



CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

Mustansiriyah University

College of Science – Department of CS/Cybersecurity

Cryptography Course

2023-2024

BMN

By

Prof. DR. Bashar AL-Esawi



bashar_sh77@uomustansiriyah.edu.iq

Page 1 of 51





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

Definition of cryptography:

Cryptography is the practice of securing communication from third-party eavesdropping by converting it into a coded language, using mathematical algorithms and secret keys, to protect the *confidentiality, integrity, and availability* of the information being transmitted.

Here are some common terms used in cryptography:

- **Plaintext:** The original message or data that is intended to be kept secret.
- **Encryption:** A method used to transform plaintext into ciphertext.
- **Ciphertext:** The encrypted message or data that has been transformed using a cipher.
- **Key:** A secret value used to encrypt or decrypt messages.
- **Encryption:** The process of transforming plaintext into ciphertext using a cipher and a key.
- **Decryption:** The process of transforming ciphertext back into plaintext using a cipher and a key.
- **Cryptosystem:** Combination of a cipher, key, and any related algorithms or protocols used to secure data.
- **Cryptanalysis:** The study of techniques used to break or weaken cryptographic systems.
- **Brute force attack:** A cryptanalysis technique where all possible keys are tried to decrypt a message.
- **Protocol:** Set of rules and procedures used to secure communication between 2 or more parties.

What is the role of cryptography in cybersecurity?

Cryptography plays a crucial role in cybersecurity by providing a means to protect sensitive information and communications from unauthorized access, interception, or modification. Here are some key roles of cryptography in cybersecurity:

1. **Confidentiality:** Cryptography helps ensure that data is kept secret and protected from unauthorized disclosure.
2. **Integrity:** Cryptography helps ensure that data is not tampered with or modified in transit, providing data integrity and trustworthiness.
3. **Authentication:** Cryptography helps establish the identity of the communicating parties, ensuring that data is exchanged only between trusted parties.
4. **Non-repudiation:** Cryptography helps ensure that parties cannot deny their involvement in the exchange of information or transactions, providing proof of the exchange.

Overall, cryptography provides the foundation for many cybersecurity technologies, including secure communication protocols, digital signatures, and data encryption, that protect sensitive information and critical systems from cyber threats.





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

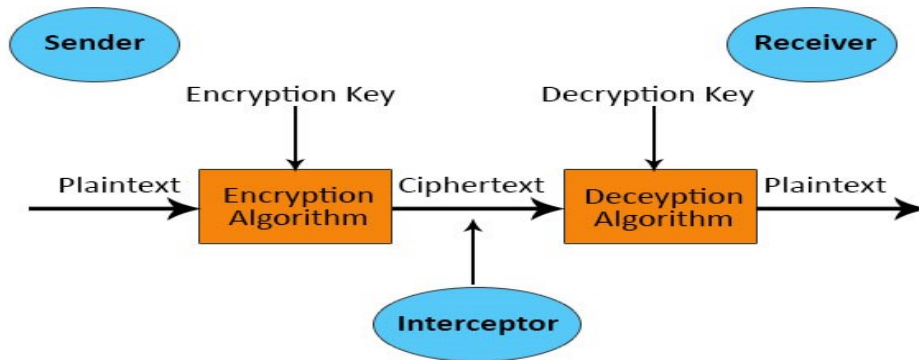


Figure (1): The Cryptosystem Architecture.

Description of Confidentiality, Integrity, and Authenticity (CIA):

Confidentiality, Integrity, and Authenticity (CIA) are three important principles that form the basis of information security.

Confidentiality refers to the protection of sensitive information from unauthorized access or disclosure. It involves ensuring that only authorized users have access to the information and that the information is protected from interception or eavesdropping during transmission.

Integrity refers to the protection of the accuracy and completeness of data and information. It involves ensuring that data is not altered or destroyed in an unauthorized or unintended manner and that the data remains consistent and accurate over time.

Availability refers to the assurance that data or information is genuine and can be trusted. It involves verifying the identity of the user or the source of the information and ensuring that the data has not been tampered with or modified.

Together, these principles form the basis of a secure and trustworthy information system that protects against unauthorized access, malicious attacks, and unintentional errors. They are critical to maintaining the confidentiality, accuracy, and reliability of sensitive information and are essential for maintaining the trust of users and stakeholders.





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

Information security, Data security, Network security, and Cyber security:

Information security, data security, network security, and cyber security are all related but distinct fields within the broader realm of computer security.

Information security refers to the protection of information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction. The goal of information security is to ensure confidentiality, integrity, and availability of information by applying a risk management process and giving assurance that information security requirements are met.

Data security, on the other hand, is the practice of protecting data, both in storage and in transit, from unauthorized access or alteration. This involves implementing various technologies, processes, and policies to secure sensitive data and prevent data breaches.

Network security focuses on protecting the infrastructure of computer networks, including the devices, protocols, and data that traverse them. The aim of network security is to prevent unauthorized access, use, disclosure, disruption, modification, or destruction of network resources and to ensure the confidentiality, integrity, and availability of data transmitted over the network.

Cyber security, also known as computer security, refers to the protection of internet-connected systems, including hardware, software, and data, from attack, damage, or unauthorized access. This field encompasses a wide range of technologies, processes, and practices aimed at safeguarding computer systems and networks from cyber threats such as malware, ransomware, and hacking.

Each of these fields is critical for protecting the security and privacy of information and data in the digital age, and they often overlap and intersect with one another, See Figure (2).

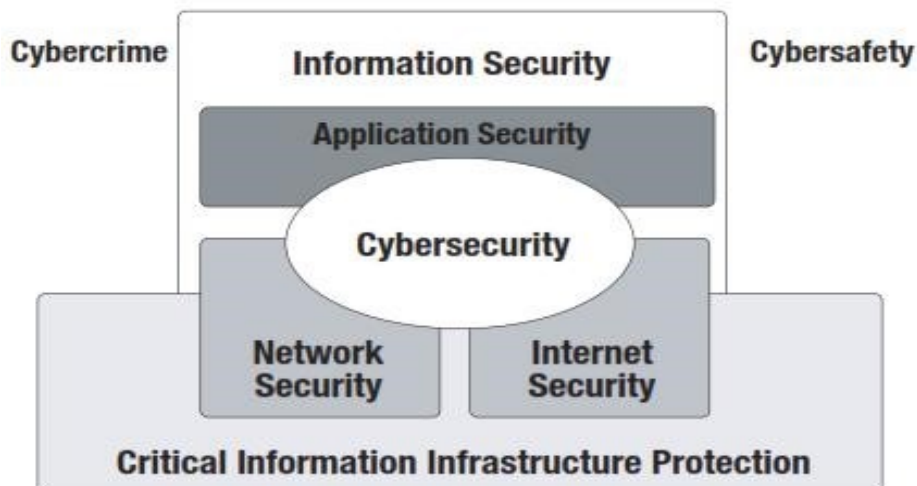


Figure (2): Critical Information Infrastructure Protection.





CRYPTOGRAPHY LECTURES

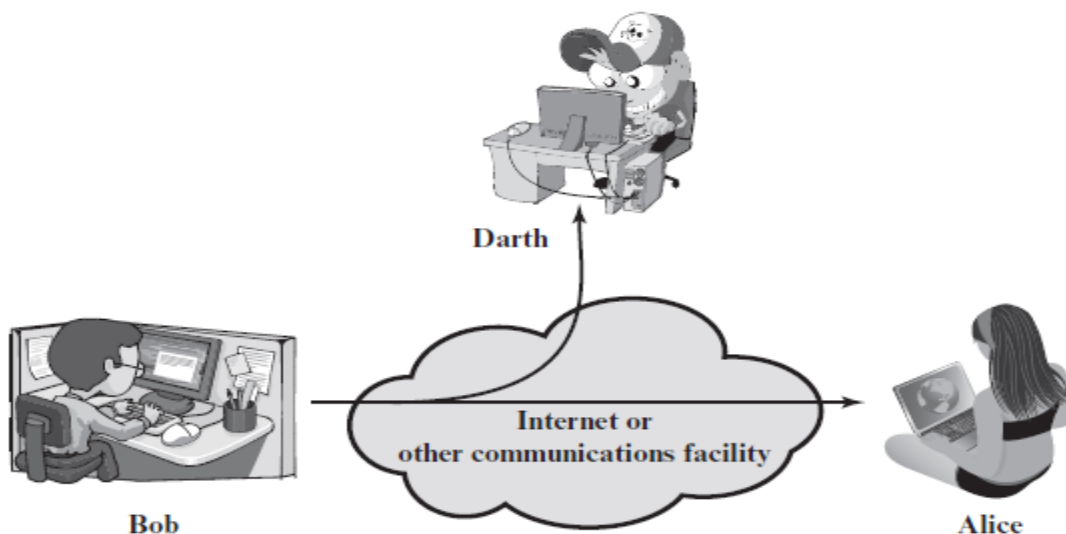
PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

OSI Security Architecture:

the OSI security architecture provides a useful, if abstract, overview of many of the concepts that this book deals with. The OSI security architecture focuses on security attacks, mechanisms, and services. These can be defined briefly as

- **Security attack:** Any action that compromises the security of information owned by an organization.
- 1. **Passive Attacks:** Passive attacks are in the nature of eavesdropping on, or monitoring of, transmissions. The goal of the opponent is to obtain information that is being transmitted. Passive attacks are very difficult to detect, because they do not involve any alteration of the data. Two types of passive attacks are:
 - a. **Release of message contents:** A telephone conversation, an electronic mail message, and a transferred file may contain sensitive or confidential information. *We would like to prevent an opponent from learning the contents of these transmissions.*
 - b. **Traffic analysis:** Suppose that we had a way of masking the contents of messages or other information traffic so that opponents, even if they captured the message, could not extract the information from the message. *The common technique for masking contents is encryption. If we had encryption protection in place, an opponent might still be able to observe the pattern of these messages. The opponent could determine the location and identity of communicating hosts and could observe the frequency and length of messages being exchanged.*
- 2. **Active Attacks:** Active attacks involve some modification of the data stream or the creation of a false stream and can be subdivided into four categories: Masquerade, replay, Modification of messages, and Denial of service.



(a) Passive attacks

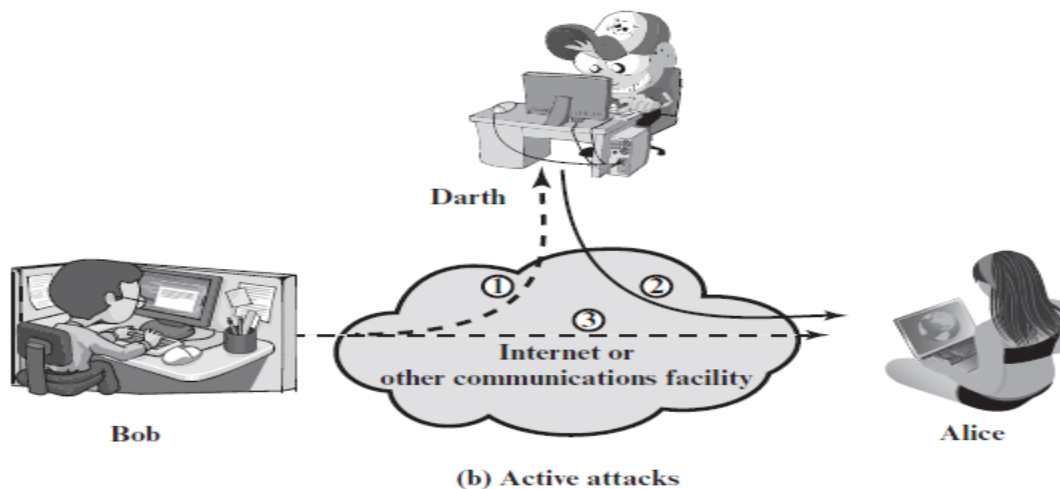




CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch



- A **Masquerade** takes place when one entity pretends to be a different entity (path 2 of Figure b is active).
- **Replay** involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect (paths 1, 2, and 3 active).
- **Modification** of messages simply means that some portion of a legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorized effect (paths 1 and 2 active).
- **Denial of service** prevents or inhibits the normal use or management of communications facilities (path 3 active). This attack may have a specific target; for example, an entity may suppress all messages directed to a particular destination (e.g., the security audit service). Another form of service denial is the disruption of an entire network, either by disabling the network or by overloading it with messages so as to degrade performance.

