Information Security Final Project

An Image Encryption Algorithm Based on Random Hamiltonian Path

朱健愷、林陽、陳冠穎

研究動機

現有圖片加密演算法中的效能瓶頸:

- 重新排列元素
- Worst case 非線性時間

觀察

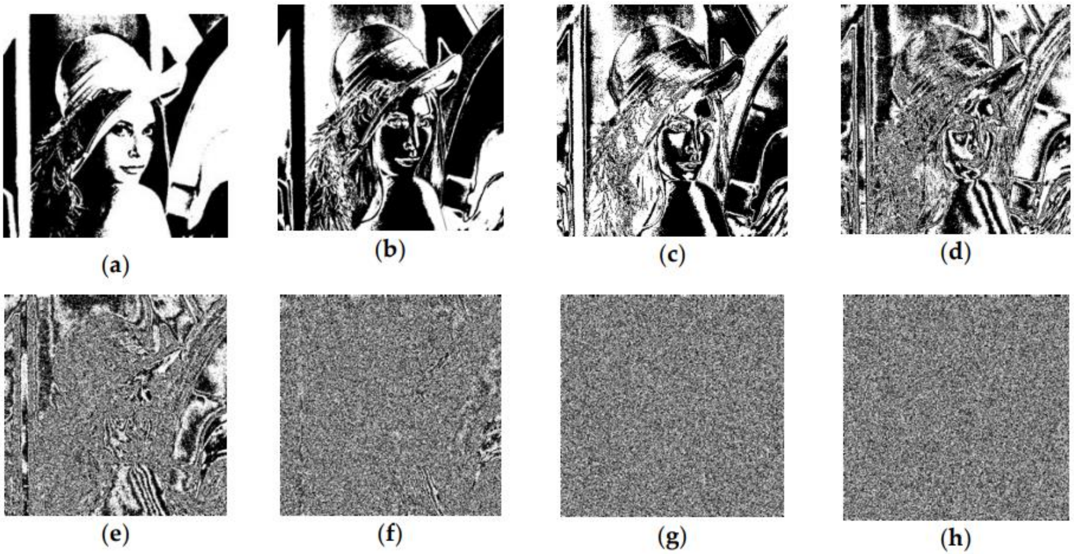


Figure 5. Bit planes of Lena, (**a**–**h**) are from the 8th bit plane to the 1st bit plane, respectively.

