International Advanced Researches & Engineering Congress-2017
http://iarec.osmaniye.edu.tr/
Osmaniye/TURKEY
16-18 November 2017

Comparative Analysis of Block Cipher Modes of Operation

Diedon Bujari[1*] and Erke Aribas[1]

[1] ITU Faculty of Computer and Informatics Engineering, Istanbul/TURKEY

* Corresponding author. Tel.: +90 551 242 23 08, E-mail address: bujari@itu.edu.tr

Abstract

In this paper, block cipher modes of operation used in cryptography, including both deterministic and probabilistic ones, are investigated in detail. A block cipher mode of operation is a particular way to use a block cipher, such as DES or AES, by combining it with some simple operations and feedback mechanism. The modes considered here are the Electronic Code Book (ECB) mode, the Cipher Block Chaining (CBC) mode, the Output Feedback (OFB) mode, the Cipher Feedback (CFB) mode, and the Counter (CTR) mode. These operation modes are analyzed, and compared in terms of their security, efficiency, and performance when implemented in MATLAB.

Keywords: Block ciphers, CBC, CFB, Cryptography, CTR, ECB, Modes of operation, OFB

1. Introduction

Cryptography, or the science of encryption, is the heart of the communication network today. It is used as an instrument to maintain the security during the exchange of data, such as text, audio, image, etc., in the presence of unauthorized attackers. Throughout history, various approaches have been practiced, such as transposition and substitution. With the developments in the field, more secure and advanced algorithms were introduced, also called as block ciphers, such as the Data Encryption Standard (DES), Advanced Encryption Standard (AES), RSA, etc. These cryptographic algorithms compose about 90% of all encryption that happens in the real world: on the Internet, cell phones, smart cards, databases, etc. [1].

As shown in Fig. 1, the application scenario is very simplistic: a data block (e.g. 128 bits) is encrypted using a key, producing the ciphertext; multiple data blocks are encrypted one after another. However, in practice, this approach is not very useful since the length of the data being encrypted is very short. For example, 128-bit data corresponds to 16 characters. In order to encrypt larger amounts of data, a block cipher is combined with some simple operations and feedback mechanism.

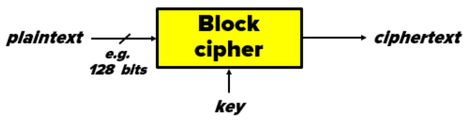

Figure 1. Block cipher scheme

In the first part of the paper, the most used and popular modes of operation are introduced. Afterwards, they are analyzed one-by-one by considering the problems they solve, their security and efficiency. By means of security, identical plaintext pattern problems, chaining reliance, and error propagation are evaluated. In addition to these, the possibility of parallelization of both encryption and decryption operations are discussed. Finally, the modes are compared in terms of performance when implemented in MATLAB. In this part, the Advanced Encryption Standard (AES) is used as the building block cipher, and the results are presented in cycles per byte (cpb).

2. Block Cipher Modes of Operation

Block ciphers, excluding as encryption algorithms, can be utilized for many other tasks in order to build different cryptographic mechanisms. Here is the list of some of its usages:

- Different encryption schemes,
- Stream ciphers,
- Pseudo-random number generator (PRNG),
- Hash functions,
- Message authentication codes (MACs), etc.

Different ways of using a block cipher for encryption, combining some simple operations, are called block cipher modes of operation. There are several modes of operation, as illustrated in Fig. 2, which are going to be discussed in the next sections. They are divided into two groups: the ones which result in deterministic encryption, and the ones in probabilistic encryption. In deterministic encryption schemes, if the key does not change, a particular plaintext is

