[CS304] Introduction to Cryptography and Network Security

Course Instructor: Dr. Dibyendu Roy
Scribed by: Sanidhya Kumar (202151138)

Winter 2023-2024
Lecture (Week 1)

1 Cryptology

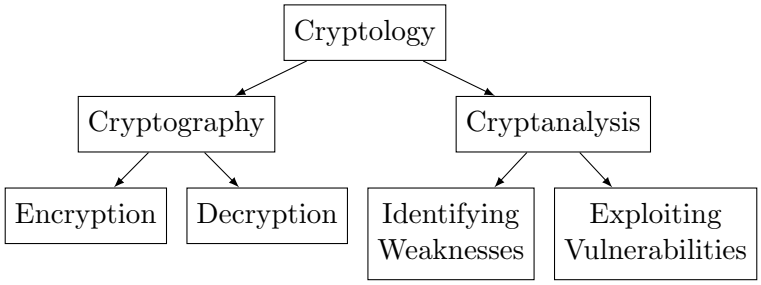

Figure 1: Cryptology Tree Structure

Cryptology is the overarching field that includes both Cryptography and Cryptanalysis. Hence, cryptology is defined as the study of cryptography as well as cryptanalysis. Now, Cryptography is essentially the process of hiding or encoding information so that only the intended recipient can read it. Cryptanalysis, on the other hand, is the process of finding weaknesses or vulnerabilities in cryptographic algorithms and exploiting them to decipher the encoded message.

2 NIST

NIST stands for the National Institute of Standards and Technology. NIST standardizes cryptographic algorithms. It also holds open competitions wherein participants can propose their cryptographic algorithms. These proposals undergo rigorous testing, and if they meet the standards, they are published to the real world.

3 An Example of Encryption

Suppose I have five ATM PINs that I want to encrypt for privacy. One approach could involve encoding each PIN using various rules, but this could complicate the decoding process due to the need to remember the specific encoding methods for each PIN. A straightforward encoding method involves adding a secret key X to all the ATM PINs. This secret key X is kept confidential. During decoding, I simply subtract X from all the encoded PINs to easily recover the original ATM PINs. This straightforward method simplifies the process and provides privacy to our ATM PINs.

$$\text{ATM1} \rightarrow \text{PIN1} + X = Y1$$
$$\text{ATM2} \rightarrow \text{PIN2} + X = Y2$$

Here,

- PIN1 is the original PIN.
- X is the secret key.
- $Y1$ is the ciphertext.
- The function PIN1 + X is known as the encryption function.

Formally, we can write the Encryption and Decryption functions as follows:

$$\textbf{Encryption:} \quad E(P, K) = C$$
$$\textbf{Decryption:} \quad D(C, K) = P$$

In this mathematical formulation:

- E is the encryption function.
- D is the decryption function.
- P is the plaintext.
- C is the ciphertext/encoded text.
- K is the secret key.

Kindly note that we will be using the above notations extensively throughout this scribe.

4 Types of Cryptography

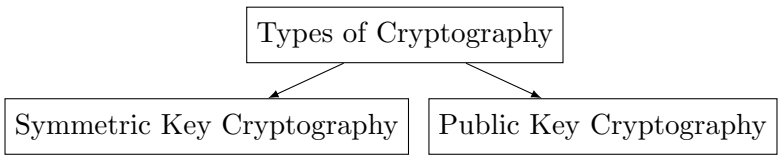

Figure 2: Types of Cryptography Tree Structure

4.1 Symmetric Key Cryptography

In this type, we use only one secret key for both encryption and decryption. Formally,

$$E(P, k) = C$$
$$D(C, k) = P$$

4.2 Public Key Cryptography

In this type, we use two keys: one is known as the public key (e_k), and the other is known as the secret key (d_k). Generally, the public key is used for encryption, and the secret key is used for decryption. Formally,

$$E(P, e_k) = C$$
$$D(C, d_k) = P$$

Note: The notations have their usual meanings as defined in the earlier sections.

