Understanding Cryptography

Answers of Homework No.1

فصل اول

1.7.

حل :

برای حل این مساله Z_4 را در نظر گرفته و جدولی می سازیم که جمع همه المان ها را در حلقه نشان دهد.

+	0	1	2	3
0	0	1	2	3
1	1	2	3	4
2	2	3	4	5
3	3	4	5	6

۱. ساخت جدول ضرب برای Z_4

×	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

۲. ساخت جدول جمع و ضرب برای Z_5

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3
×	0	1	2	3	4

0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

۳. ساخت جدول جمع و ضرب برای Z_6

+	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	2	3	4	5	0
2	2	3	4	5	0	1
3	3	4	5	0	1	2
4	4	5	0	1	2	3
5	5	0	1	2	3	4

×	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	1

۴. در Z_4 و Z_6 المان‌هایی وجود دارند که معکوس ضربی ندارند این المان ها کدام هستند؟ و چرا یک معکوس ضربی برای همه ی المان های غیر صفر در Z_5 وجود دارد؟

حل:

تعریف معکوس ضربی:

$$a * a^{-1} = a^{-1} * a = e$$

$a^{-1} \times a \equiv 1 \ mod\ m$: a has multiplicative inverse such a^{-1} if $\gcd(a, m) = 1$

یا به عبارتی دیگر در Z_m هر عدد a که $\gcd(a, m) \neq 1$ باشد معکوس ضربی ندارد.

$Z_4: a = 0, 1, 2, 3$

$\{1, 2, 3\} \rightarrow \emptyset(n) = 2$

منظور از اعداد نشان داده شده با رنگ قرمز اعداد اول نسبت به ۴ است. برای عدد صفر هم که معکوس تعریف نمی شود.

$$a^{-1} = 1^{2-1} = 1 \ mod\ 4 = 1 \quad \rightarrow 1 * 1 = 1 \ mod\ 4 = 1 \ \sqrt{}$$

$$a^{-1} = 2^{2-1} = 2 \, mod \, 4 = 2 \quad \rightarrow \quad 2 * 2 \, mod \, 4 = 0 \quad \times$$
$$a^{-1} = 3^{2-1} = 3 \, mod \, 4 = 3 \quad \rightarrow \quad 3 * 3 \, mod \, 4 = 1 \, \surd$$

پس نتیجه می‌گیریم که در Z_4 المانهای ۲ معکوس ضربی ندارد و المانهای ۱ و ۳ معکوس ضربی دارند.

Z_6: $a = 0, 1, 2, 3,4,5$

$\{1, 2, 3,4, 5\} \rightarrow \emptyset(n) = 2$

منظور از اعداد نشان داده شده با رنگ قرمز اعداد اول نسبت به ۶ است.

$$a^{-1} = 1^{-1} = 1^{2-1} = 1 \, mod \, 6 = 1 \quad \rightarrow 1 * 1 = 1 \, mod \, 6 = 1 \, \surd$$
$$a^{-1} = 2^{-1} = 2^{2-1} = 2 \, mod \, 6 = 2 \quad \rightarrow 2 * 2 \, mod \, 6 = 4 \quad \times$$
$$a^{-1} = 3^{-1} = 3^{2-1} = 3 \, mod \, 6 = 3 \quad \rightarrow 3 * 3 \, mod \, 6 = 3 \quad \times$$
$$a^{-1} = 4^{-1} = 4^{2-1} = 4 \, mod \, 6 = 4 \quad \rightarrow 4 * 4 \, mod \, 6 = 4 \quad \times$$
$$a^{-1} = 5^{-1} = 5^{2-1} = 5 \, mod \, 6 = 5 \quad \rightarrow 5 * 5 \, mod \, 6 = 1 \, \surd$$

پس نتیجه می‌گیریم که در Z_6 المان های (2, 3,4) معکوس ضربی ندارند و المان(1,5) معکوس ضربی دارند. برای عدد صفر هم که معکوس تعریف نمی شود.

Z_5:

$a = 0, 1, 2, 3,4$

$\{1, 2, 3, 4\} \rightarrow \emptyset(n) = 4$

منظور از اعداد نشان داده شده با رنگ قرمز اعداد اول نسبت به ۵ است. برای عدد صفر هم که معکوس تعریف نمی شود.

$$a^{-1} = 1^{-1} = 1^{4-1} = 1 \, mod \, 5 = 1 \quad \rightarrow 1 * 1 = 1 \, mod \, 5 = 1 \, \surd$$
$$a^{-1} = 2^{-1} = 2^{4-1} = 8 \, mod \, 5 = 3 \quad \rightarrow 3 * 2 \, mod \, 5 = 1 \, \surd$$
$$a^{-1} = 3^{-1} = 3^{4-1} = 27 \, mod \, 5 = 2 \quad \rightarrow 2 * 3 \, mod \, 5 = 1 \, \surd$$
$$a^{-1} = 4^{-1} = 4^{4-1} = 64 \, mod \, 5 = 4 \quad \rightarrow 4 * 4 \, mod \, 5 = 1 \, \surd$$

... ...

نتیجه می گیریم چون عدد ۵ نسبت به تمام اعضای غیرصفر موجود در حلقه اول است و همه ی المان های غیر صفر کوچکتر از ۵ نسبت به ۵ اول هستند پس معکوس ضربی وجود دارد.

1.8.