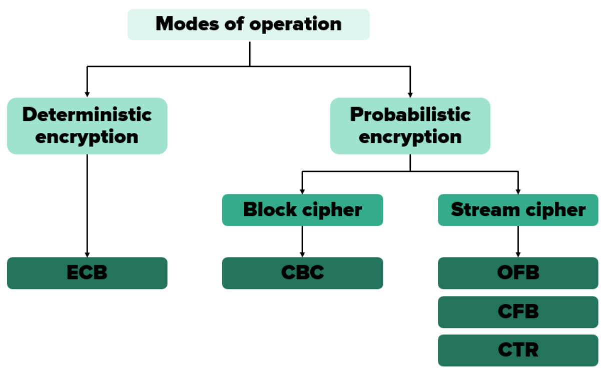

Figure 2. Modes of operation

mapped to a fixed ciphertext. On the other hand, probabilistic encryption schemes use randomness to achieve a nondeterministic generation of ciphertext. This group of schemes can function as both block ciphers and stream ciphers. All of these operation modes have one main goal: provide and maintain confidentiality and authenticity during communication. However, the security is provided by the cipher, not the mode itself [2].

2.1 Electronic Code Book (ECB) mode

The Electronic Code Book (ECB) mode is the most straightforward way of using a block cipher. The plaintext is split into n-bit blocks, and those blocks are encrypted independently using a block cipher, such as DES or AES. This means that there is no need for synchronization - blocks can be encrypted in any order, and then combined. Similarly, the decryption process is the inverse of this operation. Both encryption and decryption procedures are illustrated in Fig. 3.

Let $e(x_i)$ denote the encryption of the ith plaintext block, and $d(y_i)=e^{-1}(y_i)$ the decryption of the ith ciphertext block. Each of these blocks, as mentioned above, are of length n. We can define the encryption (1) and decryption (2) in the ECB mode as follows:

encryption: $y_i=e(x_i)$, such that $\lvert x_i \rvert=n$	(1)
decryption: $x_i=e^{-1}(y_i)=e^{-1}(e(x_i))$	(2)

As mentioned before, one of this mode's advantages is that the block sync is not necessary: the receiver can decrypt the received blocks without getting all of them. Also, bit errors related to some transmission problems have impact only on corresponding blocks. In addition, the ECB mode's implementation can be considered as fast enough, which comes from its ability of parallelization. In other words, different data blocks can be encrypted by different encryption units in parallel. Due to its speed and parallelization advantages, it has been used in database applications; addition or deletion of entries done independently of other records.

On the other hand, the ECB mode is not the best way of doing encryption. As long as the key used in encryption does not change, same plaintext blocks produce same ciphertext blocks, which makes it highly deterministic.

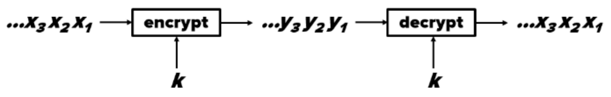

Figure 3. Electronic Code Book (ECB) mode

Therefore, it is possible to generate a code book (as the mode's name comes from), which maps plaintexts to corresponding ciphertexts. Also, if ciphertexts have fragments that repeat in the same places, such as the header and footer, the attacker uses these information in order to reach the plaintexts. This way of ciphertext-only attack is known as traffic analysis [3]. Moreover, the ECB mode is vulnerable to substitution attacks, i.e., manipulations in the ciphertext level in order to deceive the receiver.

2.2 Cipher Block Chaining (CBC) mode

As mentioned above, determinism makes encryption vulnerable to attacks; thus, it is essential to make it probabilistic. In other words, same plaintext should produce different ciphertexts every time it is encrypted. This feature is achieved using the Cipher Block Chaining (CBC) mode of operation, illustrated in Fig. 4. In this mode, blocks are considered as a whole message - blocks are "chained together" - such that the influence of each plaintext block is spread over many ciphertext blocks.

The CBC mode uses some kind of randomness, which is the initialization vector (IV) in this case, in order to make the encryption probabilistic. IV does not have to be secret, but it should be nonce - number used only once. It can be generated in different ways; for instance, using a true random number generator, assigning it a counter value, $ID_A \parallel ID_B \parallel$ TIME, etc. The first plaintext block is XORed with IV, and then encrypted using a block cipher (3). For the succeeding blocks, there is a feedback mechanism to the block cipher, as seen in Fig. 4. The previously produced ciphertext is fed back, and XORed with the plaintext block, producing the input to the block cipher. The decryption process (4) is the reverse of these operations.