5 Functionalities of Cryptography

A good cryptography method is expected to follow the following functionalities:

1. **Confidentiality** – Secrecy must be maintained throughout. This ensures that unauthorized parties cannot access the contents of the message.
2. **Integrity** – The message should be delivered from the sender to the receiver as a whole and should not be tampered with in between. This guarantees that the message remains unchanged during transmission.
3. **Authentication** – Verification must be in place to ensure that the user and the message are authentic and are coming from the user who sent them. This prevents unauthorized entities from posing as legitimate users.
4. **Non-repudiation** – This process ensures that you cannot deny that a message is not coming from you if you have really sent it. It provides evidence that the sender indeed sent the message and cannot later deny their involvement.

Note regarding mapping from Plaintext to Ciphertext and vice-versa

Encryption: $P \times \text{Encryption} \rightarrow C$
Decryption: $C \times \text{Decryption} \rightarrow P$
$P \times$ Encryption and $C \times$ Decryption are respective domains. Encryption transforms plaintext (P) into ciphertext (C), while decryption reverses this process.

6 Types of Functions

A function $f : A \rightarrow B$ is a relation between sets A and B if $a, b \in A$ and if $a = b$, then $f(a) = f(b)$. Here, the relation is a subset of $A \times B$. Let us understand a few types of functions:

1. **One-to-One functions:** Simply, $f(a) = f(b)$ if and only if $a = b$. These functions are injective, ensuring that each element in A maps to a distinct element in B.
2. **Onto functions:** If $f : A \rightarrow B$, then for all $b \in B$, there exists $a \in A$ such that $f(a) = b$. Onto functions are surjective, covering the entire set B with their mappings.
3. **Bijective functions:** Functions that are both one-to-one and onto. These functions are injective and surjective, meaning each element in A maps to a distinct element in B, and the entire set B is covered.
4. **Permutation:** Let π be a permutation on set S, then $\pi : S \rightarrow S$ is a bijection from S to S. Permutations represent rearrangements of elements in a set.

$$\pi : \begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 3 & 1 & 4 \end{bmatrix}$$

5. **One-way functions:** These functions have an inverse, but the computation for finding the inverse is computationally heavy, taking decades to calculate. For example, consider a function that takes $p_1, p_2, p_3, \ldots, p_n$ and multiplies all of them together to form N. Computing this is relatively easy. However, given N, finding $p_1, p_2, \ldots, p_n$ such that their product equals N is extremely challenging, especially for large n of the order of some thousands. Therefore, the described function is an example of a one-way function.

Substitution Box/S-Box

A Substitution Box (S-Box) is defined as $S : A \to B$ with $|B| \leq |A|$, meaning one mapping will always repeat.

$$S : \{1, 2, 3, 4\} \to \{1, 2, 3\}$$

Here, let's say 1 maps to 1, 2 maps to 2, 3 maps to 3, and 4 maps again to 1. Hence, it cannot be one-to-one. S-Boxes are often used in cryptographic algorithms.n.

7 Various Kinds of Cipher (Symmetric Key Ciphers)

7.1 Caesar Cipher

Caesar cipher involves shifting every alphabet by three characters. We can shift it by any other number as well but in Caesar cipher this agreed number is 3 and is fixed. All the alphabets are mapped sequentially from 0 to 25 as follows:

$$A \to 0, B \to 1, \ldots, Z \to 25$$

Mathematically,

$$E(X, 3) = (X + 3) \mod 26$$
$$D(C, 3) = (C + 26 - 3) \mod 26 = (C + 23) \mod 26$$

It is interesting to note that in modulus 26 system, 23 is the additive inverse of 3.
Example:

SANIDHYA

$$S \to 18, A \to 0, N \to 13, I \to 8, D \to 3,$$
$$H \to 7, Y \to 24, A \to 0$$

Encrypted message: $21, 3, 16, 11, 6, 10, 27, 3$ (VDQLGKBD)
To decrypt, perform $(C + 23) \mod 26$, which gives $18, 0, 13, 8, 3, 7, 24, 0$, which is SANIDHYA.