■ **Security Service:** a service that is provided by a protocol layer of communicating open systems and that ensures adequate security of the systems or of data transfers. A processing or communication service that enhances the security of the data processing systems and the information transfers of an organization. The services are intended to counter security attacks, and they make use of one or more security mechanisms to provide the service. These services can be described as follow:

AUTHENTICATION,

ACCESS CONTROL,

DATA CONFIDENTIALITY,

DATA INTEGRITY,

NONREPUDIATION,

AVAILABILITY SERVICE



bashar_sh77@uomustansiriyah.edu.iq





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

In the literature, the terms threat and attack are commonly used to mean more or less the same thing.

Threat

A potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm. That is, a threat is a possible danger that might exploit a vulnerability.

Attack

An assault on system security that derives from an intelligent threat; that is, an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the security policy of a system.

■ **Security mechanism:** A process (or a device incorporating such a process) that is designed to detect, prevent, or recover from a security attack.

SPECIFIC SECURITY MECHANISMS

May be incorporated into the appropriate protocol layer in order to provide some of the OSI security services.

Encipherment

The use of mathematical algorithms to transform data into a form that is not readily intelligible. The transformation and subsequent recovery of the data depend on an algorithm and zero or more encryption keys.

Digital Signature

Data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery (e.g., by the recipient).

Access Control

A variety of mechanisms that enforce access rights to resources.

Data Integrity

A variety of mechanisms used to assure the integrity of a data unit or stream of data units.

PERVASIVE SECURITY MECHANISMS

Mechanisms that are not specific to any particular OSI security service or protocol layer.

Trusted Functionality

That which is perceived to be correct with respect to some criteria (e.g., as established by a security policy).

Security Label

The marking bound to a resource (which may be a data unit) that names or designates the security attributes of that resource.

Event Detection

Detection of security-relevant events.

Security Audit Trail

Data collected and potentially used to facilitate a security audit, which is an independent review and examination of system records and activities.

Security Recovery

Deals with requests from mechanisms, such as event handling and management functions, and takes recovery actions.





Basic Number Theory

DIVISIBILITY AND THE DIVISION ALGORITHM:

Divisibility is a mathematical concept that describes the ability of one number to be divided exactly by another number without leaving a remainder. In other words, if one number is divisible by another number, it means that the first number is a multiple of the second number.

For example, 15 is divisible by 3 because 15 can be divided exactly by 3, which gives a quotient of 5 and a remainder of 0. Similarly, 10 is divisible by 5 because 10 can be divided exactly by 5, which gives a quotient of 2 and a remainder of 0.

The Division Algorithm is a method used to find the quotient and remainder when one integer (the dividend) is divided by another integer (the divisor). The algorithm states that if a and b are any two integers, with b being non-zero, then there exist unique integers q (the quotient) and r (the remainder).

THE FOLLOWING SNIP OF CODE IN C#:

```
using System;
class Program
{
    static void Main (string [ ] args)
    {
        int a = 15; // the dividend
        int b = 3; // the divisor
        if (a % b == 0)
            Console.WriteLine("{0} is divisible by {1}", a, b);
        else
            Console.WriteLine("{0} is not divisible by {1}", a, b);
        int q = a / b; // the quotient
        int r = a % b; // the remainder
        Console.WriteLine("{0} = {1}{2} + {3}", a, b, q, r);
    }
}
```





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

Greatest Common Divisor:

The Greatest Common Divisor (GCD) is the largest positive integer that divides two or more numbers without leaving a remainder. It is also known as the Highest Common Factor (HCF). The GCD of two or more integers can be found by identifying the common factors of the integers and selecting the largest one.

For example, consider the numbers 24 and 36. The factors of 24 are 1, 2, 3, 4, 6, 8, 12, and 24, while the factors of 36 are 1, 2, 3, 4, 6, 9, 12, 18, and 36. The common factors of 24 and 36 are 1, 2, 3, 4, 6, and 12. Therefore, the GCD of 24 and 36 is 12.

Another example is the numbers 18, 24, and 36. The factors of 18 are 1, 2, 3, 6, 9, and 18, the factors of 24 are 1, 2, 3, 4, 6, 8, 12, and 24, and the factors of 36 are 1, 2, 3, 4, 6, 9, 12, 18, and 36. The common factors of 18, 24, and 36 are 1, 2, 3, 6, and 9. Therefore, the GCD of 18, 24, and 36 is 3.

THE FOLLOWING SNIP OF CODE IN C#:

```
using System;
class Program
{
    static void Main(string[] args)
    {
        int a = 24;
        int b = 36;
        int gcd = FindGCD(a, b);
        Console.WriteLine("The GCD of {0} and {1} is {2}", a, b, gcd);
    }
    static int FindGCD(int a, int b)
    {
        while (b != 0)
        {
            int temp = b;
            b = a % b;
            a = temp;
        }
        return a;
    }
}
```

Because we require that the greatest common divisor be positive, $\text{gcd}(a, b) = \text{gcd}(a, -b) = \text{gcd}(-a, b) = \text{gcd}(-a, -b)$. In general, $\text{gcd}(a, b) = \text{gcd}(|a|, |b|)$.

$$\text{gcd}(60, 24) = \text{gcd}(60, -24) = 12$$





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

MODULAR ARITHMETIC:

If a is an integer and n is a positive integer, we define $a \bmod n$ to be the remainder when a is divided by n . The integer n is called the modulus.

$$11 \bmod 7 = 4; \quad -11 \bmod 7 = 3$$

To calculate $-23 \bmod 13$, we can first calculate the remainder of the division of 23 by 13, which is 10. Since the dividend is negative, we then subtract the divisor from the remainder and obtain:

$$-23 \bmod 13 = 10 - 13 = -3, \text{ Therefore, } -23 \bmod 13 \text{ is } -3.$$

Another Examples:

$$\begin{aligned} 11 \bmod 8 &= 3; 15 \bmod 8 = 7 \\ [(11 \bmod 8) + (15 \bmod 8)] \bmod 8 &= 10 \bmod 8 = 2 \\ (11 + 15) \bmod 8 &= 26 \bmod 8 = 2 \\ [(11 \bmod 8) - (15 \bmod 8)] \bmod 8 &= -4 \bmod 8 = 4 \\ (11 - 15) \bmod 8 &= -4 \bmod 8 = 4 \\ [(11 \bmod 8) \times (15 \bmod 8)] \bmod 8 &= 21 \bmod 8 = 5 \\ (11 \times 15) \bmod 8 &= 165 \bmod 8 = 5 \end{aligned}$$

One common method is the modular exponentiation algorithm, which is designed specifically for this purpose.

The modular exponentiation algorithm works by repeatedly squaring the base and taking the remainder modulo the modulus, which reduces the number of multiplications needed. Here's how the algorithm works, **using your example of calculating the remainder of 11^7 modulo 5:**

Step 1: Convert the exponent to binary form. In this case, 7 in binary is 111.

Step 2: Initialize a variable to hold the **result**, set it to 1.

Step 3: For each bit in the binary form of the exponent, from right to left:

a) **Square the result.**

b) If the current bit is 1, multiply the result by the base.

c) Take the remainder of the result modulo the modulus.





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

Here's how this works in practice:

```
Exponent: 7 (binary: 111)
Base: 11
Modulus: 5

Initialize result = 1

Starting with the rightmost bit of the exponent:
bit 1: 1 (multiply by base)
result = result * base % modulus = 1 * 11 % 5 = 1

Square the base, take remainder:
base = base^2 % modulus = 11^2 % 5 = 1

bit 2: 1 (multiply by base)
result = result * base % modulus = 1 * 1 % 5 = 1

Square the base, take remainder:
base = base^2 % modulus = 1^2 % 5 = 1

bit 3: 1 (multiply by base)
result = result * base % modulus = 1 * 1 % 5 = 1

The result is the final remainder:
result = 1

So, 11^7 MOD 5 = 1.
```





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

Solve $11^{13} \text{ mod } 53$ using the Successive Squaring Method

Step 1: Convert our power of 13 to binary notation:

Using our [binary calculator](#), we see that 13 in binary form is **1101**

The length of this binary term is 4, so this is how many steps we will take for our algorithm below

Step 2: Construct Successive Squaring Algorithm:

i	a	a ²	a ² mod p
0	11	11	11 mod 53 = 11
1	11	121	121 mod 53 = 15
2	15	225	225 mod 53 = 13
3	13	169	169 mod 53 = 10

Take a look at our binary term with values of 1 in red, this signifies which terms we use for our expansion:

$$10 \times 13 \times 11 = 1430 \text{ mod } 53 = 52$$

<https://www.mathcelebrity.com/modexp.php?>

PRIME NUMBERS:

A prime number is a positive integer greater than 1 that has no positive integer divisors other than 1 and itself. In other words, a prime number is a number that is only divisible by 1 and itself.

For example, the first few prime numbers are:

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, ...

These numbers have only two positive integer factors, which are 1 and themselves. For example, 2 can only be divided by 1 and 2, 3 can only be divided by 1 and 3, 5 can only be divided by 1 and 5, and so on.

Prime numbers are important in many areas of mathematics and computer science. They are the building blocks of the integers, and many complex mathematical structures and algorithms rely on their properties. For example, public-key cryptography, which is used to secure online transactions, relies on the fact that it is very difficult to factor large composite numbers into their prime factors.

Q) Find all Prime numbers that in range [313-31313]?

<https://planetcalc.com/9003/>



bashar_sh77@uomustansiriyah.edu.iq





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

Euler's Totient Function vs Euler Function:

"Euler's function" and "Euler's totient function" are two different names for the same mathematical concept. Both names refer to the function that counts the number of positive integers less than or equal to n that are relatively prime to n . The function is denoted by the symbol $\varphi(n)$ or sometimes by $\phi(n)$.

Determine $\phi(37)$ and $\phi(35)$.

Because 37 is prime, all of the positive integers from 1 through 36 are relatively prime to 37. Thus $\phi(37) = 36$.

To determine $\phi(35)$, we list all of the positive integers less than 35 that are relatively prime to it:

1, 2, 3, 4, 6, 8, 9, 11, 12, 13, 16, 17, 18
19, 22, 23, 24, 26, 27, 29, 31, 32, 33, 34

There are 24 numbers on the list, so $\phi(35) = 24$.

