COMPARING AND ANALYZING THE SECURITY AND EFFICIENCY OF

CHACHA20, SALSA20, AND AES.

MEMBER:

0716210 林千慧

0716215 王心柔

0713309 徐宜蓉

0716047 張景欣

簡介

- AES
- Salsa20
- ChaCha20

分析

- Security
- Efficiency

結論

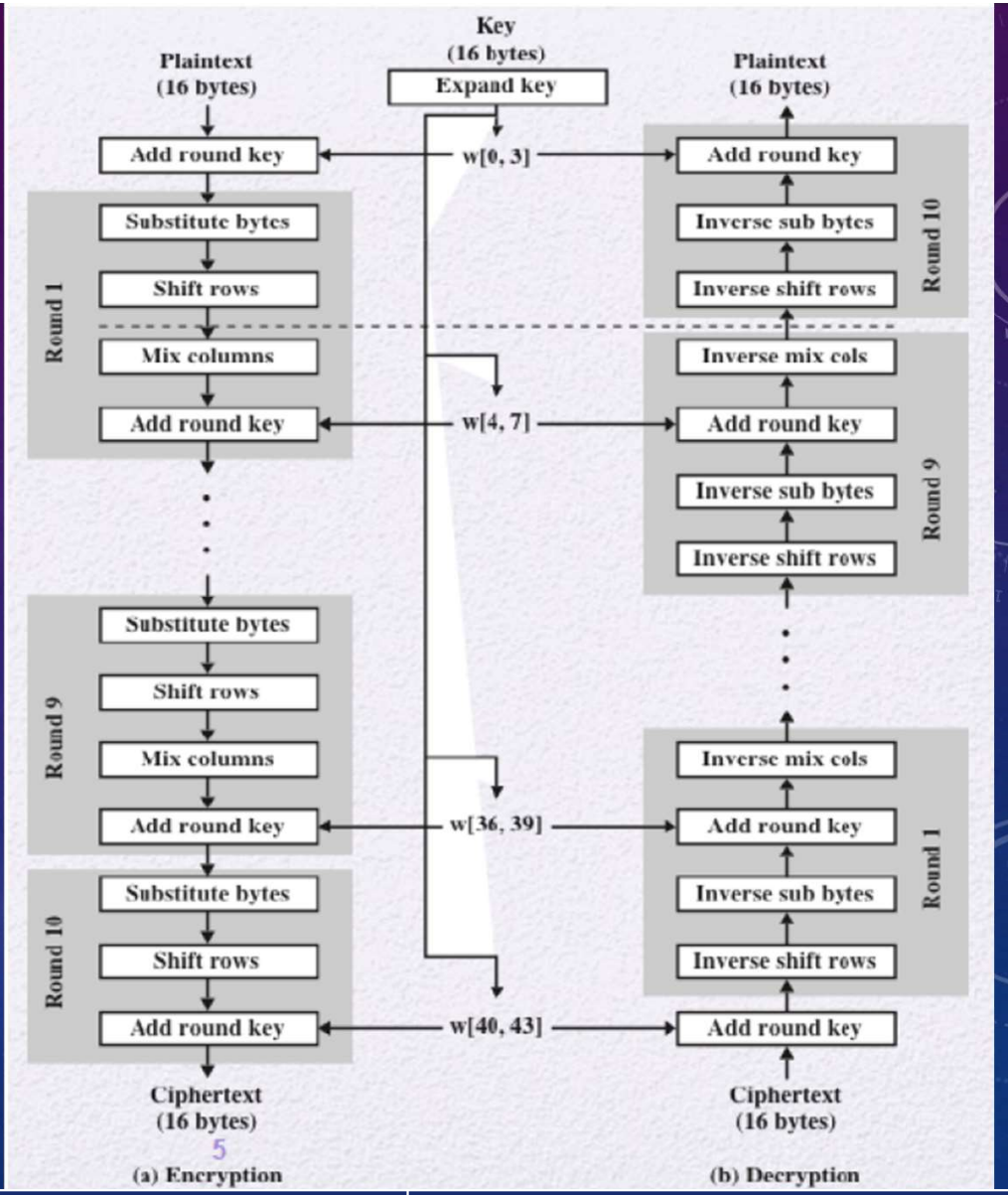
- 根據安全性、效率分析應用

AES (= RIJNDAEL演算法)

- symmetric block cipher
- NIST所舉辦的加密法選拔賽中勝出的演算法
- 區塊長度固定爲128bit
- 金鑰長度可以爲128、192、256bit，分別對應的round爲10、12、14。

