Bizeit University
Faculty of Engineering & Technology
Computer Science Department
Encryption Theory Comp 438
Final Project

Each group of students is required to select **one of the following suggested topics** and develop a project. The project should address the stated objectives and provide a comprehensive report discussing the methodology, solutions, and analysis.

1- **Steganography and Encryption Hybrid System**

This project involves developing a hybrid system that combines **Least Significant Bit (LSB) Steganography** with **symmetric encryption algorithms** like the Hill Cipher or Vernam Cipher. The system will first encrypt a message using a chosen symmetric cipher and then embed the encrypted message into an image using LSB steganography. This dual-layered approach enhances the security and privacy of hidden information. The expected workflow for the encoding and decoding as follows:

a. Encrypt the plaintext message using a symmetric encryption algorithm. Examples: **Hill Cipher:** Uses a key matrix for modular arithmetic encryption. **Vernam Cipher:** XOR operation with a randomly generated key.
b. LSB Steganography: Embed the encrypted message into the least significant bits of an image's pixel data. Supported formats BMP or PNG (due to lossless compression).
c. Decryption and Extraction, extract the hidden bits from the image to retrieve the encrypted message. Decrypt the message using the same symmetric cipher and key.
d. Security Analysis. Evaluate the security benefits of combining encryption with steganography. Analyze resistance to attacks (e.g., visual inspection, histogram analysis, brute force decryption).

2- **Cipher Analyzer and Breaker**

This project involves building a tool that can analyze encrypted messages and attempt to decrypt them by identifying the underlying classical encryption technique (e.g., Vigenère Cipher, Hill Cipher). The tool will implement cryptanalysis techniques such as **frequency analysis**, **brute force**, and **pattern matching** to break the encryption. Don’t apply cryptanalysis technique on Caeser cipher. The expected workflow for the following project as follows:

a. Users input the encrypted message and optional hints (e.g., suspected cipher type or language).
b. The tool detects the likely encryption method based on patterns or user hints.
c. The tool applies the appropriate cryptanalysis techniques to decrypt the ciphertext.

  d. The tool displays the decrypted plaintext along with details about the detected cipher and the key used.

3- **Dynamic One-Time Pad Generator**

This project focuses on developing a secure messaging application that uses the **One-Time Pad (OTP)** encryption algorithm. OTP is a theoretically unbreakable encryption method when used correctly, but it requires secure key management. The application will dynamically generate and securely exchange OTP keys between the sender and receiver. The project will demonstrate how OTP works in practice and address challenges related to key generation, distribution, and synchronization.

  a. Generate a random OTP key for each message, ensuring no key reuse.
  b. Implement a secure method (e.g., Diffie-Hellman or public key cryptography) to exchange keys between the sender and receiver.
  c. Use the OTP key to encrypt the plaintext and decrypt the ciphertext.
  d. Handle key mismatches or synchronization issues during communication.

4- **Image Encryption with Hill Cipher**

This project involves encrypting images using the **Hill Cipher**, a classical encryption algorithm based on matrix multiplication. The group will represent the pixel data of an image as a matrix and apply the Hill Cipher to encrypt the image. The encrypted image will be evaluated for its strength by attempting decryption attacks, such as **brute force or known-plaintext attacks**. The process will be as follows:

  a. Convert the image's pixel values into a matrix (e.g., grayscale or RGB values).
  b. Apply the Hill Cipher encryption using a key matrix.
  c. Reconstruct the encrypted image and analyze its visual distortion.
  d. Decryption and evaluate the encryption's robustness.

5- **Developing a Secure File Storage System**

Create a file encryption and decryption tool using a symmetric encryption algorithm like DES or RC4. The tool should include a user-friendly interface, support various file types, and focus on speed and security.

  a. Convert the image's pixel values into a matrix (e.g., grayscale or RGB values).

b. Apply the Hill Cipher encryption using a key matrix.
c. Reconstruct the encrypted image and analyze its visual distortion.
d. Attempt decryption and evaluate the encryption's robustness.

6- Custom Cryptosystem Development

In this project, the group will design and implement a new **cryptosystem** by combining features of **classical encryption techniques** (e.g., Caesar Cipher, Vigenère Cipher) and **modern encryption methods** (e.g., DES, AES). The goal is to create a unique encryption algorithm that improves security and demonstrates innovation. The process can be as follows:

a. Start by applying a classical cipher like a **Vigenère Cipher** for initial substitution. This step adds an extra layer of obfuscation to the plaintext.
b. Apply a modern encryption algorithm, such as **SDES (Simplified DES)** or **AES**, to the output of the classical cipher. This ensures strong encryption with larger key spaces and resistance to modern attacks.
c. Shuffle the encrypted output using a transposition technique to further enhance confusion and diffusion. Generate a unique key for the cryptosystem by combining elements from both classical and modern key-generation techniques (e.g., combining a user-provided passphrase with a hash function like SHA-256).
d. Reverse (Decrypt) each step (transposition, modern cipher, classical cipher) using the same key.

7- Password Manager Using DES

This project involves creating a **password manager** application that securely stores user passwords. The passwords will be **encrypted** using the **Data Encryption Standard (DES)** algorithm before being saved. The application will include features to:

a. **Store Passwords Securely:** Encrypt passwords with DES and save them to a secure database or file.
b. **Retrieve Passwords Safely:** Decrypt the stored passwords when needed, using the correct encryption key.
c. **User-Friendly Interface:** Allow users to add, view, and manage their passwords easily.

The goal of this project is to demonstrate how encryption can protect sensitive information like passwords and ensure security in a practical application.

Bizeit University
Faculty of Engineering & Technology
Computer Science Department
Encryption Theory Comp 438
Final Project

8- Cryptanalysis Simulation

This project focuses on simulating the cryptanalysis process for **DES** (Data Encryption Standard) or **SDES** (Simplified Data Encryption Standard). The objective is to implement and demonstrate how cryptanalysis techniques can be used to break these encryption algorithms, displaying their vulnerabilities. The project includes:

a. **Brute Force Attack:** This method involves trying all possible keys until the correct one is found. For SDES, where the key space is smaller, brute force is computationally feasible.
b. **Differential Cryptanalysis:** This method examines how differences in plaintext inputs affect the differences in ciphertext outputs, enabling the discovery of the encryption key.
c. **Known-Plaintext Attack (KPA):** In this attack, the attacker has access to some plaintexts and their corresponding ciphertexts.
d. **Chosen-Plaintext Attack (CPA):** The attacker can choose arbitrary plaintexts and obtain their corresponding ciphertexts to analyze the encryption process.
e. **Chosen-Ciphertext Attack (CCA):** The attacker can choose ciphertexts and observe the decrypted plaintexts to gather information about the decryption process or key.
f. **Meet-in-the-Middle Attack:** This attack targets encryption algorithms with multiple encryption layers, such as 2DES (double DES).

Each project report must include:

A- **Problem Statement:** Define the problem and its importance.
B- **Objectives:** List key goals.
C- **Approach:** Describe methods with a flowchart.
D- **Algorithms/Techniques:** Explain implementation with pseudocode.
E- **Tools:** Specify programming languages and libraries used.
F- **System Design:** Break into smaller modules for clarity.
G- **Testing:** Detail scenarios for functionality testing.
H- **Comparison:** Evaluate the system against similar solutions using metrics.

Focus on creativity and security in your solutions. Be sure to demonstrate your project during evaluation, showcasing the system's practical applications and robustness.