[CS304] Introduction to Cryptography and Network Security

Course Instructor: Dr. Dibyendu Roy
Scribed by: Chitranshi Srivastava (202051055)

Winter 2022-2023
Lecture 3 and 4 (Week #2)

In the previous week, we discussed about the basics of cryptography and some classical ciphering techniques. We continue from there in this week.

1 Classical Ciphering Techniques

1.1 Playfair Cipher

It is a multi-letter encryption technique in which pairs of letters are encrypted instead of single letters.
Encryption :
We take a 5 × 5 matrix. Each of the 25 alphabets must be unique and one letter of the alphabet (usually J) is omitted from the table as we need only 25 alphabets instead of 26. If the plaintext contains J, then it is replaced by I. Now let us see how to generate this key table using the secret key.
In a key table, the first characters (going left to right) in the table is the phrase, excluding the duplicate letters. The rest of the table will be filled with the remaining letters of the alphabet, in natural order.
<u>Note:</u>

- If there is an odd number of letters, a X is added to the last letter.
- If there are duplicates present next to each other, insert an X between the two duplicate letters.

Now let us generate the key table for the given example.
Example :
Secret Key : PLAYFAIR EXAMPLE

$$\begin{bmatrix} P & L & A & Y & F \\ I & R & E & X & M \\ B & C & D & G & H \\ K & N & O & Q & S \\ T & U & V & W & Z \end{bmatrix}$$

<u>Process</u>

- If both the letters are in the same column, take the letter below each one (going back to the top if at the bottom)
- If both letters are in the same row, take the letter to the right of each one (going back to the left if at the farthest right)
- If neither of the preceding two rules are true, form a rectangle with the two letters and take the letters on the horizontal opposite corner of the rectangle.

We use the above rules to get the cipher text.

$$\begin{pmatrix} P & L & A & Y & F \\ I & R & E & X & M \\ B & C & D & G & H \\ K & N & O & Q & S \\ T & U & V & W & Z \end{pmatrix}$$

Plain text : HIDE ⇒ HI DE
Cipher text : BM OD ⇒ BMOD
Plain text : SACHIN ⇒ SA CH IN
Cipher text : OF DB RK ⇒ OFDBRK
Decryption :

- If both the letters are in the same column, take the letter above each one (going back to the bottom if at the top)
- If both letters are in the same row, take the letter to the left of each one (going back to the right if at the farthest left)
- If neither of the preceding two rules are true, form a rectangle with the two letters and take the letters on the horizontal opposite corner of the rectangle.

Example:
Secret Key : PLAYFAIR EXAMPLE
Cipher Text : BMOD ⇒ BM OD
Since they both lie in different row and column, we form a rectangle and we use the above rules to decrypt the message.
Plain text : HI DE ⇒ HIDE

1.2 Hill Cipher

It is a multi-letter ciphering technique in which the secret key is an n×n invertible matrix.
Secret Key : A = $(a_{ij})_{n\times n}$, where $a_{ij} \in Z_{26}$
Plain text : M = $m_1 m_2 m_3 m_4 .... m_n$
Encryption :

$$C = A \cdot M = c_1 c_2 .... c_n$$

Decryption:

$$M = A^{-1} \cdot C$$

The inverse of matrix A is given as:

$$A^{-1} = \tfrac{1}{|A|} \cdot adj(A)$$

where $|A|$ is determinant of A and adj(A) is adjoint of matrix A.
Note: While calculating inverse of matrix A during decryption, all the calculations should be done under modulo 26.

1.3 Substitution Cipher

In this technique, there is a substitution from the set of alphabets to itself. It need not be a bijection.
S: {A,B.....,Z} → {A,B.....,Z}
Here, S is the secret key and Cipher text = S(Plain text)
The number of secret keys possible : $\#S = 26^{26} \approx 2^{122}$
But if the substitution is limited for bijections, then $\#S = 26!$
Substitution Ciphers are prone to brute force attacks and exhaustive search. Since the cipher text is known, frequency distribution analysis based on natural language can be used to break the code.

2 Kerchoff's Principle

It is the concept that a Cryptographic system should be designed to be secure, even if all its details, except for the key, are publicly known. It forms the basis of open security and security by design.
If the algorithm design is hidden, then only limited number of people will be able to use it.
Also, if the design is not public, then it is more prone to security vulnerabilities. Larger the number of people cryptanalyzing it, more is the probability of finding security flaws and improving the algorithm. For long term usage, those algorithms that have been published and thoroughly analyzed provide much more security.

