[CS304] Introduction to Cryptography and Network Security

Course Instructor: Dr. Dibyendu Roy
Scribed by: Archit Agrawal (202051213)

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

1 Quick Recap

In the first week, the following topics were covered:

- Cryptography, Cryptanalysis and Cryptology
- Encryption and Decryption
- Types of Cryptography: Symmetric and Asymmetric Cryptography
- Security Services
- Functions: One-One, Onto, Bijection, Permutation, One-Way, Substitution Box
- Classical Ciphers: Caesar, Transposition, Substitution, Affine and Playfair (Encryption only)
- Basic Mathematics: Multiplicative Inverse in Modular Arithmetic, Euler's Totient Function and Extended Euclidean Algorithm

2 Classical Ciphers

2.1 Playfair Cipher

Playfair Cipher uses a 5×5 matrix formed using the secret key for encryption and decryption.

Encryption: The following rules are followed for encryption:

1. Create the 5×5 matrix using secret key.Rules for creating the matrix are mentioned below:
   - Characters I and J are considered as same.
   - Fill the key characters row-wise in the matrix. If a character has already appeared, skip it.
   - Fill the remaining alphabets in lexicographic order.
2. Break the message (plain text) into groups of 2 characters.
3. Add fillers if a group has same characters or if it has only one character.
   - If length of message is odd, add X at the end of message. For example:

     Plain Text: CSE
     After Insertion: CS EX

   - If both the characters are same, add an X after the first character. For Example:

Plain Text: BALL
After Insertion: BA LX LX

4. For each group, find the characters in the matrix. Do the following according to the position of these characters:

   - If both characters are in the same row, use the next right character as cipher text. If the plain text character is the last element of that row, take the first character of that row as cipher text.
   - If both characters are in the same column, use the next down character as cipher text. If the plain text character is the last element of that column, take the first character of that column as cipher text.
   - If the characters are in different rows and columns, form a rectangle with these two characters at corners. Now, use the other row corner as the cipher text for both characters.

Example 1:

Secret Key: PLAYFAIR EXAMPLE
Plain Text: HIDE

The matrix formed using encryption rules will be:

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

Now, breaking the plain text into groups of 2 will give us:

Plain Text: HI DE

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

For the first group, the characters H and I form a rectangle shown by red colour in the figure above. Hence, the corners will be swapped.

Plain Text: HI DE
Cipher Text: BM OD

For the second group, the characters D and E are in the same column. Hence, the next down character will be the cipher text.

Decryption: The following rules are followed for decryption:

1. The first two rules, that is, the matrix construction and breaking the cipher text into groups of two is same as done during encryption.

2. For each group of cipher text, find the characters in the matrix. Do the following according to the position of these characters:

   - If both characters are in the same row, use the next left character as plain text. If the cipher text character is the last element of that row, take the first character of that row as plain text.
   - If both characters are in the same column, use the next upper character as plain text. If the cipher text character is the last element of that column, take the first character of that column as plain text.
   - If the characters are in different rows and columns, form a rectangle with these two characters at corners. Now, use the other row corner as the plain text for both characters.

Example 1:

Secret Key: PLAYFAIR EXAMPLE
Cipher Text: BMOD

Since the key is same as the encryption example, the matrix will be same as before.
Now, breaking the cipher text into groups of 2 will give us:

Cipher Text: BM OD

For the first group, the characters B and M form a rectangle shown by red colour in Figure 1. Hence, the corners will be swapped.

Cipher Text: BM OD
Cipher Text: HI DE

For the second group, the characters O and D are in the same column. Hence, the next upper character will be the cipher text.

2.2 Hill Cipher

In Hill Cipher, the secret key is a $n \times n$ invertible matrix A.

$$\text{A} = (a_{ij})_{n \times n} \rightarrow \text{Secret Key}$$

where $a_{ij} \in Z_{26}$.

$$M = m_1 m_2 ....m_n \rightarrow \text{Plain Text} \in \text{Z}_{26}^n$$

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:

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

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

2.3 Substitution Cipher

Substitution Cipher is a substitution from the set of alphabets to itself. The substitution need not be a bijection.

$$\text{S: \{A,B.....,Z\}} \rightarrow \text{\{A,B.....,Z\}}$$
$$\text{Cipher Text} = \text{S(Plain Text)}$$

S is the secret key. The number of secret keys possible is:

$$\#S = 26^{26} \approx 2^{122}$$

If the substitution box S is constrained to be a bijection, then the number of possible keys will be:

$$\#S = 26!$$

Brute Force Attack or Exhaustive Search is a way of finding the plain text if cipher text is known by exhaustively using all the possible keys.

