**[CS304] Introduction to Cryptography and Network Security**

Course Instructor: Dr. Dibyendu Roy
Scribed by: Raghav Agiwal (202151124)

Winter 2023-2024
Lecture (Week 2)

# 1 Hill Cipher

Hill cipher is an encryption algorithm used in cryptography. It operates on blocks (groups of letters) of plaintext and converts them into ciphertext using a matrix technique. The main idea is to perform a mathematical operation on the text using a matrix as the encryption key. The Hill cipher is a substitution cipher in which each letter in the text is replaced with the corresponding number and a matrix is used to check the resulting number.
The Hill cipher is an important algorithm in which the message is divided into blocks of the same size, usually a pair of letters. Each block is then treated as a vector and divided by the matrix (key encryption matrix) to create the encrypted block. The resulting block of ciphertext can be decrypted using the inverse of the encryption key matrix. Here we use the key K expression as matrix.

$$K = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

Assuming that our plaintext is M and our ciphertext is C, we can do the encryption and decryption process as follows:

Encryption :

$$C = M \cdot K \,(\text{mod } 26)$$

Decryption :

$$M = C \cdot K^{-1} \,(\text{mod } 26)$$

In these formulas C represents the ciphertext matrix, M represents the plaintext matrix, and K is the key matrix. The mod 26 denotes the modulo operation to keep the values within the range of the alphabet (A-Z).

Let's use the word "RAGHAV" as the plaintext and encrypt it using the Hill Cipher with a 2x2 matrix key. We'll choose a key matrix for illustration:

**Encryption Example:**

**Key Matrix:**

$$K = \begin{bmatrix} 2 & 3 \\ 1 & 4 \end{bmatrix}$$

**Plaintext:**

$$P = \begin{bmatrix} 17 \\ 0 \\ 6 \\ 7 \\ 0 \\ 21 \end{bmatrix}$$

(Numerical equivalent of 'R', 'A', 'G', 'H', 'A', 'V')

**Encrypted Matrix:**

$$C = K \cdot P \bmod 26$$

$$C = \begin{bmatrix} 11 \\ 17 \\ 21 \\ 1 \\ 17 \\ 17 \end{bmatrix}$$

**Ciphertext:**

Converting the numerical values back to letters:

$$\text{Ciphertext} = \text{KQUMRA}$$

So, if we encrypt the plaintext 'RAGHAV' using the key matrix K, we get the ciphertext "KQUMRA" using the hill cipher.

## 1.1 Symmatric key types

### 1.1.1 Block Cipher

A block cipher is a basic cipher that works based on the size of a file, encrypting or decrypting each block independently. In case of symmetric key encryption, block ciphering uses the same key for encryption and decryption. The Hill cipher is essentially a symmetric block cipher, although it operates on matrices rather than large blocks.

### 1.1.2 Stream Cipher

Work on each digit (bit or group of bytes) of the plaintext simultaneously to create a ciphertext stream. Stream ciphers are often used to encrypt data continuously, bit by bit, or byte by byte, and often include pseudo-random key stream generators. The keystream is combined with the plaintext using a bitwise XOR function to produce the ciphertext stream.

## 1.2 Block Cipher in detail:

### 1.2.1 Key components of Block cipher:

**Plaintext:** The original data to be encrypted.
**Ciphertext:** The encrypted result produced by the block cipher.
**Key:** A secret key shared between the sender and receiver for encryption and decryption.
**Block Size:** The fixed-size blocks into which the plaintext is divided.
**Substitution-Permutation Network (SPN):** Core structure of many block ciphers, consisting of substitution (S-boxes) and permutation (P-boxes) operations

**Example** Let's assume our block size is 3 bits. Our sacred letter is "101" and its key is "110". We will use transformation and permutation operations in a simple way.
Plaintext: 101
Key: 110

## 1.3 Substitution-Permutation Network (SPN)

### 1.3.1 Substitution (S-box):

Replace each block of plaintext with a corresponding value from the S-box.
Substitute: 101 -¿ 011

### 1.3.2 Permutation (P-box):

Permute the bits of the substituted block. Permute: 011 -¿ 101

### 1.3.3 Key Mixing (XOR with Key):

XOR the permuted block with the key.
XOR with Key: 101 XOR 110 -¿ 011
**Final Result:** Ciphertext: 011

