**INTRODUCTION TO INFORMATION SECURITY  
LAB – 2 Report**

NAME: - PRANAV MURALI

CWID: - A20555824

**Task 1: Frequency Analysis**

Frequency of the letters in the cipher text is calculated.In A screenshot of a computer

Description automatically generated

In order to reduce the computations we also need to check frequency of Trigrams

A screenshot of a computer

Description automatically generated

We see that ‘oxp’ has the most number of trigrams. We can replace it with ‘THE’ which has the highest trigram in English letters.

A screenshot of a computer

Description automatically generated

On making the required changes the cipher text is then converted into plaintext.

A screenshot of a computer

Description automatically generated

**Task 2: Encryption using Different Ciphers and Modes**

1. **DES CFB Encryption**

In DES CFB (Cipher Feedback) mode, the encryption process involves using the output of the DES encryption of the previous ciphertext block as feedback to encrypt the current plaintext block, creating a self-synchronizing stream cipher.

A screenshot of a computer

Description automatically generated

1. **DES CBC Encryption**

In DES CBC (Cipher Block Chaining) mode, each plaintext block is XORed with the ciphertext of the preceding block before undergoing DES encryption, enhancing security by introducing dependency between blocks and mitigating patterns in the encrypted output.

A screenshot of a computer

Description automatically generated

1. **BF OFB Encryption**

In Blowfish (BF) Output Feedback (OFB) mode, the encryption process involves generating a stream of pseudo-random bits using the Blowfish block cipher in output feedback mode, which is then XORed with the plaintext to produce the ciphertext, providing confidentiality and ensuring that the same plaintext block encrypted at different times results in different ciphertexts.

A screenshot of a computer

Description automatically generated

1. **BF CFB Encryption**

In Blowfish (BF) Cipher Feedback (CFB) mode, the encryption process utilizes the Blowfish block cipher to produce a stream of pseudo-random bits, which are XORed with the plaintext to generate ciphertext, creating a self-synchronizing stream cipher with an inherent block feedback mechanism for improved security.

A screenshot of a computer

Description automatically generated

1. **BF CBC Encryption**

In Blowfish (BF) Cipher Block Chaining (CBC) mode, each plaintext block is XORed with the ciphertext of the previous block before undergoing Blowfish encryption, providing a chaining mechanism that introduces dependencies between blocks, enhancing security by preventing identical plaintext blocks from producing the same ciphertext and adding resistance to patterns in the encrypted output.

A screenshot of a computer

Description automatically generated

1. **AES 256 ECB Encryption**

In AES (Advanced Encryption Standard) with a key size of 256 bits and using Electronic Codebook (ECB) mode, each 128-bit block of plaintext is independently encrypted with the same key. One significant observation is that identical plaintext blocks will result in the same ciphertext blocks, potentially revealing patterns and correlations in the encrypted data. This lack of diffusion and pattern preservation makes AES-256 ECB mode vulnerable to certain types of attacks.

A computer screen shot of a computer

Description automatically generated

1. **AES 192 OFB Encryption**

In this encryption, it employs the IV and a key for encryption in the OFB mode. Ciphertext won't be produced if plaintext is blank. The ciphertext is produced via parallel encryption. It is a stream cipher in which the plaintext and cipher block is XORed to produce the ciphertext.

A screenshot of a computer

Description automatically generated

1. **AES 128 CBC Encryption**

In AES-128 CBC (Cipher Block Chaining) mode, a notable observation is that each ciphertext block is dependent on both the corresponding plaintext block and the previous ciphertext block. This chaining effect ensures that identical plaintext blocks result in different ciphertext blocks, providing enhanced security by preventing patterns and facilitating a more effective diffusion of information throughout the encryption process.

A screenshot of a computer

Description automatically generated

1. **DES OFB Encryption**

In DES (Data Encryption Standard) using Output Feedback (OFB) mode, a key observation is that the encryption process involves generating a stream of pseudo-random bits by repeatedly encrypting the Initialization Vector (IV) with DES. The resulting stream is then XORed with the plaintext to produce the ciphertext. One important feature is that errors or changes in the ciphertext only affect the corresponding bits in the decrypted output, demonstrating the self-synchronizing property of OFB mode and providing resistance against tampering.