3 Shannon's notion of perfect secrecy

An algorithm provides perfect secrecy if the encrypted text received using that algorithm reveals absolutely nothing about the plain text(or unencrypted text). Not even a single bit of information about the actual message is revealed by the cipher text.
Suppose, we have a single but message and we are given that in the message P(0) = 0.9 and P(1) = 0.1. After encryption, if the cipher text does not reveal any additional information about guessing the message, that is, if after receiving the cipher text, the P(0) $\neq$ (0.9+ϵ)$, where \epsilon > 0$, which means the probability of guessing remains the same, then the algorithm is said to provide perfect secrecy.
Mathematically,

$$Pr[M = m|C = c] = Pr[M = m]$$
$$Pr[message|ciphertext] = Pr[message]$$

There are certain algorithms which provide perfect secrecy with some limitations. But these limitations make these algorithms impractical. For example, One Time Padding(OTP) algorithm.

4 Symmetric Key Cipher

In these ciphers, the same key is used for both encryption and decryption and hence must be kept secret from everyone other than the sender and the receiver. These are of two types

- Block Ciphers
- Stream Cipher

4.1 Block Cipher

The message to be encrypted is divided into fixed size blocks and these blocks are encrypted, that is, the encryption is done in block-wise manner. For example, DES algorithm can encrypt 64-bit messages. So if we have a message having 128 bits, it will be divided into 2 blocks of 64 bits and encryption would be done on each block.
$M = m_o||m_1||....||m_n$
length of $m_i = 1$
One way of encryption is to encrypt each block and then concatenate the encrypted blocks in the same order.
Encryption :
$C = Enc(m_0, K)||Enc(m_1, K)||....||Enc(m_n, K) = C_0||C_1||....||C_n$
There can be some other algorithms where we can do some mixing of the blocks to generate the cipher text.
Both Substitution Networks and Feistal networks are type of block ciphers

4.2 Stream Cipher

Here, the message is a stream of bits and encryption is done bitwise. $M = m_0m_1...m_l$, where $m_i \in 0, 1$
We use an algorithm to generate another bit stream using the secret key.
$Z = Z_0Z_1...Z_l$, where $Z_i \in 0, 1$
To generate the cipher text, we do XOR between bits of plain text and bits of Z.
Encryption :
$C = (m_0 \oplus Z_0, m_1 \oplus Z_1, ..., m_l \oplus Z_l)$
During decryption, the same bit stream Z will be generated using the secret key and XOR will be performed with the cipher text to get the original message.
Note : Xor function is just taken as an example. It can be another other form of computation.

4.3 Usage of Block Ciphers and Stream Ciphers

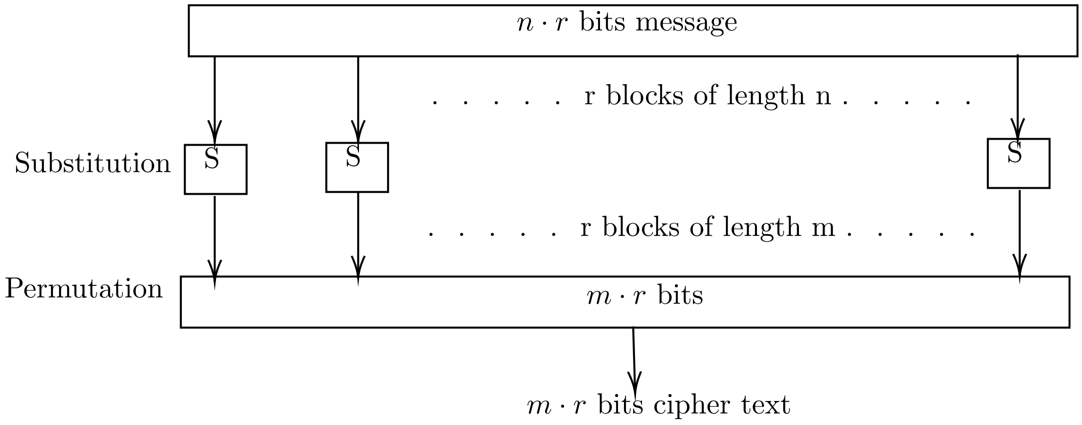
There are some algorithms which generate $2^{80} - 1$ bit long bit stream by taking 80 bit key as an input. Hence, using stream cipher, very long messages can be encrypted. However, this is not the case with block ciphers. Hence, to encrypt the whole message using block cipher, it will take 2^{74} times the time taken to encrypt one block. Hence, stream ciphers provide efficiency over block ciphers.
However, in our daily life, the message length is not so large and the software implementation of block cipher is very good. Hence, for more practical uses where the message length is small(like whatsapp), we prefer block ciphers.
Stream ciphers are used generally where the length of the message is not pre-determined or we have no idea about the length of the message. For example, voice over telephony, that is, the normal phone call uses stream ciphers as the time for which two people will communicate cannot be predicted at the starting of the call.
The hardware implementation of block ciphers is very poor as it requires a lot of gates.Hence, block ciphers are not implemented in hardware. However, all the existing stream ciphers are efficiently implemented in hardware.