灰階圖中越高位的位元，其所含有的圖片資訊量就越高

可能的解決方案

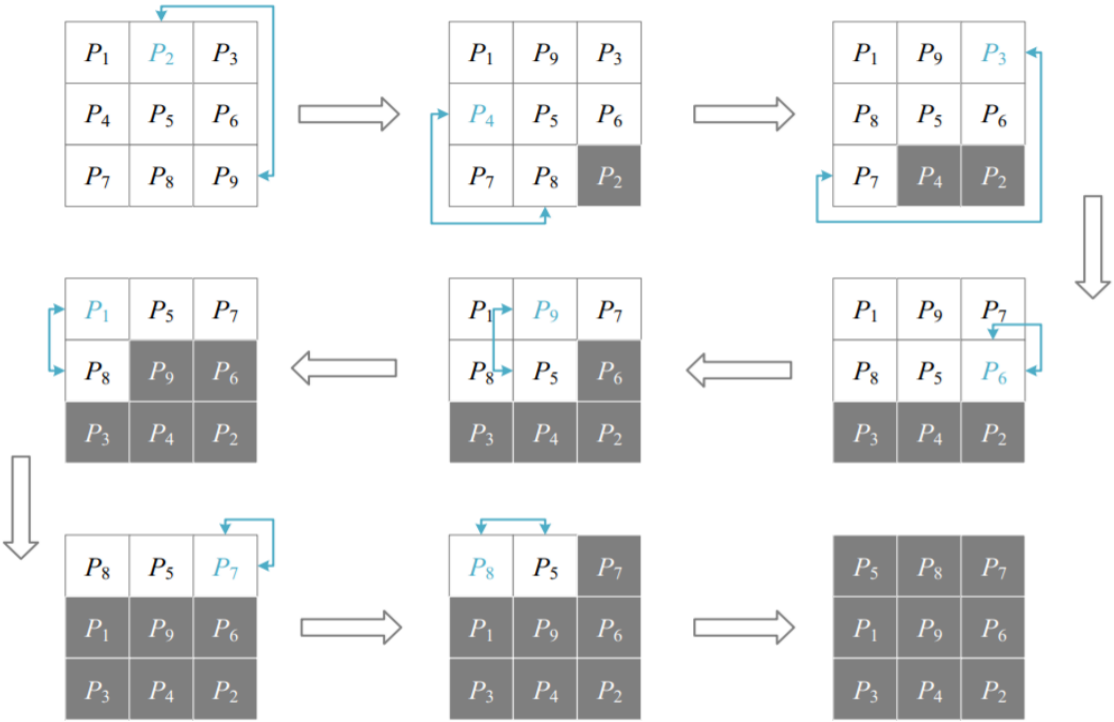
根據Dirac Theorem

$$G=(V,E),\quad d(V_i)\geq\frac{|V|}{2}\ for\ all\ V_i\in V$$

則 Hamiltonian cycle(path)存在

對於圖片而言，圖片中的每個像素皆與其他所有像素有一條邊，亦即圖片中的像素形成的圖為一完全圖，必存在至少一條Hamiltonian path。

演算法



P_1 P_2 P_3
P_4 P_5 P_6
P_7 P_8 P_9
P_1 P_9 P_3
P_4 P_5 P_6
P_7 P_8 P_2
P_1 P_9 P_3
P_8 P_5 P_6
P_7 P_4 P_2
P_1 P_9 P_7
P_8 P_5 P_6
P_3 P_4 P_2
P_1 P_9 P_7
P_8 P_5 P_6
P_3 P_4 P_2
P_1 P_5 P_7
P_8 P_9 P_6
P_3 P_4 P_2
P_8 P_5 P_7
P_1 P_9 P_6
P_3 P_4 P_2
P_8 P_5 P_7
P_1 P_9 P_6
P_3 P_4 P_2
P_5 P_8 P_7
P_1 P_9 P_6
P_3 P_4 P_2

Hamiltonian Path

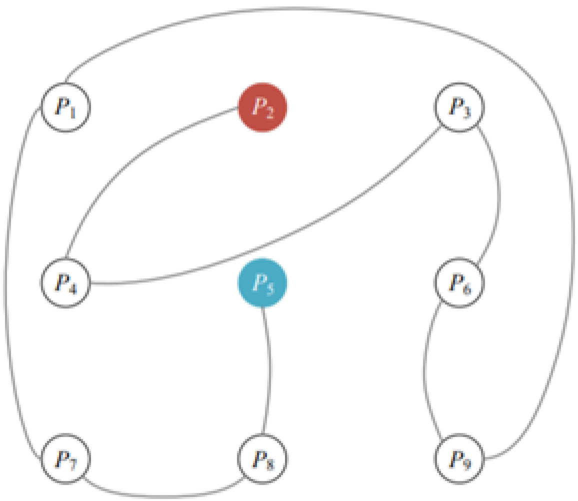


Figure 4. Generated Hamiltonian path. The red pixel is the beginning of the path and the blue pixel is the rear of the path.

Adjusted Bernoulli Map

$$x_{n+1} = 2x_n \, mod \; 1 = \begin{cases} 2x_n \; 0 < x_n < 0.5 \\ 2x_n - 1 \; 0.5 \le x_n < 1 \end{cases}$$

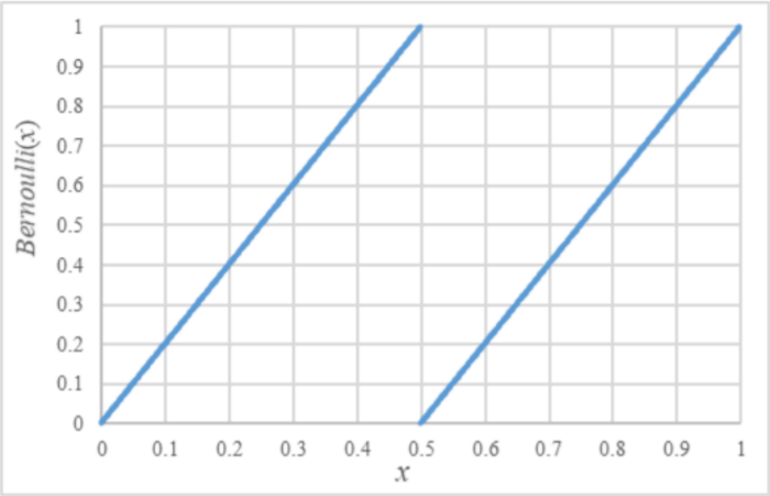


Figure 8. Bernoulli map.

