Comparative Analysis of Block Cipher Modes of Operation for Better Utilization

Sultan Almuhammadi, Ibraheem Al-Hejri
College of Computer Sciences and Engineering
King Fahd University of Petroleum and Minerals
Dhahran, Saudi Arabia
Emails: muhamadi@kfupm.edu.sa, alhejri87@gmail.com

Abstract

Cryptography plays a major role in information security. However, cryptographic algorithms consume considerable amount of resources, like memory, CPU time, encryption and decryption time. In this paper, we compare the most common block cipher modes of operation based on the recommendation of the National Institute of Standards and Technology (NIST) in terms of encryption time, decryption time, and throughput with variable data packet sizes. The results of these comparisons are summarized and our observations are highlighted to help making informative decision when choosing the mode of operations for different applications with symmetric-key ciphers.

Keywords

Mode of operation, encryption, decryption, throughput.

1. Introduction

Security is essential to transmit private data over the network. Cryptographic algorithms are an important part of information security. In symmetric-key cryptography, encryption and decryption use the same key. There are two types of cipher algorithms that can be used: stream cipher or block cipher. In a stream cipher, the ciphertext is generated by encrypting each plaintext digit one at a time with the corresponding digit of the keystream. On the other hand, a block cipher processes plaintext of fixed length, knows as the *block size*. If the length of the plaintext is larger than the block size, then the data must be divided into several blocks. Using a padding technique, typically, the last block of the plaintext must be padded to match the block size [1]. Modes of operation are formal descriptions of the way that the block encryption on a message with size larger than a block size is handled. They feature symmetric-key cryptosystems that provide desired services such as authenticity, integrity, and confidentiality. In this paper, we compare different modes of operation for best resource utilization. The modes of operations compared in this paper are: Electronic Code Book (ECB), Cipher Block Chaining (CBC), Cipher FeedBack (CFB), the Output FeedBack mode (OFB), and the Counter (CTR) modes. The performance evaluation of these modes is base on the following metrics: encryption time, decryption time, and throughput with variable data packet size.

This paper is organized as follows: Section 2 reviews the related work. Existing methodologies are presented in Section 3. In Section 4, experimental design is performed. In Section 5, we present experimental results. Finally, Section 6 provides briefly the conclusion and the recommendations for future work.

2. Background and Related Work

Based on the recommendation of the National Institute of Standards and Technology (NIST), there are five main block cipher modes of operation: the Electronic Code Book (ECB), Cipher Block Chaining (CBC), Cipher FeedBack (CFB), the Output FeedBack (OFB), and the Counter (CTR) modes [2]. In this section we will explain these common block cipher modes of operation and discuss the results obtained from other sources that are related to our study.

2.1. Electronic Code Book (ECB)

ECB describes the use of a symmetric cipher in its raw form [3]. In this mode each block is encrypted independently [4]. The main drawback of ECB mode,is that, if there are identical blocks in the plaintext, they will be encrypted into the same ciphertext blocks [5]. Thus, data patterns in ECB mode are not hiden enough. Due to this disadvantage, a lot of vulnerabilities are generated like changing of ciphered messages [6]. To compensate for this, each block of the encrypted information should be encrypted dependently. More complex modes make each ciphered message unique by combining the previous ciphered blocks as well as using Initialization Vectors (IV) [6]. We will discuss this in the next sections.

For cryptographic protocols, it is not recommended to use ECB mode because it doesn't supply data confidentiality. Also ECB makes protocols without integrity protection, especially with replay attacks [5]. ECB is the easiest and fastest mode to implement, it is the most common mode of DES algorithm used in commercial applications [7]. With this mode, Blowfish algorithm produces more throughput and consumes less memory usage, execution time based on input size of text files [8]. Figure 1 illustrates ECB mode.

