Họ và tên: Nguyễn Bắc Giang

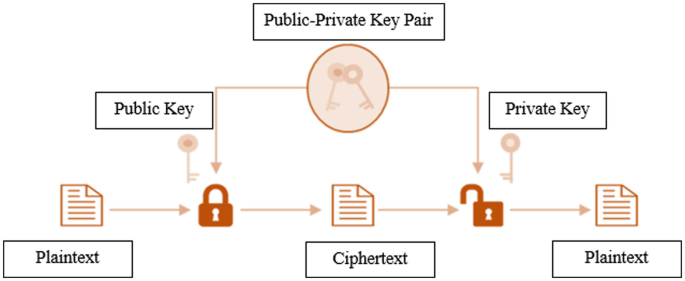
Lớp: T2208E

Question 1:

* 1. Concept

Asymmetric encryption uses a pair of keys: a public key for encryption and a private key for decryption. Data encrypted with the public key can only be decrypted with the corresponding private key, providing a secure way to share information without sharing sensitive decryption keys

* 1. Encoding and Decoding Diagram



* 1. Comparison
* Symmetric encryption uses a single key for both encryption and decryption.
* Asymmetric encryption uses a pair of keys (public and private) for encryption and decryption, providing a higher level of security but generally slower than symmetric encryption.
  1. A program to encrypt and decrypt using asymmetric by C# language

Question 2:

1. Concept of Security Vulnerability:

A security vulnerability refers to a weakness in a system that could be exploited by attackers to compromise the system's integrity, availability, or confidentiality. Vulnerabilities can exist in software, hardware, configurations, or even in human behavior. They can be unintentional flaws or errors in design, implementation, or operation that hackers can exploit to gain unauthorized access, steal data, disrupt services, or cause other types of damage.

1. SQL Injection and Prevention:

SQL Injection is a common and severe security vulnerability that occurs when attackers insert malicious SQL code into input fields of a web application, tricking the application into executing unintended SQL commands. This could lead to unauthorized access to databases, data manipulation, data exfiltration, or even complete system compromise.

Prevention of SQL Injection:

* Use Parameterized Queries (Prepared Statements): Instead of directly inserting user input into SQL queries, use parameterized queries or prepared statements provided by your database or programming language. This separates SQL code from user input, preventing direct execution of injected commands
* Input Validation and Sanitization: Validate and sanitize user inputs to ensure they match the expected format and type. Reject or sanitize inputs that contain SQL special characters that could be used in an injection attack
* Least Privilege Principle: Limit database user privileges to only what is necessary. Avoid using an account with administrative privileges in the application code.
* Escaping Special Characters: Escape or encode special characters (like quotes and semicolons) in user input before using them in SQL queries. This ensures these characters are treated as literals and not as part of the SQL command
* Regular Security Audits and Updates: Regularly audit your codebase and database configurations for potential vulnerabilities. Keep software, frameworks, and libraries updated to patch known security issues.

Question 3:

Using the password authentication method, let's know a program in C# language that encrypts

(SHA-256) a user's password string with the following information:

Username: Admin, Password: Awdx!@#$%xdwa