It should be clear that, for a prime number p , $\phi(p) = p - 1$

Now suppose that we have two prime numbers p and q with $p \neq q$. Then we can show that, for $n = p \cdot q$,

$$\phi(n) = \phi(p \cdot q) = \phi(p) \cdot \phi(q) = (p - 1) \cdot (q - 1)$$

Table 2.6 Some Values of Euler's Totient Function $\phi(n)$

n	$\phi(n)$	n	$\phi(n)$	n	$\phi(n)$
1	1	11	10	21	12
2	1	12	4	22	10
3	2	13	12	23	22
4	2	14	6	24	8
5	4	15	8	25	20
6	2	16	8	26	12
7	6	17	16	27	18
8	4	18	6	28	12
9	6	19	18	29	28
10	4	20	8	30	8





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

In number theory, the Euler Phi Function or Euler Totient Function $\varphi(n)$ gives the number of positive integers less than n that are relatively prime to n , i.e., numbers that do not share any common factors with n . For example, $\varphi(12) = 4$, since the four numbers 1, 5, 7, and 11 are relatively prime to 12.

$$\varphi(n) = n \prod (1 - 1/p_j),$$

where the p_j 's are the prime factors of n . For example, the prime factors of 12 are 2 and 3. If we use the product formula above to compute $\varphi(12)$, we get

$$\begin{aligned}\varphi(12) &= 12 \prod_{\substack{p|12 \\ p \text{ prime}}} \left(1 - \frac{1}{p}\right) \\ &= 12 \times \left(1 - \frac{1}{2}\right) \times \left(1 - \frac{1}{3}\right) \\ &= 4\end{aligned}$$

$$\begin{aligned}\varphi(16) &= 16 \prod_{\substack{p|16 \\ p \text{ prime}}} \left(1 - \frac{1}{p}\right) \\ &= 16 \times \left(1 - \frac{1}{2}\right) \\ &= 8\end{aligned}$$

<https://www.had2know.org/academics/euler-totient-function-calculator.html>

https://mathtools.lagrida.com/arithmetic/euler_totient.html



bashar_sh77@uomustansiriyah.edu.iq





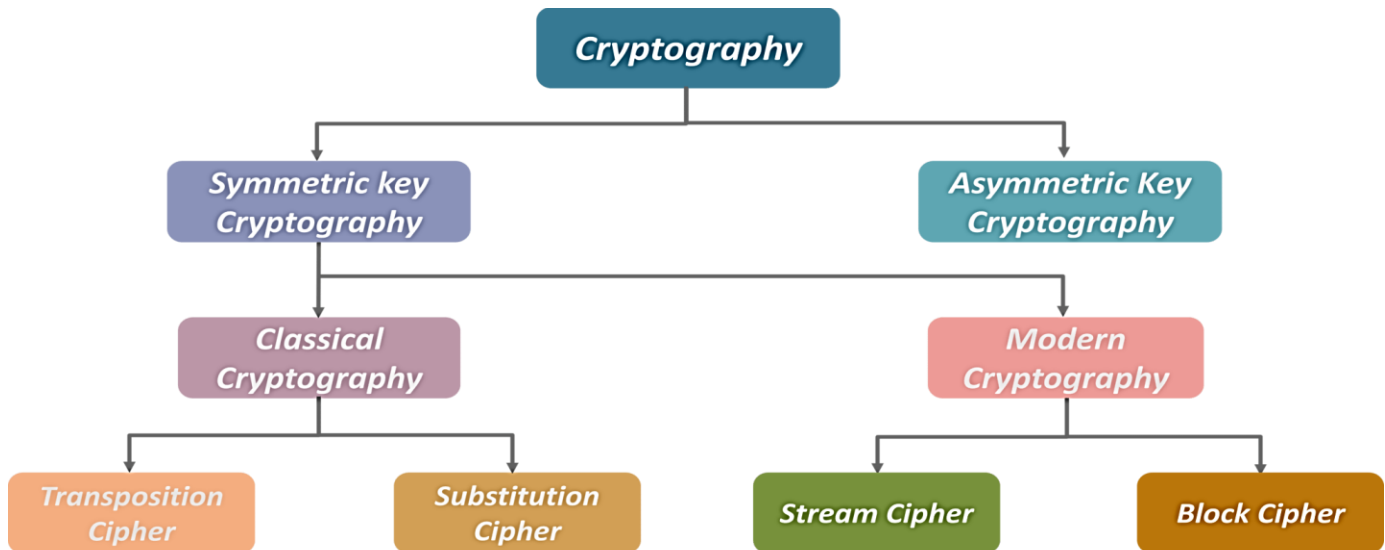
CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

Classical Encryption Techniques

We can describe Cryptography simply by the following Framework:



SYMMETRIC CIPHER MODEL:

A symmetric encryption scheme has five ingredients:

- **Plaintext:** This is the original intelligible message or data that is fed into the algorithm as input.
- **Encryption algorithm:** The encryption algorithm performs various substitutions and transformations on the plaintext.
- **Secret key:** The secret key is also input to the encryption algorithm. The key is a value independent of the plaintext and of the algorithm. The algorithm will produce a different output depending on the specific key being used at the time. The exact substitutions and transformations performed by the algorithm depend on the key.
- **Ciphertext:** This is the scrambled message produced as output. It depends on the plaintext and the secret key. For a given message, two different keys will produce two different ciphertexts. The ciphertext is an apparently random stream of data and, as it stands, is unintelligible.
- **Decryption algorithm:** This is essentially the encryption algorithm run in reverse. It takes the ciphertext and the secret key and produces the original plaintext.



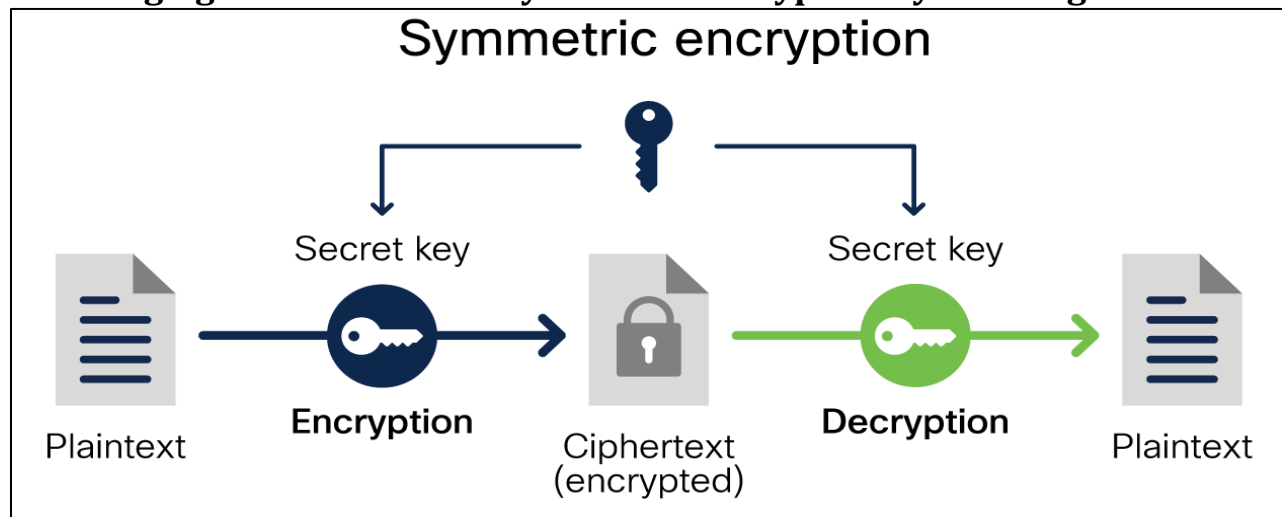


CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

The following figure describe the Symmetric Encryption System in general:



Note: All Encryption Methods depends on Alphabet, so the English alphabet as follow:

a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12

n	o	p	q	r	s	t	u	v	w	x	y	z
13	14	15	16	17	18	19	20	21	22	23	24	25

SUBSTITUTION TECHNIQUES:

A substitution cipher is a type of encryption technique where each letter in the plaintext is replaced by another letter or symbol. There are several types of substitution ciphers, and here are a few examples:

1. **Caesar Cipher:** This is one of the simplest and most well-known substitution ciphers. It involves shifting each letter in the plaintext by a fixed number of positions in the alphabet. For example, with a shift of 3, "A" would become "D", "B" would become "E", and so on.
2. **Monoalphabetic Cipher:** In this cipher, each letter in the plaintext is replaced by a corresponding letter in the ciphertext. The substitution is determined by a **fixed key** or a **predetermined pattern**. It uses a fixed key which consist of the 26 letters of a "**shuffled alphabet**".
3. **Polyalphabetic Cipher:** This type of cipher uses multiple substitution alphabets to encode the plaintext, making it more secure than monoalphabetic ciphers. One example is the **Vigenère cipher**, which uses a series of interwoven Caesar ciphers with different shift values based on a repeating keyword.





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

- Homophonic Cipher:** In this cipher, each letter in the plaintext is replaced by one or more symbols or letters in the ciphertext. This adds an extra layer of security, as there are multiple possible substitutions for each letter. However, it also makes the ciphertext longer and harder to decrypt.
- Polygraphic Cipher:** Instead of substituting letters one at a time, polygraphic ciphers substitute multiple letters or symbols at once. Examples include **Playfair cipher**, which substitutes pairs of letters, and Hill cipher, which uses matrix algebra to encrypt blocks of letters.

CAESAR CIPHER:

Caesar Cipher is a type of substitution cipher, which replaces each letter in the plaintext (the message to be encrypted) by a letter a fixed number of positions down the alphabet. For example, if the shift value is 3, then the letter 'A' in the plaintext would be replaced by the letter 'D' in the ciphertext, 'B' would be replaced by 'E', and so on.

Write down the alphabet:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Shift the letters by KEY that agree between Sender and Receiver places to the right:

Let Key = 3; Write down the plaintext to be encrypted:

Plaintext=" HELLO"; C= E (P, Key) = (P + Key) mod 26;

Replace each letter in the plaintext with the corresponding letter in the shifted alphabet:

H becomes K E becomes H L becomes O L becomes O O becomes R

The Resulting Ciphertext is "KHOOR".

To Decrypt the ciphertext, you simply shift the letters back by 3 places to the left.

P = D (C, Key) = (C - Key) mod 26; So, in this case, "KHOOR" would be **decrypted** to "HELLO".

MONOALPHABETIC CIPHER:

Using a simple monoalphabetic cipher depend on the following algorithm:

- Write down the alphabet:**

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

- Sender and Receiver agree about Create a new alphabet by randomly shuffling the letters:**

- **MAY BE AS KEYWORD OR NEW ALPHABET: Let Keyword= "MUSTANSIRIYAH"**

M	U	S	T	A	N	I	R	Y	H	B	C	D	E	F	G	J	K	L	O	P	Q	V	W	X	Z
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

- Write down the plaintext to be encrypted: HELLO**
- Replace each letter in the Plaintext=" HELLO" with the corresponding letter in the new alphabet:**

H becomes R, E becomes A, L becomes C, L becomes C, O becomes F

The Resulting Ciphertext is "RACCF".

NOTE: To decrypt the ciphertext, you simply use the inverse substitution rule to replace each letter in the ciphertext with its corresponding letter in the original alphabet.





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

PLAYFAIR CIPHER:

The **Playfair** Cipher is a type of **Polygraphic Substitution** cipher, which uses pairs of letters instead of single letters to create the ciphertext. The best-known multiple-letter encryption cipher is the Playfair, which treats digrams in the plaintext as single units and translates these units into ciphertext digrams. The Playfair algorithm is based on the use of a 5 * 5 matrix of letters constructed using a keyword.