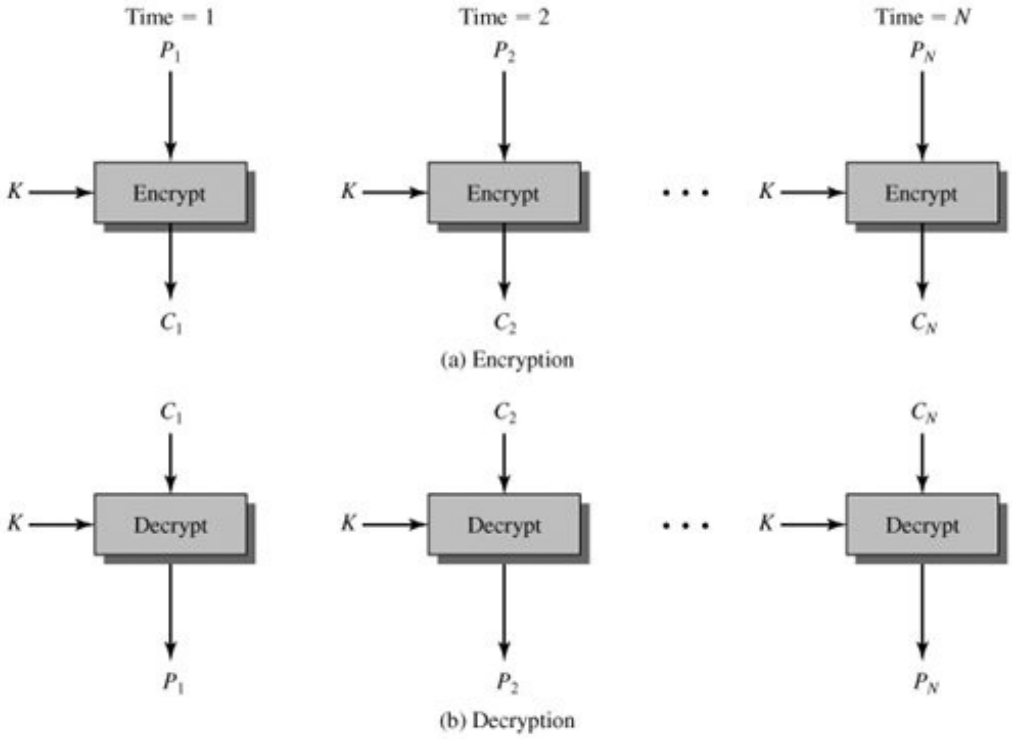

Figure 1: Electronic Code Book (ECB) [9]

ECB uses two formulae: one for encryption and the other for decryption, as follows:

$$C_i = E_k(P_i) \quad [Encryption]$$
$$P_i = D_k(C_i) \quad [Decryption]$$

2.2. Cipher Block Chaining (CBC)

The appearance of the CBC mode solves the problem in the ECB mode. It reduces the likelihood of appearing repeated patterns in the ciphertext [3]. In CBC mode, before encrypting a block, the block of plaintext is XORed with the previous ciphertext block. For the first block. It is XORed with an initialization vector(IV) [6]. In this mode, larger message size can be handled easily [10]. Thus, if the same plaintext is encrypted multiple times, the resulting ciphertexts are distinct. CBC mode requires more processing time than ECB mode due to its chaining mechanism [11]. CBC can be synchronized to avoid channel noise error propagation [4].

Moreover, Alabaichi et al. [12] reported that CBC with blowfish algorithm can be used to encrypt any type of file without restrictions on the file contents. El-Semary et al. [2] reported that although this mode provides high data confidentiality and authentication,it does not have parallelizable architecture. A secret key in CBC mode must be changed at least every 2^{10} encryptions in order to resist power analysis attacks [13]. Desai et al. [14] presented that the improved version of CBC called Interleaved Cipher Block Chaining (ICBC) mode (The Cryptographic Community proposed this mode that allows parallel implementation) shows better performance than CBC when they have been implemented by the AES algorithm.Since CBC mode does not support data level parallelism, it can not use in Disk encryption. The comparison study of Saraf et al. [15] showed that AES with CBC is the fastest algorithm among all the other algorithms (MIE, VC and N/KC) have been used for images encryption. Figure 2 shows CBC mode.CBC uses two formulae: one for encryption and the other for decryption, as follows:

$$C_i = E_k(P_i \oplus C_{i-1}), \text{ with } C_0 = IV \quad [Encryption]$$
$$P_i = D_k(C_i) \oplus C_{i-1}, \text{ with } C_0 = IV \quad [Decryption]$$

