CS641A (Modern Cryptology)

Chapter-5 Report

Instructor: Prof. Manindra Agrawal

Group Name: CMEN

Names :	Pratyush	170500
	Pravar Deep Singh	160508
	Tushar Garg	160749
	Umesh Meena	160755

Date of submission of report : 08/03/2020

In Chapter 5, as we enter we see the following message:

You are in a passage. There is very little light here. You light a matchstick... the floor is full of rocks. A cold draught hits you from behind and you shiver. You try to walk quickly but stumble. Cursing your luck, you slow down. The passage curves downwards. You put one hand on a side wall for guidance. The downward slope becomes steeper. You walk very slowly to avoid falling down. The slope has become very steep now… You throw away the matchstick to hold both side walls of the passage with your hands. It becomes completely dark! Your foot hits a small rock that rolls down the passage. Stopping, you try to hear the sound of the rock hitting the bottom and after a while you hear a very distant splashing sound. The passage seems to be leading to some deep underground well!

Since a passage was in front of us so we entered the command "**go**" to go along the passage and we obtained the following message:

You bravely decide to go ahead. With slow and deliberat esteps, you go down the slope. Suddenly, as you put down afoot on the floor, your foot finds thin air! In almost slow motion, you fall forward. You throw your hands about to holda rock but miss everything and suddenly you are in a free fall!! Desperate, you try to grab something, but there is nothing to grab …

Now since we had nothing to grab the only way which seemed to be get out of the free fall seemed to be some kind of magic. So we entered the command "**wave**" to wave the wand and we obtained the following message:

Magically, your fall downwards is arrested! Almost in slow motion, you go downwards till you hit water. It is very cold! You swim in one direction in the hope of finding dry land, but only find vertical rocky wall. Swimming in opposite direction yields the same result!

Now since swimming didn't work out we thought of diving into the water in hope of findinging something. So we entered the command "dive" and obtained the following message:

Diving under the water, you immediately notice mild yellowish light coming out of what appears like a hole in a wall under the water. Swimming quickly, you go through the hole. Now swimming upwards, you quickly hit the surface, and find yourself in a kind-of pool. The pool is circular in shape. You swim to one end of it, and climb out of water...

After a while, you look around the place. The pool is a perfect cicle! It cannot be naturally formed, you conclude. Then who could have constructed a pool in such a place? There are a number of lighted torches all around, providing ample yellow light. Someone seems to have fitted these torches and lighted them! Another unanswered fact...

A well-lit passage is carved in the wall, and seems to be leading somewhere deep inside...

Since we were in front of a well-lit passage we went into it by entering the command “go”. So we obtained the following message:

You walk down the passage. After quite a long walk, the passage ends in a small door. Entering, you stand transfixed! You are inside a large hall! Someone has constructed this deep inside the mountains!! The floor and the walls are made of marble, there are chandeliers hanging from the ceiling with burning torches inside. You rub your eyes in disbelief. But the scene stays. Recovering a little from the initial shock, you look around the hall. It has two exits, one where you enetered from and the other that is closed. It has a glass panel next to it and it is a proper door, made of carved wood...

You go around the hall, exploring its details. The marble is is very good quality. (How did someone get all this marble here, you wonder?) It feels cold. The chandeliers are made of crystal glass, so the light is reflecting in beautiful patterns. Who designed all this?

Now we entered the command “read” and obtained the following message.

You come up to the closed door and look at the screen .....
there is nothing written on it as earlier.

As you wonder if the spirit is around to help you out, you hear
it whispering in your ears ...

"This is another magical screen. And this one I remember perfectly... Consider a block of size 8 bytes as 8 x 1 vector over F_{128} -- constructed using the degree 7 irreducible polynomial x^7 + x + 1 over F_2. Define two transformations: first a linear transformation given by invertible 8 x 8 key matrix A with elements from F_{128} and second an exponentiation given by 8 x 1 vector E whose elements are numbers between 1 and 126. E is applied on a block by taking the ith element of the block and raising it to the power given by ith element in E. Apply these transformations in the sequence EAEAE on the input block to obtain the output block. Both E and A are part of the key. You can see the coded password by simply whispering 'password' near the screen..."