3 Kerkchoff's Principle

Kerckhoffs's principle of cryptography was stated by Dutch-born cryptographer Auguste Kerckhoffs in the 19th century. The principle holds that a cryptographic system should be designed to be secure, even if everything about the system, except the key, is publicly known.

- The algorithms are designed for a whole lot of people, hence, if the design is not public, then people will not be able to use the algorithm.
- Another thing is that if one is not sharing their algorithm publicly, then there are more chances of having security weaknesses in the algorithm. Every person can interpret and cryptanlayse the algorithm differently to find security flaws. These flaws can be mitigated to make the algorithm more secure.

4 Shannon's Notion of Perfect Secrecy

Let E be the encryption algorithm, M be the plain text and C be the cipher text. Hence,

$$\text{C} = \text{E(M)}$$

Perfect Secrecy is the notion that given an encrypted message from a perfectly secure encryption system, absolutely nothing will be revealed about the unencrypted message from the cipher text. No information means not even a single bit of message or a function on any single bit of message is revealed.
For Example, suppose a single bit message is encrypted using some algorithm. The probability distribution of the message is given in Table 1. An algorithm will be perfectly secure if from the

Value	$P(n)$
0	0.9
1	0.1

Table 1: Probability Distribution.

encrypted message, the probability of guessing the plain text as 0 is not $(0.9 + \epsilon)$ where $\epsilon > 0$.
The notion of perfect secrecy can be represented mathematically as:

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

There are certain algorithms which provide perfect secrecy following some limitations. However, with these limitations these algorithms are impractical and cannot be used. One such algorithm is One Time Padding(OTP).

5 Symmetric Key Cipher

It has one secret key for both encryption and decryption functions. Since the key for both encryption and decryption is same, no one else should know about the key other than the sender and the receiver. There are two different types of Symmetric Key Ciphers:

1. Block Cipher
2. Stream Cipher

5.1 Block Cipher

The message to be encrypted is divided into fixed length blocks and these blocks are encrypted, that is, the encryption is done in block-wise manner. For example, the DES algorithm can encrypt 64 bit messages.

$$M = m_o||m_1||....||m_n$$
$$\text{length of block} = \text{l}$$
$$m_0, m_1, ..., m_n \text{ are blocks of length l}$$

One way of encrypting block-wise is to encrypt each block and then concatenate each block in the same order.

$$M = m_0||m_1||....||m_n$$
$$C = Enc(m_0, K)||Enc(m_1, K)||....||Enc(m_n, K)$$
$$C = C_0||C_1||....||C_n$$

It is not necessary that these blocks will be concatenated just as they were in the original message. There can be some algorithm for mixing these blocks and generating the encrypted text.

All the block ciphers are designed using either Substituion Permutation Networks or Feistel Networks. The AES algorithm is based on Substituion Permutation Networks and the DES algorithm is based on Fiestel Networks.

5.2 Stream Cipher

The encryption in Stream Cipher is done bit-wise. The message is a bit stream represented as:

$$M = m_0m_1...m_l, \text{ where } m_i \in 0, 1$$

An algorithm will be used to generate another bit stream represented as:

$$Z = Z_0Z_1...Z_l, \text{ where } Z_i \in 0, 1$$

This bit stream can be the output of a function which takes key and message as input.
To generate the cipher text, XOR between bits of plain text and bits of Z will be performed.

$$C = (m_0 \oplus Z_0, m_1 \oplus Z_1, ..., m_l \oplus Z_l)$$

During decryption, the same bit stream Z will be generated using the function and XOR will be performed to get the original message. It should be noted that XOR is used here for the sake of example, it can be some other complex computation too.

