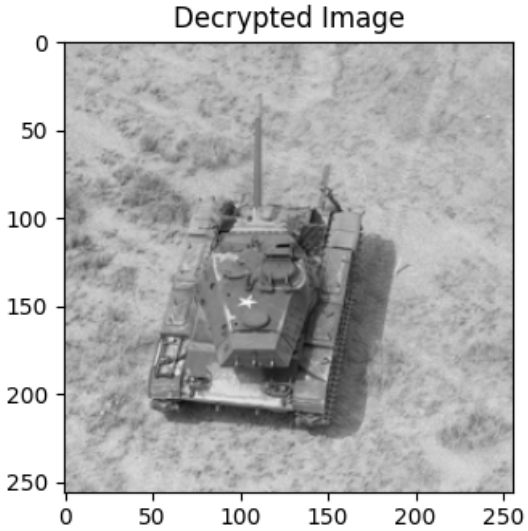
برای نشان دادن توانایی اگوریتم های رمزنگاری و رمزگشایی پیشنهادشده از یک لب تاب با پردازنده اینتل core i5 و با حافظه 12 GB استفاده می کنیم هم چنین الگوریتم توسط پایتون پیاده سازی شده است

نتایج شبیه سازی [نتایج آزمایش] [۱]

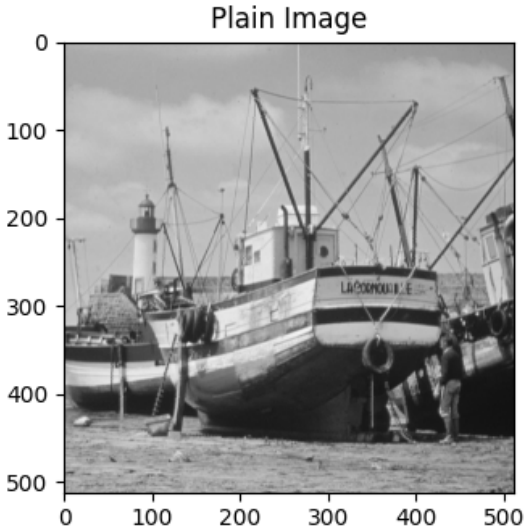
کارایی الگوریتم را با تصاویر مختلف مورد ارزیابی قرار میدهیم که به شکل ریز می باشد (به ترتیب Tank 256*256 ، Boat 512*512، Lena 512*512، Cameraman 1024*1024، CTScan 1024*1024)