2.3. Cipher Feedback(CFB)

In this mode, to produce the ciphertext,a plaintext block is XORed with the encryption module of the previous block ciphertext. The process is repeated with the following input blocks until the ciphertext is created.Generally, each consecutive input block is encrypted to generate an output block.Like CBC, in CFB mode, the block of plaintext is based on the result of the previous ciphertext block; so, multiple cipher operations cannot be carried out in parallel. CFB does not require any special measures to handle the plaintext that has a variable-length (not multiple of the block size) [5]. So, CFB is faster than CBC.Like CBC, CFB can be synchronized to avoid channel noise error propagation [4]. A secret key in CFB mode must be changed at least every 2^{12} encryptions in order to resist power analysis attacks [13]. With CFB we can get more security services such as: data integrity, authentication, confidentiality and freshness of the message at sensor node [16]. According to Thakur et al. [11] CFB mode requires less processing time than CBC. Figure 3 shows CFB mode.CFB uses two formulae: one for encryption and the other for decryption, as follows:

$$C_i = P_i \oplus E_k(C_{i-1}), \text{ with } C_0 = IV \quad [Encryption]$$
$$P_i = C_i \oplus E_k(C_{i-1}), \text{ with } C_0 = IV \quad [Decryption]$$

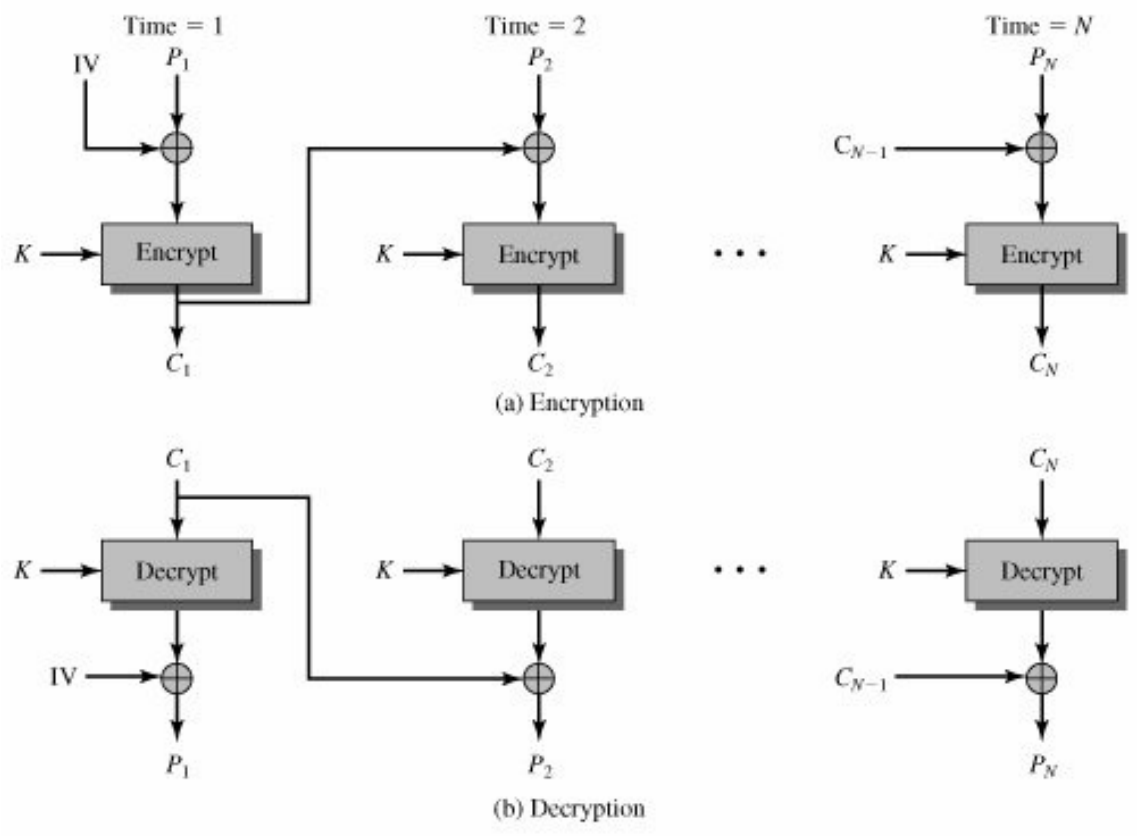

Figure 2: Cipher Block Chaining (CBC) [9]