1. First, we need to set up a key square or a key matrix, which is a 5x5 grid of letters used to encrypt the plaintext. The key square is usually created using a keyword, where all repeated letters are removed, and then the remaining letters are arranged in a 5x5 grid. In this example, we will use the keyword "LEMON":

L	E	M	O	N
A	B	C	D	F
G	H	I/J	K	P
Q	R	S	T	U
V	W	X	Y	Z

2. Write down the plaintext to be encrypted, and break it into pairs of letters. If there is an odd number of letters, add an "X" at the end:

HELLO becomes HE LX LO

3. For each pair of letters, apply the following encryption rules:
 - a. If the two letters are the same, insert an "X" between them.
 - b. If the two letters appear in the same row of the key square, replace each letter with the letter to its right (wrapping around to the beginning of the row if necessary).
 - c. If the two letters appear in the same column of the key square, replace each letter with the letter below it (wrapping around to the top of the column if necessary).
 - d. If the two letters form a rectangle, replace each letter with the letter in the same row and opposite corner of the rectangle.
4. Applying these rules to the plaintext pairs, we get the following ciphertext pairs:
HE becomes RB, LX becomes MV, LO becomes EN
The resulting ciphertext is " RBMVEN".

NOTE: To decrypt the ciphertext, you simply apply the inverse of the encryption rules to each pair of letters in the ciphertext using the same key square. In this case, " **RBMVEN** " would be decrypted to "HELLO" by replacing each pair of letters in the ciphertext with the corresponding pair of letters in the plaintext, as determined by the encryption rules used during encryption.

<https://planetcalc.com/7751/>





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

ONE TIME PAD ENCRYPTION (SUBSTITUTION- POLY-ALPHABETIC CIPHER):

One-time pad is a type of encryption where each bit or character of the plaintext is combined with a corresponding bit or character from a secret random key using the XOR operation. The key is at least as long as the plaintext and is only used once, hence the name "one-time pad".

The resulting ciphertext appears completely random and does not reveal any information about the plaintext. This makes one-time pad encryption theoretically unbreakable, as long as the key remains secret and is used only once.

- **Alphabet** such as ABCDEFGHIJKLMNOPQRSTUVWXYZ OR can be Customized.
- **Generate a key:** The key should be a random string of the same length as the plaintext. Each character in the key should be chosen uniformly and independently from the set of possible characters.
- **Convert the plaintext and key into binary:** The plaintext and key should be converted into binary strings using some fixed encoding, such as ASCII or Unicode.
- **Perform the XOR operation:** The binary plaintext and key should be XORed together bit-by-bit to produce the ciphertext. The XOR operation is performed as follows:
- **Convert the ciphertext back into a text format:** The resulting ciphertext should be converted back into a text format using the same encoding as the plaintext.

Here is an example of how this algorithm would work using the **plaintext** "HELLO" and a randomly generated **key** "KRGJK":

- Generate a **key**: "KRGJK" and in Hexa="4B 52 47 4A 4B" therefore in binary:
01001011 01010010 0100 0111 01001010 01001011
- Convert the **plaintext** into binary:
Plaintext: 01001000 01000101 01001100 01001100 01001111
Key: 01001011 01010010 0100 0111 01001010 01001011
- Perform the **XOR** operation:
Plaintext XOR Key: 00000011 00010111 00001011 00000110 00000100
- Convert the ciphertext back into a text format:
Ciphertext: 00000011 00010111 00001011 00000110 00000100
Encoded: '\x03\x17\x0B\x06\x04'=1p^o

Therefore, the resulting ciphertext is '\x03\x17\x0B\x06\x04'. This ciphertext can only be decrypted using the original key, which is known only to the sender and receiver.

<https://xor.pw/#>

<https://codebeautify.org/text-to-binary>

<https://www.rapidtables.com/convert/number/binary-to-ascii.html>





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

AFFINE CIPHER (SUBSTITUTION- MONOALPHABETIC CIPHER):

The Affine Cipher is a type of monoalphabetic substitution cipher that uses mathematical operations to encrypt and decrypt messages. It involves two keys: a multiplicative key (a number) and an additive key (another number). Here's an example of how the Affine Cipher works step by step:

1. S/R Agree about Alphabet such that:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

2. **Choose the two keys:** a multiplicative key (**a**) and an additive key (**b**). For example, let's choose **a = 5** and **b = 8**.
3. **Choose a message to encrypt.** Let's choose the message "HELLO WORLD".
4. **Convert each letter in the message to a number,** using a standard numerical mapping. For example, A=0, B=1, C=2, and so on. Using this mapping, the message "HELLO WORLD" becomes: **7 4 11 11 14 22 14 17 11 3**
5. **Apply the encryption formula to each number in the message:** $C = (a * P + b) \text{ mod } 26$ where C is the encrypted number, P is the plaintext number, and mod 26 ensures that the result is always between 0 and 25.
6. Using the keys from step 1, the encryption formula becomes: $C = (5 * P + 8) \text{ mod } 26$
7. Applying this formula to each number in the message, we get: **17 22 3 3 2 12 2 15 3 23**
8. Convert the encrypted numbers back into letters, using the same numerical mapping. For example, 0=A, 1=B, 2=C, and so on.
9. Using this mapping, the encrypted message becomes: **Cipher Text= R W D D C M C P D X**

NOTE: Affine Decryption can be computed by the following formula:

Plain Text = $(a^{-1} * (C - b)) \text{ mod } 26$, so for a=5, the $a^{-1} = 21$

HOMEWORK:

- FIND PLAINTEXT FOR ABOVE CIPHERTEXT?
- WRITE A PROGRAM TO FIND a^{-1} TO ANY VALUE OF a?
- FIND CIPHERTEXT OF MESSAGE "CYBERSECURITY" WHERE a=3 AND b=5?





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

THE AFFINE CIPHER IS A TYPE OF MONOALPHABETIC SUBSTITUTION CIPHER THAT USES A MATHEMATICAL FUNCTION TO ENCRYPT PLAINTEXT. THE FUNCTION IS OF THE FORM:

$$E(X) = (AX + B) \text{ MOD } M$$

WHERE:

E(X) IS THE CIPHERTEXT

X IS THE PLAINTEXT

A AND B ARE THE KEYS

M IS THE SIZE OF THE ALPHABET USED

TO DECRYPT THE CIPHERTEXT USING THE AFFINE CIPHER, WE NEED TO FIND THE INVERSE OF KEY1, WHICH IS THE VALUE OF A.

TO FIND THE INVERSE OF KEY1 IN THE AFFINE CIPHER, WE NEED TO FOLLOW THESE STEPS:

DETERMINE THE VALUE OF M, WHICH IS THE SIZE OF THE ALPHABET USED IN THE CIPHER. FOR EXAMPLE, IF WE ARE USING THE ENGLISH ALPHABET, $M = 26$.

CALCULATE THE GREATEST COMMON DIVISOR (GCD) OF A AND M. IF $\text{GCD}(A, M)$ IS NOT EQUAL TO 1, THEN THE INVERSE DOES NOT EXIST. IF $\text{GCD}(A, M)$ IS EQUAL TO 1, THEN THE INVERSE EXISTS.

CALCULATE THE MODULAR MULTIPLICATIVE INVERSE OF A, DENOTED AS A^{-1} , USING THE EXTENDED EUCLIDEAN ALGORITHM OR ANOTHER METHOD. THE MODULAR MULTIPLICATIVE INVERSE OF A IS A NUMBER SUCH THAT $(A * A^{-1}) \text{ MOD } M = 1$.

ONCE WE HAVE FOUND THE MODULAR MULTIPLICATIVE INVERSE OF A, WE CAN USE IT TO DECRYPT THE CIPHERTEXT USING THE FORMULA:

$$D(X) = A^{-1}(X - B) \text{ MOD } M \text{ WHERE:}$$

D(X) IS THE PLAINTEXT X IS THE CIPHERTEXT

A^{-1} IS THE MODULAR MULTIPLICATIVE INVERSE OF A

B IS THE KEY2

FOR EXAMPLE, LET'S SAY THAT WE HAVE ENCRYPTED THE LETTER "A" USING THE AFFINE CIPHER WITH KEYS $A=5$ AND $B=8$, AND THE SIZE OF THE ENGLISH ALPHABET $M=26$. TO DECRYPT THE CIPHERTEXT, WE NEED TO FIND THE INVERSE OF A.

$M = 26$ (SINCE WE ARE USING THE ENGLISH ALPHABET)

$\text{GCD}(5, 26) = 1$, SO THE INVERSE EXISTS.

TO FIND THE MODULAR MULTIPLICATIVE INVERSE OF 5, WE CAN USE THE EXTENDED EUCLIDEAN ALGORITHM:

$$26 = 5 * 5 + 1$$

$$1 = 26 - 5 * 5$$

THEREFORE, THE MODULAR MULTIPLICATIVE INVERSE OF 5 IS 21, SINCE $(5 * 21) \text{ MOD } 26 = 1$.

NOW THAT WE HAVE FOUND THE INVERSE OF A, WE CAN USE IT TO DECRYPT THE CIPHERTEXT. LET'S SAY THAT THE CIPHERTEXT IS "M". USING THE FORMULA:

$$D(X) = A^{-1}(X - B) \text{ MOD } M$$

WE CAN DECRYPT THE CIPHERTEXT AS FOLLOWS:

$$D(M) = 21(M - 8) \text{ MOD } 26$$

$$= (21M - 168) \text{ MOD } 26$$

$$= (21M + 10) \text{ MOD } 26$$

THEREFORE, THE PLAINTEXT CORRESPONDING TO THE CIPHERTEXT "M" IS THE 11TH LETTER OF THE ENGLISH ALPHABET, WHICH IS "K".





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

```
using System;
namespace AffineAlgorithm
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Modular Multiplicative Inverse in Affine Algorithm\n");

            Console.Write("Enter the value of a: ");
            int a = int.Parse(Console.ReadLine());

            Console.Write("Enter the value of m: ");
            int m = int.Parse(Console.ReadLine());

            int x = 0, y = 0;
            int gcd = ExtendedEuclideanAlgorithm(a, m, ref x, ref y);

            if (gcd != 1)
            {
                Console.WriteLine("Modular multiplicative inverse does not exist.");
            }
            else
            {
                int inverse = (x % m + m) % m;
                Console.WriteLine("The modular multiplicative inverse of {0} (mod {1}) is {2}", a, m, inverse);
            }

            Console.ReadKey();
        }

        static int ExtendedEuclideanAlgorithm(int a, int b, ref int x, ref int y)
        {
            if (b == 0)
            {
                x = 1;
                y = 0;
                return a;
            }

            int x1 = 0, y1 = 0;
            int gcd = ExtendedEuclideanAlgorithm(b, a % b, ref x1, ref y1);

            x = y1;
            y = x1 - (a / b) * y1;

            return gcd;
        }
    }
}
```





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

using System;

namespace AffineAlgorithm

```
{
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Modular Multiplicative Inverse in Affine Algorithm\n");

        Console.Write("Enter the value of a: ");
        int a1 = int.Parse(Console.ReadLine());

        Console.Write("Enter the value of m: ");
        int n1 = int.Parse(Console.ReadLine());

        int inv= modInverse(a1,n1);
        Console.WriteLine("Modular inverse is :"+inv);
        Console.ReadKey();
    }
    static int modInverse(int a, int n)
    {
        int i = n, v = 0, d = 1;
        while (a>0)
        {
            int t = i/a, x = a;
            a = i % x;
            i = x;
            x = d;
            d = v - t*x;
            v = x;
        }
        v %= n;
        if (v<0) v = (v+n)%n;
        return v;
    }
}
}
```





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

TRANSPONITION ENCRYPTION:

Transposition encryption is a type of encryption technique that involves rearranging the positions of characters in a message to make it difficult to read or understand. This technique does not change the actual characters used in the message but rather changes their positions within the message.

The basic *idea of transposition encryption is to take a message, split it into smaller segments, and then rearrange the segments in a specific order*. For example, the message "HELLO WORLD" could be split into two segments "HELLO" and "WORLD", and then rearranged as "WORLDHELLO" using a specific algorithm.