حل :

همانطور که می دانیم معکوس یک عدد integer در یک حلقه کاملا وابسته به آن حلقه است. اگر پیمانه تغییر کند معکوس هم تغییر می‌کند. یعنی معکوس یک المان به تنهایی معنایی ندارد و باید حتما پیمانه آن ذکر گردد.

$$5^{-1} \, mod \, 11 \equiv ?$$
$$5^{-1} \, mod \, 12 \equiv ?$$
$$5^{-1} \, mod \, 13 \equiv ?$$

طبق این نکته که $a^{-1} = a^{\emptyset(n)-1} \, mod \, n$

$$Z_n^* = \{ a \in Z_n \parallel \gcd(a,n) = 1 \}$$

$\emptyset(n)$ تابع فی اویلر برابر تعداد اعدادی از مجموعه ی $\{1,\dots,n\}$ است که نسبت به n اول هستند.

$$\emptyset(11) = 10 \qquad \{1,2,3,4,5,6,,7,8,9,10,11\}$$

منظور از اعداد قرمز رنگ اعداد کوچکتر از ۱۱ است که نسبت به ۱۱ اول هستند.

$$\emptyset(12) = 4 \qquad \{1,2,3,4,5,6,7,8,9,10,11,12\}$$

منظور از اعداد قرمز رنگ اعداد کوچکتر از ۱۲ است که نسبت به ۱۲ اول هستند.

$$\emptyset(13) = 12 \qquad \{1,2,3,4,5,6,,7,8,9,10,11,12,13\}$$

منظور از اعداد قرمز رنگ اعداد کوچکتر از ۱۳ است که نسبت به ۱۳ اول هستند.

1. $5^{-1} = 5^{10-1} = 5^9 \bmod 11 = 5^3 \times 5^3 \times 5^3 \bmod 11$
   $= 4 \times 4 \times 4 \bmod 11 = 64 \bmod 11 = 9 \bmod 11$
2. $5^{-1} = 5^{4-1} = 5^3 \bmod 12 = 125 \bmod 12 = 5 \bmod 12$
3. $5^{-1} = 5^{12-1} = 5^{11} \bmod 13 = 5^4 \times 5^4 \times 5^3 \bmod 13$
   $= 1 \times 1 \times 8 \bmod 13 = 8 \bmod 13$

1.9.

حل :

هدف یافتن مقدار x است.

1. $3^2 \bmod 13 = 9 \bmod 13$

2. $7^2 \bmod 13 = 49 \bmod 13 = 10 \bmod 13$

3. $3^{10} \bmod 13 = (3^2)^5 \bmod 13 = 9^5 \bmod 13 = 9^2 \times 9^2 \times 9^1 \bmod 13$
   $= 81 \times 81 \times 9 \bmod 13 = 3 \times 3 \times 9 \bmod 13 = 3^2 \times 9 \bmod 13$
   $= 81 \bmod 13 = 3 \bmod 13$

or

3. $3^{10} \bmod 13 = 3^9 \times 3 \bmod 13 = (3^3)^3 \times 3 \bmod 13 = 1^3 \times 3 \bmod 13$
   $= 3 \bmod 13$

4. $7^{100} \bmod 13 = (7^2)^{50} \bmod 100 = 49^{50} \bmod 13 = 10^{50} \bmod 13$
   $= (10^2)^{25} \bmod 13 = 100^{25} \bmod 13 = 9^{25} \bmod 13$
   $= (9^2)^{12} \times 9 \bmod 13 = 81^{12} \times 9 \bmod 13 = 3^{12} \times 9 \bmod 13$
   $= (3^3)^4 \times 9 \bmod 13 = 27^4 \times 9 \bmod 13 = 1^4 \times 9 \bmod 13$
   $= 9 \bmod 13$

5. $7^x = 11 \bmod 13 \rightarrow x = 5$

با روش سعی و خطا مقدار $x = 5$ می شود. این یک مساله لگاریتم گسسته است .

1.10.

حل :

$$m = 4 \quad \{1,2,3,4\} \quad \emptyset(4) = 2$$
$$\gcd(1,4) = 1$$
$$\gcd(3,4) = 1$$

$\emptyset(m)$ برابر است با تعداد اعدادی از مجموعه ی $\{1, \ldots, m\}$ که نسبت به m اول هستند.

$$m = 5 \qquad \{1,2,3,4,5\} \qquad \emptyset(5) = 4$$

$$\gcd(1,5) = 1$$
$$\gcd(2,5) = 1$$
$$\gcd(3,5) = 1$$
$$\gcd(4,5) = 1$$

$$m = 9 \qquad \{1,2,3,4,5,6,7,8,9\} \qquad \emptyset(9) = 6$$

$$\gcd(1,9) = 1$$
$$\gcd(2,9) = 1$$
$$\gcd(4,9) = 1$$
$$\gcd(5,9) = 1$$
$$\gcd(7,9) = 1$$
$$\gcd(8,9) = 1$$

$$m = 26 \qquad \{1,3,5,7,9,11,15,17,19,21,23,25\} \qquad \emptyset(26) = 12$$

1.13.

حل :

$$\forall\, x, y, a, b \in Z_{26}$$

در $Affine\ cipher$ داریم:

$$y = e_k(x) = ax + b \mod 26$$
$$x = d_k(y) = a^{-1}(y - b) \mod 26$$

$$k = (a, b)$$

$$= \begin{cases} y_1 - a\,x_1 = b \\ y_2 - a\,x_2 = b \end{cases} \rightarrow \begin{cases} -y_1 + a\,x_1 = -b \\ y_2 - a\,x_2 = b \end{cases}$$

$$a = (x_1 - x_2)^{-1}(y_1 - y_2) \mod m$$
$$b = (y_1 - (x_1 - x_2)^{-1}(y_1 - y_2)x_1) \mod m = y_1 - ax_1 \mod m$$

نکته : حتما باید معکوس $(x_1 - x_2)$ در پیمانه m و $\gcd((x_1 - x_2), m) = 1$ باشد.

فصل دوم

2.