2.4. Output Feedback(OFB)

In this mode"the Initialization Vectors (IV) is changed by the forward cipher function to produce the first output block. The following output blocks are created by applying the forward cipher function to the previous output blocks, and the output blocks (Xi) are XORed with the corresponding plaintext blocks to generate the ciphertext blocks". Like CFB,OFB mode does not require any special measures to handle the plaintext that has a variable-length(not multiple of the block size) [5].Thus, OFB is faster than CBC. To reach both of CFB and OFB's usages only an encryption module is needed for their cryptography applications [4]. Unlike CFB where the ciphertext is the feedback, the feedback in OFB is the output of the encryption block [2]. That means,in OFB the exclusive-OR value for each plaintext block is performed independently of both the ciphertext and plaintext. On the contrary of CBC and CFB,Ahmad et al. [16] reported that with OFB mode,bit errors during transmission do not propagate. A secret key in OFB mode must be changed at least every 2^{12} encryptions in order to resist power analysis attacks [13]. According to Thakur et al. [11] OFB shows better performance than ECB and CBC modes, but not as fast as CFB.For applications which require output feedback, OFB mode is better than the others. Figure 4 shows OFB mode.OFB uses two formulae: one for encryption and the other for decryption, as follows:

$$C_i = P_i \oplus E_k(X_{i-1}), \text{with} X_0 = IV \quad [Encryption]$$

$$P_i = C_i \oplus E_k(X_{i-1}), \text{with} X_0 = IV \quad [Decryption]$$

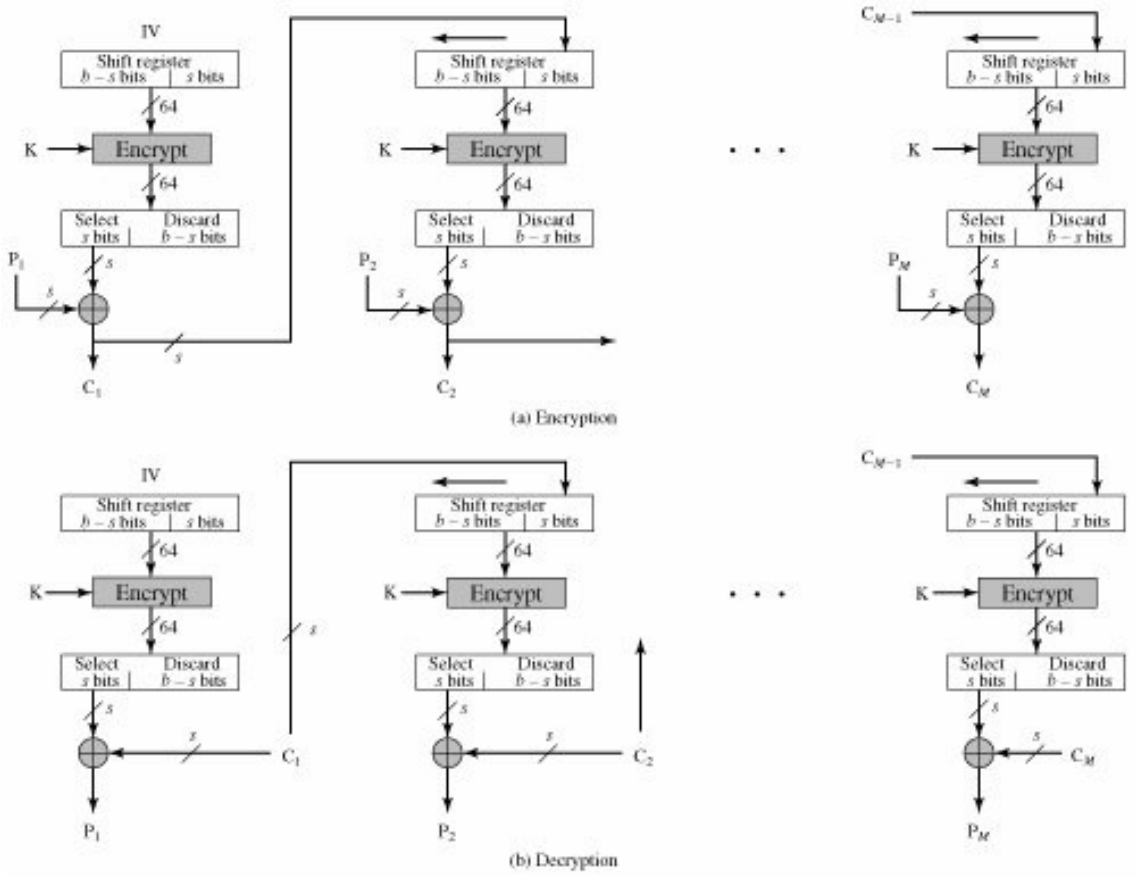

Figure 3: Cipher Feedback(CFB)) [9]