encryption: $y_1=e(x_1 \text{ XOR } IV)$ $y_i=e(x_i \text{ XOR } y_{i-1})$, for $i\geq2$	(3)
decryption: $x_1=(e^{-1}(y_1) \text{ XOR } IV)$ $x_i=(e^{-1}(y_i) \text{ XOR } y_{i-1})$, for $i\geq2$	(4)

The Cipher Block Chaining (CBC) mode has been the most commonly used mode, although its encryption operation cannot work in parallel. This is because each plaintext block affects the encryption of the next blocks. One may think that bit errors will have a huge effect on all subsequent ciphertext blocks. However, these errors are recovered in the decryption process, and produce the

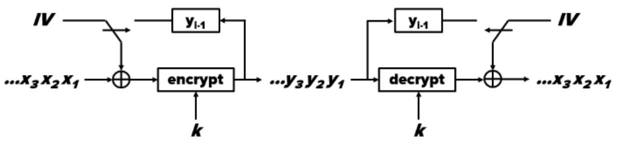

Figure 4. Cipher Block Chaining (CBC) mode

plaintext with the same bit errors. This is called as the self-recovering feature of the CBC mode, and makes the parallelization of decryption possible [4]. In contrary to the ECB mode, substitution attacks do not apply if the IV is properly chosen for every transfer. But, any alteration in the ciphertext level produces some random changes in the plaintext, which is an undesirable fact and may have negative effects.

2.3 Output Feedback (OFB) mode

As mentioned above, block ciphers can be used as stream ciphers. These encryption schemes use the block cipher as a keystream generator, as illustrated in Fig. 5. The first input to the block cipher is the initialization vector (IV). The n-bit plaintext is XORed with the n-bit key stream generated in the encryption operation, yielding an n-bit ciphertext. Forthcoming key streams are generated by feeding the previously generated ones to the block cipher used. As it can be seen from its construction, the scheme produces streams blockwisely, rather than bitwisely.

$$\textbf{encryption:}\ s_1=e(IV);\ y_1=(s_1 \text{ XOR } x_1)\quad s_i=e(s_{i-1});\ y_i=(s_i \text{ XOR } x_i),\ i\geq 2 \tag{5}$$
$$\textbf{decryption:}\ s_1=e(IV);\ x_1=(s_1 \text{ XOR } y_1)\quad s_i=e(s_{i-1});\ x_i=(s_i \text{ XOR } y_i),\ i\geq 2 \tag{6}$$

The Output Feedback (OFB) mode runs the block cipher as a synchronous stream cipher, which makes it very similar to standard stream ciphers. Neither the plaintext nor the ciphertext affect the key stream generation. Encryption (5) and decryption (6) operations are exactly the same: the XOR function during encryption is reversed by another XOR function during decryption. One of the advantages of this mode is that the feedback mechanism can work offline before the arrival of the data. On the other side, encryption and decryption cannot be parallelized since each key stream depends on all previous key streams.