One of the advantages of transposition encryption is that:

- it is relatively easy to implement, and
- can be done using simple algorithms.
- Additionally, because it only rearranges the characters in a message, it can be used to preserve the length and structure of a message.
- Finally, transposition encryption can be combined with other encryption techniques to create more complex and secure encryption systems.

There are many types of Transposition Encryption Techniques such as:

1. Rail Fence Cipher
2. Columnar Transposition Cipher
3. Double Transposition Cipher
4. Route Cipher
5. Scytale Cipher
6. Book Cipher
7. Myszkowski Transposition Cipher





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

RAIL FENCE ALGORITHM (TRANSPONITION ENCRYPTION):

Rail Fence Encryption is a type of transposition cipher that works by writing the plaintext in a zigzag pattern along a set number of "rails" or lines. Here's how the algorithm works:

1. First, choose the number of rails you want to use. This is typically a positive integer greater than 1
2. Write the plaintext message along the rails in a zigzag pattern. Start by writing the first letter in the top left corner of the first rail, then continue writing the second letter in the second rail, the third letter in the third rail, and so on. When you reach the bottom of the last rail, start writing upwards again in the next rail.
3. Once you've written the entire plaintext message in the zigzag pattern, read off the resulting ciphertext by reading each row from left to right and concatenating them together.
4. The resulting ciphertext is your encrypted message.

Let's walk through an example of **Encrypting** the message **"BASHAR MEKKE ALESAWI"** using a 4-rail fence:

Choose the number of rails: 4

B	M	L
.	A	.	.	.	R	.	E	.	.	.	A	.	E	.	.	.	I
.	.	S	.	A	.	.	.	K	.	E	.	.	.	S	.	W	.
.	.	.	H	K	A	.	.

Read off row by row, The resulting **Ciphertext** is: **"BMLAREAEISAKESWHKA"**

To **Decrypt** the ciphertext, you simply reverse the process by writing the ciphertext in the same zigzag pattern along the same number of rails and then reading off the resulting plaintext. Here's the algorithm:

- Choose the number of rails used to encrypt the message.
- Calculate the length of each row by dividing the length of the ciphertext by the number of rails and rounding up to the nearest integer.
- Write the ciphertext along the rails in a zigzag pattern, starting with the first row.
- Read off the resulting plaintext by reading each column from left to right and concatenating them together.





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

Let's walk through an example of decrypting the ciphertext " BMLAREAEISAKESWHKA " using a 4-rail fence:

- Choose the number of rails: 4
- Calculate the length of each row: $\text{ceil}(18 / 4) = 4$
- Write the ciphertext along the rails in a zigzag pattern, starting with the first row:

B	M	L
.	A	.	.	.	R	.	E	.	.	.	A	.	E	.	.	.	I
.	.	S	.	A	.	.	.	K	.	E	.	.	.	S	.	W	.
.	.	.	H	K	A	.	.

- Read off the resulting **Plaintext** by reading each column from left to right: "BASHARMEKKEALESAWI"
- And that's it! You've successfully decrypted the ciphertext using the Rail Fence Encryption method.

Overall, Rail Fence Encryption is a fun and interesting example of a transposition cipher, but it is not suitable for use in secure cryptographic systems. It's a good starting point for learning about encryption, but more sophisticated methods should be used in practice.

NOTE: The Direction of distribution is another Trick to Rail-Fence (Top-Down OR Bottom-Up)

Online Test: <http://www.online.crypto-it.net/eng/rail-fence.html>





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

```
1 using System;
2 using System.Collections.Generic;
3 class BASHAR_RailFence
4 {
5     public static string Encrypt(string messageInput, int rowNumber)
6     {
7         string messageOutput = "";
8         List<List<char>> fanceTable = new List<List<char>>();
9         for (int pos = 0; pos < rowNumber; ++pos)
10            fanceTable.Add(new List<char>());
11        int r = 0; int direction = 1;
12
13        for (int c = 0; c < messageInput.Length; ++c)
14        {
15            fanceTable[r].Add(messageInput[c]);
16            if (((r == rowNumber - 1) && (direction == 1)) ||
17                ((r == 0) && (direction == -1)))
18                direction = -direction;
19            r = r + direction;
20        }
21
22        int row = 0;
23        while (row < rowNumber)
24        {
25            for (int pos = 0; pos < fanceTable[row].Count; ++pos)
26                messageOutput = messageOutput + fanceTable[row][pos];
27            ++row;
28        }
29        return messageOutput;
30    }
31    static void Main()
32    {
33        Console.WriteLine("Enter the message:");
34        string msg = Console.ReadLine();
35        Console.WriteLine("Enter the key:");
36        int keyn = int.Parse(Console.ReadLine());
37        string ciphermsg = Encrypt(msg, keyn);
38        Console.WriteLine("Cipher text is: " + ciphermsg);
39    }
```





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

```
public static string Decrypt(string messageInput, int rowNumber)
{
    List<List<char>> fenceTable = new List<List<char>>();

    for (int pos = 0; pos < rowNumber; ++pos)
    {
        fenceTable.Add(new List<char>());
    }

    int r = 0;
    int direction = 1;

    for (int c = 0; c < messageInput.Length; ++c)
    {
        fenceTable[r].Add('*'); // use a placeholder character for now

        if (((r == rowNumber - 1) && (direction == 1)) ||
            ((r == 0) && (direction == -1)))
        {
            direction = -direction;
        }
    }

    int index = 0;
    for (int i = 0; i < rowNumber; i++)
    {
        for (int j = 0; j < fenceTable[i].Count; j++)
            fenceTable[i][j] = messageInput[index++];
    }
    string messageOutput = ""; r = 0; direction = 1;
    for (int c = 0; c < messageInput.Length; ++c)
    {
        messageOutput += fenceTable[r][0];
        fenceTable[r].RemoveAt(0);
        if (((r == rowNumber - 1) && (direction == 1)) ||
            ((r == 0) && (direction == -1)))
            direction = -direction;
        r = r + direction;
    }
    return messageOutput;
}
```





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

COLUMNAR TRANSPOSITION CIPHER(TRANSPOTION ENCRYPTION):

Columnar Transposition Cipher is a type of **transposition** cipher where the plaintext message is written in rows and then rearranged by columns, where the order of the columns is determined by a **keyword**. This results in the ciphertext being formed by reading the columns in the order specified by the keyword.

The following description of Columnar - Transposition Cipher Algorithm:

- **Choose** a keyword WHICH will determine the order of the columns in the transposition.
- **Write** the plaintext message in rows beneath the keyword, **filling** in any unused spaces with a filler character.
- **Rearrange** the columns so that they are in alphabetical order based on the keyword.
- **Read** the ciphertext by reading down each column in the new order.

To decrypt the ciphertext, use the same keyword to determine the order of the columns and then read the message row by row. Here is the algorithm in more detail:

Encryption:

Input: Plaintext message, keyword

Output: Ciphertext message

1. Remove any spaces or punctuation from the plaintext message.
2. Choose a **keyword** with no repeated letters. For example, "SECRET".
3. Rearrange the columns so that they are in alphabetical order based on the keyword.
4. Write the plaintext message in rows beneath the keyword, filling in any unused spaces with a filler character (such as "\$").

For example, with Plaintext="BASHAR AL-ESAWI" ⇒: "BASHARALESAWI"

S	E	C	R	E	T
5	2	1	4	3	6
B	A	S	H	A	R
A	L	E	S	A	W
I	\$	\$	\$	\$	\$

5. Read the ciphertext by reading down each column in the new order. Therefore, the Ciphertext= "SE\$AL\$AA\$H\$S\$BAIRW\$".

Decryption:

S	E	C	R	E	T	H	A	C	K	C	Y	B	E	R
5	2	1	4	3	6									
B	A	S	H	A	R									
A	L	E	S	A	W									
I	\$	\$	\$	\$	\$									

<https://www.boxentriq.com/code-breaking/columnar-transposition-cipher>



bashar_sh77@uomustansiriyah.edu.iq





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

NOTE: One of the optimizations using Double-Columnar Transposition.

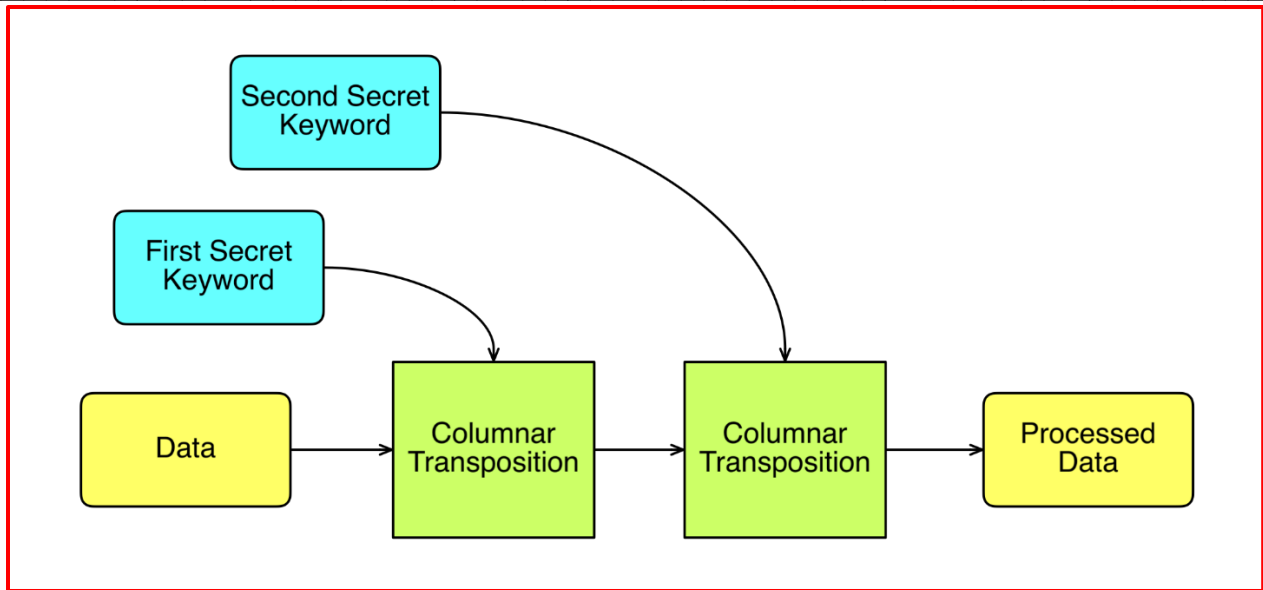
THERE ARE TWO KEYS: KEY1=BASHAR FOR ROUND1 AND KEY2=CYBERSECURITY FOR ROUND2

b	a	s	h	a	r
3	1	6	4	2	5
D	o	u	b	l	e
C	o	l	u	m	
n	a	r	T	r	
a	n	s	p	o	s
i	t	i	o	n	X

C1= oCantluTonD naibl
poemrsX

C2=abioarTeXnlXuoXDsx
t XnrXlpX uXomXCisnoX

c	y	b	e	r	s	e	c	u	r	i	t	y
2	12	1	4	7	9	5	3	11	8	6	10	13
o	C	a	n	t	l	u	T	o	n	D	n	
a	i	b	l		p	o	e	m	r	s	u	o
r	s	i	X	X	X	X	X	X	X	X	X	X





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

RC4 (RIVEST CIPHER-**STREAM CIPHER**):

RC4 (also known as Rivest Cipher 4 or ARC4) is a symmetric key stream cipher. It operates on plaintext one byte at a time and generates a key stream byte for each plaintext byte. The key stream is then XORed with the plaintext to produce the ciphertext. The security of RC4 depends on the secrecy of the key used to generate the key stream. Here are the steps for the RC4 algorithm:

1. Key Scheduling Algorithm (KSA)

The KSA initializes the permutation array S using the key K . The permutation array S is a fixed array of 256 bytes, numbered from 0 to 255.