Adjusted Bernoulli Map

為了增加Bernoulli Map非線性的特性，

我們在其中增加一次模運算，使其成為

$$x_{n+1} = \beta(\alpha x_n \ mod \ 1) \ mod \ 1$$

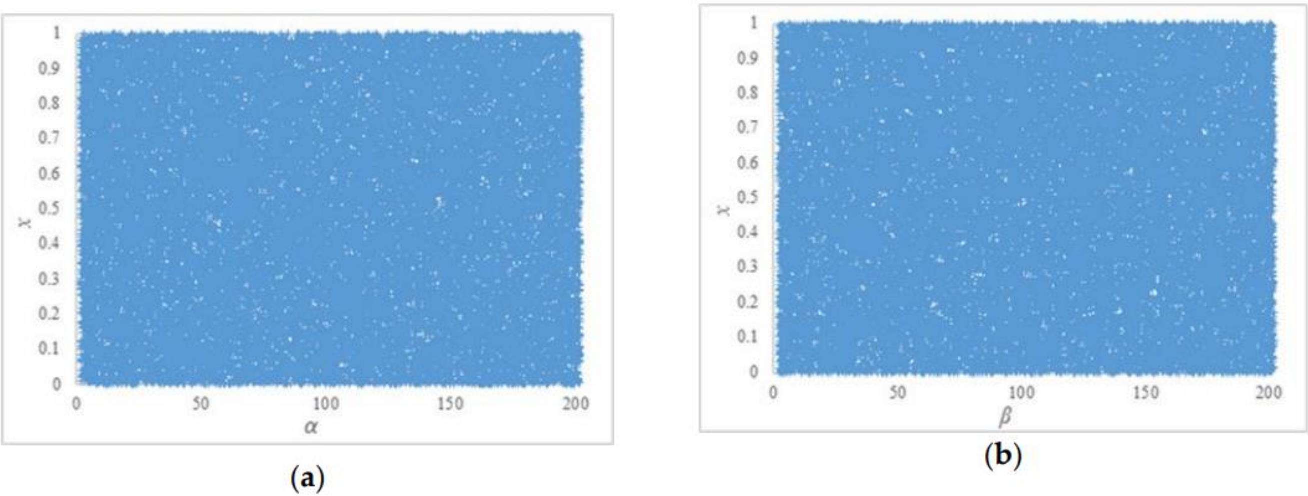


Figure 9. Bifurcation diagrams of ABM. (**a**) $\beta = 3$. The value of α is increased by 0.1, ranging from 2.1 to 202.1. (**b**) $\alpha = 3$. The value of β is increased by 0.1, ranging from 2.1 to 202.1.

加密

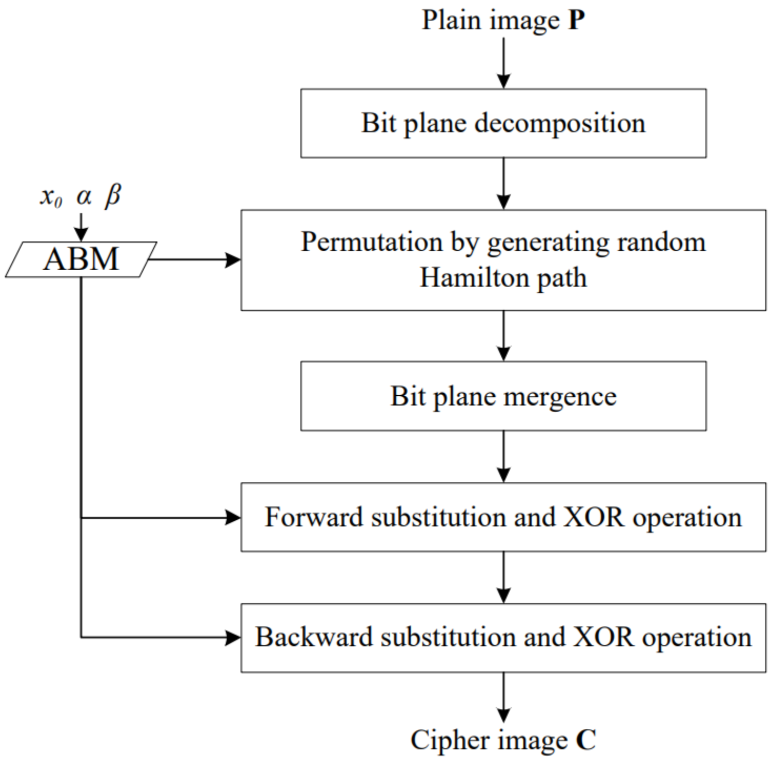


Figure 10. Encryption progress.