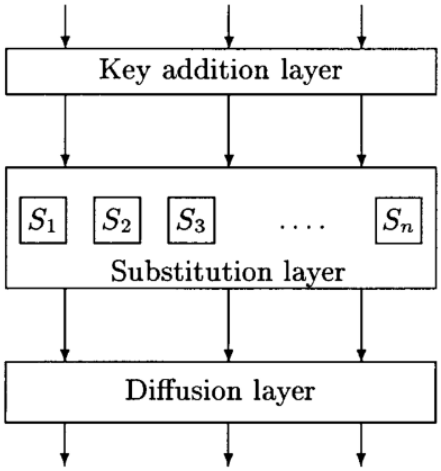


Figure 1:

The final result '011' is the ciphertext corresponding to the plaintext '101' using the key '110' and the specified S-box and P-box operations.

### 1.4 Feistel Network

Feistel networks have many functions where data is split into two parts. Each round applies an operation to one half of the data and uses the result to replace the other half. Then swap the halves and repeat the process for more matches. The final result is the ciphertext.

### Formula:

Let's denote the plaintext as $L_0R_0$, where $L_0$ is the left half and $R_0$ is the right half. After each round $i$, the Feistel function $F$ is applied to the right half, and the result is used to modify the left half. The halves are then swapped.

The Feistel Network structure can be represented as:

$$L_{i+1} = R_i$$
$$R_{i+1} = L_i \oplus F(R_i, K_i)$$

where:

- $L_{i+1}$ and $R_{i+1}$ are the left and right halves after round $i + 1$.
- $R_i$ and $L_i$ are the right and left halves after round $i$.
- $F$ is the Feistel function.
- $K_i$ is the round key for round $i$.
- $\oplus$ denotes the bitwise XOR operation.

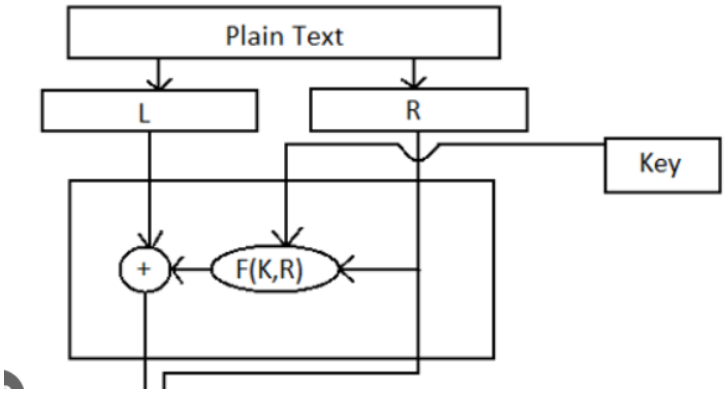


Figure 2:

### Encryption:

**Initialization:** Split the plaintext into left and right halves ($L_0$ and $R_0$).

**Rounds:** For each round $i$:

$$L_{i+1} = R_i$$
$$R_{i+1} = L_i \oplus F(R_i, K_i)$$

**Final Swap:** Swap $L_{\text{final}}$ and $R_{\text{final}}$ to get the final ciphertext.

### Decryption:

The decryption process is the same as encryption, but the round keys are used in reverse order. The process is as follows:

**Initialization:** Split the ciphertext into left and right halves ($L_{\text{final}}$ and $R_{\text{final}}$).

**Rounds (in reverse order):** For each round $i$:

$$R_i = L_{i+1}$$
$$L_i = R_{i+1} \oplus F(L_{i+1}, K_i)$$

**Final Swap:** Combine $L_0$ and $R_0$ to get the final decrypted plaintext.

## 1.5 Iterated Block Cipher:

Iterative block cipher is an important encryption algorithm that uses the block cipher multiple times in sequence (iterations) to increase security. Repeated application of the block cipher helps achieve higher levels of obfuscation and propagation, making it resistant to various cryptographic attacks.

There are many variations of a block cipher; each variation usually involves repetition of certain operations such as substitutions, substitutions, and key combinations.

### Formula:

Let's denote the plaintext as $P$, and the iterated block cipher process can be represented as:

$$C = E_K^n(P)$$

where:

- $C$ is the ciphertext.
- $E_K^n$ represents the application of the block cipher with key $K$ for $n$ rounds.
- $P$ is the plaintext.