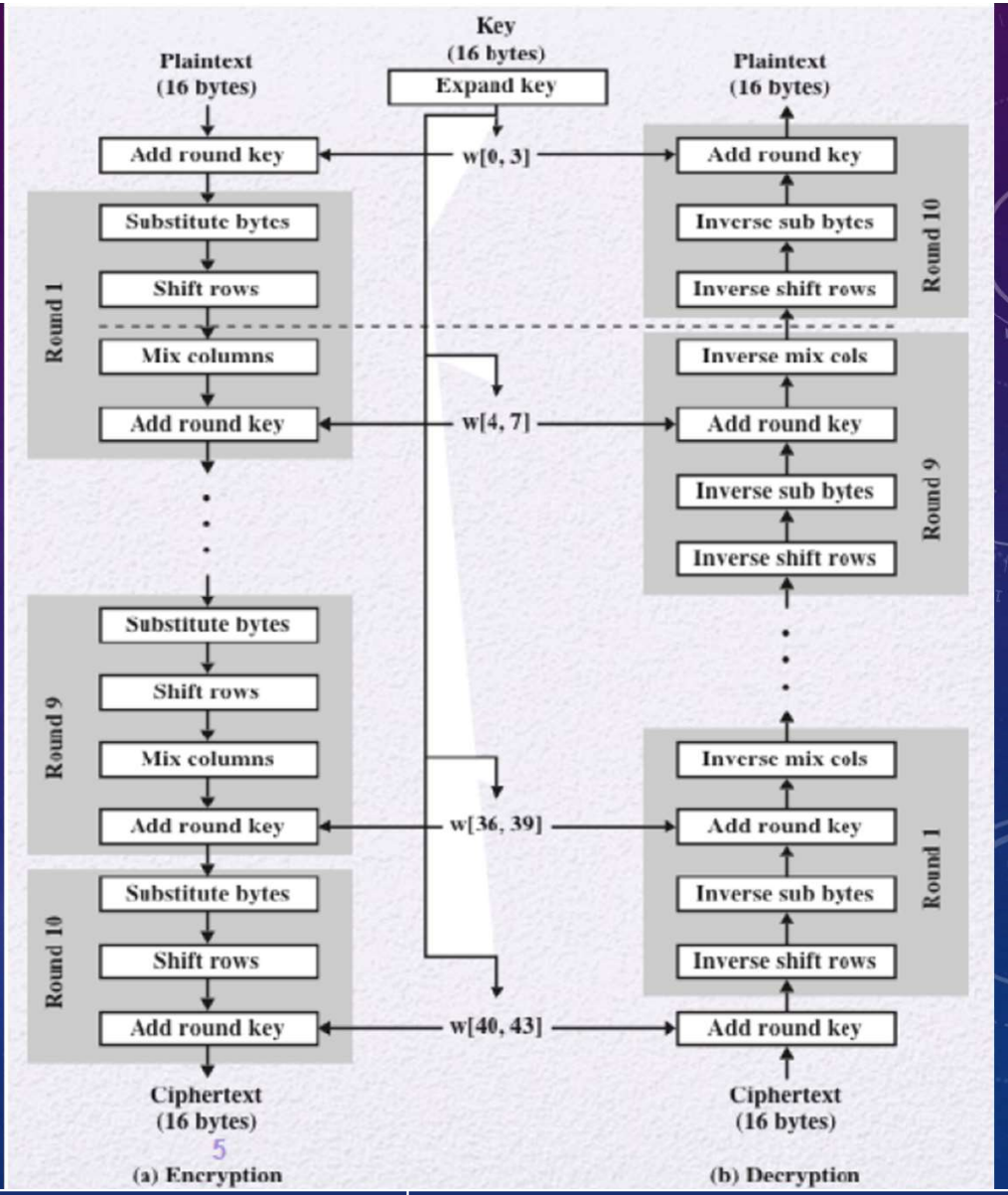
運算過程

- 加解密流程圖
- AES加密運作是在一個4x4的位元組上運作，以下是明文轉換的範例。



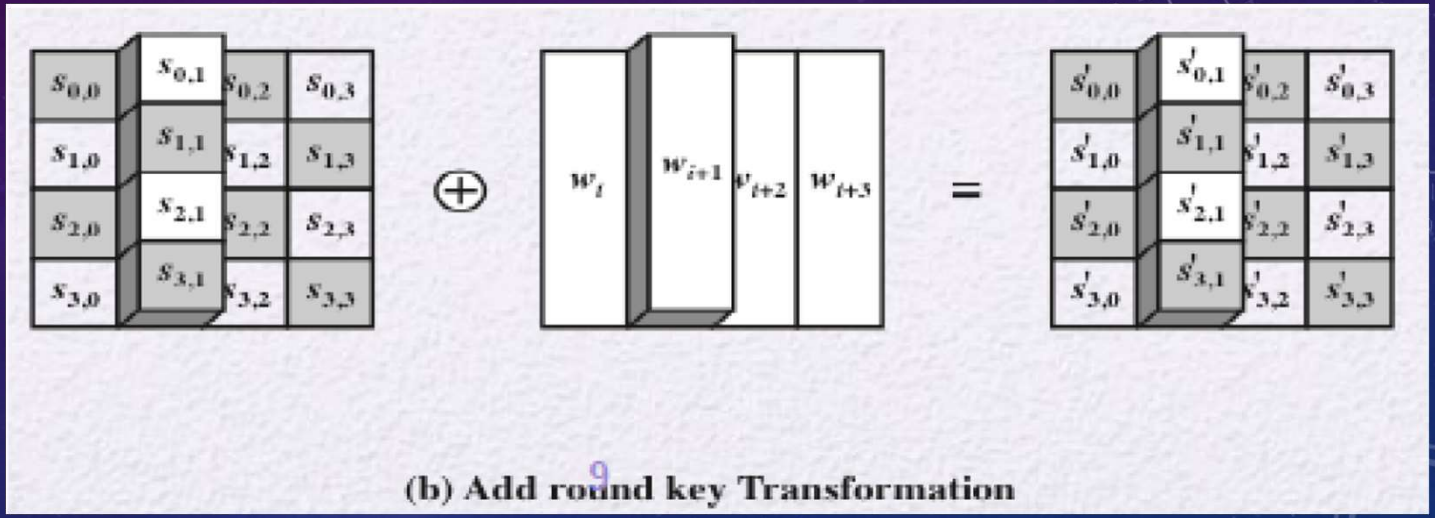
運算過程

1. AddRoundKey
2. SubBytes
3. ShiftRows
4. Mixcolumns



運算過程

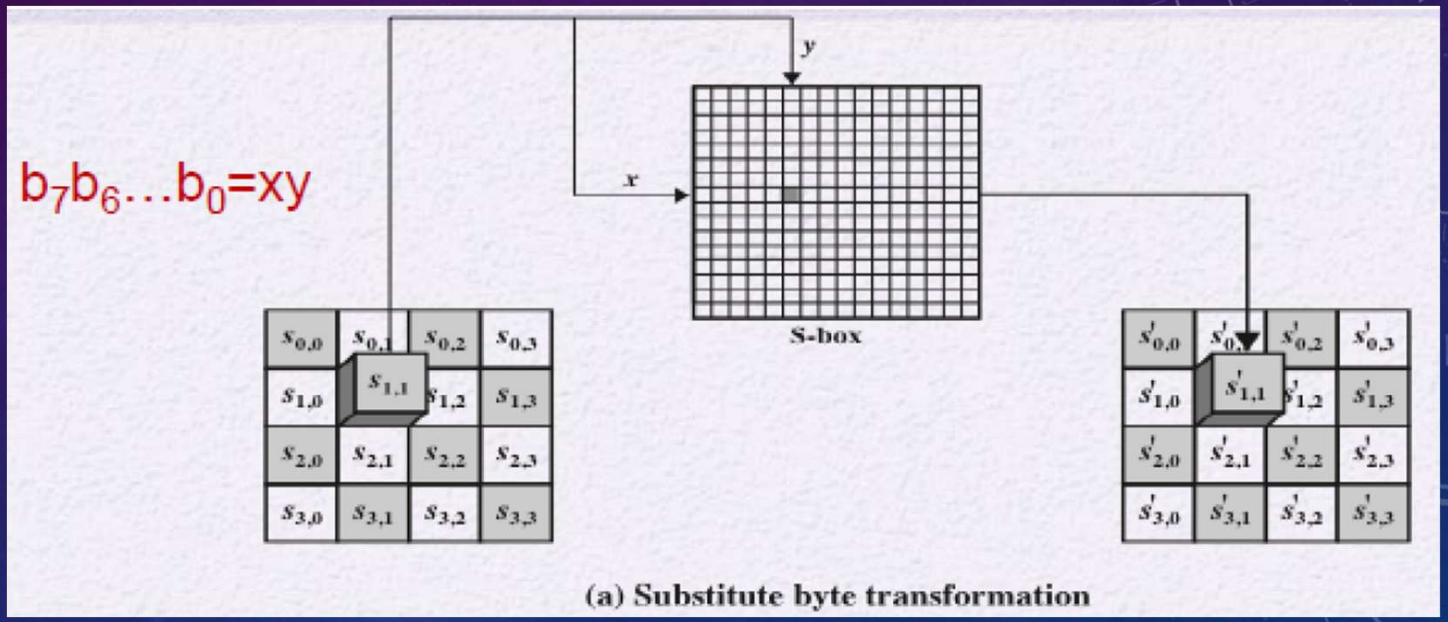
1. AddRoundKey
2. SubBytes
3. ShiftRows
4. Mixcolumns



