SaraBaradaran **HomeWork3**
9624193

Question 1)
Weak key: A DES key k is called a weak key if encryption and decryption are same.

$$Ek(Ek(x))=x$$

1.1

In this case encryption = decryption $\rightarrow$

$K_1 = K_{16}$ $K_2 = K_{15}$ …. ($K_i = K_{17-I}$) (for all i = 1, 2 … 16)

$K_1 = K_2 = \ldots = K_{16}$ (all 16 subkeys are identical)

1.2

There are four 64-bit weak keys:

$$\left.\begin{array}{l}\text{0x 0101010101010101}\\ \text{0x FEFEFEFEFEFEFEFE}\\ \text{0x E0E0E0E0F1F1F1F1}\\ \text{0x 1F1F1F1F0E0E0E0E}\end{array}\right\}\ \textbf{Weak keys}$$

1.3

2^{56} possible keys are available and 4 keys are weak

$\rightarrow$ Probability of selecting a weak key $\rightarrow$ $2^2/\,2^{56} = 2^{-54}$

1.4

When the security is important, we should not use this weak keys.

Semi-weak key: They have the property that

$$E_{k1}\,(E_{k2}\,(x))=x$$

1.5

There are 6 pairs of semi-weak key:

$$\left.\begin{array}{l}
\text{0x011F011F010E010E , 0x1F011F010E010E01}\\[1em]
\text{0x01E001E001F101F1 , 0xE001E001F101F101}\\[1em]
\text{0x01FE01FE01FE01FE , 0xFE01FE01FE01FE01}\\[1em]
\text{0x1FE01FE00EF10EF1 , 0xE01FE01FF10EF10E}\\[1em]
\text{0x1FFE1FFE0EFE0EFE , 0xFE1FFE1FFE0EFE0E}\\[1em]
\text{0xE0FEE0FEF1FEF1FE , 0xFEE0FEE0FEF1FEF1}
\end{array}\right\}\ \textbf{Semi-weak keys}$$

1.6

DES has semi-weak keys, which only produce two different subkeys, each used **eight times** in the algorithm.

1.7

There are 2^{56} possible keys for DES, of which only 4 are weak and 12 are semi-weak.

Probability of selecting a weak or semi-weak key $\rightarrow$ $(12 + 4) / 2^{56} = 2^{4} / 2^{56} = 2^{-52}$

$\rightarrow$ The possibility of this happening is very small

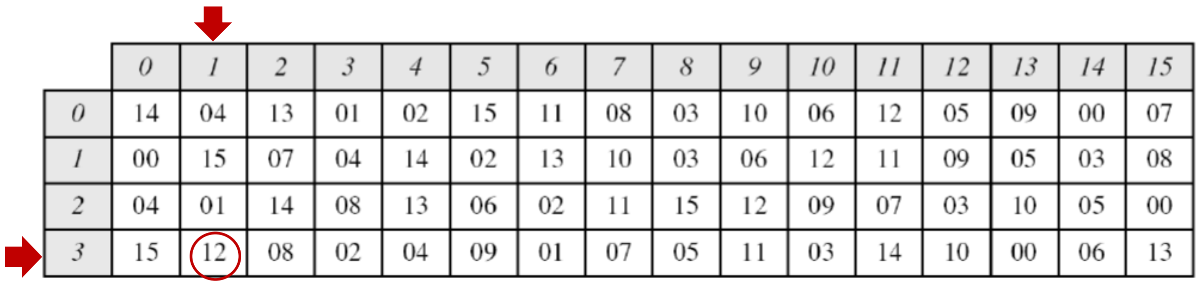
Question 2)

2.1

One important property which makes DES secure is that the S-boxes are nonlinear.