حل:

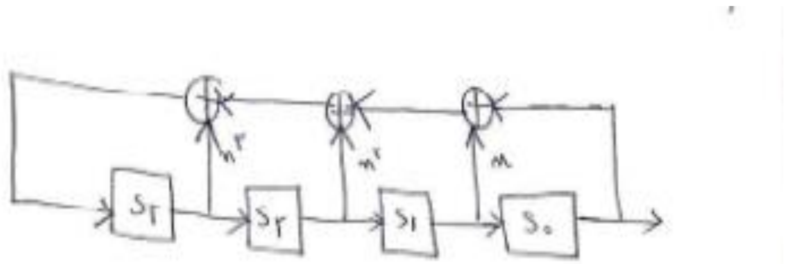
به طور کلی سه نوع LFSR موجود است . تفاوت آن ها بدین شرح است:

۱. LFSR هایی که یک دنباله واحد با طول حداکثر تولید می‌کنند این نوع $LFSR$ها به $LFSR$های مبتنی بر *primitive polynomials* معروف هستند.

۲. $LFSR$هایی که دنباله با طول حداکثر تولید نمی‌کنند و طول دنباله مستقل از مقدار اولیه رجیستر(ثبات) می باشد. این نوع $LFSR$ها به $LFSR$های مبتنی بر *irreducible polynomials* معروف هستند.

۳. $LFSR$هایی که یک دنباله با طول حداکثر تولید نمی کنند و طول آن‌ها وابسته به مقدار اولیه رجیستر است این نوع $LFSR$ها به $LFSR$های مبتنی بر *reducible polynomials* معروف هستند.

$$x^4 + x^3 + x^2 + x + 1$$



با فرض مقدار اولیه دلخواه :۰۰۰۱

مقدار اولیه	S3	S2	S1	S0	output
1	0	0	0	1	1
2	1	0	0	0	1
3	1	1	0	0	0
4	0	1	1	0	0
5	0	0	1	1	0
6	0	0	0	1	1

طول ۵

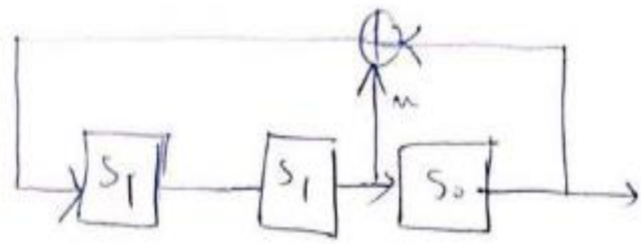
با فرض مقدار اولیه دلخواه :۱۱۱۱

مقدار اولیه	S3	S2	S1	S0	output
1	1	1	1	1	0
2	0	1	1	1	1
3	1	0	1	1	1
4	1	1	0	1	1
5	1	1	1	0	1
6	1	1	1	1	0

طول ۵

دیده می شود که ما در هر مورد یک دنباله مجزا با طول یکسان (۵) داریم. طول دنباله مستقل از مقدار اولیه رجیستر (ثبات) می‌باشد پس این چندجمله ای یک چندجمله‌ای از نوع irreducible است.

$$x^3 + \ x + 1$$

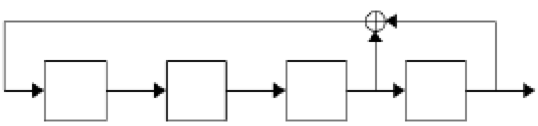


مقدار اولیه	S2	S1	S0	output
1	0	0	1	1
2	1	0	0	0
3	0	1	0	1
4	1	0	1	1
5	1	1	0	1
6	1	1	1	0
7	0	1	1	0
8	0	0	1	1

طول ۷

این نوع چندجمله ای از نوع primitive است. در اینجا یک دنباله با حداکثر طول حداکثر $2^3 - 1$ داریم.

$$x^4 + x + 1$$



با فرض مقدار اولیه دلخواه: 0001

مقدار اولیه	S3	S2	S1	S0	output
1	0	0	0	1	1
2	1	0	0	0	0
3	0	1	0	0	0
4	0	0	1	0	0
5	1	0	0	1	1
6	1	1	0	0	0
7	0	1	1	0	0
8	1	0	1	1	1
9	0	1	0	1	1
10	1	0	1	0	0
11	1	1	0	1	1
12	1	1	1	0	0
13	1	1	1	1	1
14	0	1	1	1	1
15	0	0	1	1	1
	0	0	0	1	1

این نوع چندجمله ای از نوع primitive است. در اینجا یک دنباله با حداکثر طول حداکثر $2^4 - 1$ داریم.

3.

حل :

1. The attacker needs 512 consecutive plaintext/ciphertext bit pairs x_i, y_i to launch a successful attack.
2. a. First, the attacker has to monitor the previously mentioned 512 bit pairs.
   b. The attacker calculates $s_i = x_i + y_i \bmod 2$, $i = 0, 1, \ldots, 2m-1$
   c. In order to calculate the (secret) feedback coefficients p_i, Oscar generates 256 linearly dependent equations using the relationship between the unknown key bits p_i and the keystream output defined by the equation

$$s_{i+m} \equiv \sum_{j=0}^{m-1} p_j \cdot s_{i+j} \bmod 2; s_i, p_j \in \{0,1\}; i = 0,1,2,\ldots,255$$

   with $m = 256$.
   d. After generating this linear equation system, it can be solved e.g. using Gaussian Elimination, revealing the 256 feedback coefficients.
3. The key of this system is represented by the 256 feedback coefficients. Since the initial contents of the LFSR are unalteredly shifted out of the LFSR and XORed with the first 256 plaintext bits, it would be easy to calculate them.

4.

حل:

4.1.

The full plaintext is WPIWOMBAT.