- A. Initialize the permutation array S with the values from 0 to 255:
for i from 0 to 255 $S[i] = i$;
- B. Initialize the j variable to 0: $j = 0$;
- C. For each byte in the key K , update the permutation array S :
for i from 0 to 255
 $j = (j + S[i] + K[i \bmod \text{keylength}]) \bmod 256$
swap $S[i]$ and $S[j]$
end for

2. Pseudo-Random Generation Algorithm (PRGA)

The PRGA generates the key stream by repeatedly generating a pseudo-random byte and updating the permutation array S .

- A. Initialize the i and j variables to 0:
- B. For each Plaintext byte in the input, generate a key stream byte:
for each plaintext byte P
 $i = (i + 1) \bmod 256$
 $j = (j + S[i]) \bmod 256$
swap $S[i]$ and $S[j]$
 $K = S[(S[i] + S[j]) \bmod 256]$
end for

3. Encryption or Decryption: (BEST IN BINARY WITH 8-BITS FOR EACH)

Once the pseudo-random stream is generated, it can be used to encrypt or decrypt data. This is done by XORing each byte of the data with the corresponding byte of the pseudo-random stream.

for each Plaintext byte P (OR each Ciphertext byte C)

Ciphertext byte = $P \text{ XOR } K$ /*to Do Encryption*/

OR

Plaintext byte = $C \text{ XOR } K$ /* to Do Decryption*/

end for





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

TOOLS NEEDED:

- Convert Text 2 ASCII: <https://www.browserling.com/tools/text-to-ascii>
- XOR Operation: <https://xor.pw/#>
- RC4 ONLINE: <https://www.dcode.fr/rc4-cipher>

Example: Let Plaintext="BMN", and Key=" LAMBORGHINI", Ciphertext must be ="208 163 112"

1. Key Scheduling Algorithm (KSA):

- Initialize S-Array= [0,1,2,3,4,5,6,7,8,9,10, 11,,255] for i=0 to 255;
- Let K-Array= [76, 65, 77, 66, 79, 82, 71, 72, 73, 78, 73] Ascii of KEY.
- Permutation array S,
 - **j=0; keylength=11;**
 - **for i from 0 to 255**
 - **j = (j + S[i] + K[i mod keylength]) mod 256**
 - **swap S[i] and S[j]**
 - **end for**

We will start by initializing j to 0, and then use the formula to compute the value of j for each i. Here's the step-by-step process:

1. Initialize j to 0: $j = 0$
2. Compute j for $i = 0$: $j = (j + S[0] + K[0 \text{ mod } 11]) \text{ mod } 256; = (0 + 0 + 76) \text{ mod } 256; = 76$
3. Compute j for $i = 1$: $j = (j + S[1] + K[1 \text{ mod } 11]) \text{ mod } 256; = (76 + 1 + 65) \text{ mod } 256; = 142$
4. Compute j for $i = 2$: $j = (j + S[2] + K[2 \text{ mod } 11]) \text{ mod } 256; = (142 + 2 + 77) \text{ mod } 256; = 221$
5. Compute j for $i = 3$: $j = (j + S[3] + K[3 \text{ mod } 11]) \text{ mod } 256; = (221 + 3 + 66) \text{ mod } 256; = 30$
6. And so on.....

Therefore , S-Array= [76, 142, 221,30, 113, 200, 21, 100, 181, 12, 95,.....];

- Pseudo-Random Generation Algorithm (PRGA):
 - Initialize the i and j variables to 0:
 - For each Plaintext byte in the input, generate a key stream byte:
 - for each plaintext byte P**
 - i = (i + 1) mod 256**
 - j = (j + S[i]) mod 256**
 - swap S[i] and S[j]**
 - K = S[(S[i] + S[j]) mod 256]**
 - end for**





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

```
using System;
using System.Collections.Generic;
using System.Text;

public class BasharRC4
{
    public static void Main(string[] args)
    {
        //1. Key Scheduling Algorithm KSA
        List<int> S = new List<int>();
        for (int i = 0; i < 256; i++) S.Add(i);
        foreach (int i in S) Console.Write(i + " ");
        Console.WriteLine("=====");
        string Key="BMN";
        byte[] asciiBytes = Encoding.ASCII.GetBytes(Key);
        for (int i=0;i<3;i++) Console.Write(asciiBytes[i] + " "); Console.WriteLine("ASCII Code of Key"); Console.WriteLine();

        int j=0;
        for (int i=0; i<256;i++)
        {
            j = (j + S[i] + asciiBytes[i % 3]) % 256;
            int temp = S[i]; S[i] = S[j]; S[j] = temp;
        }
        Console.WriteLine("Swaped as follow");

        foreach (int i in S) Console.Write(i + " "); Console.WriteLine();
        //2.Pseudo-Random Generation Algorithm (PRGA)
        string Plaintext="Bashar";
        byte[] asciiBytes2 = Encoding.ASCII.GetBytes(Plaintext);

        for (int i=0;i<6;i++) Console.Write(asciiBytes2[i] + " "); Console.WriteLine("ASCII Code of Plaintext");
        Console.WriteLine();

        int [] ciphertext = new int[Plaintext.Length]; int ii = 0; int jj = 0;
        for (int k = 0; k < Plaintext.Length; k++)
        {
            ii = (ii + 1) % 256;
            jj = (jj + S[ii]) % 256;
            int temp1 = S[ii]; S[ii] = S[jj]; S[jj] = temp1;

            int t = (S[ii] + S[jj]) % 256; int key = S[t];
            //3. Encryption or Decryption:
            ciphertext[k] = (asciiBytes2[k] ^ key);
        }
        Console.WriteLine("The Cipher Text of Plain= "+Plaintext+ " With KEY= "+ Key+" is: ");
        for (int i=0;i<6;i++) Console.Write(ciphertext[i] + " ");
    }
}
```





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

SDES (SIMPLE DATA ENCRYPTION STANDARD-**BLOCK CIPHER**):

SDES (Simple Data Encryption Standard) is a symmetric-key encryption algorithm that was designed to be a simpler and more efficient version of the widely used Data Encryption Standard (DES) algorithm. SDES works by taking a plaintext message and transforming it into a ciphertext message using a secret key.

The SDES algorithm operates on:

- **8-bit** blocks of **plaintext** and
- Uses a **10-bit key**, the key is used to perform **two rounds of substitution and permutation** operations on the plaintext,
- Resulting in a ciphertext that is then sent securely over the network.

Simplified DES (SDES) is a lightweight block cipher with a block size of 8 bits and a key size of 10 bits. The SDES algorithm consists of two rounds, and each round involves substitution and permutation operations. Here are the step-by-step instructions for the SDES encryption algorithm:

1) Key Generation

- Take the 10-bit key and split it into two 5-bit halves, K1 and K2.
- Permute K1 and K2 according to the P10 permutation table to obtain two new 5-bit keys, K1' and K2'.
- Shift K1' and K2' one bit to the left to obtain two more keys, K1'' and K2''.
- Permute K1'' and K2'' according to the P8 permutation table to obtain the final two 8-bit keys, K1K2.

2) Initial Permutation (IP)

- Take the 8-bit plaintext and permute it according to the IP permutation table.

3) Round 1

- Take the permuted plaintext from step 2 and split it into two 4-bit halves, L0 and R0.
- Expand R0 to 8 bits by duplicating the middle two bits, and XOR the result with K1.
- Split the result of the XOR operation into two 4-bit halves, S1 and S2.
- Look up the values of S1 and S2 in the S-boxes to obtain two new 2-bit values, S1' and S2'.
- Concatenate S1' and S2' to obtain a 4-bit value, S'.
- Permute S' according to the P4 permutation table to obtain P4(S').
- XOR P4(S') with L0 to obtain R1.
- Set L1 to R0.





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

4) Round 2

- Repeat steps 3a-3f using R1 and K2 instead of R0 and K1.

5) Inverse Initial Permutation (IP^{-1})

- Permute the concatenation of L2 and R2 according to the IP^{-1} permutation table to obtain the ciphertext.

That's it! The decryption algorithm is essentially the same as the encryption algorithm, but with the round keys used in reverse order.

here's how to encrypt the block "10110011" using the SDES algorithm with the key "1000011001":

1. Key Generation

- Split the 10-bit key into two 5-bit halves, $K1 = "10000"$ and $K2 = "11001"$.
- Permute $K1$ and $K2$ according to the P10 permutation table to obtain $K1' = "00011"$ and $K2' = "10110"$.
- Shift $K1'$ and $K2'$ one bit to the left to obtain $K1'' = "00110"$ and $K2'' = "01101"$.
- Permute $K1''$ and $K2''$ according to the P8 permutation table to obtain the final two 8-bit keys, $K1K2 = "11010011"$.

2. Initial Permutation (IP)

- Apply the IP permutation table to the block "10110011" to obtain "01101011".

3. Round 1

- Split the permuted block into two 4-bit halves, $L0 = "0110"$ and $R0 = "1011"$.
- Expand $R0$ to 8 bits by duplicating the middle two bits, yielding "10000101".
- XOR the result with $K1 = "1000011001"$ to obtain "00000000".
- Split the result of the XOR operation into two 4-bit halves, $S1 = "0000"$ and $S2 = "0000"$.
- Look up the values of $S1$ and $S2$ in the S-boxes to obtain $S1' = "00"$ and $S2' = "00"$.
- Concatenate $S1'$ and $S2'$ to obtain $S' = "0000"$.
- Permute S' according to the P4 permutation table to obtain $P4(S') = "0000"$.
- XOR $P4(S')$ with $L0 = "0110"$ to obtain $R1 = "0110"$.
- Set $L1 = R0 = "1011"$.

4. Round 2

- Repeat steps 3a-3f using $R1 = "0110"$ and $K2 = "11010011"$ instead of $R0$ and $K1$.

5. Inverse Initial Permutation (IP^{-1})

- Apply the IP^{-1} permutation table to the concatenation of $L2 = "1011"$ and $R2 = "0000"$ to obtain the ciphertext "01011000".

Therefore, the ciphertext for the block "10110011" using SDES with the key "1000011001" is "01011000"





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

To encrypt a plaintext using Simplified Data Encryption Standard (SDES), we need to follow the following steps:

1. Generate the round keys
2. Divide the plaintext into two parts
3. Apply the encryption process for each round
4. Combine the two parts to form the ciphertext

Let's follow these steps to encrypt the plaintext "10110011" using the key "1000011001".