A screenshot of a computer

Description automatically generated

**Task 3: Encryption Mode ECB vs. CBC**

The picture was given to Encrypt.

A red oval with a white background

Description automatically generated

The first 54 bytes contain the header information about the picture, so this step extracts the same.

A screenshot of a computer

Description automatically generated

Encrypted the image using ECB and CBC.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Combining Header and data together.

A screenshot of a computer

Description automatically generated

Output Files:-

A screenshot of a computer

Description automatically generated

**ECB**

A screenshot of a computer

Description automatically generated

**CBC**

A screen shot of a computer

Description automatically generated

**MY IMAGE**

****

**ECB**

**A screen shot of a computer

Description automatically generated**

**CBC**

**A screen shot of a computer

Description automatically generated**

**Task 4: Padding  
CBC Encryption and Decryption**A screenshot of a computer

Description automatically generated

**Encryption**

A screenshot of a computer screen

Description automatically generated

**Decryption**

F1.txt= 5 bytes file F1\_e.txt= f1 file encrypted F1\_d= f1 file decrypted **||** f2.txt= 10 bytes ……………. **||** f3.txt = 16 bytes……..

A screenshot of a computer

Description automatically generated

**Hexdump**

A screenshot of a computer

Description automatically generated

**ECB Encryption and Decryption**

A screenshot of a computer

Description automatically generated

**Hexdump**

A screenshot of a computer

Description automatically generated

**OFB Encryption and Decryption**A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Hexdump**

A screenshot of a computer

Description automatically generated

**CFB Encryption and Decryption**A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**Hexdump**

A screenshot of a computer

Description automatically generated

In this manner we compare the size of the original file, the encrypted text and the decrypted text to analyze the differences between the modes of encryption and decryption.

**Task 5: Error Propagation Corrupted Cipher Text**

Create a text file that is at least 1000 bytes long.

A screenshot of a computer

Description automatically generated

**CBC 128 Encryption and decryption**

A screenshot of a computer

Description automatically generated

A single bit of the 55th byte in the encrypted file corrupted

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**ECB 128 Encryption**A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Observations between all encryption modes.**

|  |  |
| --- | --- |
| ECB | All but 1 corrupted block |
| CBC | All but 2 corrupted blocks (N-2) |
| CFB | All but 2 corrupted blocks |
| OFB | All but 1 corrupted block |

The propagation of errors across multiple blocks occurs due to the block modes' reliance on the preceding block, which acts as the initialization for the current encryption. In contrast, OFB and ECB, which lack dependency on previous blocks during encryption, result in the corruption of only one block. These variations in the encryption process enable the prediction of the extent of damaged data.

**Task 6: Initial Vector (IV)**

**Uniqueness of the IV**

**SAME iV**

As demonstrated below, employing identical Initialization Vectors (IVs) leads to consistent outputs for the cipher messages, compromising security in the process.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Different iV**

As demonstrated below, employing identical Initialization Vectors (IVs) leads to consistent outputs for the cipher messages, compromising security in the process.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**Common Mistake: Use the Same IV**

We know: P1⊕ C1 = P2⊕C2

Hex(P1) = 546869732069732061206b6e6f776e206d657373616765210a

Hex(C1) = a469b1c502c1cab966965e50425438e1bb1b5f9037a4c15913

Hex(C2) = bf73bcd3509299d566c35b5d450337e1bb175f903fafc15913

**A screenshot of a calculator

Description automatically generated**

Hex(P2) = ?

Using P1 ⊕ C1 = P2 ⊕C2

**A screenshot of a computer

Description automatically generated**

Hex(P2) = ?

Using P1 ⊕ C1 = P2 ⊕C2 Hex(P2) = 4f726465723A204C61756E63682061206D647373696C6521 P2 Text -> Order: Launch a missile!

A screenshot of a calculator

Description automatically generated

A screenshot of a computer

Description automatically generated

**Common Mistake: Use a Predictable IV**

Eve must be aware of the IV that Bob will use for his next encryption because Bob employs predictable iv’s. Ciphertext(C1): bef65565572ccee2a9f9553154ed9498

iv used on P1: 31323334353637383930313233343536

Next iv used – 31323334353637383930313233343537

**A screenshot of a calculator

Description automatically generated**

**A screenshot of a calculator

Description automatically generated**

**A screenshot of a computer

Description automatically generated**