2.3

6 bit input: 1 0 0 0 1 1 → row = $(1\ 1)_2 = 3$, column = $(0\ 0\ 0\ 1)_2 = 1$ → 4 bit output : 1 1 0 0



	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	10	03	06	12	11	09	05	03	08
2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

2.2

$S(x1) \oplus Si(x2) = Si(101010) \oplus Si(010101) = 0110 \oplus 1100 = 1010 = (10)$

$S(101010)$ → row = 10 , column = 0101 → row = 2 , column = 5

→ 4bit output = (06) = 0110

$S(010101)$ → row = 01 , column = 1010 → row = 1 , column = 10

→ 4bit output = (12) = 1100

$S(x1 \oplus x2) = Si(101010 \oplus 010101) = Si(111111) = 1101 = (13)$

$S(111111)$ → row = 11 , column = 1111 → row = 3 , column = 15

→ 4bit output = (13) = 1101

Question 3)

3.2

<u>Key:</u>

0000000000000000000000000000_0000000000000000000000000000

- ➔ $K_1 = K_2 = \ldots = K_{16} = 0$

64 bit input:

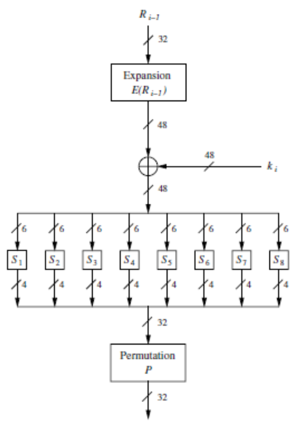
00000000000000000000000000000000_00000000000000000000000000000000

After initial permutation:

L0 R0

00000000000000000000000000000000_00000000000000000000000000000000

$Li = Ri-1,$
$Ri = Li-1 \oplus f(Ri-1,ki)$



$R_0 = 0$, $K0 = 0$ → 48 bit output after expantion:

000000_000000_000000_000000_000000_000000_000000_000000

S_boxes output:

$\mathbf{S_1 = 14}$, $S_2 = 15$, $S_3 = 10$, $S_4 = 07$, $S_5 = 2$, $S_6 = 12$, $S_7 = 04$, $\mathbf{S_8 = 13}$

1110_1111_1010_0111_0010_1100_0100_1101

After permutation p:

1101_1000_1101_1000_1101_1011_1011_1100 $\oplus L_0$ ($L_0 = 0$) =

1101_1000_1101_1000_1101_1011_1011_1100 = R1

0000_0000_0000_0000_0000_0000_0000_0000 = L1

Output after first round:

0000_0000_0000_0000_0000_0000_0000_0000_1101_1000_1101_1000_1101_1011_1011_1100

64 bit input:

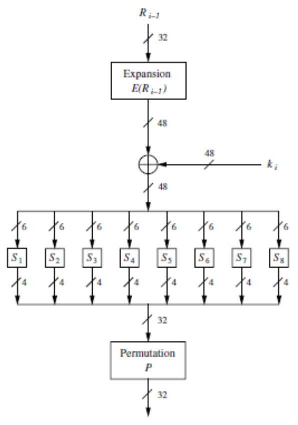
00000000000000000000000000000000_00000000000000000000000010000000

After initial permutation:

L0 R0

00000000000000000000000000000000_10000000000000000000000000000000

$Li = Ri-1,$
$Ri = Li-1 \oplus f(Ri-1,ki)$



R_0 = 0x80000000 , K_0 = 0 → 48 bit output after expantion:

010000_000000_000000_000000_000000_000000_000000_000001

S_boxes output:

S_1 = 03 , S_2 = 15 , S_3 = 10 , S_4 = 07 , S_5 = 2 , S_6 = 12 , S_7 = 04 , **S_8 = 01**

0011_1111_1010_0111_0010_1100_0100_0001

After permutation p:

1101_0000_0101_1000_0101_1011_1001_1110 $\oplus$ L_0 (L_0 = 0)

1101_0000_0101_1000_0101_1011_1001_1110 = R1

1000_0000_0000_0000_0000_0000_0000_0000 = L1

Output after first round:

1000_0000_0000_0000_0000_0000_0000_0000_1101_0000_0101_1000_0101_1011_1001_1110

3.1
In the first round 2 S-box are different: these S-boxes are S_1, S_8

3.3

The minimum number of output bits that will be changed (per S-box) as a result of a One bit change in input is Two.