(b) Add round key Transformation

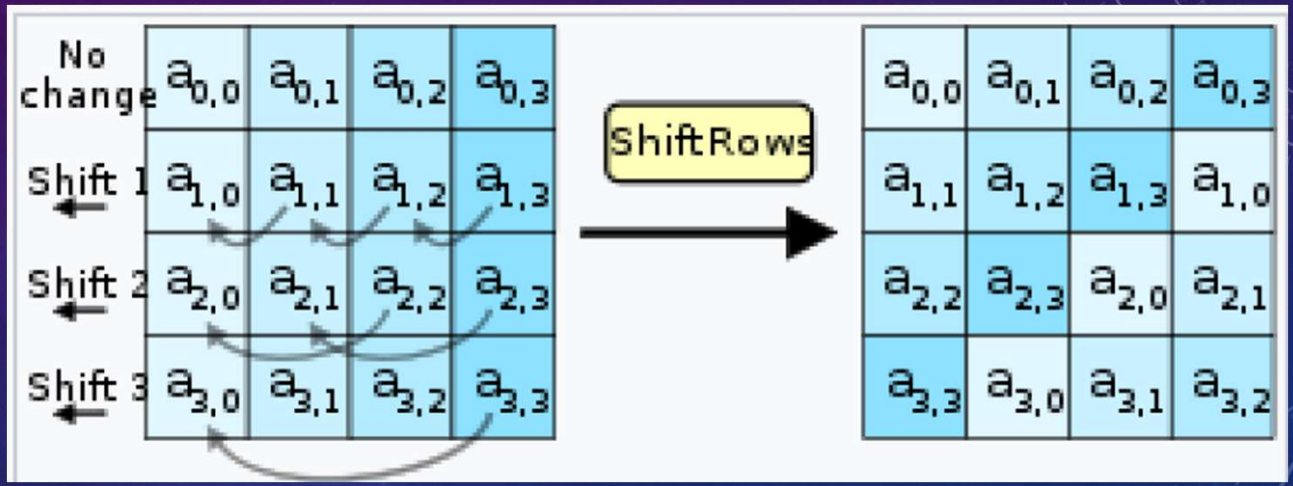
運算過程

1. AddRoundKey
2. SubBytes
3. ShiftRows
4. Mixcolumns



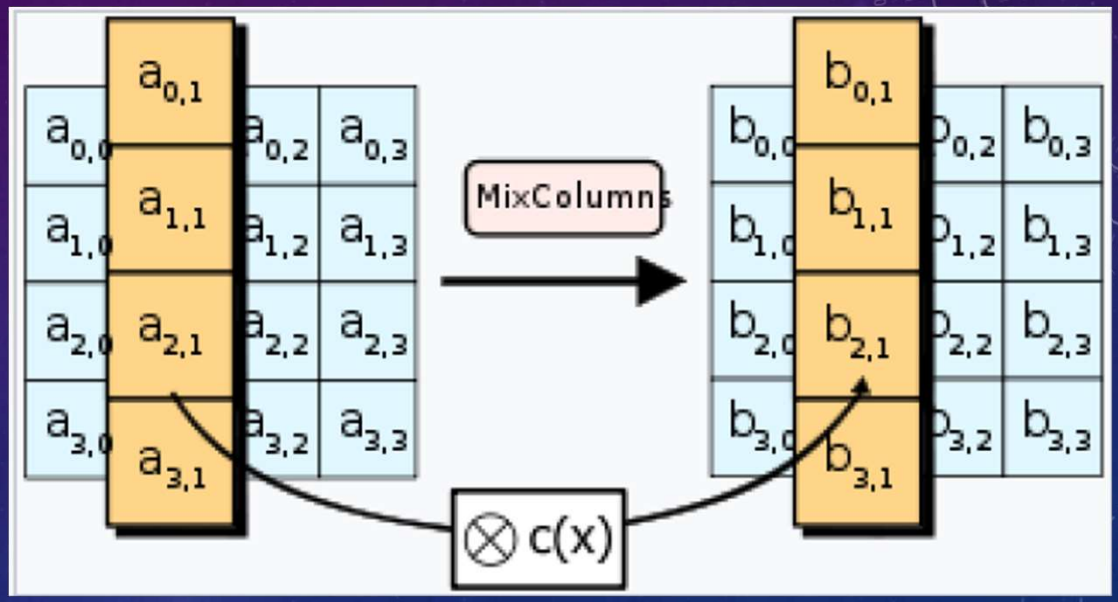
運算過程

1. AddRoundKey
2. SubBytes
3. ShiftRows
4. Mixcolumns



運算過程

1. AddRoundKey
2. SubBytes
3. ShiftRows
4. Mixcolumns



$$c(x) = 3x^3 + x^2 + x + 2$$

安全分析—AES

現有的破解方案：

- side-channel attacks：攻擊基於不安全系統上的加密系統。
  Ex. TEMPEST、power comsumption、time analysis、acoustic

- social engineering：通常被認爲是欺詐、行騙和入侵電腦系統的行爲。
  Ex.釣魚網站、釣魚郵件或是電話釣魚。

SALSA20

- Stream cipher
- 迴圈核心函數: $b \boxplus= (a \boxplus c) \lll k;$
- 金鑰長度：128或256位
- 狀態長度：512位元
- 結構：ARX
- 重複回數：20
- 初始狀態是根據金鑰的8個word、流位置的2個word、nonce的兩個word（基本上是額外的流位置）和4個固定word製成。20輪迴圈混合製成16個word的串流加密法輸出。

運算過程

```
x[ 4] ⊕= (x[ 0] ⊞ x[12])<<<7;    x[ 9] ⊕= (x[ 5] ⊞ x[ 1])<<<7;
x[14] ⊕= (x[10] ⊞ x[ 6])<<<7;    x[ 3] ⊕= (x[15] ⊞ x[11])<<<7;
x[ 8] ⊕= (x[ 4] ⊞ x[ 0])<<<9;    x[13] ⊕= (x[ 9] ⊞ x[ 5])<<<9;
x[ 2] ⊕= (x[14] ⊞ x[10])<<<9;    x[ 7] ⊕= (x[ 3] ⊞ x[15])<<<9;
x[12] ⊕= (x[ 8] ⊞ x[ 4])<<<13;   x[ 1] ⊕= (x[13] ⊞ x[ 9])<<<13;
x[ 6] ⊕= (x[ 2] ⊞ x[14])<<<13;   x[11] ⊕= (x[ 7] ⊞ x[ 3])<<<13;
x[ 0] ⊕= (x[12] ⊞ x[ 8])<<<18;   x[ 5] ⊕= (x[ 1] ⊞ x[13])<<<18;
x[10] ⊕= (x[ 6] ⊞ x[ 2])<<<18;   x[15] ⊕= (x[11] ⊞ x[ 7])<<<18;

x[ 1] ⊕= (x[ 0] ⊞ x[ 3])<<<7;    x[ 6] ⊕= (x[ 5] ⊞ x[ 4])<<<7;