2.5. Counter Mode (CTR)

CTR, also known as Segmented Integer Counter mode (SIC) [3]. In the CTR mode, similarly to the Initial Vector, a counter that starts from the preconfigured initial value together with a nonce is used to generate a key stream. Then, the key stream is XORed into the plaintext to create the ciphertext [2]. Like OFB and CFB modes, the CTR mode does not require the plaintext to be padded to the block size of the cipher. Moreover, the encryption of a plaintext block in the CTR does not depend on the result from previous blocks. Thus, CTR has become the mode of choice for high-speed applications due to its highly parallelizable architecture [2]. However, the CTR mode does not provide data integrity.Jayasinghe et al. [13] reported that comparing with other modes, CTR mode provides a balance between power and area while maintaining sufficient resistance for power analysis attacks. Li et al. [7] shows that applying an AES algorithm with CTR on an NVIDIA G80 GPU using shared memory (one of the types of on-chip memory) to store lookup tables lead to the best performance. Figure 5 shows CTR mode. CTR uses two formulae: one for encryption and the other for decryption, as follows:

$$C_i = P_i \oplus E_k(R_{i-1}), \text{ with } R_0 = IV \qquad [Encryption]$$

$$P_i = C_i \oplus E_k(R_{i-1}), \text{ with } R_0 = IV \qquad [Decryption]$$

Table 1 summarizes the encryption algorithms, the modes of operation and the performance metrics of all previous studies, and compares them to our work.

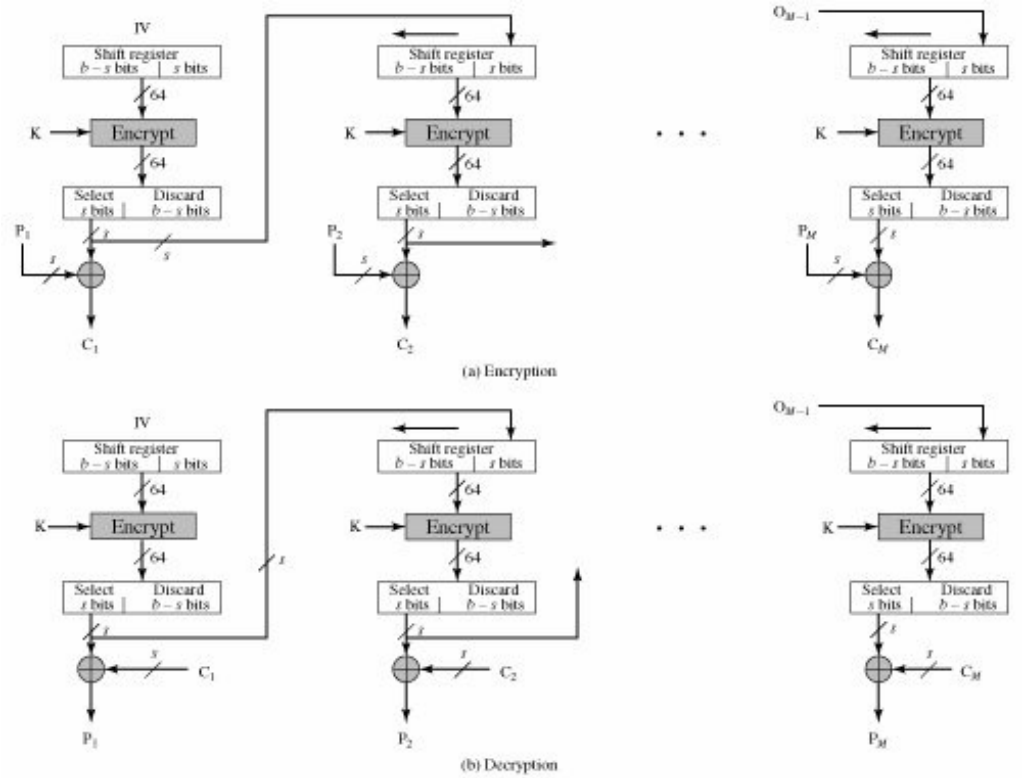

Figure 4: Output Feedback(OFB) [9]