3.4

We can xor two outputs to show difference between them:

0000_0000_0000_0000_0000_0000_0000_0000_1101_1000_1101_1000_1101_1011_1011_1100

1000_0000_0000_0000_0000_0000_0000_0000_1101_0000_0101_1000_0101_1011_1001_1110

1000_0000_0000_0000_0000_0000_0000_0000_0000_1000_1000_0000_1000_0000_0010_0010

Result: One bit of difference in inputs leads to six bits of difference in outputs .

Question 4)

*	0 (000)	1 (001)	X (010)	$X+1$ (011)	X^2 (100)	X^2+1 (101)	X^2+X (110)	X^2+X+1 (111)
0 (000)	0 (000)	0 (000)	0 (000)	0 (000)	0 (000)	0 (000)	0 (000)	0 (000)
1 (001)	0 (000)	1 (001)	X (010)	$X+1$ (011)	X^2 (100)	X^2+1 (101)	X^2+X (110)	X^2+X+1 (111)
X (010)	0 (000)	X (010)	X^2 (100)	X^2+X (110)	$X+1$ (011)	1 (001)	X^2+X+1 (111)	X^2+1 (101)
$X+1$ (011)	0 (000)	$X+1$ (011)	X^2+X (110)	X^2+1 (101)	X^2+X+1 (111)	X^2 (100)	1 (001)	X (010)
X^2 (100)	0 (000)	X^2 (100)	$X+1$ (011)	X^2+X+1 (111)	X^2+X (110)	X (010)	X^2+1 (101)	1 (001)
X^2+1 (101)	0 (000)	X^2+1 (101)	1 (001)	X^2 (100)	X (010)	X^2+X+1 (111)	$X+1$ (011)	X^2+X (110)
X^2+X (110)	0 (000)	X^2+X (110)	X^2+X+1 (111)	1 (001)	X^2+1 (101)	$X+1$ (011)	X (010)	X^2 (100)
X^2+X+1 (111)	0 (000)	X^2+X+1 (111)	X^2+1 (101)	X (010)	1 (001)	X^2+X (110)	X^2 (100)	$X+1$ (011)