1. Bit plane decomposition

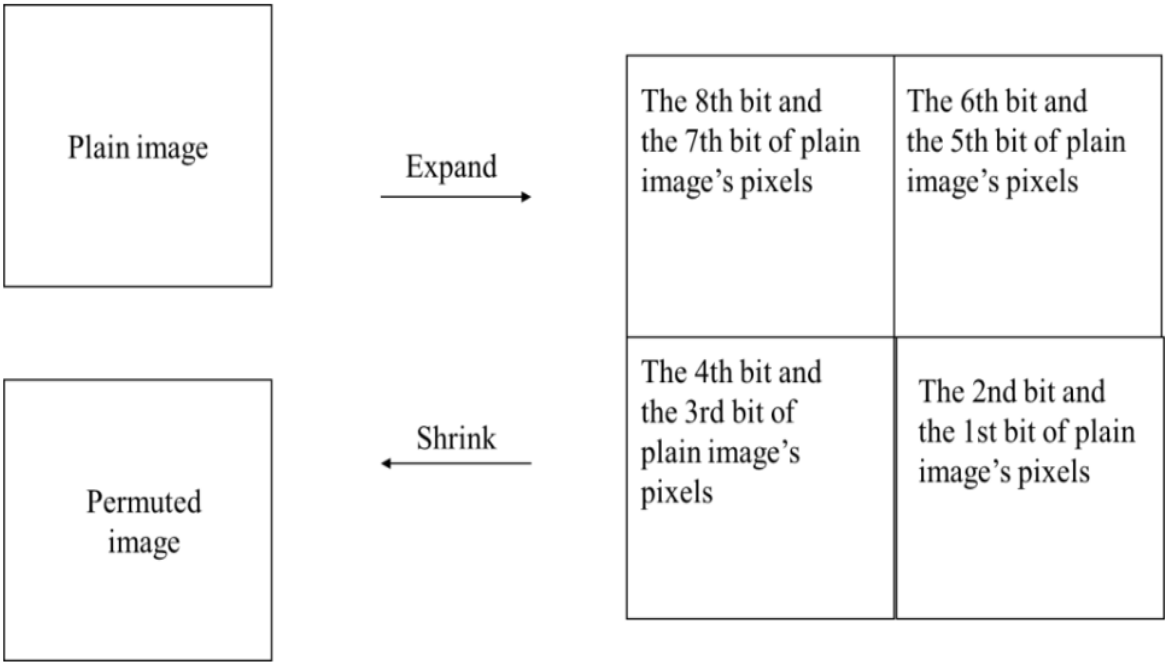


Figure 6. Modified expand–shrink strategy.

2. Permutation

將此Bit Plane套入前述漢米爾頓路徑的演算法。

將$i = 2M \times 2N, 2M \times 2N - 1, \ldots, 3, 2$依次執行，其中漢米爾頓路徑使用由ABM給定$x_0, \alpha, \beta$形成的序列$r_i$，$j = \mathrm{round}(r_i \times 10^{14})\ mod\ i + 1$來調換像素點j與i。

3. Bit plane Mergence

將排列完的圖片逆著**Bit plane decomposition**的算法，重組回原來的大小。

4.

Let i $= 0,1,\ldots,255,\ S[256], T[256]\ and\ S_i = T_i = i$.

Let i $= 255, 254, \ldots, 1$, 使用ABM依照i之順序生成Pseudo random number u_i, 並用$j = round(u_i * 10^{14})\ mod\ i$，調換 S_i, S_j 。

Let i $= 255, 254, \ldots, 1$, 使用ABM依照i之順序生成Pseudo random number p_i, 並用$j = round(p_i * 10^{14})\ mod\ i$，調換 T_i, T_j 。

5. Forward Substitution

Let $\mathrm{i} = 1,2,\ldots,MN-1$, 使用ABM依照i之順序生成Pseudo random number a_i, 將圖H的像素點

$$\mathrm{H}_{\mathrm{i}+1} = \mathrm{H}_{\mathrm{i}+1} \oplus \mathrm{S}_{\mathrm{T}_{\mathrm{H}_i}} \oplus (round(a_i \times 10^{14})\ mod\ 256)$$

使此圖達到密碼學上之擴散效果。

6. Backward Substitution

Let $\mathrm{i} = MN, MN-1, \dots, 2$, 使用ABM依照i之順序生成Pseudo random number b_i, 將圖H的像素點

$$\mathrm{H}_{\mathrm{i}-1} = \mathrm{H}_{\mathrm{i}-1} \oplus \mathrm{S}_{\mathrm{T}_{\mathrm{H}_i}} \oplus (round(b_i \times 10^{14})\ mod\ 256)$$