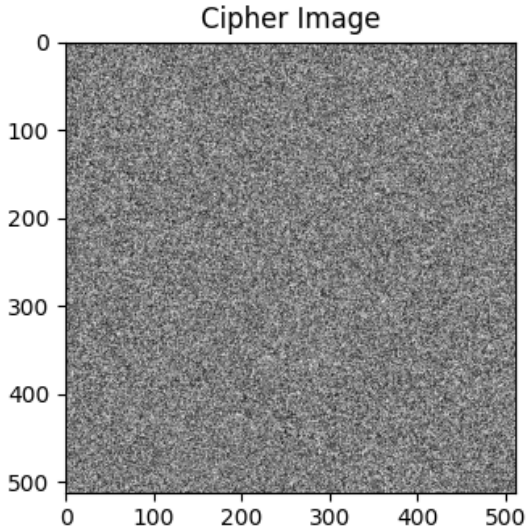


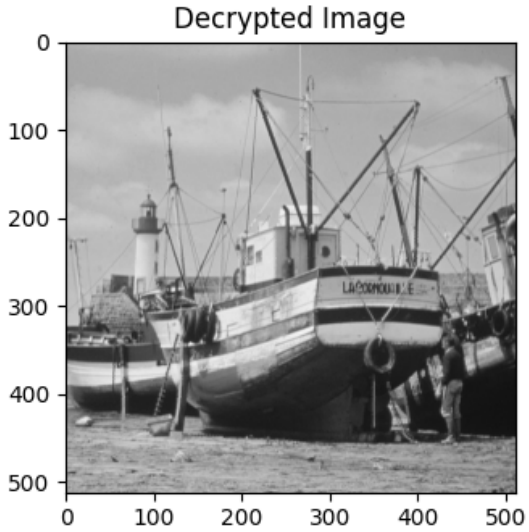


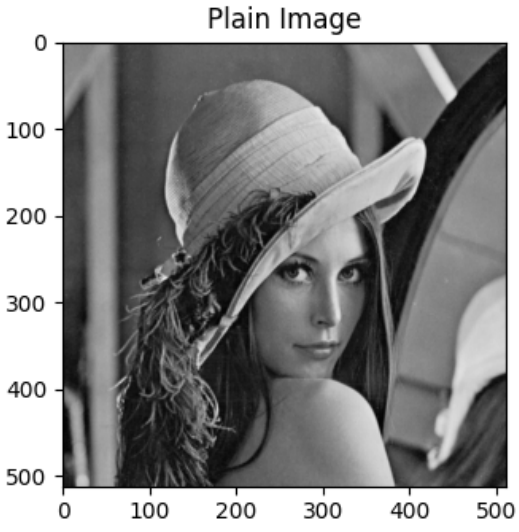


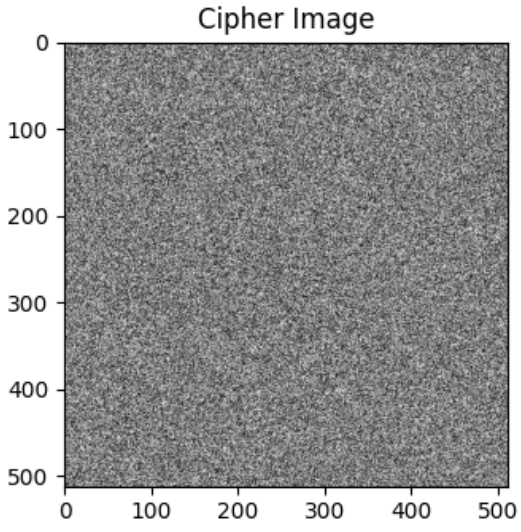
نتایج شبیه سازی [نتایج آزمایش] [۲]