$P(x) = X^3 + X + 1 \rightarrow X^3 = X + 1 \rightarrow X(X^3) = X^4 = X^2 + X$

$X(X) = X^2$

$X(X + 1) = X^2 + X$

$X(X^2) = X^3 = X + 1$

$X(X^2 + 1) = X^3 + X = X + 1 + X = 1$

$X(X^2 + X) = X^3 + X^2 = X^2 + X + 1$

$X(X^2 + X + 1) = X^3 + X^2 + X = X + 1 + X^2 + X = X^2 + 1$

$(X + 1)(X + 1) = X^2 + X + X + 1 = X^2 + 1$

$(X + 1)(X^2) = X^3 + X^2 = X^2 + X + 1$

$(X + 1)(X^2 + 1) = X^3 + X + X^2 + 1 = X + 1 + X + X^2 + 1 = X^2$

$(X + 1)(X^2 + X) = X^3 + X^2 + X^2 + X = X + 1 + X = 1$

$(X + 1)(X^2 + X + 1) = X^3 + X^2 + X + X^2 + X + 1 = X + 1 + 1 = X$

$(X^2)(X^2) = X^4 = X^2 + X$

$(X^2)(X^2 + 1) = X^4 + X^2 = X^2 + X + X^2 = X$

$(X^2)(X^2 + X) = X^4 + X^3 = X^2 + X + X + 1 = X^2 + 1$

$(X^2)(X^2 + X + 1) = X^4 + X^3 + X^2 = X^2 + X + X^3 + X^2 = X + X + 1 = 1$

$(X^2 + 1)(X^2 + 1) = X^4 + X^2 + X^2 + 1 = X^2 + X + 1$

$(X^2 + 1)(X^2 + X) = X^4 + X^3 + X^2 + X = X^2 + X + X + 1 + X^2 + X = X + 1$

$(X^2 + 1)(X^2 + X + 1) = X^4 + X^3 + X^2 + X^2 + X + 1 = X^2 + X + X + 1 + X + 1 = X^2 + X$

$(X^2 + X)(X^2 + X) = X^4 + X^3 + X^3 + X^2 = X^2 + X + X^2 = X$

$(X^2 + X)(X^2 + X + 1) = X^4 + X^3 + X^2 + X^3 + X^2 + X = X^2 + X + X = X^2$

$(X^2 + X + 1)(X^2 + X + 1) = X^4 + X^3 + X^2 + X^3 + X^2 + X + X^2 + X + 1 = X^2 + X + X^2 + 1 = X+1$

Question 5)

We have 2^4 polynomials with degree 4 over GF (2): (3 polynomials are irreducible)

$X^4 = (X)(X^3)$

$X^4 + 1 = (X + 1)(X^3 + X^2 + X + 1)$

$X^4 + X = X(X^3 + 1)$

$X^4 + X + 1$

$X^4 + X^2 = X(X^3 + X)$

$X^4 + X^2 + 1 = (X^2 + X + 1)(X^2 + X + 1)$

$X^4 + X^2 + X = X(X^3 + X + 1)$

$X^4 + X^2 + X + 1 = (X + 1)(X^3 + X^2 + 1)$

$X^4 + X^3 = X^3(X+1)$

$X^4 + X^3 + 1$

$X^4 + X^3 + X = X(X^3 + X^2 + 1)$

$X^4 + X^3 + X + 1 = (X + 1)(X^3 + 1)$

$X^4 + X^3 + X^2 = X(X^3 + X^2 + X)$

$X^4 + X^3 + X^2 + 1 = (X + 1)(X^3 + X + 1)$

$X^4 + X^3 + X^2 + X = X(X^3 + X^2 + X + 1)$

$X^4 + X^3 + X^2 + X + 1$

Question 6)

Brute-Force Attack

طول کلید برای رمز DES یک کلید ۵۶ بیتی و برای رمز AES کلید ۱۲۸و ۱۹۲ یا ۲۵۶ بیتی است پس در حمله Brute-force attack در رمز DES نیازمند بررسی $۲^{۵۶}$ حالت ولی در رمز AES حداقل نیازمند بررسی $۲^{۱۲۸}$ حالت برای کلید هستیم. پس از نظر Brute-force attack رمز AES امن تر از DES می باشد. (با توجه به امکانات امروزه رمز DES از طریق این حمله قابل شکست است.)

Statistical Attacks

با توجه به اینکه عمل diffusion در هر دو نوع رمز موجود است (bit permutation در رمز DES و mix column در رمز AES) پس می توان گفت هر دو در برابر Statical attack ها امن هستند.

Differential and linear Attacks

با توجه به اینکه حمله linear Attack نیازمند حداقل $۲^{۴۳}$ و حمله Differential and نیازمند حداقل $۲^{۴۷}$ جفت plaintext و ciphertext با ویژگی های معینی می باشد لذا در عمل می توان گفت هردو نوع رمز در برابر این حملات امن است. چرا که در عمل شرایط لازم برای این حملات با احتمال بسیار پایینی ممکن است برقرار شود.

Question 7)

A stream cipher is a symmetric key cipher (method of encryption) where plaintext digits are combined with a pseudorandom cipher digit stream. A stream cipher encrypts plaintext with a key and algorithm applied to every binary digit (one and zeros) for every bit in the data stream. The pseudorandom cipher digits are generated through a number of random seed values that use digit shift registers.