برنامه در فایل پیوست آمده است.

$$WPI \rightarrow \begin{cases} W \rightarrow 22 \rightarrow 10110_2 \\ P \rightarrow 15 \rightarrow 01111_2 \\ I \rightarrow 8 \rightarrow 01000_2 \end{cases} \rightarrow \ plaintext: 10110\ \ 01111\ \ 01000$$

$$J5A \rightarrow \begin{cases} J \rightarrow 9 \rightarrow 01001_2 \\ 5 \rightarrow 31 \rightarrow 11111_2 \\ A \rightarrow 0 \rightarrow 00000_2 \end{cases} \rightarrow \ ciphertext: \ 01001 \quad 11111 \quad 00000$$

4.2.

$$x_i \oplus \mathrm{y}_i = \mathrm{z}_i$$

$$\begin{aligned} xi &= 10110\ 01111\ 01000 \\ yi &= 01001\ 11111\ 00000 \\ zi &= 11111\ 10000\ 01000 \end{aligned}$$

$$keystream: \ S_0S_1S_2S_3S_4 \quad S_5S_6S_7S_8S_9 \quad S_{10}S_{11}S_{12}S_{13}S_{14} \ldots \ldots \ldots ..$$

$$keystream: 11111 \quad 10000 \quad 01000 \quad \ldots \ldots$$

initialization vector در واقع همان m بیت ابتدای $keystream$ می‌باشد پس داریم:

$$Initialization\ Vector: \xrightarrow{m=6} (Z0 \ = \ 1{,}1{,}1{,}1{,}1{,}1)$$

4.3.

$$S_{i+m} = \sum_{j=0}^{m-1} C_j S_{i+1} \ mod\ 2$$

$$m = 6$$

$$\begin{aligned} i = 0&: \quad S_6 \equiv C_0S_0 + C_1S_1 + C_2S_2 + C_3S_3 + C_4S_4 + C_5S_5 \\ i = 1&: \quad S_7 \equiv C_0S_1 + C_1S_2 + C_2S_3 + C_3S_4 + C_4S_5 + C_5S_6 \\ i = 2&: \quad S_8 \equiv C_0S_2 + C_1S_3 + C_2S_4 + C_3S_5 + C_4S_6 + C_5S_7 \\ i = 3&: \quad S_9 \equiv C_0S_3 + C_1S_4 + C_2S_5 + C_3S_3 + C_4S_4 + C_5S_5 \\ i = 4&: \quad S_{10} \equiv C_0S_4 + C_1S_5 + C_2S_6 + C_3S_7 + C_4S_8 + C_5S_9 \\ i = 5&: \quad S_{11} \equiv C_0S_5 + C_1S_6 + C_2S_7 + C_3S_8 + C_4S_9 + C_5S_{10} \end{aligned}$$

$$\forall: \ \ (S_0, S_1, S_2, S_3, S_4, S_5) = 111111_2$$

$$\forall: \ \ (S_6, S_7, S_8, S_9, S_{10}, S_{11}) = 000001_2$$

$$
\begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^{-1} \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}
= \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}
$$

4.4.

$$Ciphertext \oplus Keystream = plaintext$$

$Keystream$: 11111 10000 01000 01100 01010 01111 01000 11100 10010

$Ciphertext$: 01001 11111 00000 11010 00100 00011 01001 11100 00001

$Plaintext$: 10110 01111 01000 10110 01110 01100 00001 00000 10011

که متناظر است با:

$WPIWOMBAT$

The $WOMBAT$ is an animal that lives in Tasmania and South-Eastern Australia.

4.5. Known-plaintext Attack.

5.

حل :

هدف اول ما به دست آوردن $keystream$ است. می توان متن اصلی شناخته شده را به نمایش کد اسکی تبدیل نمود. به دلیل ویژگی‌هایی که عملگر XOR دارد، اگر $a \oplus b = c$ باشد در نتیجه $a \oplus c = b$ می‌تواند صحیح باشد. پس XOR بیت‌های متن اصلی معلوم با بیت های متن رمزشده خروجی معلوم برای ما $keystream$ را تولید می‌کند که می توان در رمزنگاری از آن استفاده نمود.

Plaintext ASCII	B	A	R	A	C	K	O	B	A	M	A
Plaintext bits	01000010	01000001	01010010	01000001	01000011	01001011	01001111	01000010	01000001	01001101	01000001
Ciphertext bits	01000011	00011011	00010010	00110000	11111000	10100111	10001110	11101001	00010100	00011101	01100100
Keystream + Nonce	0000001	01011010	01000000	01110001	10111011	11101100	11000001	10101011	01010101	01010000	00100101

به دلیل اینکه نانس ۱ به هر بایت برای این پیام خاص اضافه شده بود، ما می توانیم آن را از هر بایت از $keystream$ کم کنیم تا کلید ثابت را بدست آوریم. البته ما نیاز نداریم که اینکار را انجام دهیم به دلیل اینکه ما می دانیم دومین پیام با استفاده از یک نانس ۲ رمز شده است. $keystream$ مورد استفاده برای رمزنگاری دومین پیام می تواند صرفا با افزودن ۱ در پیمانه ۲۵۶ به اولین $keystream$ به دست آید.

Fixed Key + 1	00000001	01011010	01000000	01110001	10111011	11101100	11000001	10101011	01010101	01010000	00100101
Fixed Key + 2	00000010	01011011	01000001	01110010	10111100	11101101	11000010	10101100	01010110	01010001	00100110

می‌توانیم دومین متن رمزشده را با استفاده از این $keystream$ جدید رمزگشایی کنیم. می‌توانیم هر بیت از $keystream$ جدید را با هر بیت از دومین متن رمزشده XOR کنیم تا هر بیت از متن اصلی به دست آید.

