Submission 3: Cryptographic Solutions

1) Vignère Cipher

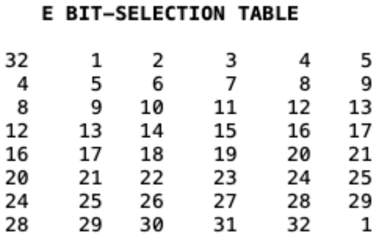
Message	D	O	N	A	L	D	T	R	U	M	P
Key	H	F	T	R	O	C	K	S	H	F	T
+	10	19	6	17	25	5	3	9	1	17	8
Cipher Text	K	T	G	R	Z	F	D	J	B	R	I

Solution: **KTGRZFDJBRI**
All Numbers higher than 26 are calculated with the following formula: $(a + b)\,\%\,length$

2) DES encryption

To calculate f, we first expand each block R_{i-1} from 32-bits to 48-bits. This is done by using a selection table the function E. Thus $E(R_{i-1})$ has 32-bit input block, and a 48-bit output block.

Let E be such that the 48 bits of its output, written as 8 blocks of 6 bits each, are obtained by selecting the bits in its inputs in order according to the following table:



E BIT-SELECTION TABLE

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

We calculate $E(R_{i-1})$ from R_{i-1} as follows:

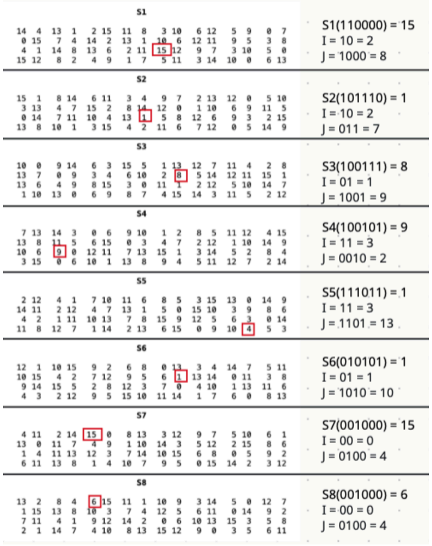
$$R_{i-1} = 10100111\ 00111011\ 11111111\ 00000001$$
$$E(R_{i-1}) = 110100\ 001110\ 100111\ 110111\ 111111\ 111110\ 100000\ 000011$$

(Note that each block of 4 original bits has been expanded to a block of 6 output bits.)

Next in the function f calculation, we XOR the output $E(R_{i-1})$ with the Key K_i:

$$K_i = 000100\ 100000\ 110100\ 010010\ 000100\ 101011\ 101000\ 001011$$
$$E(R_{i-1}) = 110100\ 001110\ 100111\ 110111\ 111111\ 111110\ 100000\ 000011$$
$$K_i + E(R_{i-1}) = 110000\ 101110\ 010011\ 100101\ 111011\ 010101\ 001000\ 001000$$

Each group of six bits will give us an address I a different S-Box. Located at that address will be a 4 bit number. This 4 bit number will replace the original 6 bits. The net result is that the eight groups of 6 bits are transformed ino eight groups of 4 bits for 32-bits total.



S1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S1(110000) = 15
I = 10 = 2
J = 1000 = 8

S2

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S2(101110) = 1
I = 10 = 2
J = 011 = 7

S3

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S3(100111) = 8
I = 01 = 1
J = 1001 = 9

S4

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S4(100101) = 9
I = 11 = 3
J = 0010 = 2

S5

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S5(111011) = 1
I = 11 = 3
J = 1101 = 13

S6

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S6(010101) = 1
I = 01 = 1
J = 1010 = 10

S7

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S7(001000) = 15
I = 00 = 0
J = 0100 = 4

S8

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

S8(001000) = 6
I = 00 = 0
J = 0100 = 4

From:

$$K_i + E(R_{i-1}) = B_1B_2B_3B_4B_5B_6B_7B_8$$

To:

$$S_1(B_1)S_2(B_2)S_3(B_3)S_4(B_4)S_5(B_5)S_6(B_6)S_7(B_7)S_8(B_8)$$

As seen in figure m. Result:

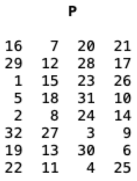
$$S_1(B_1)S_2\ldots = 1111\,0001\,1000\,1001\,0001\,0001\,1111\,0110$$

The final stage in the calculation of f is to do a permuation P of the S-Box output to optain the final value of f:

$$f = P(S_1(B_1)\ldots S_8(B_8))$$

The permutation P is defined in the following table. P yields a 32-bit output from a 32-bit input by permuting the bits of the input block.