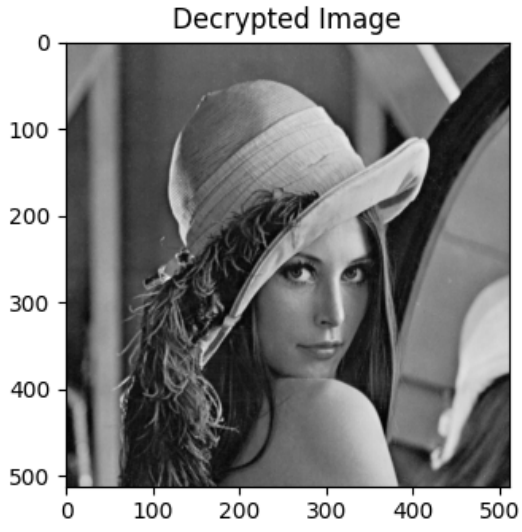




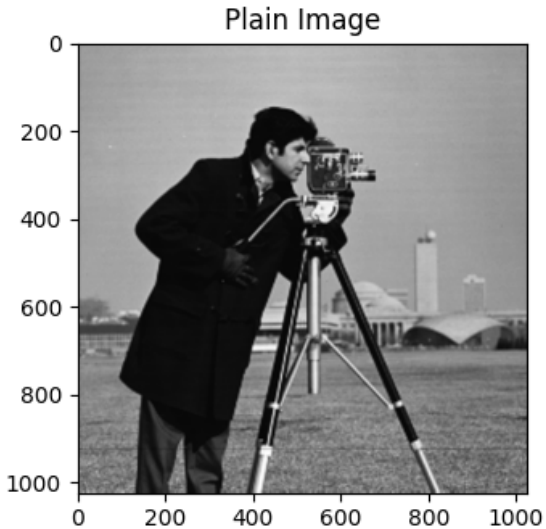


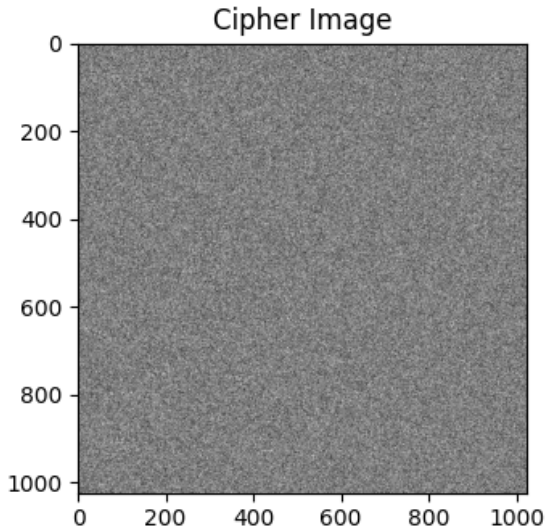


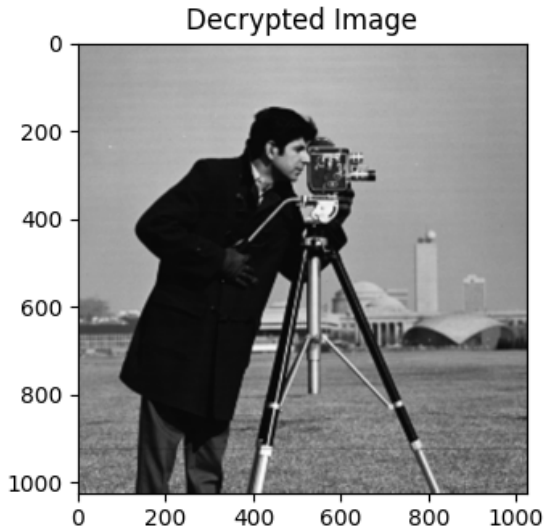


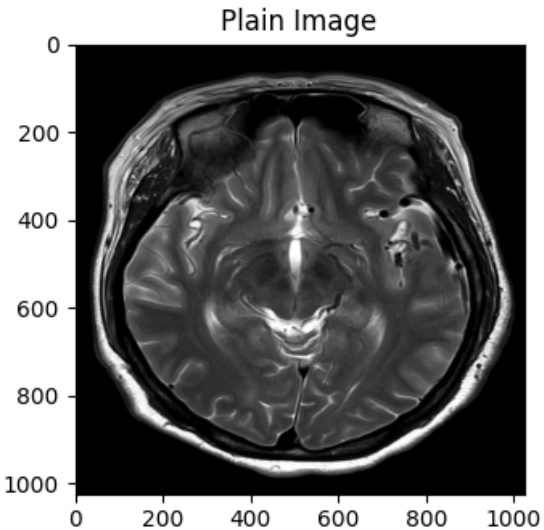


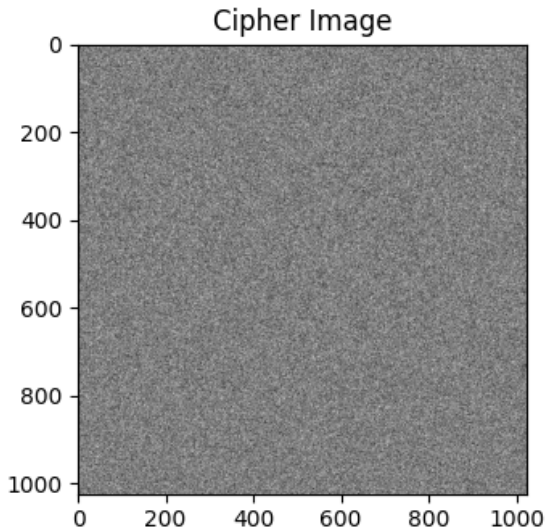
نتایج شبیه سازی [نتایج آزمایش] [۳]

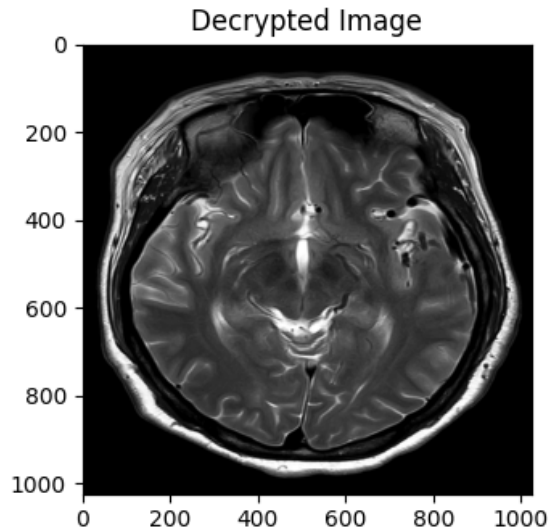












تحلیل امنیت [مقدمه]