5.3 Stream Ciphers Versus Block Ciphers

There are known algorithms which can 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. Suppose an algorithm that encrypts 64 bit blocks at a time. 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, practically the messages are not very large and there are very good software implementation of block ciphers. Hence, where there are small messages, block ciphers are preferred over stream ciphers.

Stream ciphers are used generally where the length of the message is not pre-determined or there can nothing be said about how long the message will become. 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 judged at the starting of the call.

The implementation of block ciphers in hardware 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.

6 Product Cipher

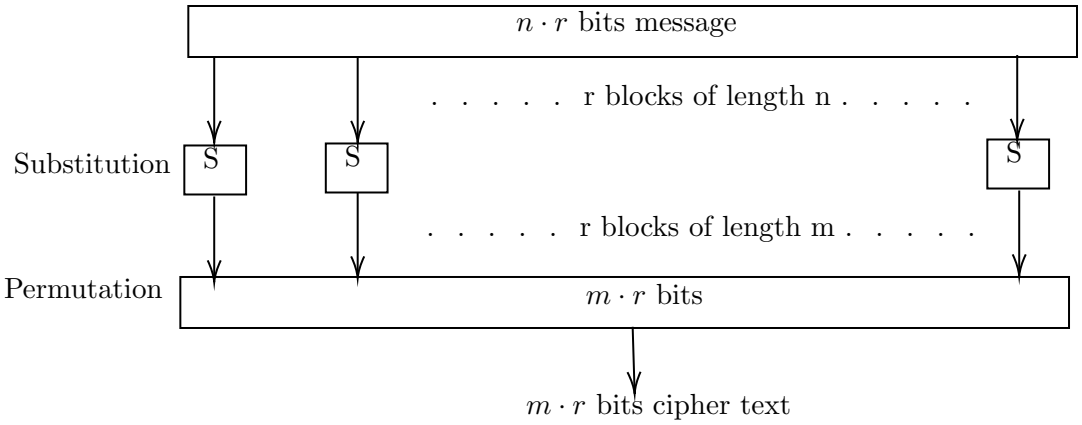
A Product Cipher combines two or more transformations in a manner intending that the resulting cipher is more secure than the individual transformations. A Product Cipher can be better than individual transformation in terms of both security and efficiency. One such cipher is a Substitution Permutation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

6.1 Substitution Permutation Network (SPN)

It is a product cipher based on a substitution box and a permutation box. Let's assume a message of length $n \cdot r$, a substituion S and a permutation P be defined as:

$$S : \{0,1\}^n \rightarrow \{0,1\}^m$$
$$P : \{0,1,...,m \cdot r - 1\} \rightarrow \{0,1,...,m \cdot r - 1\}$$

A SPN can be defined as performing 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.



7 Feistel Network

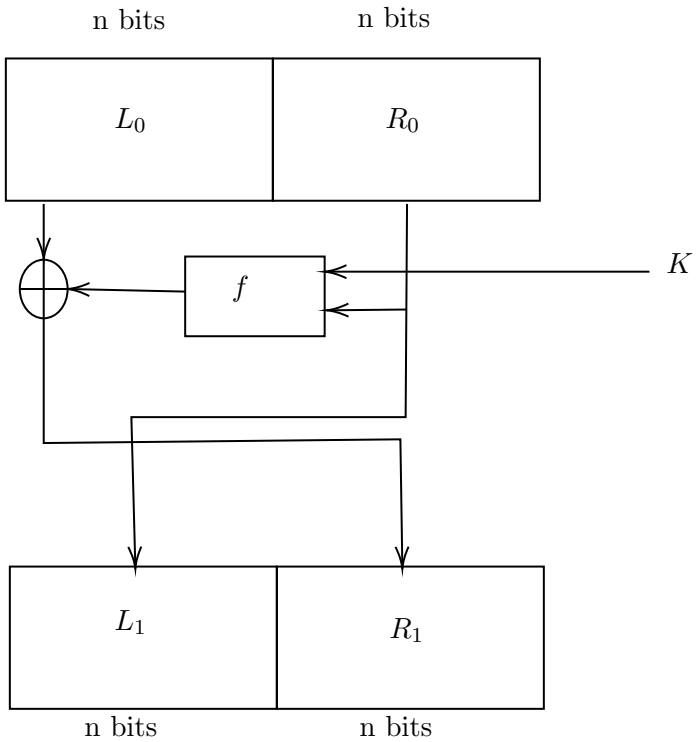
In a Feistel Network, let's say there is a message P of $2n$ bits. This message is divided into two equal parts L_0 and R_0.