x[11] ⊕= (x[10] ⊞ x[ 9])<<<7;    x[12] ⊕= (x[15] ⊞ x[14])<<<7;
x[ 2] ⊕= (x[ 1] ⊞ x[ 0])<<<9;    x[ 7] ⊕= (x[ 6] ⊞ x[ 5])<<<9;
x[ 8] ⊕= (x[11] ⊞ x[10])<<<9;    x[13] ⊕= (x[12] ⊞ x[15])<<<9;
x[ 3] ⊕= (x[ 2] ⊞ x[ 1])<<<13;   x[ 4] ⊕= (x[ 7] ⊞ x[ 6])<<<13;
x[ 9] ⊕= (x[ 8] ⊞ x[11])<<<13;   x[14] ⊕= (x[13] ⊞ x[12])<<<13;
x[ 0] ⊕= (x[ 3] ⊞ x[ 2])<<<18;   x[ 5] ⊕= (x[ 4] ⊞ x[ 7])<<<18;
x[10] ⊕= (x[ 9] ⊞ x[ 8])<<<18;   x[15] ⊕= (x[14] ⊞ x[13])<<<18;
```

⊞：模加2^32

⊕：互斥

<<<：左旋操作

安全分析—SALSA20

- 截至2015年，沒有已知的對Salsa20/12或完整Salsa20/20的攻擊被發布；已知的最佳攻擊是打破12輪或20輪迴圈中的8輪。
- 在很多CPU上 Salsa20都是花constant time，所以對power attacks和其他side-channel attacks的抵抗性很好。

CHACHA20

- stream cipher
- Salsa20的改良版，更能抵抗密碼分析攻擊。
- 安全性取決於PRNG強度：
  - 加密/解密過程：把plaintext/ ciphertext和keystream做xor運算
- 有被運用在FreeBSD, OpenBSD和NetBSD等作業系統中的arc4random亂數生成器，取代已經脆弱的RC4。
- 不能保證authenticity
- 和Message Authentication Code一起使用，來達成網路安全通訊，例如ChaCha20-Poly1305。
- 在沒有對AES指令做加速的CPU上ChaCha20的performance通常比AES好

運算過程

1. 初始矩陣
2. 置換
3. QUARTERROUND
4. 生成keystream
5. 加密
6. 解密

運算過程

1. 初始矩陣
2. 置換
3. QUARTERROUND
4. 生成keystream
5. 加密
6. 解密

constant[0]	constant[1]	constant[2]	constant[3]
0x61707865	0x3320646e	0x79622d32	0x6b206574
key[0]	key[1]	key[2]	key[3]
key[4]	key[5]	key[6]	key[7]
counter	nonce[0]	nonce[1]	nonce[2]

