## Applied Crypto

## Exploring Modes of Operations for Block Ciphers and the Meet in the Middle Attack

**Problem 1** Explore different modes of operation through manual encryption and decryption.

**Objective:** Gain practical insights into the modes of operation discussed in class by manually encrypting and decrypting a given plaintext.

**1. Encryption Setup:**

a) Use a hypothetical block cipher with a block length of 4, defined as $E_k(b_1b_2b_3b_4) = (b_2b_3b_1b_4)$.

b) Convert English plaintext into a bit string using the table provided (A=0000 to P=1111). Assume we have a language that uses 16 letters only. If we want a more realistic exercise, we can have block size of 5 bits that can represent 32 cases (more than 26 letters) or even size of 8 bits that use the ASCII. Here we just use the size of 4 bit.

| A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|
| 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
| I | J | K | L | M | N | O | P |
| 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |

**2. Encryption Modes:** Encrypt the plaintext 'FOO' using the following modes. Convert the final ciphertexts into letters. Show your work

a) ECB (Electronic Codebook)

b) CBC (Cipher Block Chaining) with IV=1010

c) CTR (Counter) with ctr=1010

**3. Decryption Task:**

Assume the ciphertexts from 2 are received by an intended receiver.

a) Manually decrypt each ciphertext to recover the original plaintext. Show your work.

**Problem 2 Implementing a Meet-in-the-Middle Attack on a Mini Block Cipher**

**Project Overview:**

This project involves students in the implementation and analysis of a meet-in-the-middle (MITM) attack against a simplified block cipher called mini block cipher based on the idea of the SAES (FYI, it is better to use the simplified DES). The primary goal is to provide hands-on experience with the MITM attack, showcasing its effectiveness against certain cryptographic algorithms and understanding why modern ciphers like AES are designed to be immune to such attacks.

MITM or "Meet-in-the-Middle Attack" is an exhaustive key search attack[1]. It is a cryptanalytic attack applicable to ciphers based on composition of multiple rounds of substitutions and permutations. It works by finding plaintext-ciphertext pairs that map to the same intermediate value after partial encryption/decryption[2].

SAES starts with the key expansion, then works on encryption to get ciphertext and on decryption to recover the plaintext. The key expansion generates three keys. The first key, Key0, is used for the add round key to the plaintext. The second key, Key1, is used to perform Round 1 transformation on state, defined as encrypt_round1(). The third key, Key2, is used to perform Round 2 transformations on state, defined as encrypt_round2(). For decryption, it is reverse. Key2 is used to perform inverse Round 2 transformations on ciphertext, defined as decrypt_round2(). Key1 is used to perform inverse Round 1 transformations on state, defined as decrypt_round1(). So, the pseudo-code for SAES would be the following. It is assumed the key size is 16 bit.

```
1. Get (Key0, Key1, Key2) from Key K using the key expansion.
2. Two rounds of Encrypt to get the ciphertext.
     I.   AddRoundKey Key0
     II.  encrypt_round1() using Key1
            i.   Subsititute()
            ii.  Shift()
            iii. Mix()
            iv.  AddRoundKey()
     III. encrypt_round2() using Key2
            i.   Subsititute()
            ii.  Shift()
            iii. AddRoundKey()

3. Two rounds of decryption to recover the plaintext
     I.   decrypt_round2() using Key 2
            i.   AddRoundKey()
            ii.  Shift()
            iii. Subsititute()
     II.  decrypt_round1() using Key1
```

[1] https://en.wikipedia.org/wiki/Meet-in-the-middle_attack

[2] For example, https://youtu.be/S-EhbhDXUwM (It is a bit long…)

```
                i.   AddRoundKey()
                ii.  Mix()
                iii. Shift()
                iv.  Subsititute()
        III.    AddRoundKey Key0
```