By whispering ‘password’ near the screen we obtained the following message: ‘gkftmrfrfolimtgrlrihgniummflhkio’

So we found the EAEAE encryption which we had to break and we also got the password whose plaintext we had to find.

Encryption method :

In the last passage it was given that the input is represented using the EAEAE scheme where E is an exponentiation operation and A is a Linear transformation operation both of which assumes that input is divided into bytes and is written as an 8x1 vector.

Some Properties of F_{128} field:-

Definition of Field:- A structure $< F, +, \cdot >$ is a field if the following two conditions are satisfied:

1. $< F, +, \cdot >$ is a commutative ring.
2. For all elements of F , there is an inverse element in F with respect to the operation '$\cdot$', except for the element 0, the neutral element of $< F, + >$.

$$F_{128} = \{a_6x^6+a_5x^5+a_4x^4+a_3x^3+a_2x^2+a_1x+a_0 \mid a_i \in F_2\}$$

Here the operation '.' is multiplication of polynomials modulo irreducible polynomial x^7+x+1. Following are the properties which the above definition of F_{128} satisfies:-

1. In the multiplicative group of field F_{128} we will have 127 number of elements, as 0 is excluded in the multiplicative group. Since the order of multiplicative group is 127 which is a prime, from Lagrange Theorem we know that the order of every element will either be 1 or 127. So if the element is the identity of the multiplicative group then order is 1 otherwise the order will be 127.

2. Every finite abelian group satisfies the property that $(ab)^n = a^n b^n$ and from definition fields are abelian.

3. If we raise all the elements by the same power which is not a multiple of 127 then we will obtain all the elements to be different i.e. map f: $F_{128} \longrightarrow F_{128}$ such that $f(g) = g^n$ is a one-one and onto map. First of all the map is well defined since a group is closed under group action hence g^n will lie inside the multiplicative group $F_{128}\backslash\{0\}$. Now if,

$$f(g) = f(h)$$
$$g^n = h^n$$
$$g^n h^{-n} = 1$$
$$g^n (h^{-1})^n = 1$$
$$(gh^{-1})^n = 1 \qquad \{\text{from property 2}\}$$

Since n is not a multiple 127 and the order of each element is either 127 or 1 we conclude that $gh^{-1} = 1$, i.e. g = h. Now since g is a map from F_{128} to F_{128} and is one-one it will have to be onto.

4. The map $T_a : F_{128} \longrightarrow F_{128}$ such that $T_a(b) = ab$ where $a \neq 0$ is one-one and onto.

Proof: Assume $T_g(b) = T_g(c)$

$$\Rightarrow \quad ab = ac$$

Pre-multiplying with a^{-1} we get,

$$a^{-1}(ab) = a^{-1}(ac)$$
$$(a^{-1}a)\, b = (a^{-1}a)\, c \qquad \{\text{Using associativity}\}$$
$$eb = ec \qquad \{\text{Since } a^{-1}a = e\}$$
$$b = c \qquad \{\text{Since e is the identity}\}$$

Hence b and c will have to be the same for the above condition to hold.Hence T_a is one-one function. Now since T_a is a map from F_{128} to F_{128} and is one-one it will have to be onto.

Breaking the EAEAE:-

From the passage we know that the input to the EAEAE encryption is a block of 8 byte represented as a vector of size 8x1 with each element from the field F_{128}. So we thought if we give some input which is less than the size i.e. 8x1 = 8 bytes then it will assume some inherent padding to the input so that it becomes 8 byte long and EAEAE can be operated on that input. So to find out answer to these question we started to put random input on the site and we observed as follows:-

1. If the input string is of less than or equal to 16 characters then the output string was of 16 characters. Now as discussed above we thought that there must be some kind of padding to the input to make it 16 characters long.

2. Now If the input plaintext was given a longer string (17 to 32 characters) then the number of characters in the output became 32. So, we thought that the output may represent encryption of two blocks of the input.