運算過程

1. 初始矩陣
2. 置換
3. QUARTERROUND
4. 生成keystream
5. 加密
6. 解密

行置換 ： $M_{next}[i][j] = M_{current}[j][i]$

X[0]	X[1]	X[2]	X[3]
X[4]	X[5]	X[6]	X[7]
X[8]	X[9]	X[A]	X[B]
X[C]	X[D]	X[E]	X[F]

→

X[0]	X[4]	X[8]	X[C]
X[1]	X[5]	X[9]	X[D]
X[2]	X[6]	X[A]	X[E]
X[3]	X[7]	X[B]	X[F]

列置換 ： $M_{next}[i][j] = M_{current}[(i+j)\%4][j]$

X[0]	X[4]	X[8]	X[C]
X[1]	X[5]	X[9]	X[D]
X[2]	X[6]	X[A]	X[E]
X[3]	X[7]	X[B]	X[F]

→

X[0]	**X[5]**	**X[A]**	**X[F]**
X[1]	X[6]	X[B]	X[C]
X[2]	X[7]	X[8]	X[D]
X[3]	X[4]	X[9]	X[E]

運算過程

1. 初始矩陣
2. 置換
3. QUARTERROUND
4. 生成keystream
5. 加密
6. 解密

ex. 做完第一次行置換的矩陣如下 -

X[0]	**X[4]**	**X[8]**	**X[C]**
X[1]	X[5]	X[9]	X[D]
X[2]	X[6]	X[A]	X[E]
X[3]	X[7]	X[B]	X[F]

依序執行QUARTERROUND -

```
QUARTERROUND（X[0]，X[4]，X[8]，X[C]）；
QUARTERROUND（X[1]，X[5]，X[9]，X[D]）；
QUARTERROUND（X[2]，X[6]，X[A]，X[E]）；
QUARTERROUND（X[3]，X[7]，X[B]，X[F]）；
```

運算過程

1. 初始矩陣
2. 置換
3. QUARTERROUND
4. 生成keystream
5. 加密
6. 解密

QUARTERROUND做的事情如下(令輸入爲a, b, c, d)-

```
a ⊞= b; d ⊕= a; d <<<= 16;
c ⊞= d; b ⊕= c; b <<<= 12;
a ⊞= b; d ⊕= a; d <<<= 8;
c ⊞= d; b ⊕= c; b <<<= 7;
```

⊞ ： bitwise add,
⊕ ： bitwise xor,
<<< ： constant-distance rotation operation

例如a=0x11111111，b=0x01020304，c=0x77777777，d=0x01234567

```
c ⊞= d; b ⊕= c; b <<<= 7
```

$c \boxplus= d = 0x77777777 \boxplus 0x01234567 = 0x789abcde$

$b \oplus= d = 0x01020304 \oplus 0x789abcde = 0x7998bfda$

$b <<< 7 = 0x7998bfda <<< 7$

$= 01111001100110001011111111011010 \lll 7$

$= 11001100010111111101101001111 00 = 0xcc5fed3c$

運算過程

1. 初始矩陣
2. 置換
3. QUARTERROUND
4. 生成keystream
5. 加密
6. 解密

運算過程

1. 初始矩陣
2. 置換
3. QUARTERROUND
4. 生成keystream
5. 加密
6. 解密

運算過程

1. 初始矩陣
2. 置換
3. QUARTERROUND
4. 生成keystream
5. 加密
6. 解密

實驗方法

[Security]

- 從網路上隨便抓六萬多字的明文拿來用AES加密，丟進nist 測試，全部通過。證明產生的密文是夠亂的。
- Salsa20跟ChaCha20是stream cipher，安全性取決於PRNG的強度，故可以透過NIST SP 800-22及AIS-31兩種test-suite來作爲衡量其安全性的標準之一。

[Efficiency]

- 透過pyRAPL來測量加解密過程中的energy consumption，作爲衡量效率的標準之一。

Security-- AES

- NIST SP 800-22：

(1) 每個sequence有512000bit，共264bitstream

(2) 部分結果

RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES

generator is <hp.bin>

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
23	25	23	26	23	35	22	28	30	29	0.747592	261/264	Frequency
14	30	25	34	24	32	22	24	36	23	0.086796	263/264	BlockFrequency
20	28	18	25	31	38	35	15	27	27	0.030965	261/264	CumulativeSums
22	26	24	22	26	28	29	40	25	22	0.344294	261/264	CumulativeSums
27	33	21	21	20	36	27	35	25	19	0.116957	262/264	Runs
29	17	27	37	27	17	23	31	25	31	0.136680	261/264	LongestRun
24	25	30	28	34	22	23	30	22	26	0.777709	263/264	Rank
29	39	32	19	29	23	21	27	16	29	0.066882	258/264	FFT
21	32	26	22	25	25	31	31	26	25	0.834308	263/264	NonOverlappingTemplate
24	31	22	25	32	21	22	33	29	25	0.637119	262/264	NonOverlappingTemplate
18	26	27	28	27	19	31	26	36	26	0.402554	261/264	NonOverlappingTemplate
29	29	34	24	21	23	26	27	26	25	0.860264	261/264	NonOverlappingTemplate
37	31	30	21	21	25	18	31	24	26	0.226634	261/264	NonOverlappingTemplate
26	30	27	19	32	28	31	14	29	28	0.270473	261/264	NonOverlappingTemplate
26	21	25	26	28	26	25	33	27	27	0.957882	260/264	NonOverlappingTemplate
23	32	31	29	27	32	19	22	24	25	0.621105	263/264	NonOverlappingTemplate
29	19	30	33	30	21	23	29	24	26	0.621105	261/264	NonOverlappingTemplate
24	24	21	34	30	31	21	26	29	24	0.661132	261/264	NonOverlappingTemplate
21	27	17	19	29	28	28	33	38	24	0.114350	263/264	NonOverlappingTemplate
19	27	35	30	25	29	21	25	28	25	0.613107	262/264	NonOverlappingTemplate
27	20	30	37	20	23	27	29	28	23	0.409370	259/264	NonOverlappingTemplate
21	29	27	29	30	25	32	25	23	23	0.878500	261/264	NonOverlappingTemplate
22	37	21	20	30	28	25	27	24	30	0.416251	263/264	NonOverlappingTemplate
25	28	24	35	28	19	29	29	22	25	0.653133	264/264	NonOverlappingTemplate
18	29	22	36	25	23	28	30	36	17	0.080899	262/264	NonOverlappingTemplate
30	24	25	27	35	24	30	23	26	20	0.692983	260/264	NonOverlappingTemplate
24	19	24	22	25	24	33	33	29	31	0.526413	264/264	NonOverlappingTemplate