使此圖達到密碼學上之擴散效果。

解密

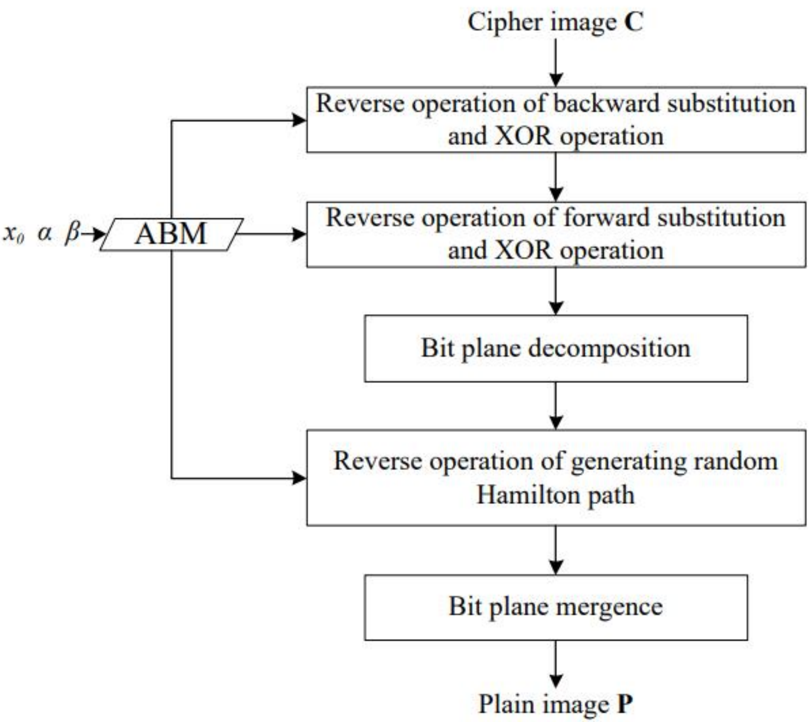


Figure 11. Decryption progress.

Implemented Source Code

https://github.com/jack111331/RHPImageEncryptionSuite

分析

運用NIST SP800-22 test suite用統計的方式的測試ABM序列的亂度。

Table 1. Randomness test using NIST SP800-22 test suite.

Statistical Tests	P-value	Pass Rate (%)
Frequency	0.798139	100.00
Block frequency	0.108791	99.33
Cumulative Sums *	0.282804	99.83
Runs	0.588652	99.67
Longest run	0.245072	99.33
Rank	0.319084	100.00
FFT	0.280306	99.00
Non overlapping template *	0.468139	98.95
Overlapping template	0.425059	98.00
Universal	0.449672	99.33
Approximate entropy	0.561227	99.67
Random excursions *	0.533005	98.95
Random excursions variant *	0.419542	99.27
Serial *	0.464632	98.83
Linear complexity	0.915745	99.33

* Average value of multiple tests.

密鑰的安全性

通常建議密鑰空間不低於2^{100}，而我們使用的密鑰是ABM的參數，亦即x_0, α, β，為3個倍精度浮點數，且按照IEEE 754的浮點數標準，每個倍精度浮點數的小數點部位都用了52 bits，所以很顯然我們使用的密鑰的空間是大於2^{100}。

Sensitivity

$$位元改變率：\mathrm{NBCR}(\mathrm{C}_1, C_2) = \frac{Ham(C_1, C_2)}{M \times N \times d}$$

Table 2. Key sensitivity ($\Delta = 0.00000000000001$).

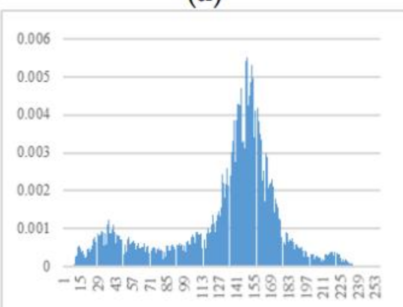
		Boat (512 × 512)	Couple (512 × 512)	Tank (512 × 512)	Male (1024 × 1024)	Clock (256 × 256)
Key sensitivity in encryption process	$(x_0 + \Delta, \alpha, \beta)$	0.499321	0.499997	0.500057	0.499904	0.501156
	$(x_0, \alpha + \Delta, \beta)$	0.499923	0.500155	0.499765	0.499937	0.500164
	$(x_0, \alpha, \beta + \Delta)$	0.49994	0.500076	0.500499	0.500078	0.500856
Key sensitivity in decryption process	$(x_0 + \Delta, \alpha, \beta)$	0.500289	0.499741	0.500082	0.499858	0.500328
	$(x_0, \alpha + \Delta, \beta)$	0.500154	0.499415	0.5001	0.49992	0.501308
	$(x_0, \alpha, \beta + \Delta)$	0.499747	0.500337	0.499742	0.49986	0.499378

Histogram 原圖

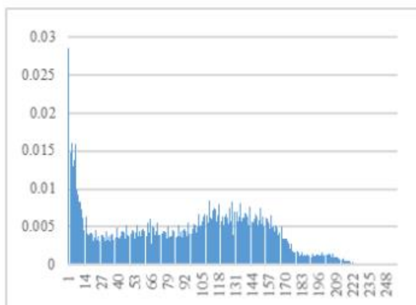
(a)

(b)

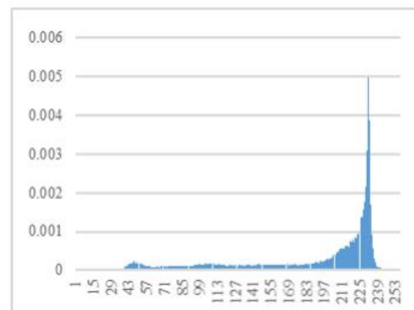
(c)



(d)



(e)



(f)