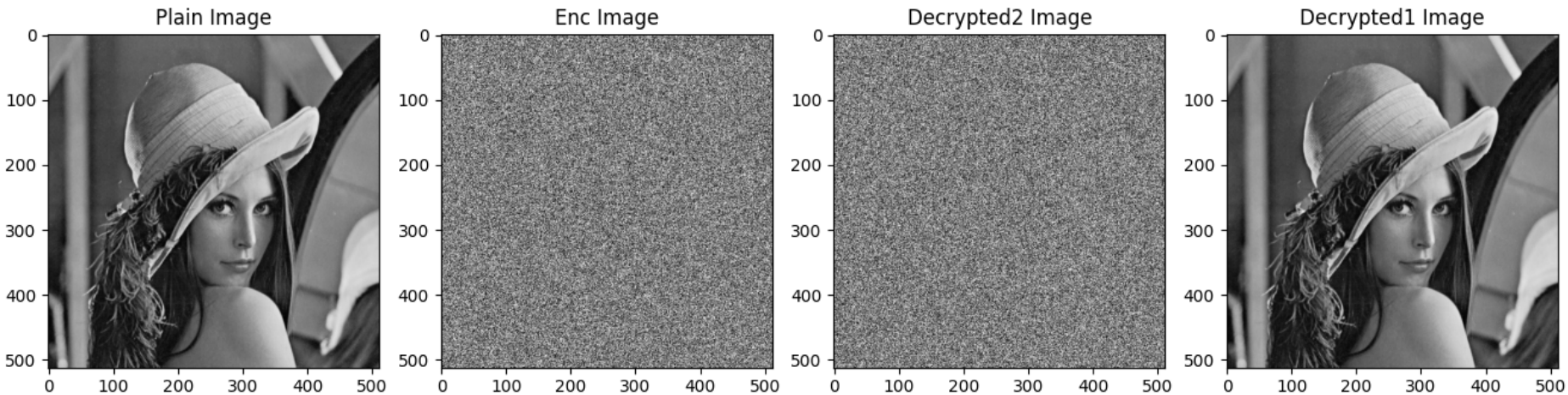
این بخش چندین معیار شناخته شده را توصیف می کند که معمولاً برای ارزیابی امنیت الگوریتم های جدید استفاده می شود. برای ارزیابی عملکرد و جنبه‌های امنیتی الگوریتم پیشنهادی، تحلیلی از روش پیاده‌سازی شده در زیر ارائه می‌کنیم. سرعت الگوریتم پیشنهادی نیز در آخرین بخش مورد بررسی قرار گرفته است. از نمونه عکس Lena 512*512 استفاده خواهیم کرد

تحلیل امنیت [تحلیل فضای کلید]

اندازه کلید در فرآیند رمزگذاری اهمیت دارد. این کلید باید به اندازه کافی بزرگ باشد تا در برابر حملات Brute Force مقاومت کند کلید خصوصی توسط مقادیر $a, b, c, d, e, f, g, h, x1, x2, x3, x4, x5, x6, x7, N_0$ و در سیستم پر آشوب ۷ بعدی ایجاد می‌شود. اگر دقت تخمین مقدار اولیه را برابر با 10^{16} در نظر بگیریم، بنابراین کل کلید خصوصی $N_0 \times 10^{240}$ است که به اندازه کافی بزرگ است تا در برابر حملات Brute Force مقاومت کند.

تحلیل امنیت [حساسیت کلید]

یک روش رمزگذاری خوب طراحی شده باید نسبت به هرگونه تغییر کوچک در شرایط اولیه کلید خصوصی استفاده شده بسیار حساس باشد. هنگامی که این کلید کمی تغییر می کند، تصویر بازیابی شده نویز و نامفهوم می شود یکی از پارامتر های کلید را تغیر می دهیم مثلا 0.0001 به آن اضافه می کنیم نتیجه به شکل زیر خواهد بود



تحلیل امنیت [تحلیل هیستگرام]

هیستوگرام یک تصویر فرکانس هر پیکسل را نشان می دهد. اگر هیستوگرام تصویر رمزگذاری شده به طور یکنواخت توزیع شود، طرح رمزنگاری می تواند به طور موثر در برابر حملات آماری مقاومت کند.

تحلیل امنیت [تحلیل ضریب همبستگی] [۱]

یکی از مهمترین ویژگی ها در زمینه رمزگذاری تصویر، همبستگی بین هر دو پیکسل مجاور است. پیکسل ها در یک تصویر ساده دارای ضرایب همبستگی بالایی با همسایگان خود هستند. برای کاهش احتمال حملات، همبستگی بین پیکسل های همسایه یک تصویر رمزی باید بسیار نزدیک به صفر باشد.

همبستگی عمودی، افقی و مورب بین هر جفت پیکسل، x و y را می توان با

Type	Plain Image	Chiper Image
V	0.965104	-0.19866
H	0.989273	0.0484015
D	0.928971	0.127113

تحلیل امنیت [تحلیل ضریب همبستگی] [۲]

$$R_{x,y} = \frac{cov(x,y)}{\sqrt{D(x)} \times \sqrt{D(y)}}$$

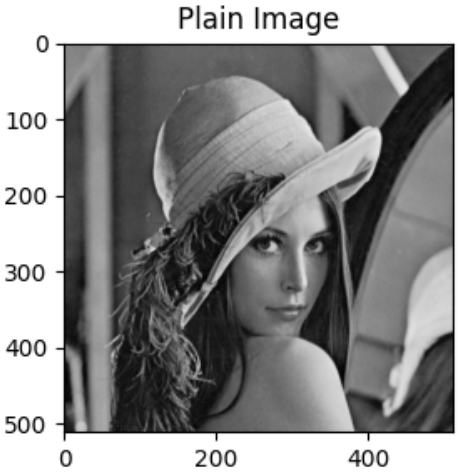
$$cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y - E(y_i))$$