A block cipher is an encryption method that applies a deterministic algorithm along with a symmetric key to encrypt a block of text, rather than encrypting one bit at a time as in stream ciphers. In block cipher, text is divided in relatively large blocks, typically 64 or 128 bytes long and that each block is encoded separately. Plaintext is used during the encryption and the resulting encrypted text is referred to as a cipher text.

Let us discuss some of the major key differences between Stream Cipher vs Block Cipher:

1. In stream cipher, the encryption is done bit by bit whereas, in block cipher, it is done block by block.
2. In stream cipher, the decryption is also done by bit by bit whereas in block cipher it is done by block by block.
3. Stream cipher relies on substitution techniques like Caesar cipher, modified Caesar cipher, monoalphabetic cipher, homophonic cipher, polygram substitution cipher, polyalphabetic cipher, Playfair cipher, and hill cipher.
4. Block cipher relies on transposition techniques like rail-fence technique, columnar transposition technique, Vernam cipher, and book cipher.
5. Stream cipher uses confusion to ensure that it does not give clues about plain text whereas block cipher uses both confusion and diffusion.
6. A stream cipher is faster than block cipher whereas block cipher is slower.
7. In a stream cipher, one key is used for one time whereas in block cipher key can be reused.
8. In stream cipher we need a key with the length of plain text whereas in block ciphers we can use 64bit key (for DES) and 128 or 192 or 254bit key (for AES).
9. Stream cipher requires less code than block cipher.
10. Stream Cipher doesn't consist of a complex algorithm or process as a Block Ciphers.
11. It is simple to implement Stream cipher in Hardware than that of Block cipher.
12. Stream cipher uses XOR function for converting the plain text into cipher text that is the reason why it is easy to reverse the XORed bits. Whereas Block cipher does not use XOR for doing so.

13. Stream Ciphers does not require large memory because they only work on small bits at a time unlike block ciphers that require a relatively large memory because they work on a large chunk of data.
14. Redundancy is less in stream cipher whereas block cipher increases the redundancy.

15. A stream cipher is used for SSL secure connection for web whereas block cipher is used for database, file encryption.
16. Encryption can be implemented bit by bit in stream ciphers and instantly when new data is available for processing, so an incoming bit will automatically generate an outgoing bit without buffering the input. On the other hand, block ciphers require a complete data block by applying a padding scheme to be collected before the first output bit can be generated.
17. Stream ciphers do not provide integrity protection or authentication. On the contrary, some block ciphers (depending on the mode) can provide integrity protection, in addition to confidentiality.

CRYPTOOL :

Encryption/Decryption

1

i. Key : 00 00 00 00 00 00 00 00
Output file → Q7_1_i_firstEncrypthionWithWeakKey.hex
Output file → Q7_1_i_secondEncrypthionWithWeakKey.hex

ii. Key 1 : 01 1F 01 1F 01 0E 01 0E
Output file → Q7_1_ii_firstEncryptionWithSemiWeakKey.hex
Key 2 : 1F 01 1F 01 0E 01 0E 01
Output file → Q7_1_ii_secondEncryptionWithSemiWeakKey.hex

2

Triple DES:

Keying option 1

All three keys are independent. Sometimes known as 3TDEA or triple-length keys.
This is the strongest, with 3 × 56 = 168 independent key bits. It is still vulnerable to meet-in-the-middle attack, but the attack requires $2^{2 \times 56}$ steps.

Keying option 2

K_1 and K_2 are independent, and $K_3 = K_1$. Sometimes known as 2TDEA or double-length keys.
This provides a shorter key length of 112 bits and a reasonable compromise between DES and Keying option 1, with the same caveat as above. This is an improvement over "double DES" which only requires 2^{56} steps to attack.

Keying option 3

All three keys are identical, i.e. $K_1 = K_2 = K_3$.

