The usage and the impact of shift registers on the CFB mode of operation

HW1 - CNS Sapienza

Andrea Fioraldi 1692419

2018-10-12

1 Introduction

With the aim to provide confidentiality and authenticity of information, *Block Ciphers* are widely used in cryptography. Block Ciphers operates on fixed-length messages, called *blocks*, and *Modes of Operations* are the techniques used to apply block ciphers to messages longer than a block.

Cipher Feed Back (CFB) is a popular mode of operation. In the following sections, we will discuss the usage and the impact of CFB and, in particular, of its variant that uses shift registers.

2 CFB Overview

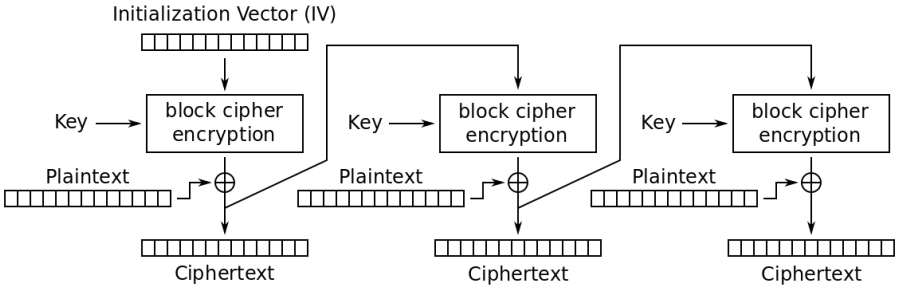

Figure 1: CFB encryption, from [1]

In this technique, regards the encryption, the produced ciphertext block is forwarded to the next encryption unit to produce a block of encrypted data that, xored[1] with the correspondent plaintext block, generates the next ciphertext block.

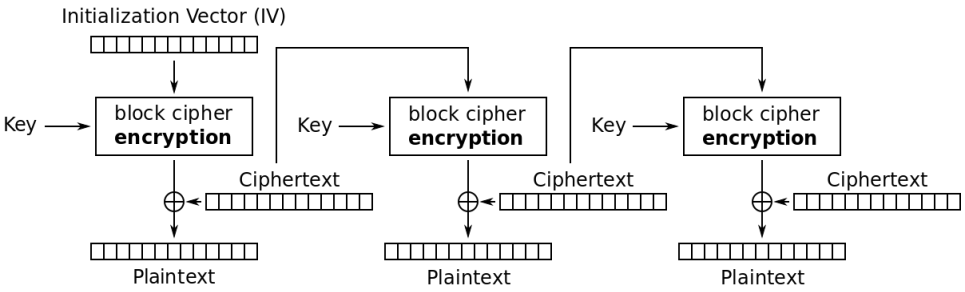

Figure 2: CFB decryption, from [1]

Viceversa, during the decryption, the output of the encryption unit using the previous ciphertext block is xored with the current ciphertext block to produce the corresponding plaintext block.

This mode of operation requires an initialization vector (IV) as the initial input block. As you can see in figure 2 the encryption unit is used also for decryption.

Accordingly to [1], CFB encryption and decryption can be expressed with the following formulas:

- Encryption: $C_0 = IV$, $C_i = CIPH_K(C_{i-1}) \oplus P_i \ \forall i = 1...n$;
- Dencryption: $P_0 = IV$, $P_i = CIPH_K(C_{i-1}) \oplus C_i \ \forall i = 1...n$;

The used symbols are defined in the following way:

- IV := the initialization vector;
- $CIPH_K$:= the encryption process using the key K;
- P_i := the i-th block of the plaintext;
- C_i := the i-th block of the ciphertext;
- n := number of blocks composing the plaintext/ciphertext;

[1] exclusive-ORed

An additional property is the possibility to use parallelism in the decryption process.

3 CFB with Shift Registers

One of the most used variants of CFB introduces shift registers as input for the encryption unit.

3.1 Usage

Before describing the usage of the shift registers in the CFB mode we must provide some additional definitions:

- $b :=$ the size of a block in bits;
- $s :=$ the size of a plaintext/ciphertext segment in bits ($1 \leq s \leq b$);
- $SR :=$ the content of the input shift register;
- $P_i :=$ the i-th segment of the plaintext;
- $C_i :=$ the i-th segment of the ciphertext;

We define a *segment* as a block of the plaintext/ciphertext on which CFB operates. It can be smaller than the type of blocks used by $CIPH_K$, so we use the term segment to distinguish between them.