3. To confirm the size of block length, we tried the input, given in table 1, on the site and observed that the ciphertext corresponding to 'aaaaaaaaaaaaaaab' is the same as ciphertext corresponding to 'aaaaaaaaaaaaaaa' appended with the ciphertext of 'b'. So it was apparent from this that the encryption scheme does the encryption by encrypting 16 characters at a time starting from the start of the string and if the number of characters in the last block is less than 16 then it assumes some kind of padding in the last block.

Input Length	plaintext	ciphertext	Output Length
1	b	iqhufrmqmnmogtmq	16
16	aaaaaaaaaaaaaaaa	grfjjhhnkuigglmt	16
17	aaaaaaaaaaaaaaaab	grfjjhhnkuigglmtiqhufrmqmnmogtmq	32

Table 1: Plaintexts and their corresponding ciphertexts.

4. It was said that the input of 8 bytes is represented as 8x1 vector which is input to the EAEAE encryption method. We thought that 16 characters should represent the 8 bytes of the input. This structure seemed to be exactly the same as what we encountered before in chapter-4 (DES).

5. Now to confirm if the EAEAE also follows the same structure as chapter-4, we generated 2,000 input text randomly and obtained their corresponding cipher text and did a frequency analysis. What we found was that those 2,000 ciphertext only consisted of letters from the alphabets 'f' to 'u'. Which is exactly the same as DES so we assumed that two letters correspond to one byte, with the mapping given in Table 2.

f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Table 2: number in decimal corresponding to the characters

6. Now since every element of the 8x1 vector is an element of F_{128} which is modulo irreducible polynomial x^7+x+1, it can not contain x^7 term hence its 8th bit must always be zero. Hence, to represent any element in F_{128} we require 7 bits so total number of possible output from 7 bits = $2^7 = 128$ = number of elements in the field. So assuming the structure assumed in previous point where two characters represents one byte we thought that in the output text the first letter can never take character which has its 4th bit on. So we did frequency analysis of the characters at the odd position (assuming indexing from 1) and observed characters only from 'f' to 'm' which confirmed our assumption on the structure of output.

The Square Attack on EAEAE:-

Let us assume that the A matrix to be $[A_{ij}]$ and E matrix to be $[E_i]$. Now we take 128 column vectors with P in i^{th} row and C in all other rows. Here P represents that the i^{th} row in all the 128 vectors take different values and C represents that the i^{th} row in all the 128 vectors takes the same value. So for i = 3 the input vector looks as follows:-

$$S = \begin{bmatrix} C \\ C \\ P \\ C \\ C \\ C \\ C \\ C \end{bmatrix}$$

1. **First E operation:-** The map f s.t. $f(g) = g^n$ is one-one and onto from property 3, So when we will apply the vector E on the input vector S we will obtain C wherever C was there and in the row where P was there we will obtain the P as we have 128 different initial values which will map to 128 different values.

2. **First A operation:-** Now we consider the operation A on vector S obtained after applying E. After multiplying by the matrix A, the j^{th} row will be $A_{j1}S_1 + A_{j2}S_2 + ... + A_{ji}S_i + ... + A_{j8}S_8$. Now the A matrix is fixed for all the inputs so for a row all terms except $A_{ji}S_i$ will be some fixed constant (say D) for all the 128 input vectors. Now for 128 input vectors S_i represents a P i.e. all of them are distinct elements, so if A_{ji} is zero all the values will be D otherwise we claim that they will be P. Let us assume if there exist two different inputs which maps to the same output i.e. $A_{ji}P_1 + D = A_{ji}P_2 + D$ which implies $A_{ji}P_1 = A_{ji}P_2$ but from property 4 we know if A_{ji} is not zero it can not map two different inputs to the same output. Hence we know that we will get P in the output.

   So if the j^{th} row has i^{th} entry as non-zero then we will obtain P after the first A matrix operation otherwise we will obtain a C in the j^{th} row and vice versa.

$$S_j = \begin{cases} C & \text{if } A_{ji} = 0 \\ P & \text{otherwise} \end{cases}$$

3. **Second E operation:-** Similar to the 'First E operation' we will obtain a P if it was initially a P and will obtain a C if it was initially a C.

4. **Second A operation:-** Now we were not able to maintain any specific structure except that after operating by matrix A for each row the sum of all the entries will be zero because after the the multiplication each term with a C will come in each of the input and so will come 128 time which is essentially not contributing anything and all the P terms will also sum to zero. Hence the output after operating A will be 'Z' where Z represents that the sum of all 128 entries is zero.