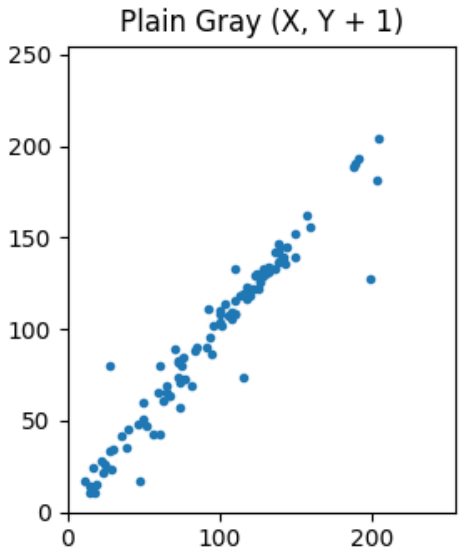
$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2 \qquad (۱۱)$$

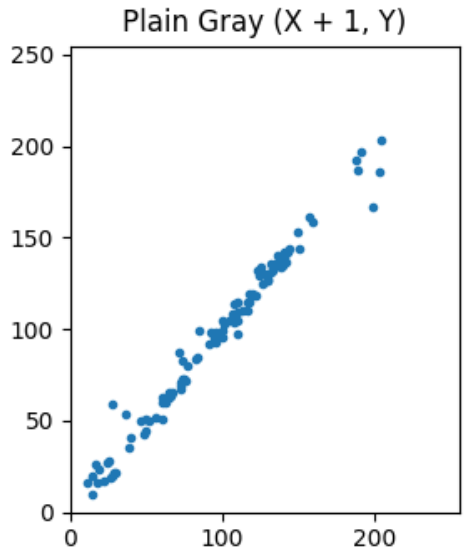
$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i$$

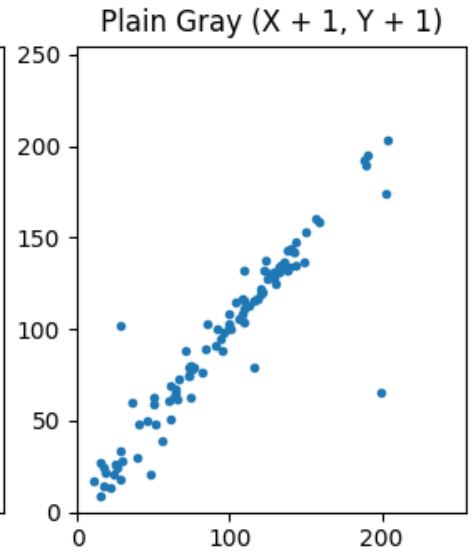
به طوری که و N تعداد کل پیکسل های درگیر در محاسبات است

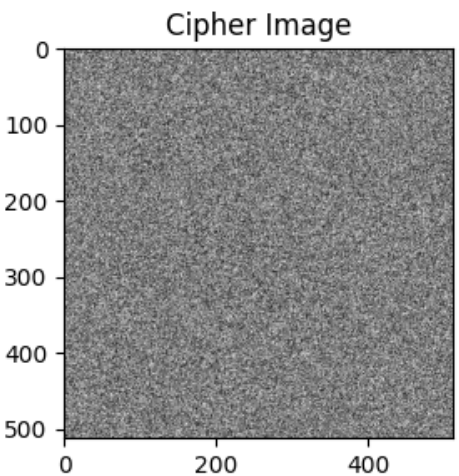
تحلیل امنیت [تحلیل ضریب همبستگی] [۳]