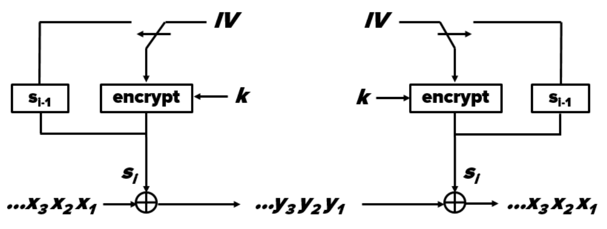

Figure 5. Output Feedback (OFB) mode

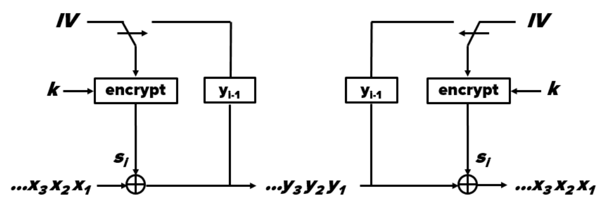

Figure 6. Cipher Feedback (CFB) mode

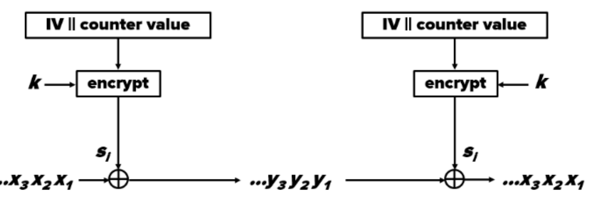

Figure 7. Counter (CTR) mode

2.4 Cipher Feedback (CFB) mode

The Cipher Feedback (CFB) mode is quite similar to the Output Feedback (OFB) mode: it runs a block cipher as a stream cipher generator; but instead of the previous key stream, the ciphertext is fed back to the block cipher in order to produce the next stream, as in Fig. 6. The first n-bit key stream is generated by encrypting the initialization vector (IV), which is then XORed with the n-bit data to yield the n-bit ciphertext. Subsequent streams are generated by feeding the ciphertext back to the block cipher. Like in the OFB operation mode, encryption (7) and decryption (8) processes are exactly the same process. Also, the encryption cannot be parallelized. In contrary to OFB, the CFB mode is an asynchronous stream cipher generator since the key stream generation is a function of the ciphertext, and the parallelization of decryption is possible.

$$\textbf{encryption:}\ y_1=(e(IV) \text{ XOR } x_1)\quad y_i=(e(y_{i-1}) \text{ XOR } x_i),\ \text{for } i\geq 2 \tag{7}$$
$$\textbf{decryption:}\ x_1=(e(IV) \text{ XOR } y_1)\quad x_i=(e(y_{i-1}) \text{ XOR } y_i),\ \text{for } i\geq 2 \tag{8}$$

2.5 Counter (CTR) mode

The Counter (CTR) mode, introduced by Diffie and Hellman in 1979 [5], is very similar to the Output Feedback (OFB) mode and the Cipher Feedback (CFB) mode. It uses a block cipher as its stream generator, whose input is a counter value, as illustrated in Fig. 7. The value of the counter should change every time a new key stream is generated. In order to produce such a counter, the following approach is often practiced: a nonce initial vector smaller than the block length, followed by the counter (CTR) initialized to zero. Although some argue that this systematic approach can risk its security, the CTR mode is widely used and recommended nowadays. In addition, parallelization of its encryption (9) and decryption (10) operations is possible.

$$\textbf{encryption:}\ y_i=(e(IV\ ||\ CTR_i) \text{ XOR } x_i) \tag{9}$$
$$\textbf{decryption:}\ x_i=(e(IV\ ||\ CTR_i) \text{ XOR } y_i) \tag{10}$$

3. Comparison Between Block Cipher Modes of Operation

The Electronic Code Book (ECB) mode is the most straightforward way of using a block cipher, but not the best way of encryption. It should not be used while encrypting multiple data blocks with the same key since same plaintext blocks produce same ciphertext blocks, making it highly deterministic. Also, the mode is vulnerable to traffic analysis or ciphertext-only and substitution attacks, as discussed above. On the other hand, the ECB mode has an advantage over other modes of operation, speed, which is made possible due to it parallelization ability. This paper does not consider the running time of this mode since it does not achieve the security goals desired from a block cipher mode of operation.

The Cipher Block Chaining (CBC) mode solves the determinism problem of the ECB mode: using the same key, same plaintext blocks produce different ciphertext blocks. Compared to the ECB mode, substitution attacks do not apply to the CBC mode if the initialization vector (IV) is properly chosen. On the other hand, even though it has been the most commonly used one, its encryption process lacks the parallelization feature - in MATLAB, it takes 4.07 cpb for 1KB of random data. The decryption runs at 1.29 cpb since it can operate in parallel [6].

The Output Feedback (OFB) and the Cipher Feedback (CFB) modes are very similar to each other: they both run a block cipher as a synchronous stream cipher generator. Since the encryption and decryption processes are exactly the same, it saves code space. However, as in the CBC mode, parallelization of the encryption process is not possible. In MATLAB, 1KB data is encrypted at 4.39 cpb using the OFB mode, and 5.47 cpb using the CFB mode [6].