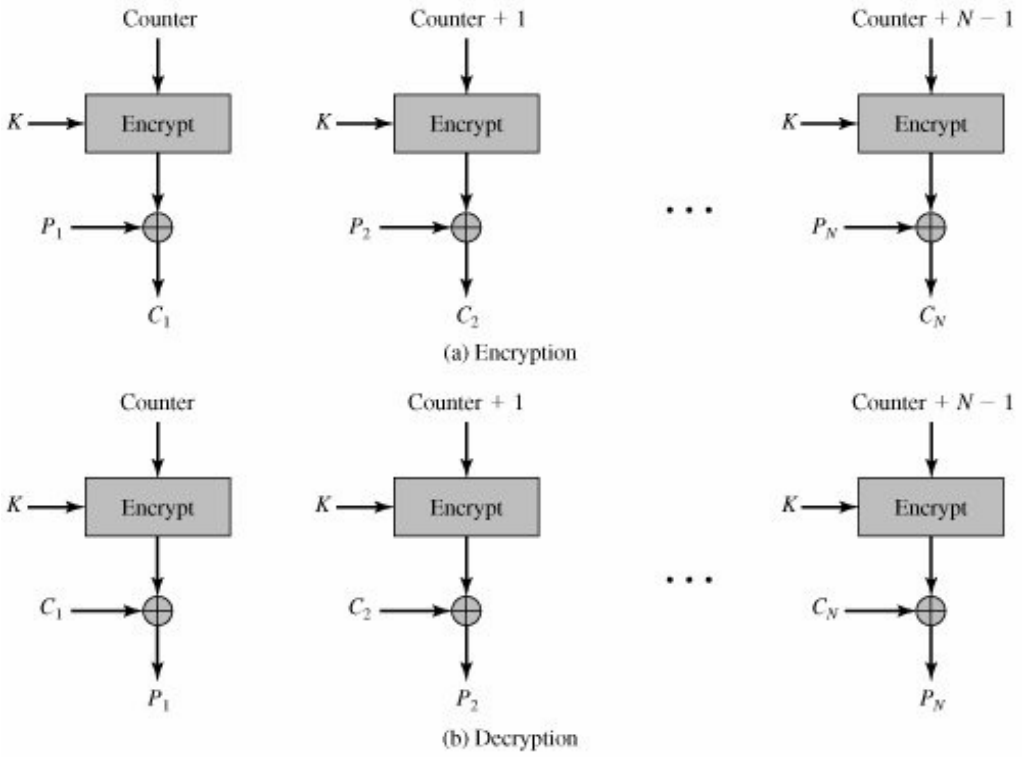

Figure 5: Counter Mode (CTR) [9]

Table 1: Summary of existing studies on modes of operation

Reference	Algorithm	Mode	Performance Metric
[7]	AES	ECB, CBC	Throughput, Data transfer cost
[8]	DES, AES Blowfish	ECB	Execution Time, Throughput Memory Usage
[11]	DES, AES Blowfish	ECB, CBC CFB, OFB	Execution Time
[14]	AES	CBC, ICBC	Encryption Time Decryption Time
[15]	AES	CBC	Encryption Time
[17]	AES, RC4	ECB, CBC CFB	Throughput, Encryption Time Decryption Tim, CPU Time Memory Utilization
[18]	DES, 3DES AES, RC2 RC6, Blowfish	ECB	CPU Clock Cycle , CPU Time Encryption Time, Battery Power
This paper	AES	ECB, CBC CFB , OFB CTR	Encryption Time, Decryption Time, Throughput

3. Research Methodology

We adopted the research methodologies used in performance evaluation of AES & RC4 [17] and performance analysis of encryption algorithms [8].

3.6. Performance Evaluation of AES & RC4 Algorithms :

In [17], the authors compare AES and RC4 algorithms. A laptop with 2.99 GHz CPU and 2 GB RAM is used to conduct the experiments.The laptop encrypts files with different sizes in the ranges 100KB to 50MB. The performance of the two algorithms is evaluated based on the following metrics:

1. CPU process time: time dedicated to a specific process by CPU. It reflects the load of the CPU. When CPU time is increased, the CPU loading is also increased.
2. Encryption time: It is considered as the time consuming to generate a cipher text from a plaintext by an encryption algorithm. The encryption time is varied proportionally according to the size of data [19].
3. Decryption time: It is considered as the time consuming to reproduce a plaintext from a cipher text by a decryption algorithm. The decryption time is varied proportionally according to the size of data [19].
4. Throughput: It is calculated as the whole encrypted plaintext in Kilobytes divided by encryption time (KB/sec).For encryption scheme, the throughput indicates the speed of encryption. When the throughput is increased, the power consumption is decreased [8].
5. Memory Utilization: It indicates how much of memory is needed during the encryption process or the decryption process.