Keystream	00000010	01011011	01000001	01110010	10111100	11101101	11000010	10101100	01010110	01010001	00100110
Ciphertext bits	01000110	00010100	00001111	00110011	11110000	10101001	10010110	11111110	00000011	00011100	01110110
Plaintext bits	01000100	01001111	01001110	01000001	01001100	01000100	01010100	01010010	01010101	01001101	01010000

به محض اینکه ما بیت های متن اصلی را داشته باشیم، می توان به آسانی هر بایت رابه اسکی موردنظر تبدیل نمود تا متن اصلی به دست آید.

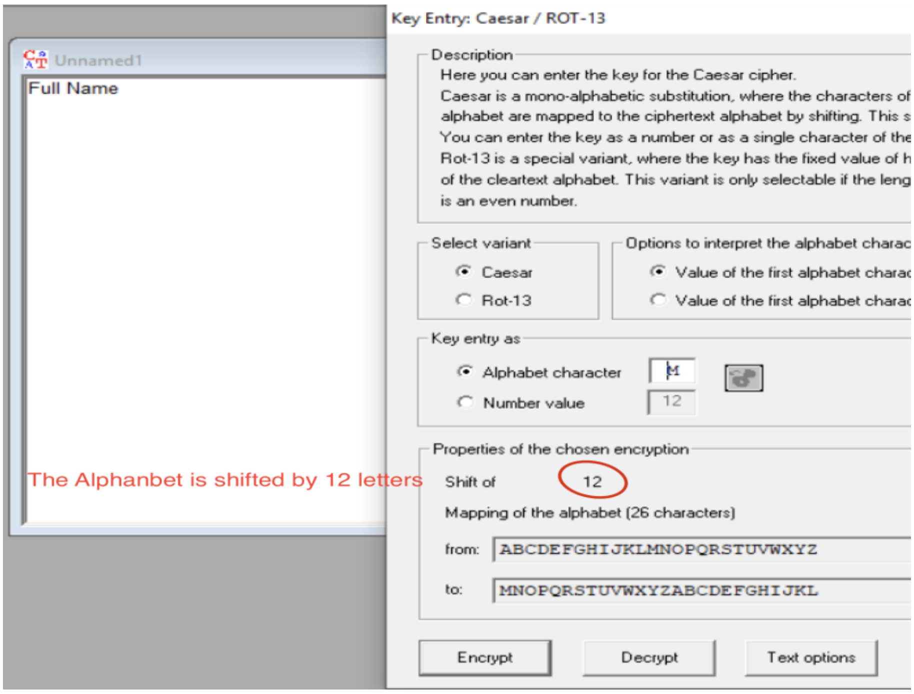
Plaintext bits	01000100	01001111	01001110	01000001	01001100	01000100	01010100	01010010	01010101	01001101	01010000
Plaintext ASCII	D	O	N	A	L	D	T	R	U	M	P

پیام $Blake$:

DONALDTRUMP

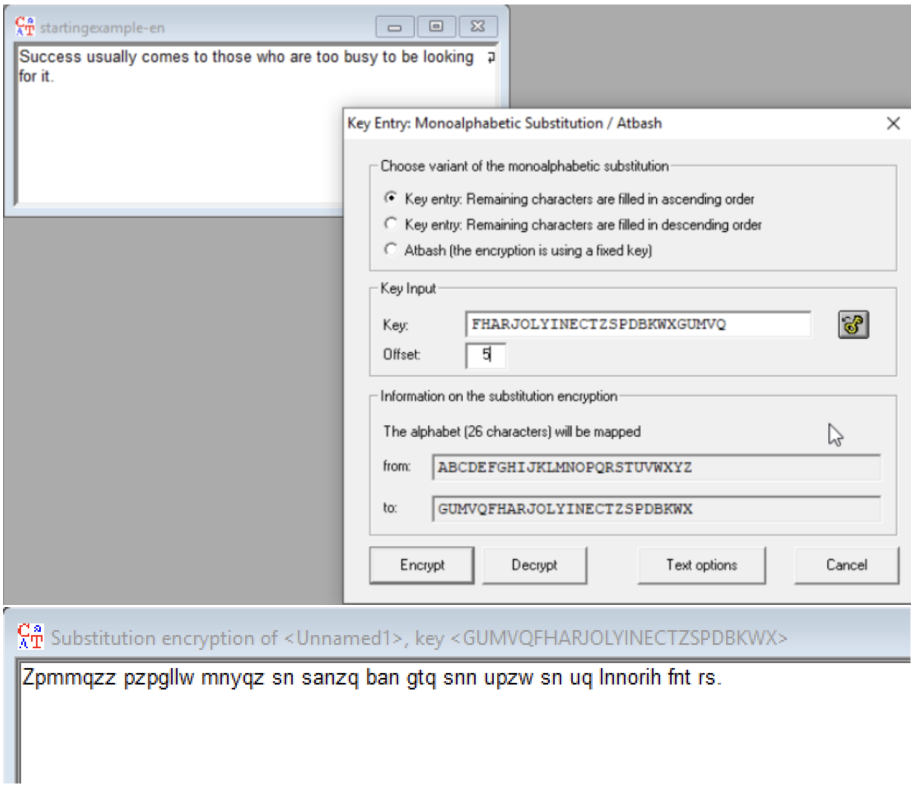
حل :

6.1.