Security-- SALSA20

- NIST SP 800-22：

(1) 每個sequence有512000bit，共264bitstream

(2) 部分結果

RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES

generator is <Salsa20.bin>

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
26	32	27	24	32	24	25	25	32	17	0.565427	261/264	Frequency
22	30	24	22	27	37	25	18	33	26	0.270473	262/264	BlockFrequency
29	25	26	26	34	24	23	24	30	23	0.878500	264/264	CumulativeSums
24	23	27	34	26	24	32	22	22	30	0.700892	263/264	CumulativeSums
35	20	28	33	30	26	28	21	24	19	0.326155	263/264	Runs
38	20	31	27	24	30	28	18	24	24	0.235870	258/264	LongestRun
24	32	24	27	35	22	33	26	20	21	0.375964	264/264	Rank
31	23	31	26	29	29	25	20	22	28	0.820696	261/264	FFT
25	28	18	26	25	29	24	33	23	33	0.605118	263/264	NonOverlappingTemplate
36	23	25	25	24	20	28	27	26	30	0.677093	263/264	NonOverlappingTemplate
20	30	21	27	30	26	25	26	38	21	0.338175	263/264	NonOverlappingTemplate
25	36	24	26	23	18	29	30	30	23	0.458843	260/264	NonOverlappingTemplate
26	20	21	26	29	32	25	31	28	26	0.813745	262/264	NonOverlappingTemplate
31	22	24	25	26	29	24	27	31	25	0.950837	262/264	NonOverlappingTemplate
23	26	32	21	27	31	28	29	23	24	0.860264	263/264	NonOverlappingTemplate
29	33	29	26	31	20	32	24	19	21	0.409370	263/264	NonOverlappingTemplate
33	22	18	18	30	27	25	31	29	31	0.320255	260/264	NonOverlappingTemplate
22	24	38	24	29	26	27	20	27	27	0.503460	262/264	NonOverlappingTemplate
19	25	28	23	22	25	39	26	23	34	0.192485	262/264	NonOverlappingTemplate
24	24	27	24	30	17	31	30	29	28	0.708772	261/264	NonOverlappingTemplate
19	27	35	28	24	18	23	31	32	27	0.308675	263/264	NonOverlappingTemplate
20	27	28	33	19	30	31	24	20	32	0.363083	261/264	NonOverlappingTemplate
28	31	15	26	26	23	22	35	34	24	0.188522	260/264	NonOverlappingTemplate
23	24	29	16	27	24	37	30	27	27	0.332128	261/264	NonOverlappingTemplate
15	19	22	27	30	32	28	34	33	24	0.127904	262/264	NonOverlappingTemplate
29	23	38	30	17	30	18	27	32	20	0.070165	262/264	NonOverlappingTemplate
24	30	32	31	22	23	33	24	19	26	0.534146	262/264	NonOverlappingTemplate
29	22	34	29	26	19	31	25	26	23	0.637119	263/264	NonOverlappingTemplate
27	25	27	25	28	24	22	22	34	30	0.853943	261/264	NonOverlappingTemplate
23	25	28	32	23	20	33	23	24	33	0.541916	260/264	NonOverlappingTemplate
29	23	17	26	27	34	23	16	36	33	0.062216	263/264	NonOverlappingTemplate
28	22	26	27	35	28	20	26	33	19	0.416251	261/264	NonOverlappingTemplate
23	21	30	30	32	32	26	22	23	25	0.708772	258/264	NonOverlappingTemplate
29	28	25	29	35	25	25	26	18	24	0.669119	263/264	NonOverlappingTemplate
26	18	19	24	34	25	21	27	34	36	0.111793	264/264	NonOverlappingTemplate
23	23	38	28	27	28	29	28	22	18	0.338175	261/264	NonOverlappingTemplate
22	24	35	33	35	24	23	17	21	30	0.119615	260/264	NonOverlappingTemplate
24	26	23	33	25	35	29	25	13	31	0.145961	261/264	NonOverlappingTemplate
27	28	20	33	19	28	29	26	26	28	0.739918	261/264	NonOverlappingTemplate
22	29	24	29	31	23	19	28	32	27	0.716618	262/264	NonOverlappingTemplate

Security-- CHACHA20

- NIST SP 800-22：

(1) 使用randomgen裡的ChaCha，可以改變round數(必須>=2, 且爲偶數)

