Topic: Symmetric Ciphers. Stream Ciphers. Block Ciphers.

Forors Valentin

November 24, 2022

1 Introduction

Symmetric Cryptography deals with the encryption of plain text when having only one encryption key which needs to remain private. Based on the way the plain text is processed/encrypted there are 2 types of ciphers:

1. Stream ciphers:
   - The encryption is done one byte at a time.
   - Stream ciphers use confusion to hide the plain text.
   - Make use of substitution techniques to modify the plain text.
   - The implementation is fairly complex.
   - The execution is fast.
2. Block ciphers:
   - The encryption is done one block of plain text at a time.
   - Block ciphers use confusion and diffusion to hide the plain text.
   - Make use of transposition techniques to modify the plain text.
   - The implementation is simpler relative to the stream ciphers.
   - The execution is slow compared to the stream ciphers.

2 Objectives

1. Get familiar with the symmetric cryptography, stream and block ciphers.
2. Implement an example of a stream cipher.
3. Implement an example of a block cipher.
4. The implementation should, ideally follow the abstraction/contract/interface used in the previous laboratory work.
5. Please use packages/directories to logically split the files that you will have.
6. As in the previous task, please use a client class or test classes to showcase the execution of your programs.

3 Implementation description:

RC4 Cipher: RC4 (also known as Rivest Cipher 4) is a form of stream cipher, that encrypts messages

one byte at a time. It relies on:

1. Key inputs. This tool generates an eight-bit number (cipher) that's impossible to guess.
2. Keystreams. The cipher scrambles plain text.
3. Product. An X-OR operation combines the keystream with the cipher.

RC4 relies on two mathematical concepts:

1. KSA: A key-scheduling algorithm initializes the process in an array typically referred to as "S." That "S" is processed 256 times, and bytes from the key are mixed in too.
2. PRGA: Data is fed in byte by byte, and a mathematical model modifies it. The model looks up values, add them to 256, and uses the sum as the byte within the keystream. It swaps each element with another at least once every 256 rounds.

Here are presented the two functions KSA and PRGA, that were implemented for the RC4 cipher:

```
static int [] KSA(int[] S, final String key) {
    for (int i = 0; i < N; ++i) {
        S[i] = i;
    }

    int j = 0;
    int kLen = key.length();

    for (int i = 0; i < N; ++i) {
        j = (j + S[i] + (int)key.charAt(i % kLen)) % N;
        temp = S[i];
        S[i] = S[j];
        S[j] = temp;
    }

    return S;
}
```

In this function we perform the initialization of the initial state S with values from 0 to -1 and the initialization of the iterator j with the initial value and length of the text kLen. We repeat (N-1) times the permutation in state S, define the new value of the iterator j and make a permutation in state S with iterators i and j using a variable temp.

```
static String PRGA(int[] S, String keyStream, final int textLen){
    int i = 0, j = 0;
    for (int k = 0; k < textLen; ++k) {
        i = (i + 1) % N;
        j = (j + S[i]) % N;
        temp = S[i];
        S[i] = S[j];
        S[j] = temp;
        keyStream += (char) ((S[(S[i] + S[j]) % N]));
    }

    return  keyStream;
}
```

In this function we initialize the iterators with an initial value and determine new values of iterators i and j. After this we perform the permutation in state S with iterators i and j using temp variable and generate the key.

RC4 relies on random number generators. But unlike other stream ciphers, RC4 doesn't need linear-feedback shift registers. Despite its complexity, RC4 is remarkably fast.

DES Cipher:

DES stands for Data Encryption Standard. It is a symmetric-key block cipher algorithm used to encrypt and decrypt data. It is based on LUCIFER (also known as Feistel block cipher algorithm) which is a direct predecessor of the DES algorithm. It encrypts the data using the first key (k1), decrypts the data by using the second key (k2) and again encrypts the data by using the third key (k3). It is used still but considered as a legacy algorithm.

The algorithm performs 16 rounds of encryption and for each round, a unique key is generated. Before moving to the steps, it is important to know that in plaintext the bits are labeled from 1 to 64 where 1 is the most significant bit and 64 is the least significant bit.

The algorithm includes the following steps:

1. As an input we provide 64-bit plain text.
2. The text is parsed into a function called the Initial Permutation (IP) function.
3. The initial permutation (IP) function breaks the plain text into the two halves of the permuted block. These two blocks are known as Left Plain Text (LPT) and Right Plain Text (RPT).
4. The 16 round encryption process is performed on both blocks LPT and RPT. The encryption process performs the following:
   - Key Transformation
   - Expansion Permutation
   - S-Box Permutation
   - P-Box Permutation
   - XOR and Swap
5. After performing the encryption process, the LPT and RPT block are rejoined. After that, the Final Permutation (FP) is applied to the combined block.
6. Finally, we obtain the 64-bit ciphertext of the plaintext.

4 Conclusion

Symmetric encryption is a type of encryption where only one key (a secret key) is used to both encrypt and decrypt electronic data. The entities communicating via symmetric encryption must exchange the key so that it can be used in the decryption process. There are two types of symmetric encryption algorithms:

- **Block algorithms.** Set lengths of bits are encrypted in blocks of electronic data with the use of a specific secret key. As the data is being encrypted, the system holds the data in its memory as it waits for complete blocks.
- **Stream algorithms.** Data is encrypted as it streams instead of being retained in the system's memory.

While working on this laboratory work, I got familiar with the symmetric ciphers (RC4 and DES) and how are they implemeted and used. While implementation process, I discovered that RC4 is way easier for understanding than DES cipher.