5 Product Ciphers

A Product Cipher combines two or more transformations in a manner intending that the resulting cipher is more secure than the individual transformations. It can be better in terms of both security and efficiency over the individual function. One such cipher is a Substitution Permutation Network (SPN).

5.1 Substitution Permutation Network (SPN)

It is a product cipher based on a substitution box and a permutation box. Suppose we have a message of length $n \cdot r$. A substituion S and a permutation P be defined as:
$S : \{0,1\}^n \rightarrow \{0,1\}^m$
$P : \{0,1,...,m \cdot r-1\} \rightarrow \{0,1,...,m \cdot r-1\}$
We do a substitution on r blocks of length n, which will generate a $m \cdot r$ length text. Then, performing a permutation on this $m \cdot r$ length text to get the cipher text.


$n \cdot r$ bits message
. . . . . r blocks of length n . . . . .
Substitution
S
S
S
. . . . . r blocks of length m . . . . .
Permutation
$m \cdot r$ bits
$m \cdot r$ bits cipher text

6 Feistel Network

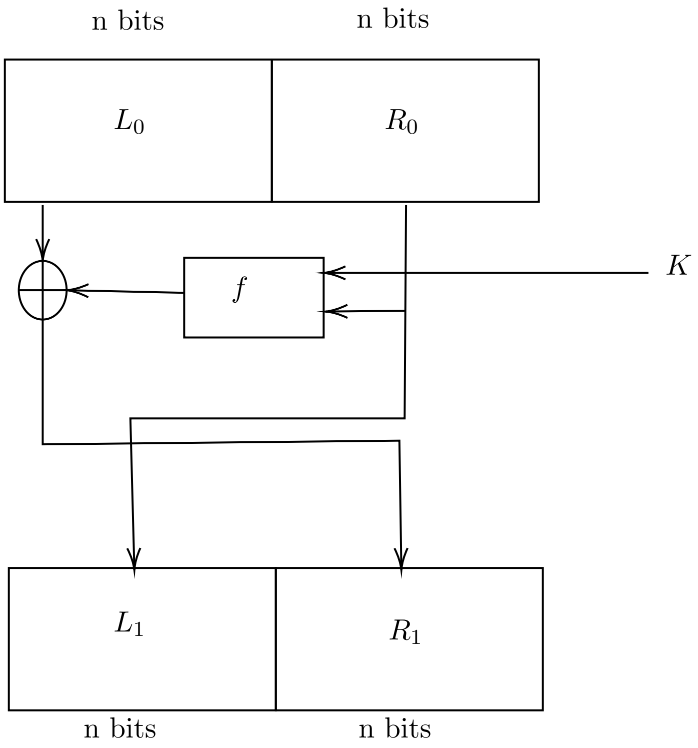
In a Feistel network, we have an even-length message. Suppose we have a message of 2n bits, we divide it into two equal parts $L_0 and R_0$.

$$\text{P} = L_0||R_0.$$

L_0 and R_0 are of n bits each and are calculated using right and left shifting of P by n bits respectively.
Let K be the secret key having length l.
A fuction f, called as round function, is defined as:

$$f : \{0,1\}^n \times \{0,1\}^l \rightarrow \{0,1\}^n$$

Encryption:
The diagram for encryption is given below:



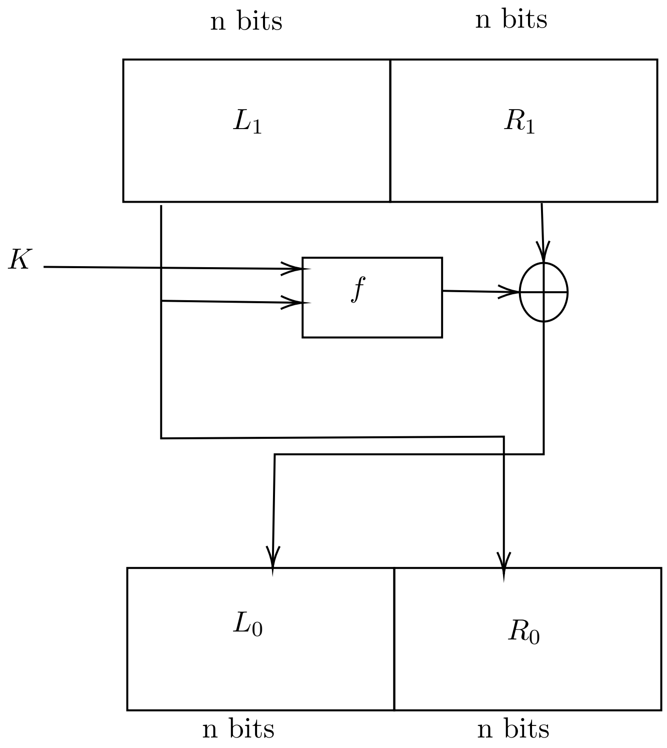
Mathematically:

$$L_1 = R_0$$
$$R_1 = L_0 \oplus f(R_0, K)$$
$$\text{C (cipher text)} = L_1 || R_1$$

Decryption:
The decryption can be done as follows:

$$R_0 = L_1$$
$$L_0 \oplus f(R_0, K) = R_1$$
$$L_0 \oplus f(R_0, K) \oplus f(R_0, K) = R_1 \oplus f(R_0, K)$$
$$L_0 = R_1 \oplus f(R_0, K)$$
$$L_0 = R_1 \oplus f(L_1, K)$$

Therefore, we have expressions for both L_0 and R_0 in terms of L_1, R_1 and K.
Diagramatically,

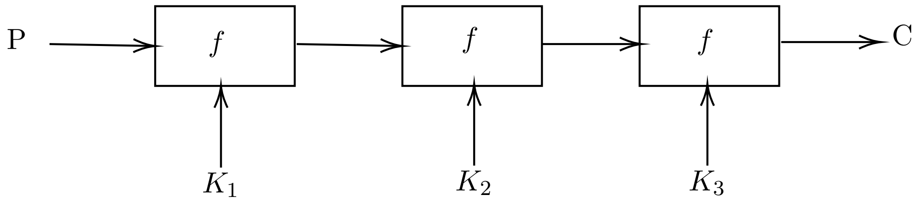


<u>**Note**</u> **:**

- The inverse of f is not neede in the decryption process. Hence, it does not matter whether f is invertible or not. So, the round function may or may not be invertible.
- It can be seen that one half of the plain text remains similar in the cipher text. However, here we have shown only one round of implementation. In actual usage, there are multiple rounds and the whole message is encrypted in later rounds.

7 Iterated Block Cipher

An Iterated Block Cipher is a block cipher involving the sequential repetition of an internal function (called the **Round Function**). The parameters include **the number of rounds r, the block size n and the round keys K_i of length l generated from the original secret key K.** For example, we see a 3-round block cipher such that :
f → Round Function
P → Plain Text Block
K → Secret Key



C → cipher text
$G(K) \rightarrow K_1, K_2, K_3$ (round keys)

The function G, which is known as the **Key Scheduling Function** takes the secret key as input generates the round keys.

8 One Time Padding

One Time Padding (OTP) provides perfect secrecy under some conditions.
Encryption:
$Enc(P, K) = P \oplus K = C$
where,
P → Plain Text
K → Secret Key
$\oplus$ → xor operation
Decryption:
$Dec(C, K) = C \oplus K = P$
To ensure that OTP provides perfect secrecy, that is,

$$P(message|Ciphertext) = P(message)$$

certain conditions have to be fulfilled. These are as follows : The conditions under which OTP provides perfect secrecy are as follows:

1. The secret key K cannot be used to encrypt two messages, that is, the key can not be reused.
2. The length of key must be greater than or equal to the length of message.
   $length(K) \geq length(P)$
3. Key K is uniformly selected from the key space.