5. **Third E operation:-** After operating E all the structure will be lost and we won't be able to say anything about the final output.

So, as we gave our input as P in i^{th} row and C in rest of rows to the system, we expected output ciphertext to be all rows X (none of C,P or Z). However we realised that if our input has P in the i^{th} row, output has C in first i-1 rows. We gave different input but the output had the same pattern. Since the final output is C for first i rows , it means that the output after first A operation should also have C in first i rows (i^{th} row of input is P), Which means as argued above in point 2 , A_{ji} is zero for all $j < i$. This means our matrix A is a lower triangular matrix.

Obtaining the value of E and A matrix:-

Now we know that the A matrix is lower triangular assuming that the A matrix operates on the input vector from left. So we know that the first row of the output will only depend on the first row of A and will be related to input by the following expression:-

$$(A_{11}(A_{11}(S_1^{E_1}))^{E_1})^{E_1}$$

$$A_{11}^{E_1+E_1^2}S_1^{E_1^3}$$

So we iterated over all possible combinations of A_{11} and E_1 and obtained 3 possible pairs of A_{11} and E_1 which satisfies the generated mapping of 128 input texts to 128 output ciphertexts.

Now we wanted to apply the same analogy to the other elements. So by inspecting the structure of the A matrix, we realized that the first j entries of output will be zeros after each of the steps if the first j entries of the input were zero.

Proof:- P(1): For the first row we know it is true.

P(n): Let us assume it is true till the n^{th} row.

Since the first n entries are zero they will remain zero after each operation. Now for $(n+1)^{th}$ zero, it will be zero after operating first E. Now since the output after operating A on row k only depends on the first k entries of the input we know that the output in row n+1 after operating A will be zero as the first n+1 entries are zero. So we can repeat the same argument for the other two E and A operations and hence it is also true for P(n+1).

So we know that if first i entries are zero in input they will remain zero in the output as well. Which tells us that the output in the $(i+1)^{th}$ row will only depend on the input in the $(i+1)^{th}$ rows if first i entries are zero. So if we generate 128 inputs which had 'ff' in all the rows except i^{th} in which P runs over all possible 128 values the output looks as follows:

$$A_{ii}^{E_i+E_i^2}S_i^{E_i^3}$$

So we worked on the input of the form described above and ran through all the possible values of A_{ii} and E_i for all the 128 input output pair for each i and obtained 3 possible pairs for (A_{ii}, E_i) in the intersection of solution space for each 128 io pairs. The following Table summarizes the possible pair.

Entry	First possible pair	Second possible pair	Third possible pair
$\mathbf{E_1, A_{11}}$	26, 48	113, 43	115, 125
$\mathbf{E_2, A_{22}}$	23, 125	48, 58	56, 4
$\mathbf{E_3, A_{33}}$	70, 67	124, 28	60, 80
$\mathbf{E_4, A_{44}}$	125, 15	40, 45	89, 31
$\mathbf{E_5, A_{55}}$	111, 22	77, 119	66, 115
$\mathbf{E_6, A_{66}}$	74, 111	44, 66	9, 19
$\mathbf{E_7, A_{77}}$	52, 37	103, 126	99, 12
$\mathbf{E_8, A_{88}}$	84, 15	72, 96	98, 73