(2) bit stream length = 512000, number of bit streams = 264

(3) 因爲據說ChaCha20, ChaCha12, ChaCha8是比較安全的，所以先分別測試round = 20/12/8的結果，發現都可以通過全部測試。

(4) 之後再遞減round數做測試：

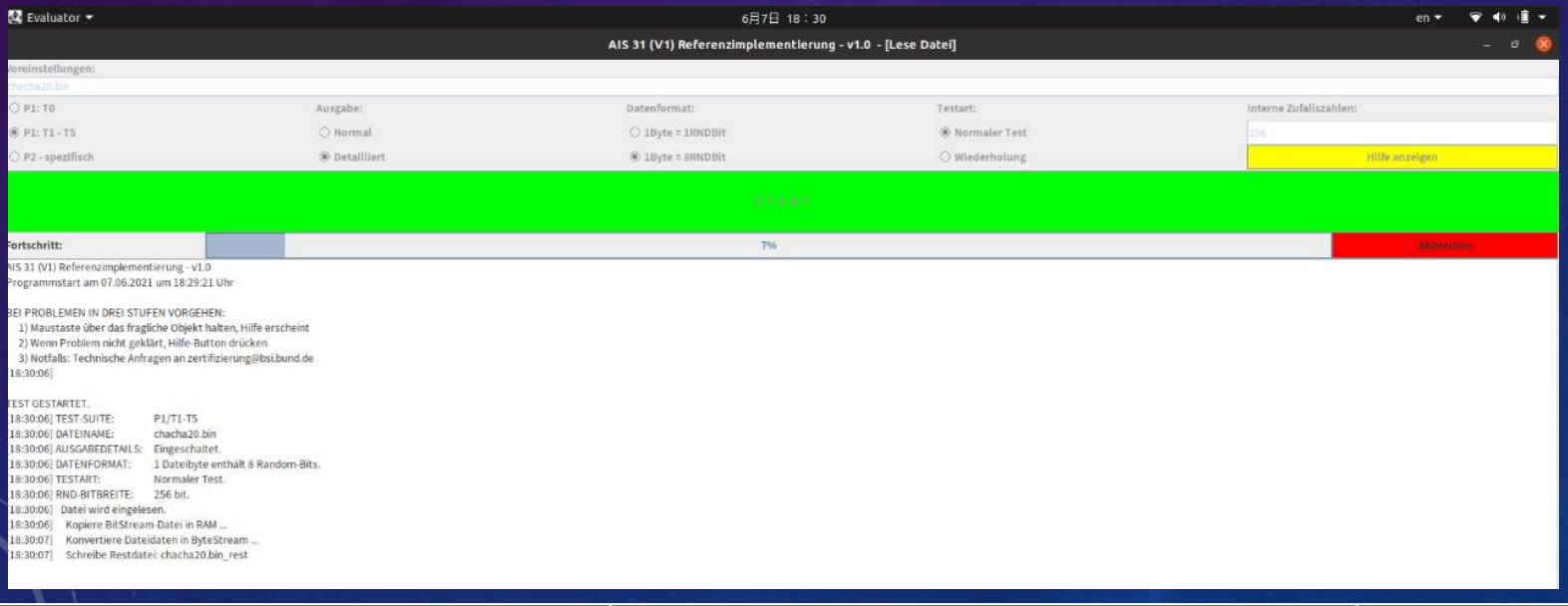
  - 發現ChaCha6和ChaCha4基本上可通過全部測試，偶爾會發生有1, 2項測試沒過的情形；
  - ChaCha2則完全無法通過全部測試。

Security-- CHACHA20

- AIS-31：

(1) 使用randomgen裡的ChaCha，round = 20。

(2) 執行截圖如下



(3)ChaCha20所產生的random bits, 通過了所有測試(P1: T0, P1:T1-T5, P2)

1

[18:44:07]TEST STARTED.
[18:44:07] TEST SUITE: P2 (specific tests)
[18:44:07] Filename: chacha20.bin
[18:44:07] ISSUE DETAILS: Enabled.
[18:44:07] DATA FORMAT: 1 file byte contains 8 random bits.
[18:44:07] TEST START: Normal test.
[18:44:07] RND BIT WIDTH: 256 bits.
[18:44:07] The file is being read.

2

[18:44:07] Copy BitStream file to RAM ...
[18:44:08] Convert file data to ByteStream ...
[18:44:09] Write remaining file: chacha20.bin_rest
[18:53:13] 7200000 elements copied to RAM.
[18:53:13] The file was read.
[18:53:13] Test procedure T6a for verification of P2.i) (vii.a) started.
[18:53:13] | P (1) - 0.5 | = 2.2999999999995246E-4
[18:53:13] Last element: 100000
[18:53:13] Test procedure T6a passed.

[18:53:13] Test procedure T6b for verification of P2.i) (vii.b) started.

[18:53:13] p (01) = 0.49882

[18:53:13] p (11) = 0.50245

[18:53:13] | p_ (01) - p_ (11) | = 0.0036299999999999666

[18:53:13] Last element: 500396

[18:53:13] Test procedure T6b passed.

[18:53:13] Test procedure T7a for verification of P2.i) (vii.c) started.

[18:53:13] Test size [0] = 0.6055202799925774

[18:53:13] Test size [1] = 3.362037956063709

[18:53:13] Last element: 1706360

[18:53:13] Test procedure T7a passed.

[18:53:13] Test T8 to verify P2.i) (vii.e) started.

[18:53:14] Test size = 7.999502481982367

[18:53:14] Last element: 6994840

[18:53:14] Test T8 passed.

[18:53:14] Run finished successfully.

[18:53:13] Test procedure T7b for verification of P2.i) (vii.d) started.

[18:53:13] Test size [0] = 0.0028800007787522105

[18:53:13] Test size [1] = 0.8323235320481406

[18:53:13] Test size [2] = 0.05832000149299203

[18:53:13] Test size [3] = 1.1045001512060706

[18:53:13] Last element: 4926360

[18:53:13] Test procedure T7b passed.