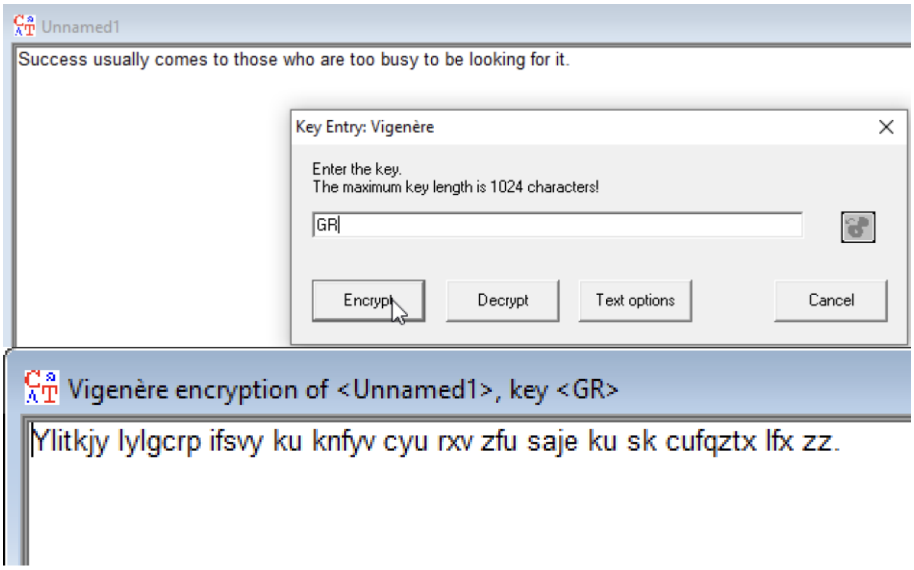
6.2.

My Student No is 9527393 which equals 26*366438+5. Meaning that I should use the number 5 as my offset to the substitution cipher.

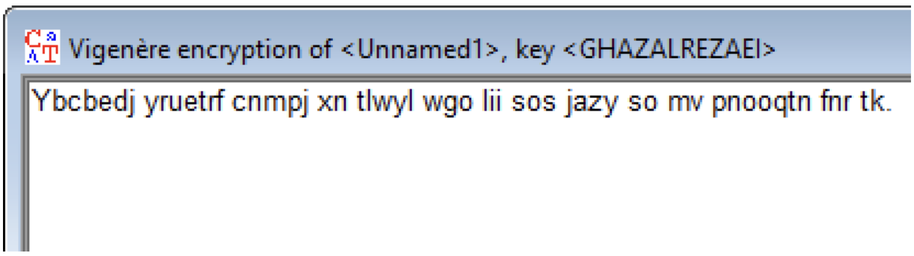

startingexample-en
Success usually comes to those who are too busy to be looking for it.
Key Entry: Monoalphabetic Substitution / Atbash
Choose variant of the monoalphabetic substitution
Key entry: Remaining characters are filled in ascending order
Key entry: Remaining characters are filled in descending order
Atbash (the encryption is using a fixed key)
Key Input
Key: FHARJOLYINECTZSPDBKWXGUMVQ
Offset: 5
Information on the substitution encryption
The alphabet (26 characters) will be mapped
from: ABCDEFGHIJKLMNOPQRSTUVWXYZ
to: GUMVQFHARJOLYINECTZSPDBKWX
Encrypt
Decrypt
Text options
Cancel
Substitution encryption of <Unnamed1>, key <GUMVQFHARJOLYINECTZSPDBKWX>
Zpmmqzz pzpgllw mnyqz sn sanzq ban gtq snn upzw sn uq lnnorih fnt rs.

6.3.

a.

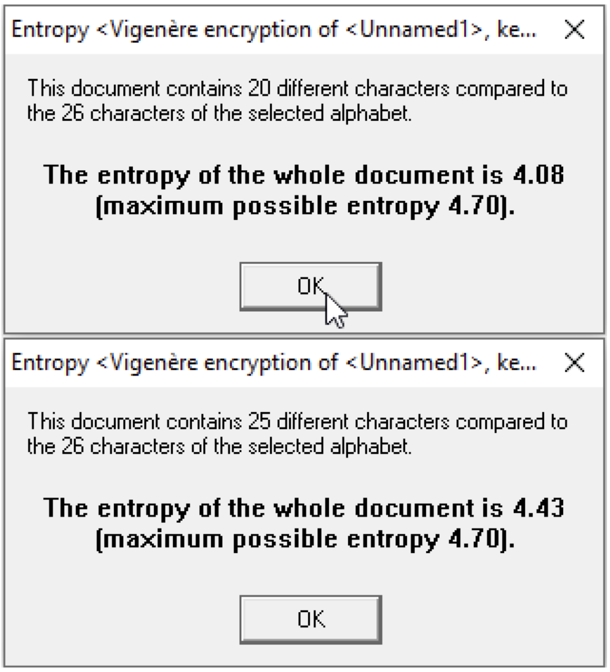

Unnamed1
Success usually comes to those who are too busy to be looking for it.
Key Entry: Vigenère
Enter the key.
The maximum key length is 1024 characters!
GR
Encrypt
Decrypt
Text options
Cancel
Vigenère encryption of <Unnamed1>, key <GR>
Ylitkjy lylgcrp ifsvy ku knfyv cyu rxv zfu saje ku sk cufqztx lfx zz.

b.


Vigenère encryption of <Unnamed1>, key <GHAZALREZAEI>
Ybcbedj yruetrf cnmpj xn tlwyl wgo lii sos jazy so mv pnooqtn fnr tk.