7.2 Transposition Cipher

The message M is written into components $m_1 m_2 m_3 \ldots m_t$. We define a permutation C which is a permutation on t elements. For encryption, choose a permutation which can be given by:

$$\text{Encryption: } C = m_{e(1)} m_{e(2)} \ldots m_{e(t)}$$

For decryption, choose a permutation such that it reverses the effect of encryption permutation:

$$\text{Decryption: } M = C_{e^{-1}(1)} C_{e^{-1}(2)} \ldots C_{e^{-1}(t)}$$

Example: Let M be $CAESAR$ and secret key e be a mapping $[1, 2, 3, 4, 5, 6] \to [6, 4, 1, 3, 5, 2]$. Using this secret key, we encode as follows:

$$\text{M: CAESAR} \to \text{C: RSCEAA}$$

To decode, use the inverse e which is a mapping $[1, 2, 3, 4, 5, 6] \to [3, 6, 4, 2, 5, 1]$.Using this secret key, we decode:

$$\text{C: RSCEAA} \to \text{M: CAESAR}$$

7.3 Substitution Cipher

A substitution cipher replaces each letter in the plaintext with another letter based on a predetermined mapping, providing a simple form of encryption. This means that in this cipher, we simply substitute one alphabet with another alphabet:

$$A' = \{A, B, C, \ldots, Z\}$$

$$e : \text{substitution from } A' \text{ to } A'$$

$$C = e(m_1)e(m_2)\ldots e(m_t)$$

Example: Let $e(A) = Z, e(B) = D, e(C) = A$. Hence, using this cipher, the message ABC can be encoded as ZDA. To decrypt, perform the back mapping to get the plaintext.

7.4 Affine Cipher

Affine cipher uses the below mapping from alphabets to numbers and then performs some mathematical function to encode those numbers. The alphabets A to Z are mapped from 0 to 25 as,

$$A \to 0, B \to 1, \ldots, Z \to 25$$

We define the following,

$$A' = \text{Set of all alphabets}$$

$$\mathbb{Z}_{26} = \{0, 1, \ldots, 25\}$$

$$A' \to \mathbb{Z}_{26} \text{ with the mapping as defined above}$$

$$k = \text{secret key which is a tuple } (a, b) \in \mathbb{Z}_{26} \times \mathbb{Z}_{26}$$

$$x = \text{plaintext and } x \in \mathbb{Z}_{26}$$

For encryption we use the following function –

$$\text{Encryption: } e(x, k) = (ax + b) \mod 26 = c$$

For decryption:

$$\text{Decryption: } d(c, k) = ((c–b)a^{-1}) \mod 26$$

Here a^{-1} is such that $a \cdot a^{-1} \equiv 1 \mod 26$. We will be able to use affine cipher if and only if we find some a whose a^{-1} exists given the condition that $\gcd(a, 26) = 1$.

7.4.1 Multiplicative Inverse

Let y be the multiplicative inverse of x modulo m. Hence,

$$x \cdot y \equiv 1 \mod m$$

Extended Euclidean Algorithm is used to find the GCD of two integers. Let us first compute GCD of $x = 3$ and $y = 17$ using Euclid's Division Algorithm:

$$17 = 3 \cdot 5 + 2$$

$$3 = 2 \cdot 1 + 1$$

$$2 = 1 \cdot 2 + 0$$

Hence, $\text{GCD}(3, 17) = 1$. Now, going in reverse direction of this will lead us to the values of a and b.

$$1 = 1 \cdot 3 - 1 \cdot 2$$

$$1 = 1 \cdot 3 - 1 \cdot (1 \cdot 17 - 5 \cdot 3)$$

$$1 = 6 \cdot 3 - 1 \cdot 17$$

Hence, $a = 6$ and $b = -1$.