The original DES cipher's key size of 56 bits was generally sufficient when that algorithm was designed, but the availability of increasing computational power made brute-force attacks feasible. Triple DES provides a relatively simple method of increasing the key size of DES to protect against such attacks, without the need to design a completely new block cipher algorithm.

In general, Triple DES with three independent keys (keying option 1) has a key length of 168 bits (three 56-bit DES keys), but due to the meet-in-the-middle attack, the effective security it provides is only 112 bits. Keying option 2 reduces the effective key size to 112 bits (because the third key is the same as the first) .However, this option is susceptible to certain chosen-plaintext or known-plaintext attacks, and thus it is designated by NIST to have only 80 bits of security.

i. Triple DES is more secure because it's effective key length is 112 bit (we know effective key length of DES is only 56 bit).

ii. In brute-force attack the keyspace is 2^{112}.
Because the effective key length is 112 so we need search among 2^{112} key

iii. DES vs TripleDES
Key length of DES = 56 but key length of Triple DES = 168 (in option 1) or 112 (in option 2)
Effective key length of DES = 56 but Effective key length of TripleDES = 112
TripleDES is more secure than DES
The number of round in DES = 16 but the number of round in TripleDES = 48

iv. Key : 0x 11 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Output file → Q7_2_iv_TripleDESEncryption.hex

v. Key1 : 0x 11 00 00 00 00 00 00 00
Output file → Q7_2_v_DESEncryption_first.hex
key2 : 0x 00 00 00 00 00 00 00 00
Output file → Q7_2_v_DESEncryption_second.hex
Key3 : 0x 11 00 00 00 00 00 00 00
Output file → Q7_2_v_DESEncryption_third.hex

vi. Key : 0x 11 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Output → Q7_2_vi_TripleDESDecryption.hex

3

i. Key :0x 11 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Output file → Q7_3_i_DESXEncryption.hex

ii. First 8 byte of key is 11 00 00 00 00 00 00 00

4

Key : 0x 11 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Output → Q7_4_AESEncryption.hex

Analysis

5

i.

$1.8 * 10^{25}$ years →

$1.8 * 10^{25} * 365$ day →

$1.8 * 10^{25} * 365 * 24$ hours →

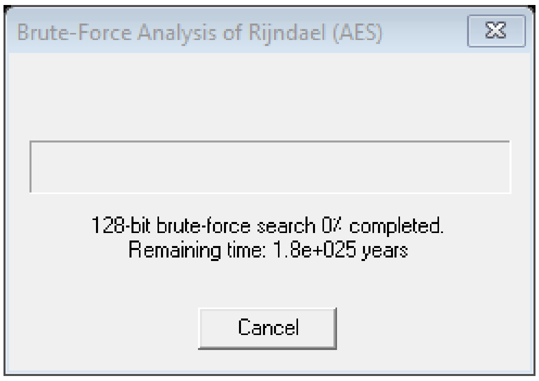
$1.8 * 10^{25} * 365 * 24 * 60$ minutes →