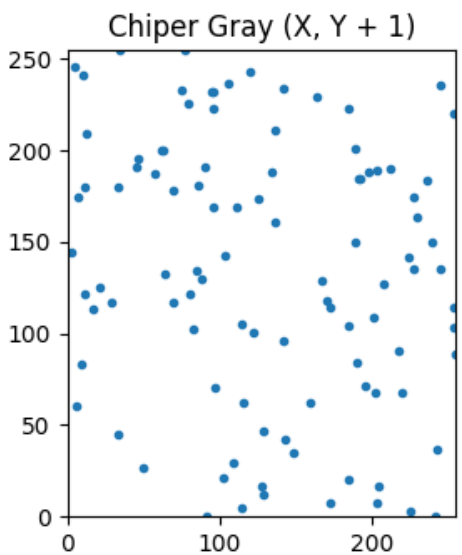


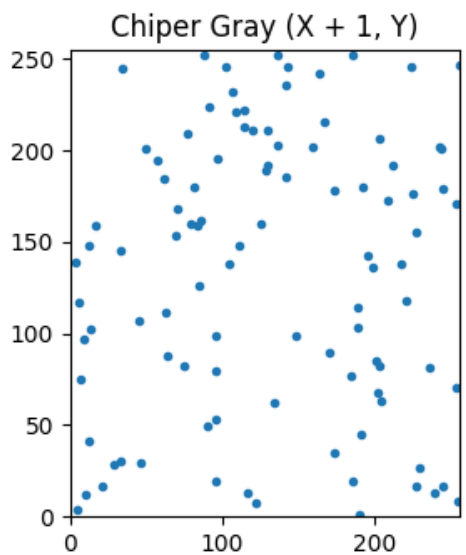


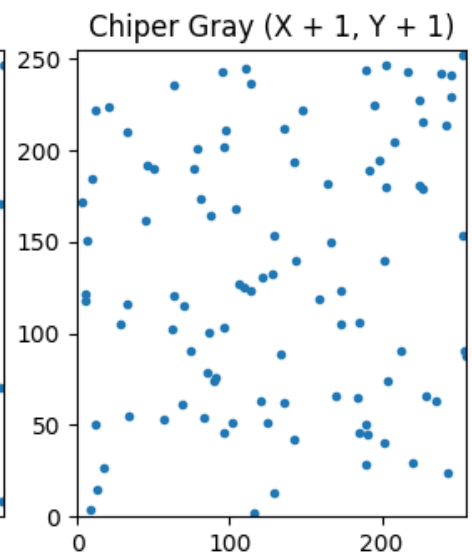












تحلیل امنیت [آنتروپی اطلاعات]

تست آنتروپی اطاعات میزان درجه تصادفی بودن و غیر قطعی بودن در یک تصویر را اندازه گیری می کند

$$H(m) = \sum_{i=1}^{2^N-1} p(m_i) \log \frac{1}{p(m_i)} \quad (۱۲)$$

به طوری که N تعداد بیت های نماد m_i و $p(m_i)$ احتمال هر کدام می باشد

```
PlainImage  Entropy = 7.592928651429443
EncImage Entropy = 7.999321402592455
```

بهترین مقدار آنتروپی برای تصویر رمزگذاری شده ۸ می باشد

تحلیل امنیت [تحلیل حملات دیفرانسیل] [۱]

هدف از این حمله کرک کردن تصویر رمزشده توسط ارتباط تصویر رمز شده با تصویر اصلی بدون استفاده از کلید می باشد برای ارزیابی این حمله از دو پارامتر $\mathrm{UACI}, \mathrm{NPCR}$ استفاده می شود

$$\mathrm{UACI} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}|C_2(i,j) - C_1(i,j)|}{255 \times M \times N} \times 100\% \quad (۱۳)$$

$$\mathrm{NPCR} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} h(i,j)}{M \times N} \times 100\% \quad (۱۴)$$

$$h(i,j) = \begin{cases} 0 & \text{if} C_2(i,j) = C_1(i,j) \\ 1 & \text{if} C_2(i,j) \neq C_1(i,j) \end{cases} \quad (۱۵)$$

تحلیل امنیت [تحلیل حملات دیفرانسیل] [۲]

```
Before: Pixel at(485,162) = 36
After: Pixel at (485,162) = 37
NPCR = 99.61357116699219   UACI = 33.445109947054995
```

برای اطمینان از امنیت الگوریتم، مقدار UACI الگوریتم های رمزنگاری تصویر باید بزرگتر از 0.33 و مقدار NPCR باید بیشتر از 0.99 باشد.

تحلیل امنیت [حمله نویز فلفل نمکی] [۱]

تفاوت بین تصویر اصلی و تصویر رمزگذاری شده با استفاده از نسبت سیگنال به نویز پیک (PSNR) اندازه گیری می شود. می توان آن را محاسبه کرد