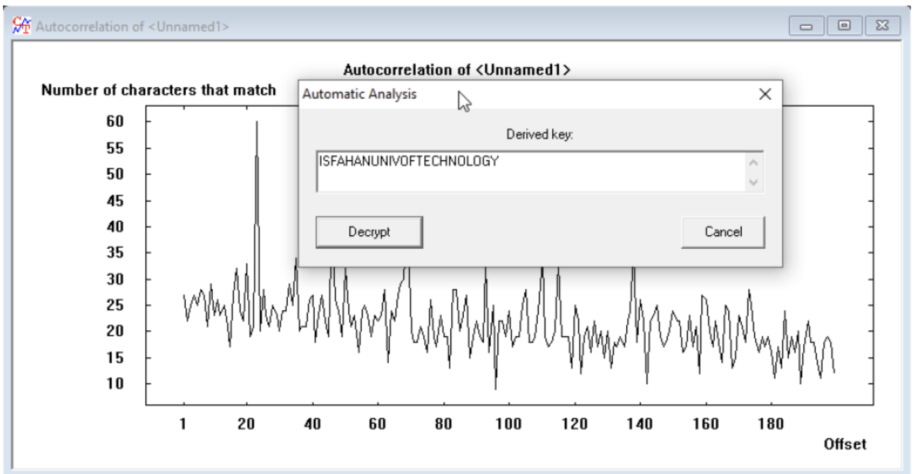
c.

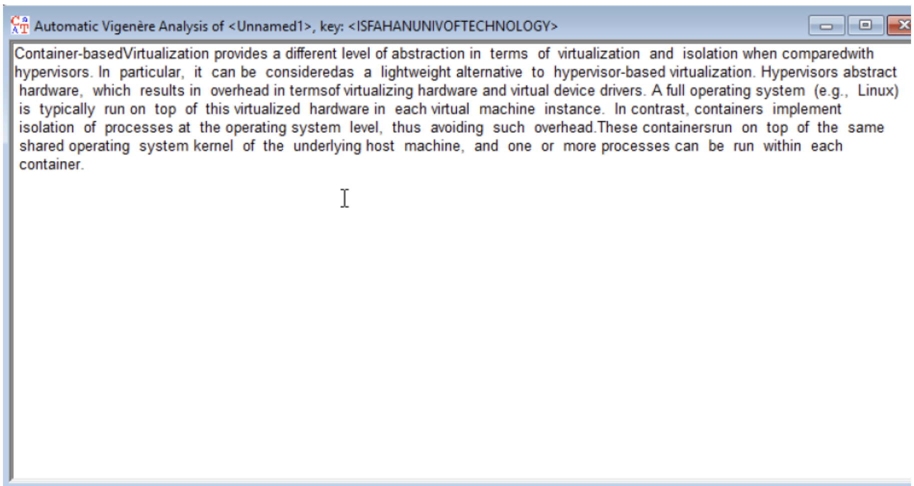
As observable in the following pictures the entropy of the document encrypted with the shorter key is 4.08, which is way smaller compared to the ciphertext with the longer key (4.43).



Entropy is a measure of the "randomness" of the data in a file, where typical text files will have a low value, and encrypted or compressed data will have a high measure. If data is encrypted, it will have a higher entropy value compared to one that isn't. Actually, in encrypted files, character distribution is random, or at least much more random than a "normal" data file. Therefore, the lower its entropy, the more probable a file is a normal or weakly encrypted one.

6.4. The autocorrelation tool analyzes different parts of a message and compares them to find similarities. It is possible to derive the length of the key using this tool, when the message is encrypted with the Vigenère cipher.

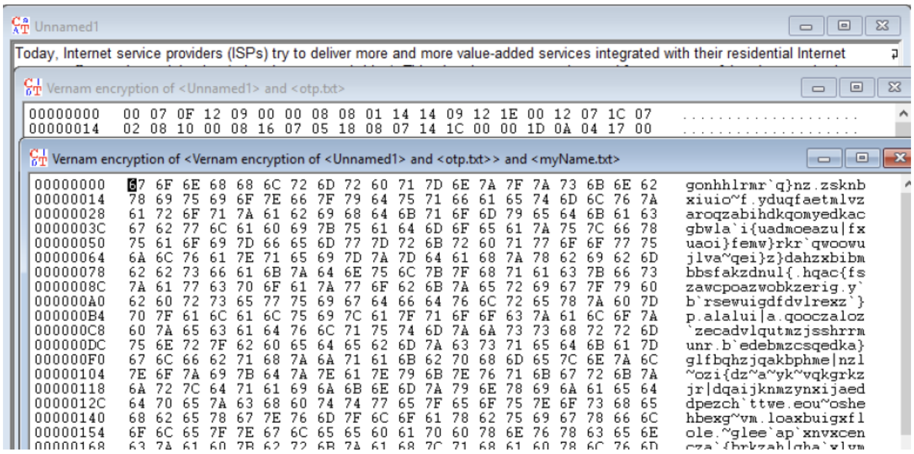




6.5.