### Encryption:

The encryption process involves applying the block cipher multiple times to the plaintext:

$$C = E_K^n(P)$$

### Decryption:

The decryption process is the reverse of encryption, applying the block cipher in the reverse direction (using the inverse operations or the decryption algorithm) for the same number of rounds:

$$P = D_K^n(C)$$

where:

- $D_K^n$ represents the decryption of the block cipher with key $K$ for $n$ rounds.

### Example:

Let's consider a simplified iterated block cipher with three rounds, denoted as $E_K^3$. The encryption process would involve applying the block cipher three times:

$$C = E_K^3(P) = E_K(E_K(E_K(P)))$$

And the decryption process would involve applying the block cipher in the reverse direction for three rounds:

$$P = D_K^3(C) = D_K(D_K(D_K(C)))$$

It's important to note that in practical applications, the number of rounds, the choice of block cipher, and the key schedule are crucial factors in determining the security of the iterated block cipher. Common examples of iterated block ciphers include the Advanced Encryption Standard (AES) and the Data Encryption Standard (DES).

## Data Encryption Standard (DES)

### Definition:

The Data Encryption Standard (DES) is a symmetric-key block cipher that was widely used as a standard for secure communication and data protection. Developed by IBM and adopted as a federal standard in the United States in the 1970s, DES was later replaced by the Advanced Encryption Standard (AES) due to concerns about its key size.

### Encryption Process:

**Initial Permutation (IP):**

$$IP(PT)$$

**Key Schedule:** The 64-bit key is expanded and used to generate 16 subkeys, one for each round.

**Rounds (16 rounds):** Each round consists of:

- **Expansion:** Expanding the 32-bit right half to 48 bits.
- **Substitution:** Substituting using S-boxes.
- **Permutation:** Permuting the 32-bit result.
- **XOR with Left Half:** XOR the result with the left half of the data.

The process is repeated for 16 rounds.

**Final Permutation (FP):**

$$FP(R_{16}L_{16})$$

### Decryption Process:

The decryption process is the reverse of encryption. The ciphertext undergoes the same processes, but the subkeys are used in reverse order.

**Initial Permutation (IP):**

$$IP(C)$$

**Key Schedule:** The same key schedule generates subkeys in reverse order.

**Rounds (16 rounds):** Each round's processes are applied in reverse order.

**Final Permutation (FP):**

$$FP(R_{16}L_{16})$$

### Key Generation (Key Schedule):

**Permutation Choice 1 (PC-1):**

$$PC-1(Key)$$

**Key Splitting:** The 56-bit key is split into two 28-bit halves.

**Round Key Generation (Shifts and Permutations):**

$$K_i = PC-2(Shift(PC-1(Key)))$$

### Formulas:

**Initial Permutation (IP):**

$$IP(PT)$$

**Final Permutation (FP):**

$$FP(R_{16}L_{16})$$

**Round Encryption (Example for Round i):**

$$R_{i+1} = L_i \oplus f(R_i, K_i)$$

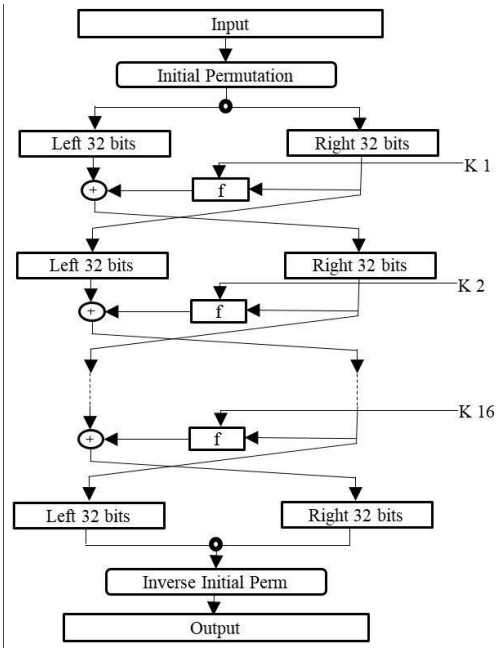


Figure 3:

$$L_{i+1} = R_i$$

where:

- $L_i$ and $R_i$ are the left and right halves of the data at round $i$.
- $K_i$ is the subkey for round $i$.
- $f$ is the round function combining expansion, substitution, permutation, and XOR operations.

**Round Key Generation:**

$$K_i = PC - 2(Shift(PC - 1(Key)))$$

DES is considered insecure by today's standards due to its small key size. For secure communication, it has been largely replaced by more advanced encryption algorithms like AES.