Step 1: Generate the round keys

To generate the round keys, we need to perform the following steps:

1. Permute the 10-bit key using the P10 permutation: 3 5 2 7 4 10 1 9 8 6 Key: 1 0 0 0 0 1 1 0 0 1
2. Split the permuted key into two parts, each of 5 bits: Left part: 10000 Right part: 11001
3. Shift each part one bit to the left: Left part after shift: 00001 Right part after shift: 10011
4. Combine the shifted parts to form a 10-bit string: Combined key: 0000110011
5. Permute the combined key using the P8 permutation: 6 3 7 4 8 5 10 9 Round key 1: 00010110
6. Shift each part of the combined key two bits to the left: Left part after shift: 00110 Right part after shift: 01100
7. Combine the shifted parts to form a 10-bit string: Combined key: 0011001100
8. Permute the combined key using the P8 permutation: 6 3 7 4 8 5 10 9 Round key 2: 01100010

So the round keys are: Round key 1: 00010110 Round key 2: 01100010

Step 2: Divide the plaintext into two parts

The plaintext is "10110011". We need to divide it into two parts, each of 4 bits: Left part: 1011 Right part: 0011

Step 3: Apply the encryption process for each round

We have two round keys, so we need to apply the encryption process twice. The encryption process is as follows:





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

1. Apply the initial permutation (IP) to the right part of the plaintext: 2 4 3 1 Right part after IP: 0011 -> 1100
2. Expand the right part to 8 bits using the E/P table: 4 1 2 3 2 3 4 1 Expanded right part: 11000011
3. XOR the expanded right part with the first round key: Expanded right part after XOR: 11010101
4. Split the result into two parts, each of 4 bits: Left part: 1101 Right part: 0101
5. Apply the S-boxes to each part: Left part after S-boxes: 0110 Right part after S-boxes: 0010
6. Combine the two parts into a 4-bit string: Combined part: 01100010
7. Permute the combined part using the P4 permutation: 2 4 3 1 Result after P4 permutation: 0011
8. XOR the result of step 7 with the left part of the plaintext: Result after XOR: 1100
9. Swap

Example:

To find the ciphertext for the plaintext "10110011" using the key "1000011001" and Simplified Data Encryption Standard (SDES), we need to follow the encryption steps:

Step 1: Generate the round keys

We have already generated the round keys in the previous question:

Round key 1: 00010110 Round key 2: 01100010

Step 2: Divide the plaintext into two parts

The plaintext is "10110011". We need to divide it into two parts, each of 4 bits:

Left part: 1011 Right part: 0011

Step 3: Apply the encryption process for each round

We have two round keys, so we need to apply the encryption process twice. The encryption process is as follows:

Round 1:

1. Apply the initial permutation (IP) to the right part of the plaintext: 2 4 3 1 Right part after IP: 0011 -> 1100
2. Expand the right part to 8 bits using the E/P table: 4 1 2 3 2 3 4 1 Expanded right part: 11000011
3. XOR the expanded right part with the first-round key: Expanded right part after XOR: 11010101
4. Split the result into two parts, each of 4 bits: Left part: 1101 Right part: 0101
5. Apply the S-boxes to each part: Left part after S-boxes: 0110 Right part after S-boxes: 0010
6. Combine the two parts into a 4-bit string: Combined part: 01100010
7. Permute the combined part using the P4 permutation:
2 4 3 1 Result after P4 permutation: 0011
8. XOR the result of step 7 with the left part of the plaintext: Result after XOR: 1100
9. Swap the left and right parts: Left part after swap: 0011 Right part after swap: 1100





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

Round 2:

1. Apply the initial permutation (IP) to the right part of the plaintext: 2 4 3 1 Right part after IP: 1100 -> 0011
2. Expand the right part to 8 bits using the E/P table: 4 1 2 3 2 3 4 1 Expanded right part: 01010101
3. XOR the expanded right part with the second round key: Expanded right part after XOR: 00110111
4. Split the result into two parts, each of 4 bits: Left part: 0011 Right part: 0111
5. Apply the S-boxes to each part: Left part after S-boxes: 1000 Right part after S-boxes: 0100
6. Combine the two parts into a 4-bit string: Combined part: 10000100
7. Permute the combined part using the P4 permutation: 2 4 3 1 Result after P4 permutation: 0010
8. XOR the result of step 7 with the left part of the plaintext: Result after XOR: 1001
9. Combine the left and right parts: Combined parts: 00111001

Step 4: The resulting ciphertext is "00111001".





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

PUBLIC KEY CRYPTOGRAPHY (ASYMMETRIC CRYPTOGRAPHY):

INTRODUCTION:

Public key cryptography is a cryptographic system that enables secure communication between two parties without the need for a shared secret key. Instead, the system uses two mathematically related keys: a public key and a private key. These keys are used to encrypt and decrypt messages in such a way that only the intended recipient can read the message.

BASIC CONCEPTS:

Public key cryptography involves the use of two mathematically related keys: a public key and a private key. **The public key is made available to anyone who wants to send a message to the recipient. The private key, on the other hand, is known only to the recipient and is used to decrypt the message.**

EXAMPLE:

Alice wants to send a message to **Bob**, but she wants to ensure that only Bob can read the message. Bob generates a pair of keys, a public key and a private key. **Bob** gives **Alice** his public key, which she uses to encrypt the message. Once Alice encrypts the message with Bob's public key, only Bob can decrypt it with his private key. This ensures that only Bob can read the message.

ADVANTAGES AND DISADVANTAGES:

Advantages:

- Provides a secure means of communication without the need for a shared secret key.
- Enables digital signatures, which can be used to ensure the authenticity and integrity of a message.

Disadvantages:

- Computationally intensive, which means that it can be **slower than** symmetric key cryptography.
- Vulnerable to attacks such as brute force attacks and man-in-the-middle attacks.





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

PRACTICAL APPLICATIONS:

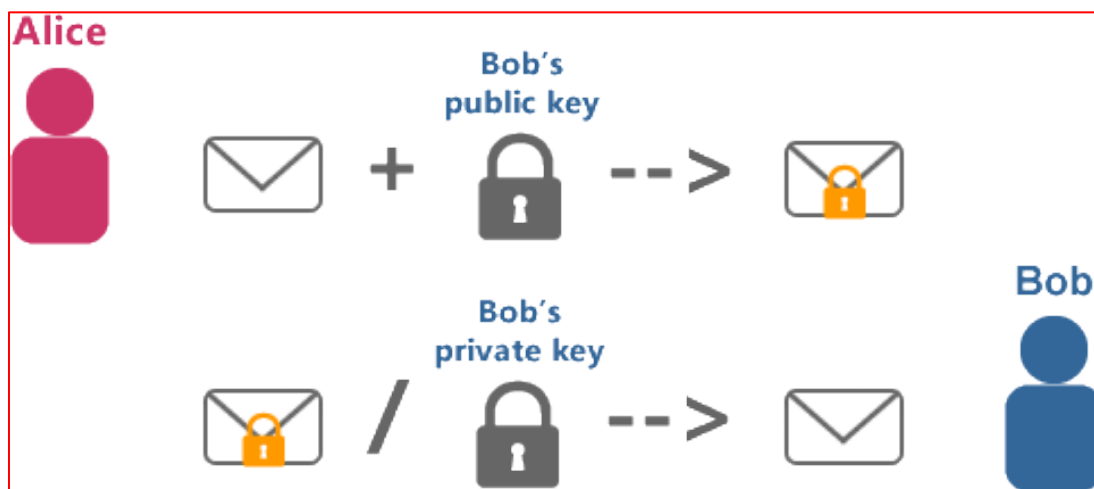
- 1) **Secure communication** over the internet: HTTPS, the protocol used to secure web traffic, uses public key cryptography to ensure the confidentiality and integrity of data sent between a web server and a client.
- 2) **Digital signatures**: used to ensure the authenticity and integrity of electronic documents.

Public key cryptography is a powerful tool that enables secure communication and digital signatures without the need for a shared secret key. While it has some disadvantages, its advantages make it ideal for many applications, including secure communication over the internet and digital signatures.

RSA ALGORITHM:

The RSA algorithm is a widely used public key cryptosystem that was developed by Ron Rivest, Adi Shamir, and Leonard Adleman in 1977. The algorithm is based on the difficulty of factoring large prime numbers.

The RSA algorithm involves the use of a public key and a private key. The public key is made available to anyone who wants to send a message to the recipient. The private key is known only to the recipient and is used to decrypt the message.



RSA GENERAL STRUCTURE









CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

Advantages of RSA:

	No need of sharing secret keys		Proof of owner's authenticity
	Faster Encryption than DSA		Data can't be modified in transit

KEY GENERATION:

The RSA algorithm involves the generation of a public key and a private key. The key generation process involves the following steps:

- 1) Choose two large prime numbers p and q .
- 2) Calculate $n = p \times q$.
- 3) Calculate the totient of n , $\phi(n) = (p-1) \times (q-1)$.
- 4) Choose an integer e such that $1 < e < \phi(n)$ and $\text{gcd}(e, \phi(n)) = 1$.
- 5) Calculate the value of d such that $d \times e = 1 \pmod{\phi(n)}$.
- 6) The **public** key is (n, e) and the **private** key is (n, d) .

TO GENERATE THE PUBLIC AND PRIVATE KEYS USING THE RSA ALGORITHM, WE NEED TO FOLLOW THESE STEPS:

- 1) Choose two distinct **prime** numbers, let's say $P=11$ and $Q=13$.
- 2) Compute $N = P \times Q$. In this case, $N=11 \times 13=143$.
- 3) Compute $\phi(N) = (P-1) \times (Q-1)$. In this case, $\phi(143) = (11-1) \times (13-1) = 120$.
- 4) Choose an integer e as **public key**, such that:
 - a. $1 < e < \phi(N)$ and,
 - b. $\text{gcd}(e, \phi(N)) = 1$. In this case, we can choose $e=7$.
- 5) Compute the **private key** d such that:
 - a. $(d \times e) \pmod{\phi(N)} = 1$.
 - b. In this case, we need to find d such that $(d \times 7) \pmod{120} = 1$.
One possible solution is $d=103$.
- 6) The **public key** is (N, e) , which in this case is $(143, 7)$.
- 7) The **private key** is (N, d) , which in this case is $(143, 103)$.
- 8) Therefore, the public key is $(143, 7)$ and the private key is $(143, 103)$.





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

ENCRYPTION AND DECRYPTION:

ENCRYPTION:

To encrypt a message m using the RSA algorithm, the sender performs the following steps:

- Represent the message as a number between 0 and $n-1$.
- Calculate the ciphertext $c = m^e \pmod{n}$.
- Send the ciphertext c to the recipient.

Okay Let Alice send message "RSA" to Bob,

- we can represent each letter of the message "RSA" as a number using its ASCII code, and encrypt each number separately using RSA.
- The ASCII codes for "R", "S", and "A" are 82, 83, and 65, respectively.
- To encrypt each number using RSA, we can use Bob's public key (N, e) , which we obtained in the previous step. $N = 143$ and $e = 7$, so we have:
 - For the letter "R", the corresponding number is 82. **The ciphertext for 82 is $(82^7) \pmod{143} = 34--69$.**
 - For the letter "S", the corresponding number is 83. **The ciphertext for 83 is $(83^7) \pmod{143} = 22--8$.**
 - For the letter "A", the corresponding number is 65. **The ciphertext for 65 is $(65^7) \pmod{143} = 107--65$.**
- Therefore, the encrypted message is "69 08 65". Alice would send this sequence of three numbers to Bob as the ciphertext.

[HTTPS://WWW.OMNICALCULATOR.COM/MATH/POWER-MODULO](https://www.omnicalculator.com/math/power-modulo)

DECRYPTION:

To decrypt the ciphertext c using the RSA algorithm, the recipient performs the following steps:

- Use the private key (n, d) to calculate $m = c^d \pmod{n}$.
- Convert the resulting number back to the original message.

SECURITY:

The security of the RSA algorithm is based on the difficulty of factoring large prime numbers. The strength of the RSA algorithm depends on the length of the keys used. Longer keys are more secure, but they also require more computation to perform encryption and decryption.