Efficiency

- Energy Consumption

(1) 使用pyRAPL裡的measureit，可以指定某段程式碼要執行、測量幾次。

(2) 測量時，把所有其他程式關掉，以免影響結果。

(3) 一次執行包含加密和解密，各執行、測量100次。

AES部分結果截圖

1	label	timestamp	duration	pkg	dram	socket
2	AES256_once	1622991118	989.46951	12862.51	625.6	0
3	AES256_once	1622991118	965.35615	12302.83	461.43	0
4	AES256_once	1622991118	940.7191	12787.44	551.76	0
5	AES256_once	1622991118	883.43556	10488.26	324.09	0
6	AES256_once	1622991118	844.65902	9647.19	280.76	0
7	AES256_once	1622991118	821.18122	9354.83	242.31	0
8	AES256_once	1622991119	851.76651	9520.24	253.3	0
9	AES256_once	1622991119	778.18349	9583.11	225.83	0
10	AES256_once	1622991119	842.52266	9697.85	246.58	0
11	AES256_once	1622991119	776.33543	9372.54	221.55	0
12	AES256_once	1622991119	824.16641	9724.09	239.87	0
13	AES256_once	1622991119	805.86312	9510.48	240.48	0
14	AES256_once	1622991119	831.6076	10435.76	529.17	0
15	AES256_once	1622991119	838.86173	10289.89	355.84	0
16	AES256_once	1622991119	849.03332	9577.61	249.63	0

平均：

duration	pkg	dram
811.4415904	10687.72	253.6797

Duration：執行一次經過的時間 單位：μs

PKG : CPU energy consumption 單位：μJ

DRAM : RAM energy consumption 單位：μJ

Salsa20部分結果截圖

	label	timestamp	duration	pkg	dram	socket
2	salsa_once	1.62E+09	46.51984	607.91	26.25	0
3	salsa_once	1.62E+09	35.66452	473.63	27.46	0
4	salsa_once	1.62E+09	29.4149	372.92	21.97	0
5	salsa_once	1.62E+09	28.54736	408.94	13.43	0
6	salsa_once	1.62E+09	28.52664	408.32	11.6	0
7	salsa_once	1.62E+09	33.74674	529.17	29.91	0
8	salsa_once	1.62E+09	37.36614	479.13	28.07	0
9	salsa_once	1.62E+09	38.34652	416.87	21.98	0
10	salsa_once	1.62E+09	29.82987	377.81	20.75	0
11	salsa_once	1.62E+09	42.61142	465.7	23.19	0
12	salsa_once	1.62E+09	37.52747	526.73	20.76	0
13	salsa_once	1.62E+09	36.3921	517.58	20.14	0
14	salsa_once	1.62E+09	44.13968	599.36	52.49	0
15	salsa_once	1.62E+09	34.05505	417.48	26.24	0
16	salsa_once	1.62E+09	27.59431	412.6	18.31	0

平均：

duration	pkg	dram
33.40873	407.0973	15.6251

Duration：執行一次經過的時間 單位：μs

PKG : CPU energy consumption 單位：μJ

DRAM : RAM energy consumption 單位：μJ

ChaCha20部分結果截圖

	A	B	C	D	E	F
1	label	timestamp	duration	pkg	dram	socket
2	chacha_once	1622986486	37.31137	623.16	36.01	0
3	chacha_once	1622986486	38.15675	558.47	29.29	0
4	chacha_once	1622986486	33.37269	378.42	17.7	0
5	chacha_once	1622986486	43.62725	484.62	20.75	0
6	chacha_once	1622986486	39.40691	532.23	33.57	0
7	chacha_once	1622986486	33.43253	494.99	24.41	0
8	chacha_once	1622986486	39.27992	548.71	28.08	0
9	chacha_once	1622986486	38.93365	519.4	31.74	0
10	chacha_once	1622986486	44.98399	617.68	32.35	0
11	chacha_once	1622986486	40.81101	546.26	22.58	0
12	chacha_once	1622986486	42.54895	515.14	35.4	0
13	chacha_once	1622986486	50.47185	657.34	53.1	0
14	chacha_once	1622986486	36.83879	582.28	34.79	0
15	chacha_once	1622986486	31.70293	411.98	20.75	0
16	chacha_once	1622986486	36.26632	509.03	18.31	0

平均：

duration	pkg	dram
33.57176	424.3978	13.0768

Duration：執行一次經過的時間 單位：μs

PKG : CPU energy consumption 單位：μJ

DRAM : RAM energy consumption 單位：μJ

CONCLUSION

應用—AES

- 無線網路架構：
  - Wi-Fi Protected Access (WPA)、WPA2。
  - ZigBee。
- 電子商務層面：
  - Secure Sockets Layer (SSL)安全通訊端層
  - Transport Layer Security(TLS)傳輸層安全性
- 軟硬體的實現：
  - 語音、影像
  - IC卡(Smart Card)

應用—SALSA20、CHACHA20

- 因爲在路由器等性能不強的設備上使用 aes 加密方式會影響性能，使用rc4-md5又加密強度不夠，所以人們創造了 Salsa20 這個加密算法，它比前輩rc加密算法速度更快而加密強度更高，後來，Google 又在這個算法的基礎上開發了 chacha20 這個更快加密更強的算法。
- 基本上，它現在算是性能不強的設備使用最佳的算法了。

參考資料

- https://zh.wikipedia.org/wiki/Salsa20
- https://baike.baidu.com/item/chacha20-poly1305
- https://bashtage.github.io/randomgen/devel/bit_generators/generated/randomgen.chacha.ChaCha.random_raw.html#randomgen.chacha.ChaCha.random_raw
- https://pycryptodome.readthedocs.io/en/latest/src/cipher/chacha20.html
- https://pypi.org/project/pyRAPL/
- https://searchsecurity.techtarget.com/definition/Advanced-Encryption-Standard
- https://cms.aaasec.com.tw/index.php/2019/06/04/b_05/
- https://www.kryptall.com/index.php/2015-09-24-06-28-54/how-safe-is-safe-is-aes-encryption-safe
- https://cms.aaasec.com.tw/index.php/2019/06/04/b_05/
- https://www.itread01.com/content/1539952689.html?fbclid=IwAR0IgYBOtjtyVgsf7cxqjKmdGEZ1knoke69xAyrVvlFJnmqKMXlnrVuEHUI
- https://www.csoonline.com/article/3388647/what-is-a-side-channel-attack-how-these-end-runs-around-encryption-put-everyone-at-risk.html

THANK YOU~