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

L_0 and R_0 are of n bits each and can be calculated using right and left shifting of P by n bits respectively.
Let K be the secret key and the length of the secret key be 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:



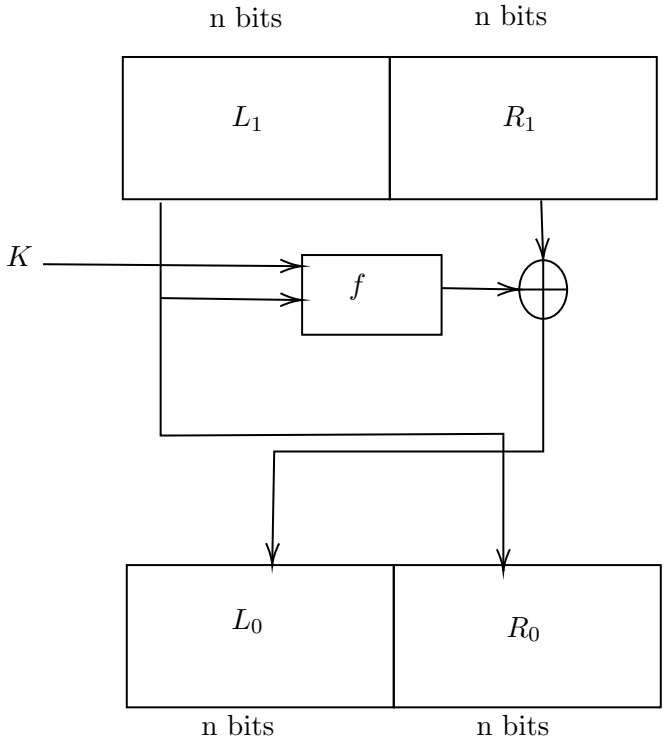
From the diagram, it is clear that:

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

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. The diagram for decryption is given below:



As it can be seen that the inverse of f is not required in the decryption process. This means that it does not matter whether f is invertible or not. In Fiestel networks, the round function may or may not be invertible.

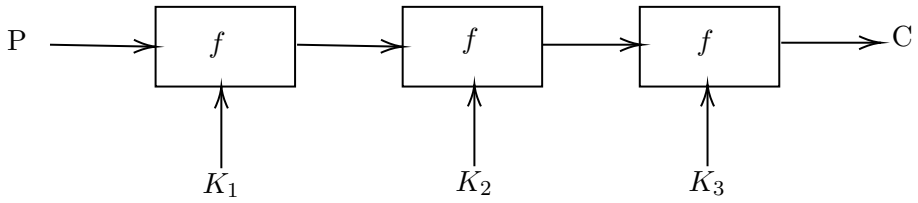
It can be seen that one half of the plain text remains similar in the cipher text. However, only one round of implementation is shown. In general, there are multiple rounds and the whole message can be encrypted in later rounds.

8 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 includes **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. A 3-round block cipher is shown below for example.

$f \rightarrow$ Round Function
$P \rightarrow$ Plain Text Block
$K \rightarrow$ Secret Key



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

The function G, knows as **Key Scheduling Function** generates the round keys by taking the secret key as input.

9 One Time Padding

One Time Padding (OTP) provides perfect secrecy under some conditions.

Encryption:

P $\rightarrow$ Plain Text
K $\rightarrow$ Secret Key
$Enc(P, K) = P \oplus K = C$
$\oplus \rightarrow$ xor operation

Decryption:

$$Dec(C, K) = C \oplus K = P$$

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.
3. Key K is uniformly selected from the key space.

Stream Cipher is a generalisation of OTP which does not provide perfect secrecy.