The tasks that are performed as follows:

(i) For each algorithm, the encryption and decryption time are calculated using input files with different sizes.
(ii) For each algorithm, the throughput is computed in KB/Sec.

(iii) For each algorithm, the CPU time is calculated with different file sizes.
(iv) The effect of changing file size on memory consumption is considered.
(v) The effect of changing the key size on execution time is considered .

3.7. Performance Analysis of Encryption Algorithms:

In [8],the authors compare AES, DES and Blowfish algorithms.This experiment is conducted using nine files with different sizes. A laptop with 1.8 GHz Dual processor is used to conduct the experiments. Performance evaluation task is performed based on execution time, memory usage and Throughput..The algorithms' settings are given in Table II.

Table 2: Blowfish, AES and DES Algorithms' Settings

Algorithm	Key Size (in bits)	No of rounds	Block Size (in bits)
BF 128	128	16	64
BF 192	192	16	64
BF 256	256	16	64
AES 128	128	10	128
AES 192	192	12	128
AES 256	256	14	128
DES	64	16	64

4. EXPERIMENTAL DESIGN

Performance evaluation of block cipher modes is based on the existing classes in java environment. The implementation uses managed wrappers for AES-128 available in java.cypto and java.security[CryptoSpec] that wraps unmanaged implementations available in JCE (Java Cryptography Extension) JCA (Java Cryptography Architecture). The functionality of a cryptographic cipher which is used for encryption and decryption is provided by the Cipher class. It is considered the core of the JCE framework.

A laptop 2.30 GHz CPU and 4 GB RAM is used to conduct the experiment.The laptop encrypts files with different sizes in the ranges 500KB to 200MB.

Performance evaluation task is performed based on encryption time, decryption time and Throughput. The encryption time is considered as the time consuming to generate a cipher text from a plaintext by an encryption algorithm. The decryption time is considered as the time consuming to reproduce a plaintext from a cipher text by a decryption algorithm. Regarding the throughput, it is calculated as the whole encrypted plaintext in Kilobytes divided by the total encryption time in second (KB/sec). For encryption scheme, throughput indicates the speed of encryption.When the throughput is increased, the power consumption is decreased.

5. EXPERIMENTAL RESULTS AND ANALYSIS

In this section we show the results of our experiments. The results show the effect of changing file size on each mode.

5.8. Performance Evaluation Based on Encryption Time:

In Fig 6, the performance of block cipher modes is shown in terms of encryption time. We compare the encryption time with different file sizes.We note that ECB takes less time

than other modes. With file more than 100MB, CTR takes slightly more time than CFB, OFB and CBC.Generally, the differences between the modes are negligible especially with file less than 50MB.Table III shows the comparative encryption time among block cipher modes.

5.9. Performance Evaluation Based on Decryption Time:

In Fig 7, the performance of block cipher modes is shown in terms of decryption time. We compare the decryption time with different file sizes. We note that ECB takes less time than other modes. OFB takes less time than CBC and CFB .Generally, the differences between the modes are negligible especially with files less than 50MB. The comparative decryption time among block cipher modes is shown in Table IV.

Table 3: Comparative Encryption Time Among Block Cipher Modes

File Size (kbytes)	ECB(ms)	CBC(ms)	CFB(ms)	OFB(ms)	CTR(ms)
530	94.3	97.7	109.3	107.6	95.3
2,115	99.9	105	115.8	115.7	104.9
10,479	184.5	210.8	217.1	217.1	215.7
52,382	613.8	713.5	732.2	735.8	735.1
108,095	1204.5	1401.2	1419.9	1413	1422.9
216,189	2348.5	2744.6	2874.3	2788	2794.9