For the project, we need to modify the pseudo code to fit our class project. The pseudo-code for this mini block cipher based on the SAES is as follows:

```
    1. Get (Key0, Key1, Key2) from Key K using the key expansion.

    2. Two rounds of Encrypt to get the ciphertext.
        I.      encrypt_round1() using Key1 on plaintext, P, and get intermediate state X
                i.   Subsititute()
                ii.  Shift()
                iii. Mix()
                iv.  AddRoundKey()
        II.     encrypt_round2() using Key2 on intermediate state X, and get the ciphertext C.
                i.   Subsititute()
                ii.  Shift()
                iii. AddRoundKey()

    3. Two rounds of decryption to recover the plaintext
        I.      decrypt_round2() using Key 2 on the ciphertext, C, and get intermediate state, Y
                i.   AddRoundKey()
                ii.  Shift()
                iii. Subsititute()
        II.     decrypt_round1() using Key1 on Y to get plaintext P.
                i.   AddRoundKey()
                ii.  Mix()
                iii. Shift()
                iv.  Subsititute()
```

The meet in the middle attack strategy to mini block cipher:

A. Calculate X = encrypt_round1(Key1, P)
B. Calculate X’ = decrypt_round2(Key2, C).
C. Find out a pair (Key1, Key2) such at X = X’
D. For one specific (P, C), there will be many matched pairs, we need to use another plaintext and ciphertext pair to eliminate some of the matched pairs. Ideally, we shall get one matched key pair. This pair can be used to get plaintext from any cyphertext. In other words, we cracked the block cipher.

**Project Tasks**

**Task 1: Implementing Mini Block Cipher with key size 16 bit and block size 16 bit:**

a) Students will implement the Mini Block Cipher encryption and decryption functions (1, 2I, 2II, 3I, 3II) using jupyter notebook.
b) Make at least ten pairs of plaintexts and ciphertexts.

**Task 2: Meet-in-the-Middle Attack Implementation:**

a) Students need to implement the meet in the middle attack strategy to mini block cipher (a-d).
b) Show key pair(s) that works for the pair of plaintext and ciphertext from task1 (b). Ideally, it should have only one key pair works.

**Task 3: Analyze the time and memory complexity of the attack compared with the naive exhaustive key search.**

a) What is the key space for the mini block cipher?
b) Image the mini block cipher is executed twice to generate a cipher text. It is called double mini cipher block. We need a key in 32 bits. The first 16 to the first mini block cipher, the remaining 16 to the second mini block cipher. The meet in the middle attack is to match the state for the first encryption of mini block cipher and the second decryption mini block. How many operations are needed to such attack?
c) If we do exhaustive key search for the double mini block cipher, how many operations are needed?
d) What is the tradeoff for the MITM attack (speed, memory, etc.)?

**Deliverables and Submission**

- Written Report
- Code: Use Jupyter Notebook

**Evaluation Criteria**

- Correctness.
- Attack Strategy.
- Analysis.

**Grading rubrics**

| | Max Point | Expectations from the description. Show your work |
|---|---|---|
| **Problem 1** | **4** | |
| 2a | 0.5 | Accurately perform ECB encryption for 'FOO', converting ciphertext into letters. |
| 2b | 1 | Accurately perform CBC with IV=1010 encryption for 'FOO', converting ciphertext into letters. |
| 2c | 1 | Accurately perform CTR with IV=1010 encryption for 'FOO', converting ciphertext into letters. |
| 3a | 1.5 | Successfully decrypts each ciphertext, demonstrating understanding of decryption processes and converting plaintext back into letters. |

| | | |
|---|---|---|
| **Problem 2** | **8** | |
| Task 1a | 2.5 | Accurate implementation of Mini Block Cipher functions in Python, following provided specifications. |
| Task 1b | 0.5 | Creation of at least ten distinct plaintext-ciphertext pairs, clearly presented. |
| Task 2a | 2 | Correct implementation of meet-in-the-middle attack strategy on the mini block cipher, demonstrating understanding. |
| Task 2b | 1 | Clear presentation of meet-in-the-middle attack results using plaintext-ciphertext pairs from Task 1b. |
| Task 3a | 0.5 | Correct Calculation of key space for the mini block cipher. |
| Task 3b | 0.5 | Correct calculation of operations required for the meet-in-the-middle attack on the double mini block cipher |
| Task 3c | 0.5 | Correct calculation of operations needed for exhaustive key search on the double mini block cipher. |
| Task 3d | 0.5 | Clear discussion of trade-offs between meet-in-the-middle attack and exhaustive key search, demonstrating critical thinking. |