When speed is essential, as it is in this case, the Counter (CTR) mode gives the best results. Its parallelization ability makes it fast enough, and therefore, widely used and recommended nowadays. It takes 1.28 cpb to encrypt 1KB of random data in MATLAB [6].

Table 1. Comparison of modes' run time in MATLAB (1KB of data)

Mode of operation	**Running time in MATLAB**	
	Encryption	***Decryption***
CBC	*4.07 cpb*	*1.29 cpb*
OFB	*4.39 cpb*	*4.39 cpb*
CFB	*5.47 cpb*	*5.55 cpb*
CTR	*1.28 cpb*	*1.28 cpb*

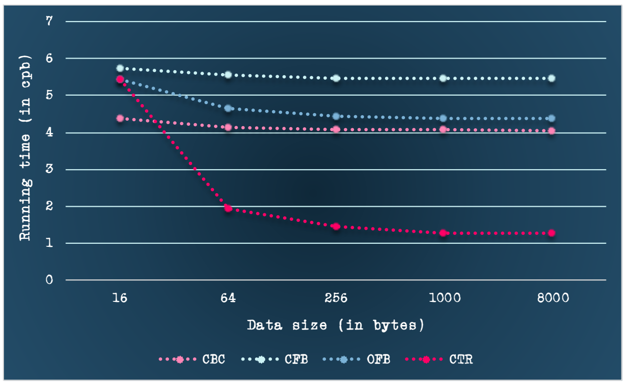

Figure 8. Graphical representation of the trendlines for encryption operations as the size of data increases

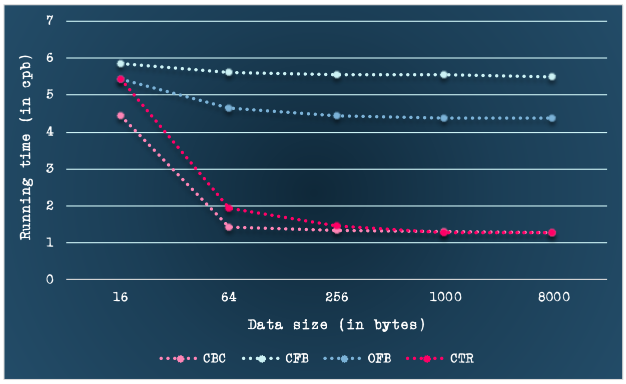

Figure 9. Graphical representation of the trendlines for decryption operations as the size of data increases

4. Conclusion

As discussed above, the Electronic Code Book (ECB) mode should not be practiced for general purposes since it lacks essential security requirements. If the initialization vector is a nonce, the Cipher Block Chaining (CBC) mode can be considered as a secure encryption scheme; however, it does not outperform the Counter (CTR) mode - it is about 3 times slower. On the other hand, the CBC decryption performs well in larger amounts of data because of its parallelization feature. The Output Feedback (OFB) and the Cipher Feedback (CFB) modes are the worst considering their software performce.

All in all, the Counter (CTR) mode is the most secure, efficient, and fastest way of doing encryption. The performance advantages of the CTR mode can be seen from the graphical representations of the trendlines for both encryption and decryption operations, particularly in Fig. 8 and Fig. 9.

References

1. Paar, Ch. & Pelzl, J. (2010). *Understanding Cryptography*.
2. National Institute of Standards and Technology. (1998). *DES Modes of Operation*. FIPS PUB 81. Retrieved from http://csrc.nist.gov/publications/fips/fips81/fips81.htm
3. Schneier, B. (1996). *Applied Cryptography*.
4. National Institute of Standards and Technology. (2001). *Recommendation for Block Cipher Modes of Operation*. NIST Special Publication 800-38A. Retrieved from http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf
5. Lipmaa, H., Rogaway, Ph. & Wagner, D. (n.d.). *CTR-Mode Encryption*. Retrieved from http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/ctr/ctr-spec.pdf
6. Rogaway, Ph. (2011). *Evaluation of Some Blockcipher Modes of Operation*. Retrieved from http://web.cs.ucdavis.edu/~rogaway/papers/modes.pdf