7.4.2 Euler Totient Function ($\phi(n)$)

This function is used to find the number of positive integers which are relatively prime to n and also smaller than n. Mathematically,

$$\phi(n) = \begin{cases} (p-1)(q-1) & \text{if } n \text{ is non-prime where } p \text{ and } q \text{ are co-prime factors of } n \\ (p-1) & \text{if } p \text{ is prime} \end{cases}$$

In our case where $n = 26$ (since 26 alphabets are in the English alphabet), we get $\phi(n) = (2 - 1)(13 - 1) = 12$. Therefore, the possible values for a in the key are 12 out of 26, and there are 26 possible values for b. Consequently, there is a total of $12 \cdot 26 = 312$ keys possible for the Affine Cipher. This limitation arises from the requirement for a to have a multiplicative inverse modulo 26.

7.5 Playfair Cipher

This cipher consists of a 5×5 matrix constructed using the secret key chosen and the alphabets of the English alphabet. Since there are 26 alphabets in the English language and this cipher involves the use of 25 (a 5×5 matrix), it is assumed in this cipher that $i = j$. It would be easier to understand this cipher using an example.

7.5.1 Matrix Construction

Let's consider our secret key as "PLAYFAIREXAMPLE." We will construct a 5×5 matrix by entering alphabets in a row-wise fashion ((0,0), (0,1), (0,2), (0,3), (0,4), (1,0), (1,1),...,(4,4)). We will start by entering all the unique alphabets from our secret key into the matrix without repetition and then fill the remaining English alphabets in the matrix. Note that $i = j$ during this process.

For the secret key "PLAYFAIREXAMPLE," the matrix would be as follows:

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

7.5.2 Encryption and Decryption Process

Now, let's understand how encryption and decryption are done using the Playfair cipher. We will break our plaintext into blocks of two and then process each block.

Example If the message is HIDE, we will break it into two blocks: HI and DE. Now consider the first block HI, where H and I form a rectangle in the matrix. We will replace each letter with the opposite corner letter of the same row. So, H becomes B, and I becomes M. Similarly, consider the second block DE, where D and E form a column in the matrix. We will replace each letter with the letter just below it, wrapping to the top if needed. So, D becomes O, and E becomes D. Hence, we successfully encoded HIDE as BMOD.

H	I
D	E

$\Rightarrow$

B	M
O	D

To decode, we will do the opposite of what we did while encoding. Using the ciphertext BMOD, we break it into two blocks: BM and OD. For the first block BM, the letters form a rectangle in the matrix. We replace each letter with the opposite corner letter of the same row. So, B becomes H, and M becomes I. Similarly, for the second block OD, the letters form a column. We replace each letter with the letter just above it, wrapping to the bottom if needed. Therefore, O becomes D, and D becomes E. Hence, we successfully decoded BMOD back to HIDE.

B	M
O	D

$\Rightarrow$

H	I
D	E

7.5.3 Rules for Playfair Cipher

The following conditions are to be followed in the Playfair cipher:

To Encode

1. If letters of a block form a rectangle: Replace the letters with letters in the opposite corner of the same row.
2. If letters of a block form a row: Replace the letters with the letters just right to it and wrap it to the left if needed.
3. If letters of a block form a column: Replace the letters with the letters just below it and wrap it to the top if needed.

To Decode

1. To decode (1): follow (1).
2. To decode (2): Replace the letters with the letters just left to it and wrap it to the right if needed.
3. To decode (3): Replace the letters with the letters just above it and wrap it to the bottom if needed.

Handling Odd-Length and Repetition

To ensure a smooth encryption process and accommodate odd-length and repeated characters in the Playfair cipher, we follow these guidelines:

1. For even-length texts, divide them into blocks of 2 as usual.

2. For odd-length texts, add an X to the end of the plaintext to make it even length before dividing it into blocks.

3. Address repetition by adding an extra X after every repeated alphabet in the plaintext.

For example, with the plaintext "BALL," we add an X after the first L, resulting in the blocks BA, LX, LX. The encoding process is then performed as defined above.

7.5.4 Issues with Playfair Cipher

There are various issues with the Playfair cipher. For example, HIDE is encoded as BMOD using the secret key PLAYFAIREXAMPLE, and while decoding, we will get two options: HIDE and HJDE because we assumed $i = j$. So, we will have to manually check which of the possible combinations is correct.