Table 3: Possible E_i and A_{ii} pairs for $1 \leq i \leq 8$

First we need to reduce possible pairs of E_i and A_{ii} to one. Now consider an input in which i^{th} row is P and the $(i-1)^{th}$ row is some nonzero C and the rest of the rows are zeros. Now consider any one of the 128 input blocks of text, the i^{th} element of output ciphertext for this block will depend only on E_i , A_{ii} and $A_{i,\, i-1}$ and i^{th} element of input block (one can easily see this by expanding the matrix multiplication). So we can iterate over 3 (E_i, A_{ii}) pairs and 128 possible values of $A_{i,\, i-1}$ for an input block and output ciphertext block pair. Taking 128 such input-output pairs will give a unique trio of E_i ,A_{ii} and $A_{i,\, i-1}$. Furthermore if our 128 blocks input has P on i^{th} row, some known nonzero C on j,j+1,.....i-1 rows and the rest of the rows are zero, then for any input text block the i^{th} element of output ciphertext for this input block will depend only on A_{kl} (where $0 < k < i$ and $0<l<i$) , A_{ip} (where $j < p \leq i$) , E_q ($0 < q < i$) ,A_{ij} and the i^{th} element of input block. So we can find A_{ij} by iterating over 128 possibilities of A_{ij} and 128 io pairs to find unique value of A_{ij} if we know the values of A_{kl} (where $0 < k < i$ and $0 < l < i$) and A_{ip} (where $j < p \leq i$) and E_q ($0 < q < i$) . Since we have till now already found A_{ii}, $A_{i,i-1}$ and E_i for $1 \leq i \leq 8$, we can recursively find the rest of elements of the matrix A.

Using the above technique we obtained the A and E matrix as follows:

$$A = \begin{bmatrix} 125 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 87 & 125 & 0 & 0 & 0 & 0 & 0 & 0 \\ 59 & 96 & 67 & 0 & 0 & 0 & 0 & 0 \\ 13 & 102 & 28 & 15 & 0 & 0 & 0 & 0 \\ 59 & 92 & 15 & 24 & 115 & 0 & 0 & 0 \\ 100 & 43 & 96 & 37 & 89 & 111 & 0 & 0 \\ 44 & 85 & 5 & 50 & 58 & 87 & 126 & 0 \\ 37 & 29 & 61 & 124 & 38 & 121 & 106 & 96 \end{bmatrix}$$

$$E = \begin{bmatrix} 115 \\ 23 \\ 70 \\ 125 \\ 66 \\ 74 \\ 103 \\ 72 \end{bmatrix}$$

Now we inverted all the five operations done on the ciphertext '**gkftmrfrfolimtgrlrihgniummflhkio**' and obtained the corresponding deciphered text '**mkmlmgmhlololulnmllulkmolpllmpmh**', but when we tried to enter the deciphered text, we were unable to proceed to the next level. So we explored the possibility of encoded text i.e. earlier we were preprocessing the output by converting 8 bits into two letters of 4 bits, but now we thought that the binary text corresponding to the deciphered text might give us some idea. So instead we tried to read it by assuming 8 block input to be a character represented by its ASCII encoding. Finally we reached the next level by entering the password obtained as '**uvqriiohvoeyjfzr**'.

Explanation of the Codes:-

1. The file ‘pattern.py’ generates 2000 random plaintext find their corresponding ciphertexts and saves them in the file ‘random_output.txt’.
2. The file ‘pattern.py’ gives the frequency of the different letters in the output saved in the file ‘random_output.txt’.
3. The file ‘function.py’ contains the definition of the different functions which were used by the different files.
4. The folder ‘lower_triangular’ contains the file which was used to determine that the A matrix is upper diagonal.
    a. This folder has 8 folders numbered from 1 to 8 in which the name of the folder represents the position of ‘P’ in the input vector. Each of the folders contains 256 files which were initially used to generate as many linearly independent equations as possible.
    b. The file ‘query.py’ was used to generate the files described above.
    c. The file ‘equations.py’ was used to find the rank of the equations generated using square attack.
5. The folder ‘findind_vars’ contains the file which was used to find the value of A and E matrix and used to decipher the password.
    a. This folder has 8 folders numbered from 1 to 8 in which the name of the folder represents the position of ‘P’ in the input vector and in i^{th} folder there are i files. If i^{th} folder has file named ‘k.txt’ in it then it represents that the we took input in which first i-k-1 row were ‘ff’ and then from i-k to i-1 the rows were filled with ‘fg’ and i^{th} row runs over all possible values from F_{128} and the remaining rows to be filled with ‘ff’.
    b. The file ‘query.py’ was used to generate the input described above.
    c. The file ‘A_E_and_key.py’ was used to find the values of A, E and the key of chapter-5.
6. Since moodle does not allow us to upload more than 1MB we have deleted the text files in the lower_triangular folder which can be generated by the code submitted.