Histogram 加密圖

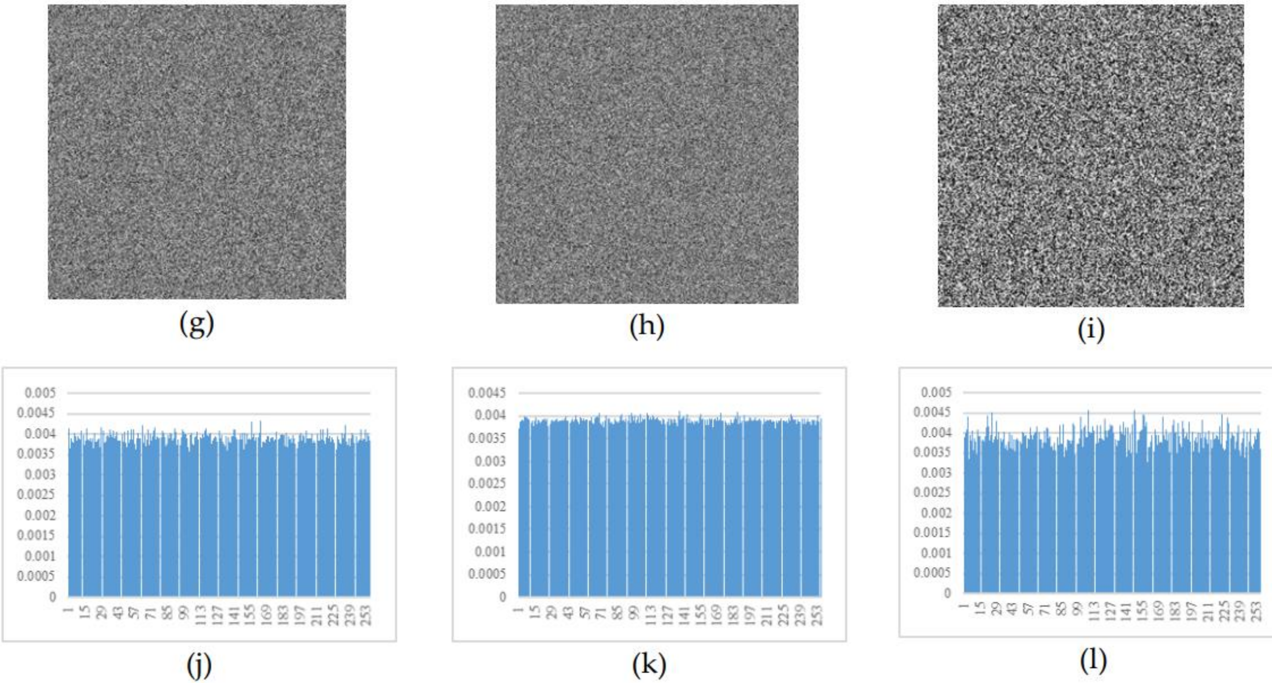


Figure 13. Histograms. (**a**) Plain image boat; (**b**) plain image male; (**c**) plain image clock; (**d**) histogram of plaintext boat; (**e**) histogram of plaintext male; (**f**) histogram of plaintext clock; (**g**) cipher image boat; (**h**) cipher image male; (**i**) cipher image clock; (**j**) histogram of cyphertext boat; (**k**) histogram of cyphertext male; (**l**) histogram of cyphertext clock.

Entropy

理想情況下從一張加密圖中計算出的Entropy應該要是每個像素所使用的Bit數d。

Table 4. Information entropy.

	Plain Image	Proposed Scheme	[2]	[4]	[28]
Chemical plant(256 × 256)	7.34243	7.99725	7.99716	7.99692	7.99683
Clock(256 × 256)	6.70567	7.99727	7.99726	7.99692	7.99705
Moon surface(256 × 256)	6.70931	7.99725	7.99738	7.9974	7.9972
Boat(512 × 512)	7.19137	7.99922	7.99934	7.9994	7.99921
Couple(512 × 512)	7.20101	7.99931	7.99934	7.99931	7.99936
Lena(512 × 512)	7.44551	7.99932	7.99929	7.99934	7.99932
Tank(512 × 512)	5.49574	7.99928	7.99934	7.99923	7.99934
Airplane(1024 × 1024)	5.64145	7.99984	7.99984	7.99983	7.99981
Airport(1024 × 1024)	6.83033	7.99984	7.99983	7.99981	7.99983
Male(1024 × 1024)	7.52374	7.99981	7.99978	7.99981	7.99981

分析

NPCR: Number of pixel change

UACI: Unified average changed intensity

$$\mathrm{NPCR} = \left[\frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N} D(i,j)\right] \times 100\%$$

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

$$\mathrm{D(i,j)} = \begin{cases} 0, C_1(i,j) = C_2(i,j) \\ 1, C_1(i,j) \neq C_2(i,j) \end{cases}$$

分析

隨機挑選一些像素，將這些像素的最後一位bit反轉，並使用該演算法加密

Table 5. Results of NPCR and UACI.