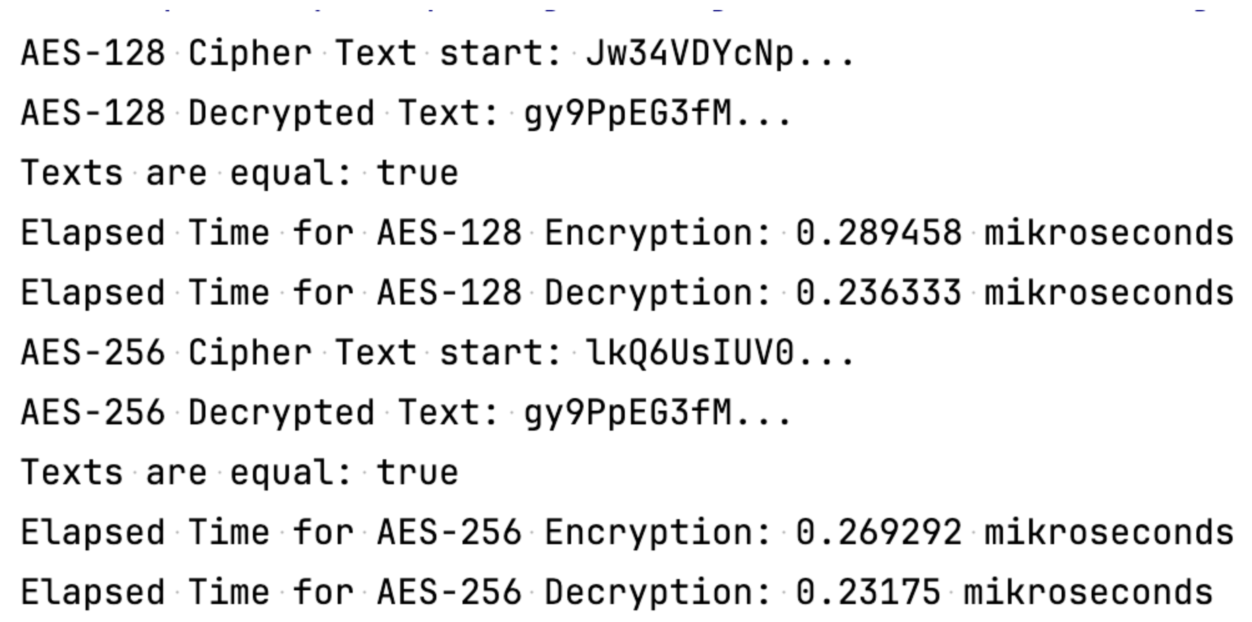
$$f = 1010\,0010\,1001\,0010\,1110\,0111\,0110\,0010$$



P

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

3) AES Performance in Java



```
AES-128 Cipher Text start: Jw34VDYcNp...
AES-128 Decrypted Text: gy9PpEG3fM...
Texts are equal: true
Elapsed Time for AES-128 Encryption: 0.289458 mikroseconds
Elapsed Time for AES-128 Decryption: 0.236333 mikroseconds
AES-256 Cipher Text start: lkQ6UsIUV0...
AES-256 Decrypted Text: gy9PpEG3fM...
Texts are equal: true
Elapsed Time for AES-256 Encryption: 0.269292 mikroseconds
Elapsed Time for AES-256 Decryption: 0.23175 mikroseconds
```

Setup

The encryption and decryption assessments were executed on a MacBook Pro, featuring a robust 10-core ARM CPU. For these tasks, a randomly generated string comprising 1,000 characters, including the 26 alphabetic letters and digits from 0 to 9, was utilized. The same input string was employed to compare the performance of AES-128 and AES-256 encryption algorithms. To verify the accuracy and reliability of the encryption processes, it was confirmed that the decrypted text exactly matched the original input, ensuring the proper functioning of both encryption and decryption mechanisms.

Analysis

The performance evaluation consistently shows that the encryption times for AES-128 and AES-256 are very similar, with only minor differences observed across multiple tests. This indicates that the enhanced security offered by the longer key length in AES-256 has a negligible effect on encryption speed. Conversely, the decryption performance reveals a significant difference, with AES-256 decryption being notably slower than AES-128. This pronounced disparity in decryption times implies that while AES-256 provides greater security, it imposes a performance cost during decryption, which is a crucial factor to consider in scenarios where decryption speed is a priority.

Source Code

https://github.com/tim-herbst/it-sec2-submissions