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

using System;
class Program

```
{
    static void Main(string[] args)
    {
        // Define the message as a string
        string message = "RSA";

        // Define Bob's public key
        int N = 143; int e = 7;

        // Encrypt each letter of the message using RSA
        string ciphertext = "";
        foreach (char c in message)
        {
            // Convert the letter to its ASCII code
            int m = (int) c;
            Console.WriteLine();
            Console.WriteLine("ASCII is: "+m);

            // Compute the ciphertext for the ASCII code using RSA
            int ctext = ModPow(m, e, N);

            // Append the ciphertext to the output string
            ciphertext += ctext.ToString() + " ";
        }
        // Print the encrypted message
        Console.WriteLine("Encrypted message: " + ciphertext.Trim());
    }
    // Compute the modular exponentiation a^b mod n
    static int ModPow(int a, int b, int n)
    {
        int result = 1;
        for (int i = 0; i < b; i++)
            result = (result * a) % n;
        return result;
    }
}
```

<https://www.programiz.com/csharp-programming/online-compiler/>

<https://www.cryptool.org/en/cto/rsa-step-by-step>



bashar_sh77@uomustansiriyah.edu.iq





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

AUTHENTICATION AND DIGITAL SIGNATURES:

INTRODUCTION:

In today's digital world, where transactions and communications are largely conducted online, it is essential to ensure that the parties involved are who they claim to be. This is where authentication and digital signatures come into play.

AUTHENTICATION:

Authentication is the process of verifying the **identity of a user, device, or application**. It is the first line of defense against unauthorized access to a system, network, or data. There are three types of authentication:

- **Something you know:** This type of authentication requires the user to enter a password, PIN, or some other secret information that only they know.
- **Something you have:** This type of authentication involves a physical object such as a smart card, token, or a mobile device that contains a unique identifier.
- **Something you are:** This type of authentication involves biometric identification such as fingerprints, facial recognition, or iris scans.

تلعب المصادقة دورًا مهمًا في الأمن السيبراني لأنها تساعد في التحقق من هوية المستخدم أو الجهاز أو التطبيق قبل منح اذن الوصول إلى **نظام أو شبكة أو بيانات**. إنه خط الدفاع الأول ضد الوصول غير المصرح به، لأنه يضمن السماح للمستخدمين المصرح لهم فقط بالوصول إلى المعلومات الحساسة. توفر الأنواع الثلاثة للمصادقة اعلاه - طبقات مختلفة من الأمان للتأكد من أن الشخص أو الجهاز الذي يحاول الوصول إلى النظام أو البيانات هو بالفعل من يدعي أنه المخول بذلك.

تعد كلمات المرور وأرقام التعريف الشخصية والبطاقات الذكية والرموز المميزة ومعرفات القياسات الحيوية كلها أمثلة على طرق المصادقة التي تساعد على منع الوصول غير المصرح به إلى المعلومات الحساسة. بدون المصادقة المناسبة، تتعرض البيانات والأنظمة الحساسة لخطر الاختراق، مما يؤدي إلى خروقات البيانات وسرقة الهوية وغيرها من الجرائم الإلكترونية. لذلك، تعد المصادقة مكونًا مهمًا للأمن السيبراني يساعد على حماية الأفراد والمؤسسات من التهديدات السيبرانية.

The infographic is a blue rectangular box with a white border. At the top, it has the title "What Is the Best Authentication Method? 5 Types of Authentication" in white text. Below the title, there are five dark blue square icons arranged horizontally. Each icon contains a white graphic and a label below it: 1. A speech bubble with "SMS" and an envelope icon, labeled "SMS/Email codes". 2. A profile of a person with sound waves, labeled "Voice". 3. A yellow pill-shaped icon with four dots, labeled "Passwords". 4. A fingerprint icon, labeled "Fingerprint". 5. A person icon with a face and a green checkmark, labeled "Face Verification".





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024





Mustansiriyah University – College of Science -Cybersecurity Branch

Scenario for an authentication process in an HR system used via the internet:

- 1) An employee wants to log in to the HR system.
- 2) The employee navigates to the HR system's login page on their web browser.
- 3) The login page prompts the employee to enter their username and password.
- 4) The employee enters their username and password, **which are sent over an encrypted connection to the HR system's server.**
- 5) The server validates the username and password against the HR system's database to confirm the employee's identity.
- 6) If the username and password are correct, **the server generates a unique session ID and sends it back to the employee's browser.**
- 7) The employee's browser stores the session ID as a cookie, which will be sent back to the server with each subsequent request.
- 8) The employee is now logged in and can access their **personal information in the HR system.**
- 9) If the employee logs out or if the session expires, the session ID is invalidated, and the employee will need to re-enter their credentials to log back in.

To enhance the security of this authentication process, the HR system could implement additional security measures, such as two-factor authentication or IP address filtering, to ensure that only authorized users can access the system.

Biometric Technology Types Features Overview

Features	 Facial Recognition	 Iris Scanning	 Fingerprint Identification	 Voice Verification
Accuracy	Medium-low	High	High	Medium
Attack's precaution	Medium	Very high	High	Medium
Verification time	About 3 seconds	Less than 5 seconds	Less than 3 seconds	About 5 seconds
Devices Required	Camera	Camera	Scanner	Microphone
Interference	Lighting, glasses, hair, moving, age	Lighting, inaccurate eye positioning in relation to the device	Dryness, dirt, deep injury, age	Noise, cold symptoms





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

TWO-FACTOR AUTHENTICATION:

Two-factor authentication (2FA) is a security measure that requires two different forms of authentication in order to grant access to a system or service. Typically, this involves something the user knows, such as a password, and something the user possesses, such as a smartphone or hardware token.

- ❖ For example, let's say you have enabled 2FA on your **online banking account**. When you log in to your account, you will be prompted to enter your password as usual. However, in order to complete the login process, you will also need to provide a second form of authentication, such as a one-time code generated by a mobile app or sent to your phone via SMS.
- ❖ Another example of 2FA is when you use your **debit card to withdraw cash from an ATM**. In addition to your PIN, the machine might also require you to insert a physical token or smart card that has been issued to you by your bank.

By requiring multiple forms of authentication, 2FA helps to prevent unauthorized access to sensitive information and reduce the risk of identity theft or other security breaches.

```
1 import vonage
2 # Initialize the client object
3 client = vonage.Client(key="7115001", secret="Gyu1500150")
4 # Initialize the SMS object
5 sms = vonage.Sms(client)
6 # Use the SMS object to send a message
7 response = sms.send_message({
8     "from": "Bashar",
9     "to": "+9647715352885",
10    "text": "The 2FA code=1977"
11 })
12 # Check the response
13 if response["messages"][0]["status"] == "0":
14     print("Message sent successfully.")
15 else:
16     print(f"Message failed with error: {response['messages'][0]['error-text']}")
```





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

HOW TO USE SSH PUBLIC KEY AUTHENTICATION OVERVIEW

Public key authentication is a way of logging into an SSH/SFTP account using a cryptographic key rather than a password.

If you use very strong SSH/SFTP passwords, your accounts are already safe from brute force attacks. However, using public key authentication provides many benefits when working with multiple developers. For example, with SSH keys you can

- allow multiple developers to log in as the same system user without having to share a single password between them;
- revoke a single developer's access without revoking access by other developers; and
- make it easier for a single developer to log in to many accounts without needing to manage many different passwords.

HOW PUBLIC KEY AUTHENTICATION WORKS:

Keys come in pairs of a public key and a private key. Each key pair is unique, and the two keys work together. These two keys have a very special and beautiful mathematical property: if you have the private key, you can prove you have it without showing what it is. It's like proving you know a password without having to show someone the password.

Public key authentication works like this:

1. Generate a key pair.
2. Give someone (or a server) the public key.
3. Later, anytime you want to authenticate, the person (or the server) asks you to prove you have the private key that corresponds to the public key.
4. You prove you have the private key.
5. You don't have to do the math or implement the key exchange yourself. The SSH server and client programs take care of this for you.





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

DIGITAL SIGNATURES:

A digital signature is an ~~electronic method of verifying~~ the authenticity of a digital document or message. It is a cryptographic technique that provides integrity, non-repudiation, and authenticity. Digital signatures use a combination of a private key and a public key to encrypt and decrypt data. The private key is kept secret by the owner, while the public key is available to anyone who wants to verify the signature.

The digital signature process involves the following steps:

- The sender creates a digital document or message.
- The sender's private key is used to create a unique digital signature for the document.
- The digital signature is attached to the document.
- The recipient uses the sender's public key to verify the digital signature.
- If the digital signature is valid, the recipient can be sure that the document has not been tampered with and that it came from the sender.

BENEFITS OF DIGITAL SIGNATURES:

Digital signatures offer several benefits over traditional signatures:

- 1) Security: Digital signatures use cryptography to ensure that the signature is secure and cannot be tampered with.
- 2) Efficiency: Digital signatures can be created and verified quickly and easily.
- 3) Non-repudiation: Digital signatures provide non-repudiation, which means that the sender cannot deny having signed the document.
- 4) Cost Savings: Digital signatures eliminate the need for paper-based signatures and reduce the cost of printing, scanning, and shipping documents.

APPLICATIONS OF DIGITAL SIGNATURES:

Digital signatures have several applications, including:

- 1) Legal Documents: Digital signatures can be used to sign legal documents such as contracts, agreements, and deeds.
- 2) Financial Transactions: Digital signatures can be used to sign and verify financial transactions such as online banking, stock trading, and e-commerce.
- 3) Government Transactions: Digital signatures can be used to sign and verify government transactions such as tax returns, applications, and licenses.

CONCLUSION:

In conclusion, authentication and digital signatures are essential tools for ensuring the security and authenticity of digital transactions and communications. By using these techniques, individuals and organizations can protect their data and reduce the risk of fraud and cybercrime.





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

How to Install Kali Linux:

<https://www.kali.org/docs/wsl/wsl-preparations/#setting-up-wsl>

We can check our build version by pressing WIN+R and then typing in 'winver'. We will be shown a page like this:



What we are looking for is the number after 'OS Build'. If we see the number '19041' or a number higher, like is shown in the screenshot, we note this down and proceed to [Windows 10](#). If we see a number lower than this, we will be unable to use WSL2.

Setting up WSL



bashar_sh77@uomustansiriyah.edu.iq

Page 49 of 51





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

Windows 11

To install and setup WSL we can run the command `wsl --install -d kali-linux`. We may need to perform a computer restart, but once complete we will have the latest version of Kali Linux installed.

Windows 10

We will need to open Powershell as Administrator for the following.

We first will run the following command to enable WSL:

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```

If we are setting up WSL2 we will want to skip to the next section. If we don't then we can restart now and install Kali through the Microsoft Store.

Setting up WSL2

We can choose to enable WSL2 if we are the proper version of Windows. [WSL2](#) enables some very helpful features and so it is recommended to enable it if your system supports it.

If we have a Windows 11 system or are build 19041 and higher, which we discovered in the previous section, then we are able to skip to [enabling WSL2 with set version](#). Otherwise, we press WIN+R and launch 'winver' again. This time we are checking to see if we fall under the following:

For x64 systems: Version 1903 or later, with Build 18362 or later. For ARM64 systems: Version 2004 or later, with Build 19041 or later.





CRYPTOGRAPHY LECTURES

PROF. DR. BASHAR AL-ESAWI – 2023/2024

Mustansiriyah University – College of Science -Cybersecurity Branch

Enabling WSL2 with set version

We first check to see if we are already set to WSL2 with the following command ran through Powershell:

```
Get-ItemPropertyValue `
    -Path HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Lxss `
    Name DefaultVersion
```

The output will either be a 1 or a 2. If it is two then we can leave it be. Otherwise we run the following command:

```
wsl --set-default-version 2
```

Once we are happy we can choose to install Kali through the command `wsl --install -d kali-linux` if we are using Windows 11 or otherwise we can install it through the Microsoft Store!

Enabling WSL2 manually

```
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
```

After this we restart our computer.

Next we download an update for Windows from the following: [Here if we are on an x64 system](#) or [Here if we are on an ARM64 system](#)

Now we launch Powershell as administrator and run the following command:

```
wsl --set-default-version 2
```

Now just install Kali through the Microsoft Store to finish setup!

Thanks, all Students

BMN



bashar_sh77@uomustansiriyah.edu.iq

Page 51 of 51