$$PSNR = 10 \log \left(\frac{max^2}{MSE} \right) \quad (۱۶)$$

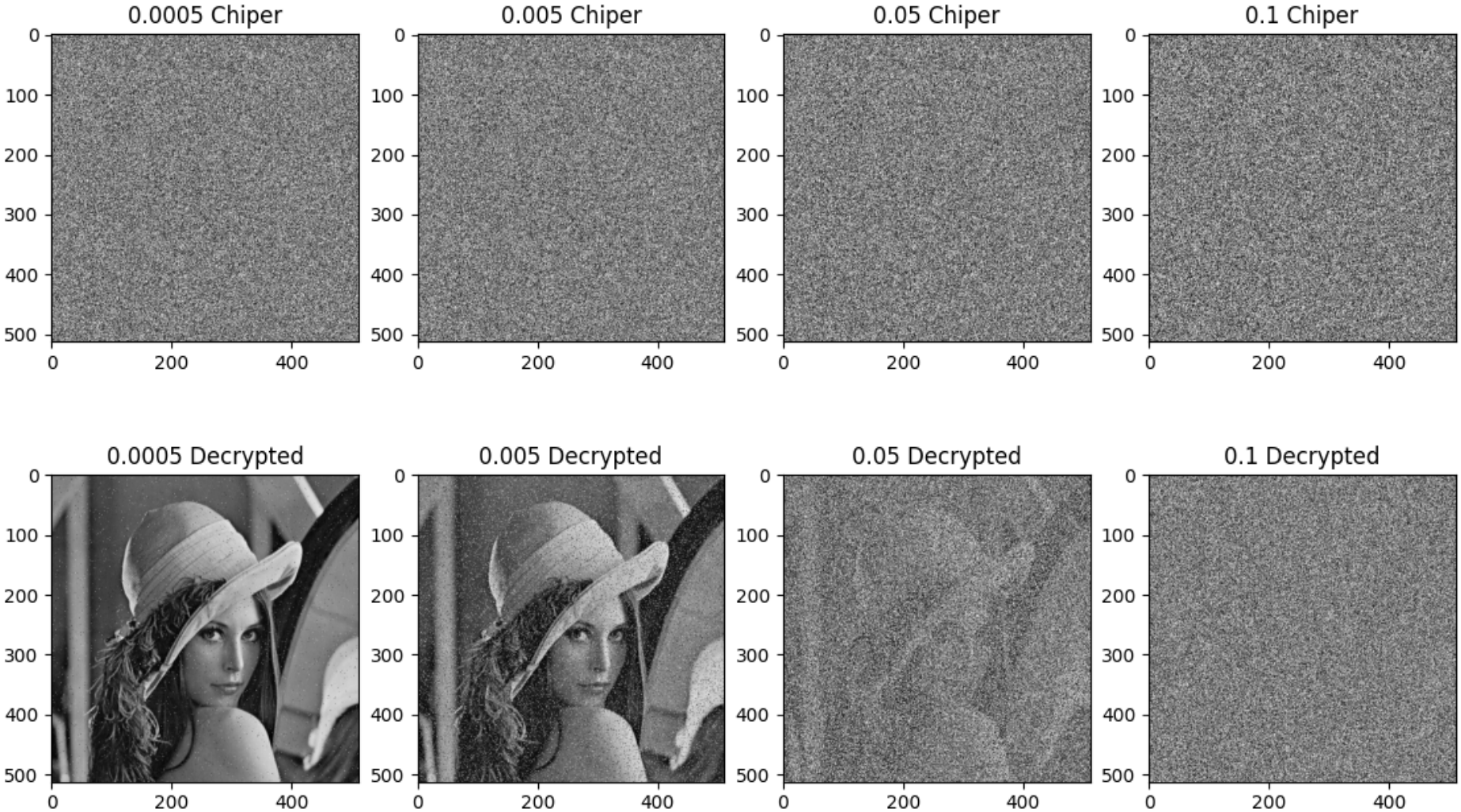
$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (O(i,j) D(i,j))^2 \quad (۱۷)$$

تحلیل امنیت [حمله نویز فلفل نمکی] [۲]

