LAB #16 encryption & decryption concept

Class M02 Name Trần Đăng Khoa
Students ID B2014926

Check system environment

Resource	Detail
Client	OS: Window Browser: Chrome Language: Python 3,Java, C
Web Server	None

Choose one practice model for exercise

- Exercise model A => Describe encryption & decryption full process and explain

Exercise model A => Describe encryption & decryption full process and explain

- Encryption

+ Plain text: The original, human-readable data that needs protection is known as plain text.

+ Key generation: a key is generated through an algorithm. In symmetric key cryptography, the same key is used for both encryption and decryption, there is a pair of keys: a public key for encryption and a private key for decryption.

+ Encryption Algorithm: The plain text is combined with the encryption key using an encryption algorithm. This process results in ciphertext, an unreadable and seemingly random sequence of characters. + Ciphertext: the result of the encryption process is

ciphertext, which is
the information to be transmitted or stored. It should be computationally infeasible to derive the original plain text from the ciphertext without the proper key.

- Decryption

+ Ciphertext: The encrypted data, or ciphertext, is received by the intended recipient

+ Key Input: In symmetric key cryptography, the same key used for encryption is used for decryption. In asymmetric key cryptography, the recipient uses their private key for decryption.

+ Decryption algorithm: The decryption key is applied to the ciphertext

using a decryption algorithm. This process transforms the ciphertext back into the original plain text.

+ Plain text: the result of the decryption process is the original, readable plain text.

- Example:

+ Encryption Process:

(+) Plain text: Suppose we have the plain text message: "HELLO" (+) Key generation: Generate a secret key, for example, " KEY123"

(+) Encryption Algorithm: Use a symmetric encryption algorithm to combine the plain text and the secret key. The result might be something like: "VczUq+uM5rXJ8IKFbPvIUw=="

(+) Ciphertext: The final encrypted message, or ciphertext, is "VczUq+uM5rXJ8IKFbPvIUw==". This is what would be transmitted of stored.

+ Decryption Process:

(+) Ciphertext: suppose we receive the ciphertext " VczUq+uM5rXJ8IKFbPvIUw==".

(+) Key input: Use the same secret key, "KEY123," for decryption. (+) Decryption Algorithm: Apply the decryption algorithm with the secret key to the ciphertext. The result is the original plain text: "HELLO".

(+) Plain text: The decrypted message is " HELLO", which is the original information.

- Explanation:

+ Security: The security of this process relies on keeping the secret key "KEY123", confidential. Without the key, it should be computationally to derive the original plain text from the ciphertext.

+ Use case: This example represents a simplified scenario. In real-world applications, more robust encryption algorithm and key management would be employed.

• Exercise model B => Describe Secret Key Cryptography and public key - Secret Key cryptography: also known as Symmetric Cryptography, operates on the principle of using a single shared secret key for both encryption and decryption. In this model, the communicating parties must

agree upon and securely distribute the secret key before initiating secure communication. The sender employs this secret key to transform plaintext into ciphertext, and the recipient uses the same key to decrypt and retrieve the original message. Example of secret key cryptography algorithms include DES(data encryption Standard) and AES(Advanced encryption standard). While this approach is efficient for large-scale data encryption, the challenge lies in securely distributing the secret key among all involved parties.

- Public Key: In contrast, public key cryptography, or asymmetric key cryptography, introduces a pair of mathematically related keys for secure communication: a public key and a private key. Each user possesses their unique pair of keys. The public key can be openly shared, allowing anyone to encrypt messages or verify digital signatures, while the corresponding private key is kept confidential. This model eliminates the need for secure key distribution, making it particularly advantageous. Public key cryptography is commonly used for secure key exchanges and digital signatures. Examples of algorithms in this category include RSA(Rivest Shamir-Adleman) and ECC(Elliptic curve Cryptography). Despite being generally slower for large-scale data encryption, public key cryptography plays a crucial role in ensuring secure communication over insecure channels. In practice, a combination of both symmetric and asymmetric cryptography is often employed to address various security requirements.