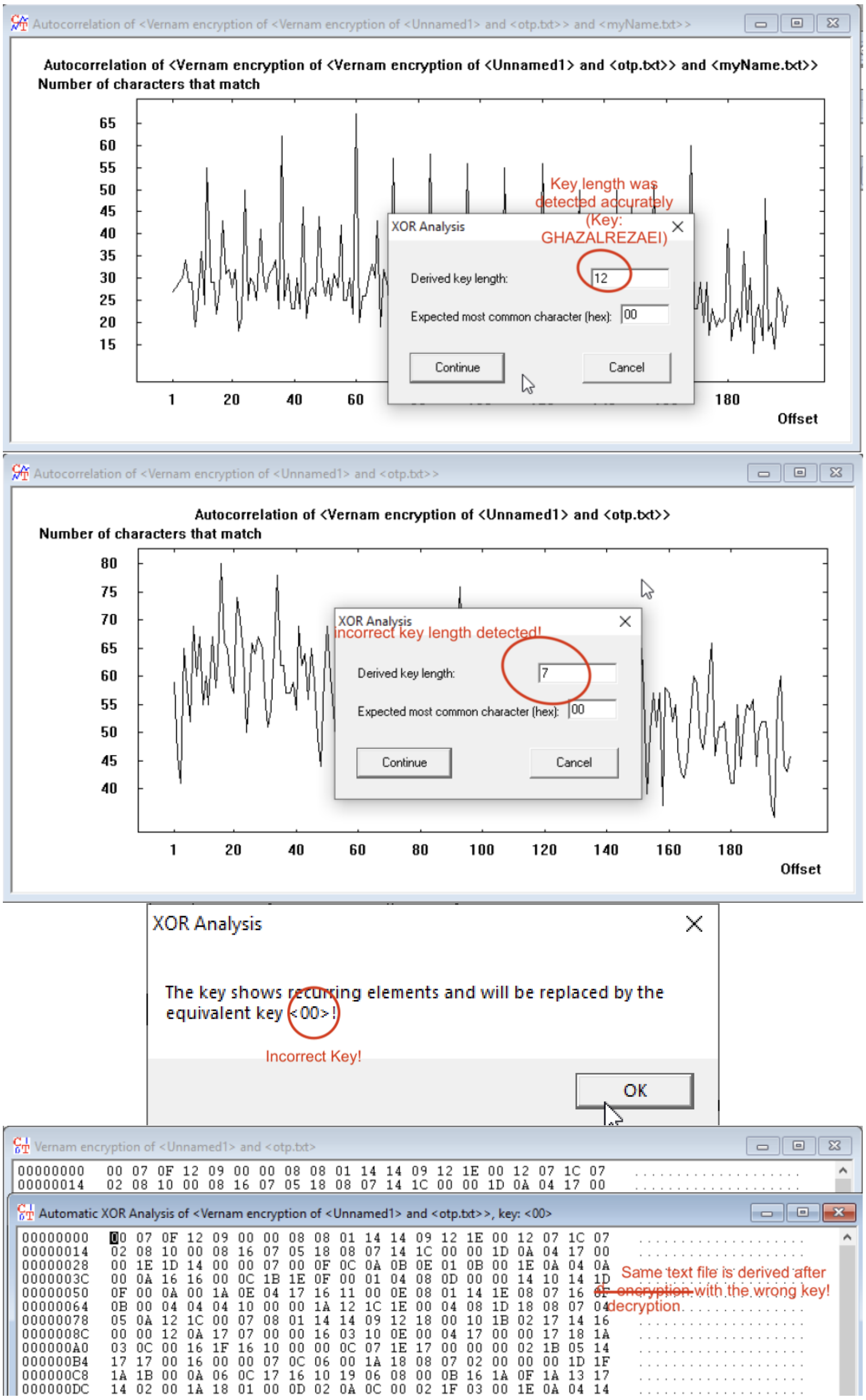
a & b.

Simply follow the instructions in the question.



c.

In One Time Pad, if key is shorter than message, it means that some part of text will be encrypted with same part of key, two or more time. In this case, by XORing two part encrypted with the same key, adversary can receive tiny pieces of information about plaintext. If key is significantly shorter than plaintext, adversary can apply frequency analysis to discover the whole message, or a part of it. As displayed in below figure, the frequency analysis tool has successfully discovered the key length of the shorter OTP key.


Autocorrelation of <Vernam encryption of <Vernam encryption of <Unnamed1> and <otp.txt>> and <myName.txt>>
Number of characters that match
Offset
XOR Analysis
Derived key length: 12
Expected most common character (hex): 00
Continue
Cancel
Key length was detected accurately (Key: GHAZALREZAEI)
Autocorrelation of <Vernam encryption of <Unnamed1> and <otp.txt>>
Number of characters that match
Offset
XOR Analysis
Incorrect key length detected!
Derived key length: 7
Expected most common character (hex): 00
Continue
Cancel
XOR Analysis
The key shows recurring elements and will be replaced by the equivalent key <00>!
Incorrect Key!
OK
Vernam encryption of <Unnamed1> and <otp.txt>
00000000 00 07 0F 12 09 00 00 08 08 01 14 14 09 12 1E 00 12 07 1C 07
00000014 02 08 10 00 08 16 07 05 18 08 07 14 1C 00 00 1D 0A 04 17 00
Automatic XOR Analysis of <Vernam encryption of <Unnamed1> and <otp.txt>>, key: <00>
00000000 00 07 0F 12 09 00 00 08 08 01 14 14 09 12 1E 00 12 07 1C 07
00000014 02 08 10 00 08 16 07 05 18 08 07 14 1C 00 00 1D 0A 04 17 00
00000028 00 1E 1D 14 00 00 07 00 0F 0C 0A 0B 0E 01 0B 00 1E 0A 04 0A
0000003C 00 0A 16 16 00 0C 1B 1E 0F 00 01 04 08 0D 00 00 14 10 14 1D
00000050 0F 00 0A 00 1A 0E 04 17 16 11 00 0E 08 01 14 1E 08 07 16 0F
00000064 0B 00 04 04 04 10 00 00 1A 12 1C 1E 00 04 08 1D 18 08 07 04
00000078 05 0A 12 1C 00 07 08 01 14 14 09 12 18 00 10 1B 02 17 14 16
0000008C 00 00 12 0A 17 07 00 00 16 03 10 0E 00 04 17 00 00 17 18 1A
000000A0 03 0C 00 16 1F 16 10 00 00 0C 07 1E 17 00 00 00 02 1B 05 14
000000B4 17 17 00 16 00 00 07 0C 06 00 1A 18 08 07 02 00 00 00 1D 1F
000000C8 1A 1B 00 0A 06 0C 17 16 10 19 06 08 00 0B 16 1A 0F 1A 13 17
000000DC 14 02 00 1A 18 01 00 0D 02 0A 0C 00 02 1F 03 00 1E 0A 04 14
Same text file is derived after ~~encryption~~ with the wrong key! decryption