Index of Modified Pixel	NPCR (1 Round)	UACI (1 Round)	NPCR (2 Rounds)	UACI (2 Rounds)
0	0.996983	0.3349	0.995941	0.335169
255	0.99733	0.335224	0.996063	0.3338
511	0.996616	0.334727	0.996143	0.333902
65,151	0.99897	0.3355	0.996078	0.333911
65,407	0.998333	0.335641	0.996254	0.334896
130,560	0.996365	0.333719	0.995861	0.334763
130,816	0.99995	0.335614	0.996147	0.334734
131,071	0.999985	0.335876	0.996216	0.335797
196,096	0.999943	0.336146	0.995804	0.333875
196,352	0.999031	0.335429	0.996269	0.334645
261,632	0.9981	0.335204	0.996181	0.333829
261,888	0.999249	0.335551	0.996037	0.334519
262,143	0.997608	0.335361	0.995998	0.334605
Theoretical value	0.996094	0.334635	0.996094	0.334635

效率

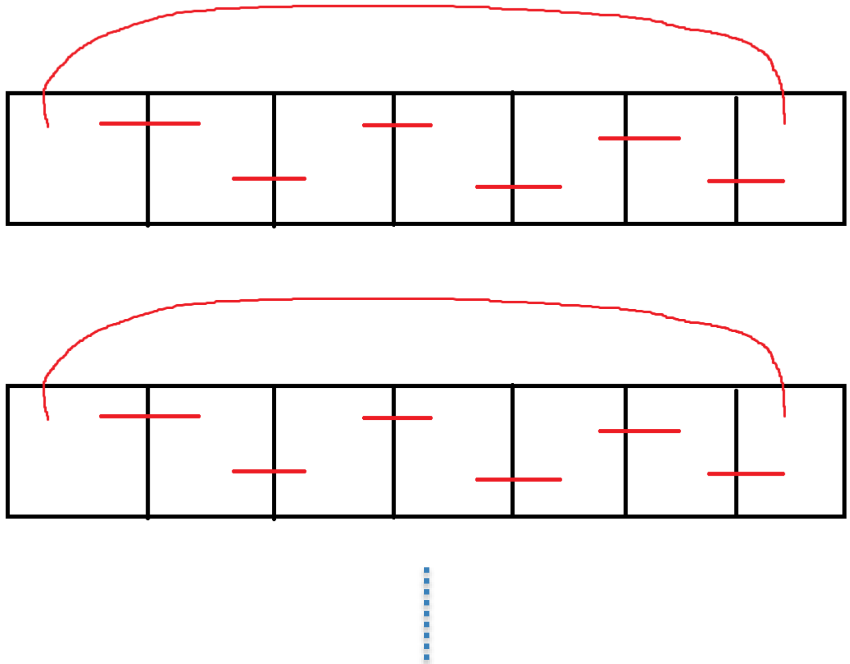
從上述演算法可以看出重排列與前向、後向替代等步驟的時間複雜度都是線性，因此整個圖片加密演算法為線性時間複雜度。

Correlation Coefficient Analysis(CCA) Improve

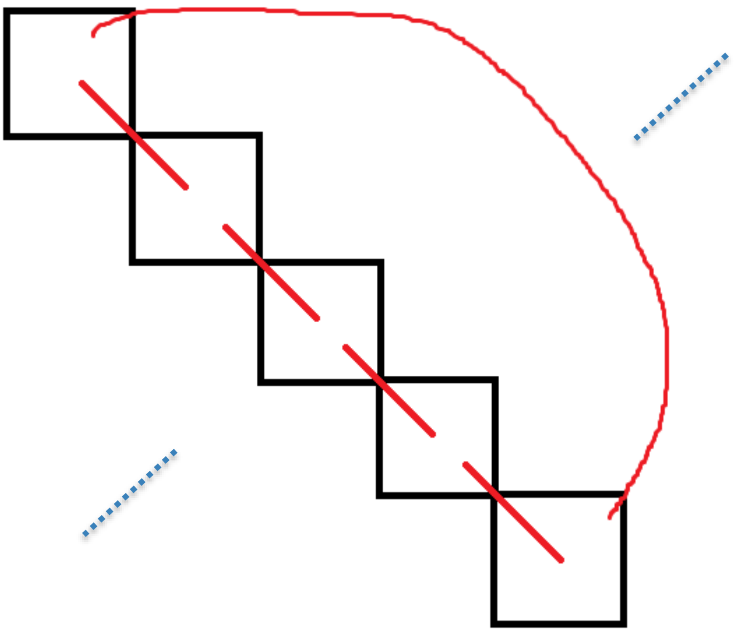
傳統研究上對於圖片相鄰像素的CCA分析使用的是:

- 水平線
- 垂直線
- 對角線

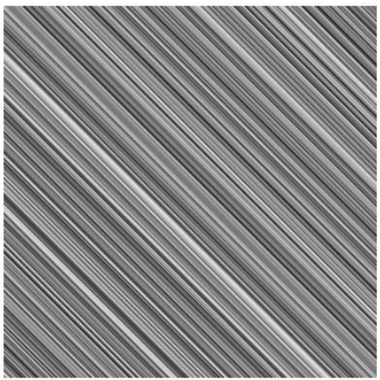
Horizontal CCA



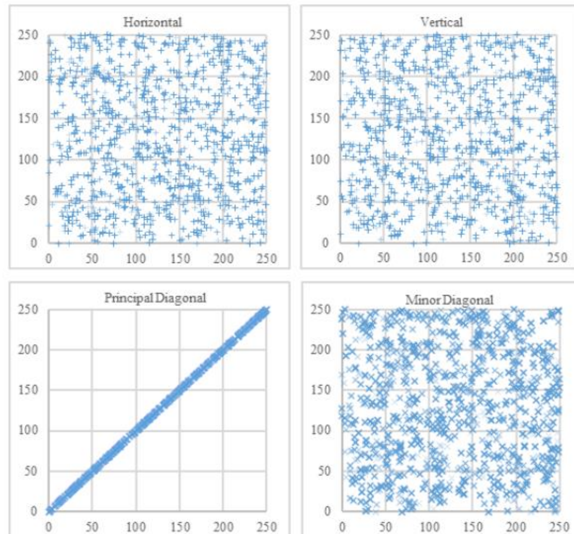
Diagonal CCA



Correlation Coefficient Analysis(CCA) Improve



(a)



(b)

Figure 15. Another example. (**a**) An image in which all adjacent pixels of principal diagonal direction are equal; (**b**) its scatter plots.

Correlation Coefficient Analysis(CCA) Improve

這篇論文提出研究上對於圖片相鄰像素的CCA分析使用的應為:

- 水平線
- 垂直線
- 對角線
- 反對角線

Correlation Coefficient Analysis(CCA) Improve

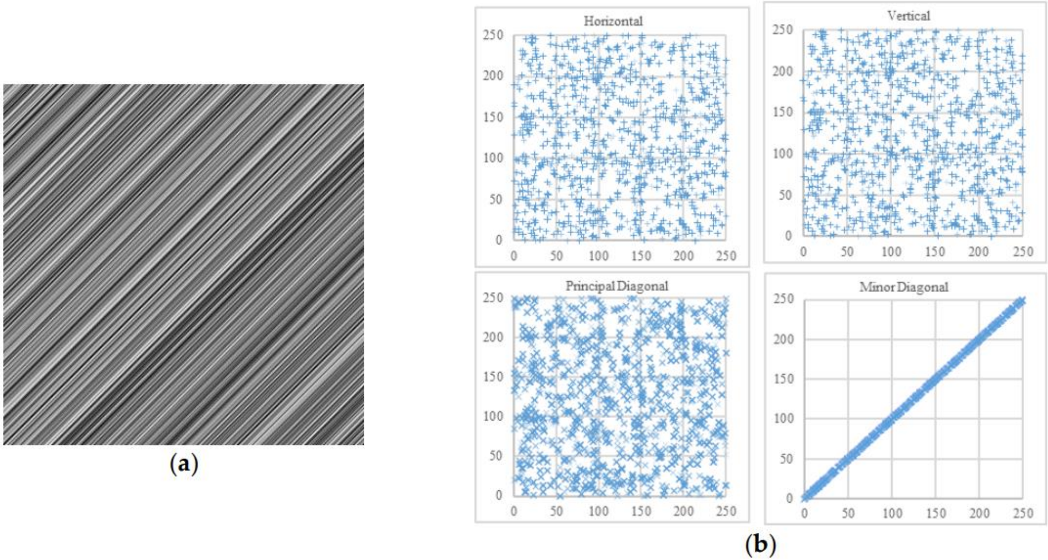


Figure 14. One example. (**a**) An image in which all adjacent pixels of minor diagonal direction are equal; (**b**) its scatter plots.

拓展-彩色圖

上述運用的情境是8-bit per pixel的黑白圖，但若要拓展運用情境到32-bit per pixel的彩色圖…

拓展-彩色圖

將32-bit彩色圖拆分為4個8-bit黑白圖來加解密:D

Trying To Decrypt With Known Info

假設攻擊者擁有α、β、以及經過整個加密過程後的x…

Trying To Decrypt With Known Info

假設攻擊者擁有α、β、以及經過整個加密過程後的x…

Impossible.. 因為在ABM序列生成過程中，x_0乘以α或β時，因為IEEE 754定義的浮點數會捨去浮點數中的部分Mantisa，

因此IEEE754定義中浮點數的缺點在此成為防止逆推x的優點。

Trying To Decrypt With Known Info

假設攻擊者擁有α、β、以及經過整個加密過程後的x...

但是還是可以暴力破出來喔:D

因為x只有64-bit，可以窮舉x來破解