$1.8 * 10^{25} * 365 * 24 * 60 * 60$ seconds →

$1.8 * 10^{25} * 365 * 24 * 60 * 60 * 10^{6}$ microseconds

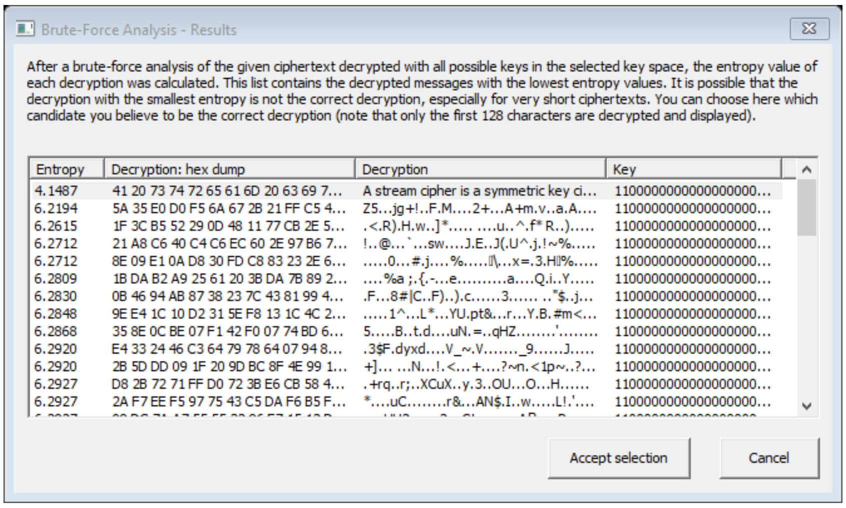
There are 2^{128} keys → $567648 * 10^{33} / 2^{128}$

~ 1.5 microseconds for each key



128-bit brute-force search 0% completed.
Remaining time: 1.8e+025 years

ii. آنتروپی میزان عدم اعتماد به یک متغیر تصادفی می باشد به عبارتی در میان کلید های بدست آمده کلیدی احتمالا درست تر است که آنتروپی مربوط به آن کمتر باشد.



After a brute-force analysis of the given ciphertext decrypted with all possible keys in the selected key space, the entropy value of each decryption was calculated. This list contains the decrypted messages with the lowest entropy values. It is possible that the decryption with the smallest entropy is not the correct decryption, especially for very short ciphertexts. You can choose here which candidate you believe to be the correct decryption (note that only the first 128 characters are decrypted and displayed).

Entropy	Decryption: hex dump	Decryption	Key
4.1487	41 20 73 74 72 65 61 6D 20 63 69 7...	A stream cipher is a symmetric key ci...	1100000000000000000...
6.2194	5A 35 E0 D0 F5 6A 67 2B 21 FF C5 4...	Z5...jg+!..F.M....2+...A+m.v..a.A....	1100000000000000000...
6.2615	1F 3C B5 52 29 0D 48 11 77 CB 2E 5...	.<.R).H.w..]*..... ....u..^.f* R..).....	1100000000000000000...
6.2712	21 A8 C6 40 C4 C6 EC 60 2E 97 B6 7...	!..@...`...sw....J.E..J(.U^.j.!~%.....	1100000000000000000...
6.2712	8E 09 E1 0A D8 30 FD C8 83 23 2E 6...	.....0...#.j....%.....\...x=.3.H%.....	1100000000000000000...
6.2809	1B DA B2 A9 25 61 20 3B DA 7B 89 2...	....%a ;.{.-...e.........a....Q.i..Y.....	1100000000000000000...
6.2830	0B 46 94 AB 87 38 23 7C 43 81 99 4...	.F...8#|C..F)..).c......3...... .."$..j...	1100000000000000000...
6.2848	9E E4 1C 10 D2 31 5E F8 13 1C 4C 2...	.....1^...L*...YU.pt&...r...Y.B.#m<...	1100000000000000000...
6.2868	35 8E 0C BE 07 F1 42 F0 07 74 BD 6...	5.....B..t.d....uN.=..qHZ........'........	1100000000000000000...
6.2920	E4 33 24 46 C3 64 79 78 64 07 94 8...	.3$F.dyxd....V_~.V......._9......J.....	1100000000000000000...
6.2920	2B 5D DD 09 1F 20 9D BC 8F 4E 99 1...	+]... ...N...!.<...+....?~n.<1p~..?...	1100000000000000000...
6.2927	D8 2B 72 71 FF D0 72 3B E6 CB 58 4...	.+rq..r;..XCuX..y.3..OU...O...H......	1100000000000000000...
6.2927	2A F7 EE F5 97 75 43 C5 DA F6 B5 F...	*....uC........r&...AN$.I..w.....L!.'....	1100000000000000000...