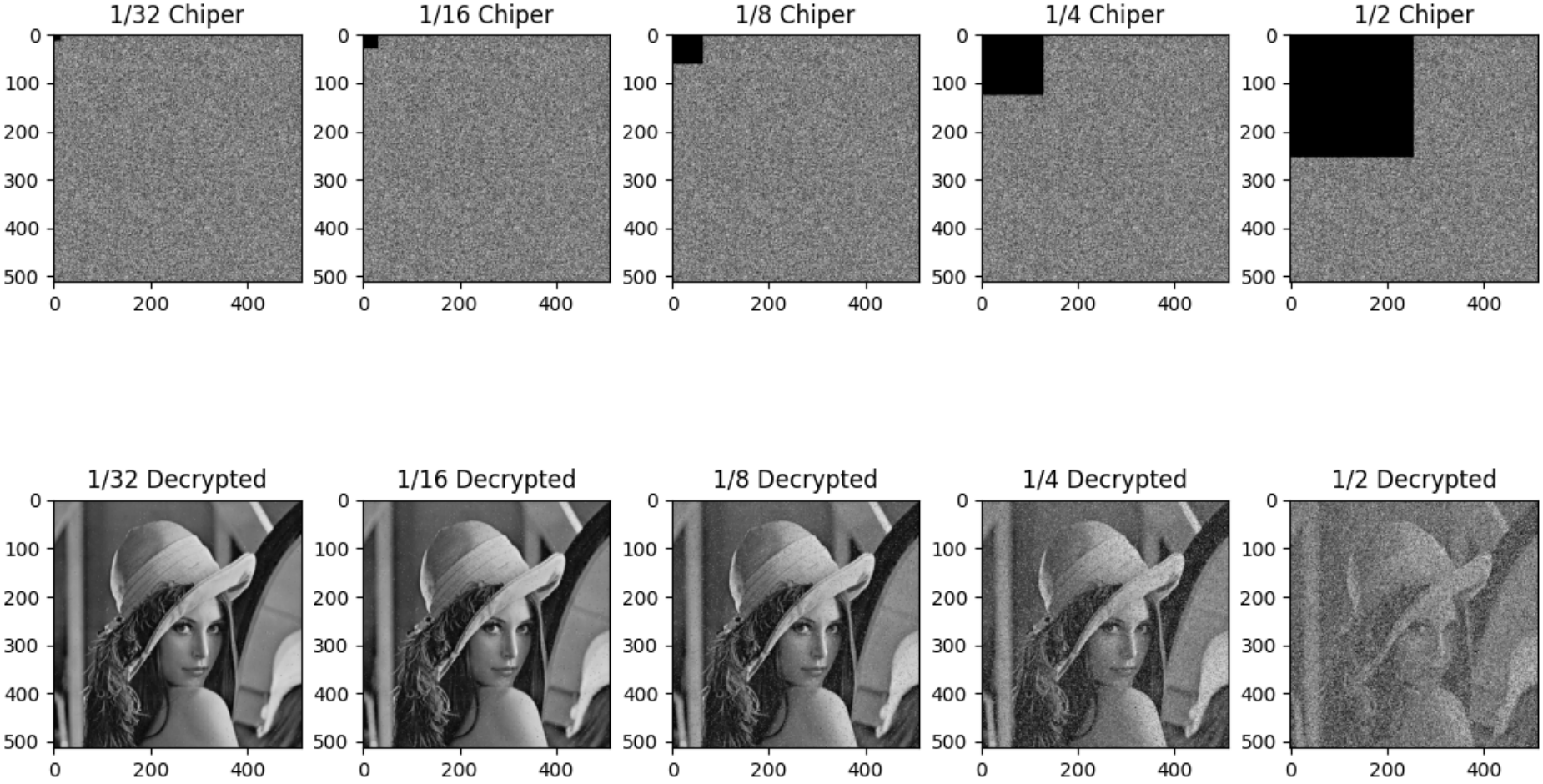
تصاویر رمزگذاری شده هنگام انتقال از طریق کانال های ارتباطی فیزیکی مستعد نویز یا تداخل هستند. تکنیک رمزگذاری رمزی باید به اندازه کافی انعطاف پذیر باشد تا بتواند تصاویر رمز شده را با وجود انباشته شدن نویز رمزگشایی کند.

```
Noise Level = 0.0005, PSNR of nosiy cipher image = 26.338162627358 29
Noise Level = 0.005, PSNR of nosiy cipher image = 16.683452549564024
Noise Level = 0.05, PSNR of nosiy cipher image = 9.438350993765841
Noise Level = 0.1, PSNR of nosiy cipher image = 8.676358168000617
```

تحلیل امنیت [حمله نویز فلفل نمکی] [۳]



تحلیل امنیت [حمله برش تصویر]



تحلیل امنیت [تست سرعت]

هنگام توسعه یک روش رمزگذاری تصویر قوی، سرعت اجرا به همان اندازه نگرانی های امنیتی بسیار مهم است.

Images	Size	Encryption Time	Decryption Time
Lena	512×512	2.5423987944sec	2.725500211sec
Cameraman	1024×1024	10.9839755196sec	8.5778788464sec
Tank	256×256	1.3027955124sec	1.5723612726sec
Boat	512×512	2.6061833132sec	2.4872887996sec

نتیجه گیری

در این کار، ما یک تکنیک جدید رمزگذاری تصویر در مقیاس خاکستری را پیشنهاد کردیم. ماتریس پاسکال از مرتبه ۴ با یک سیستم پر آشوب هفت بعدی در این تکنیک ترکیب شده است. ما در ابتدا از یک سیستم پر آشوب هفت بعدی برای تولید توالی های تصادفی استفاده کردیم و سپس سه تا از این توالی ها را برای تغییر موقعیت پیکسل انتخاب کردیم. تصویر منتشر شده در مقیاس خاکستری به مجموعه ای از ۴ بلوک فرعی تقسیم می شود و ماتریس پاسکال برای تغییر مقادیر پیکسل ها برای هر بلوک فرعی استفاده می شود. برای تقویت امنیت، دو بار فرآیندهای سردرگمی و انتشار اجرا می شوند

مراجع

Ammar Ali Neamah, Journal of King Saud University - Computer and Information SciencesVolume 35 Issue 3 Mar 2023 pp 238–248 https://doi.org/10.1016/j.jksuci.2023.02.014

Alghamdi, Y., Munir, A., Ahmad, J., 2022. A Lightweight Image Encryption Algorithm Based on Chaotic Map and Random Substitution. Entropy 24 (10), 1344. https://doi.org/10.3390/e24101344.