In this variant of the technique, only the s most significant bits of the E_K output are xored with the plaintext segment to produce a ciphertext segment. Then, to form the next input block, the $b - s$ least significant bits of the previous input block are shifted to left of s positions and thus remaining space is filled with the s bits of the previous ciphertext segment. The IV is used as usual. The decryption works in a similar way, the input ciphertext segment is placed in the input block after shifting the $b - s$ least significant bits.

The mathematical formulas to CFB with shift registers are the following:

- Encryption: $C_i = extract(E_K(SR_{i-1}), s) \oplus P_i \ \forall i = 1...n$;
- Dencryption: $P_i = extract(E_K(SR_{i-1}), s) \oplus C_i \ \forall i = 1...n$;

Where $extract(x, y)$ are the most significant y bits of a message x. The evolution of the content of SR in both encryption and decryption is described by:

$S_0 = IV$, $S_i = ((S_{i-1} << s) + C_i) \ mod \ 2^n \ \forall i = 1...n$.

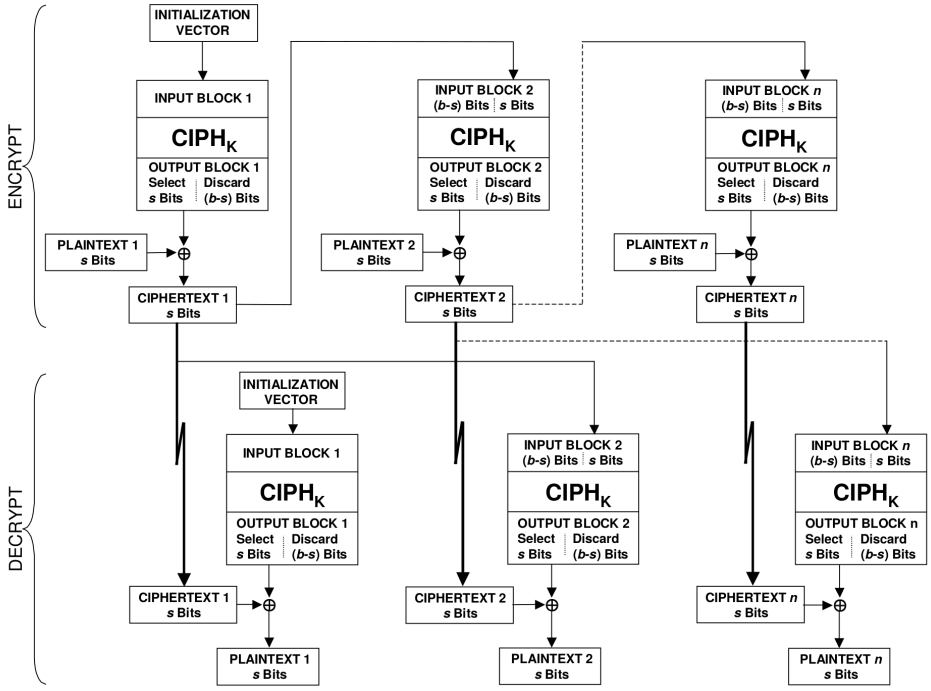

Figure 3: CFB with shift registers, from [2]

3.2 Impact

The usage of shift registers in CFB mode provides a big change over the traditional technique: it makes CFB self-synchronizing. A cipher is called self-synchronizing when, if a part of the ciphertext is lost, the decryption can continue correctly after a finite number of incorrect decryptions of blocks. In the standard CFB if a bit of the ciphertext is lost the technique is not able to synchronize. With the shift registers, instead, if we lost x bits of a ciphertext segment C_j, the decryption will produce wrong plaintext segments until that there are some bits of C_j in SR. The number of the corrupted plaintext segments is n/s.

4 Conslusion

Introducing shift registers in the CFB mode is a game-changing choice with a low impact on performance. In fact, the implementations in software and in hardware are not more complicated than the standard technique and with

a reasonable overhead of $b/s * n$ calls to E_K versus the n calls performed in the traditional CFB.

References

[1] *Block cipher mode of operation - Wikipedia.* `https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation` Accessed: 2018-10-12.

[2] Morris Dworkin. *Recommendation for Block Cipher Modes of Operation. Methods and Techniques.* Tech. rep. 800-38A. National Institute of Standards and Technology, 2001. `url: http://www.dtic.mil/docs/citations/ADA400014`