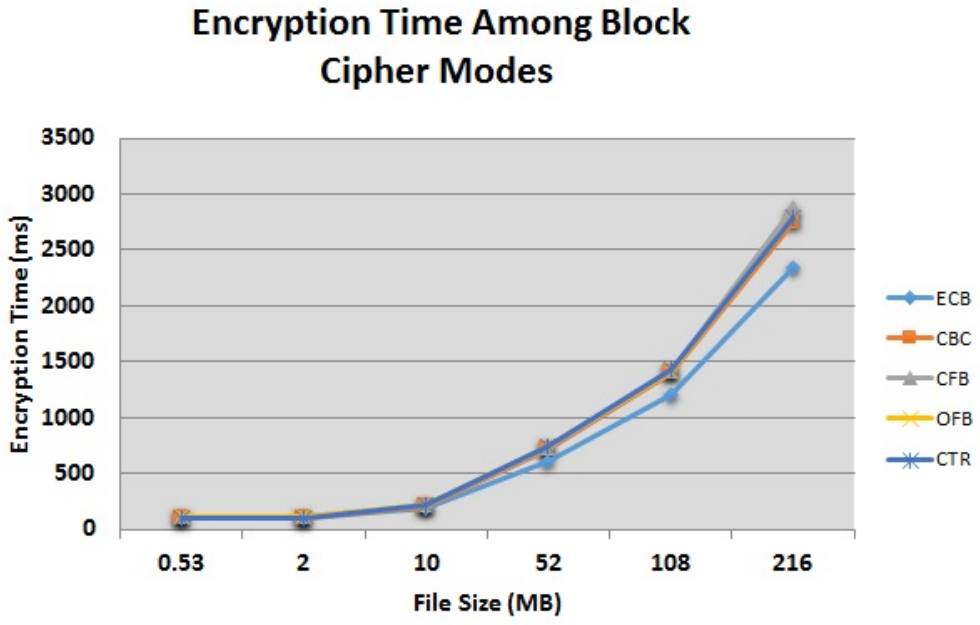

Figure 6: Encryption Time of Block Cipher Modes

Table 4: Comparative Decryption Time Among Block Cipher Modes

File Size (kbytes)	ECB(ms)	CBC(ms)	CFB(ms)	OFB(ms)	CTR(ms)
530	92.2	92. 3	37.7	16.8	18.4
2,115	112.6	108	53.1	49.9	37. 3
10,479	193.9	220. 3	165.7	153.3	146.8
52,382	656.4	764.3	731.1	704.7	703.2
108,095	1415.8	1617.4	1641.7	1582.7	1435. 9
216,189	2480.9	2894.1	2970.3	2897.2	3006.7

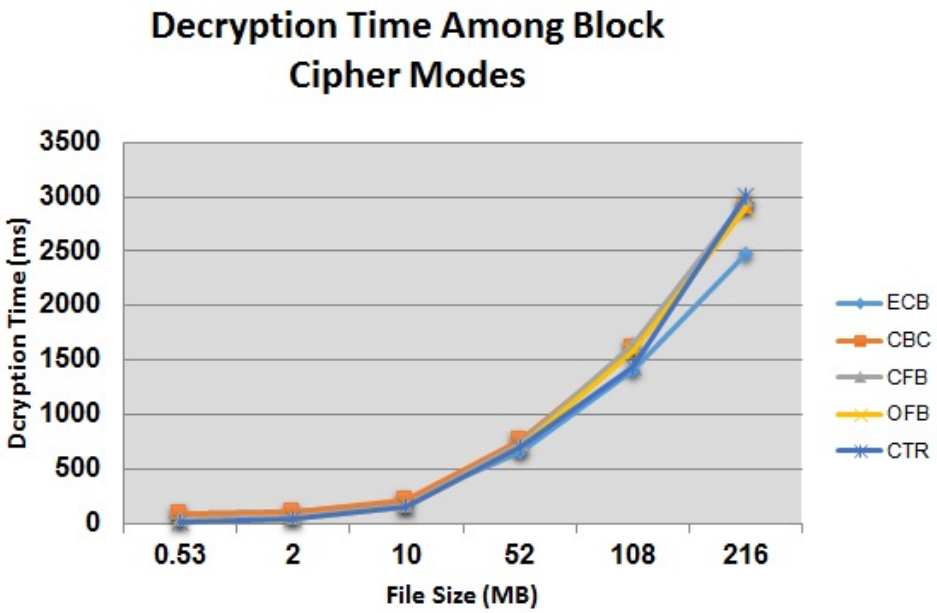

Figure 7: Decryption Time of Block Cipher Modes

5.10. Performance Evaluation Based on Throughput :

In our experiment, we use AES with the same key size(128 bit). So, the difference between the modes is relatively small and negligible.However, ECB mode is the fastest, and consumes less power than others. CTR is slightly faster in decryption than encryption.Another point can be noticed here is that slightly the results are better in encryption w.r.t. decryption.Comparative encryption throughput among block cipher modes is shown in Table V and Fig 8. and comparative decryption throughput is shown in Table VI and Fig 9.

Table 5: Comparative Throughput (Encryption)

Mode	Throughput (kbytes/ms)
ECB	85.75294
CBC	73.92467
CFB	71.27784
OFB	72.4894
CTR	72.60282

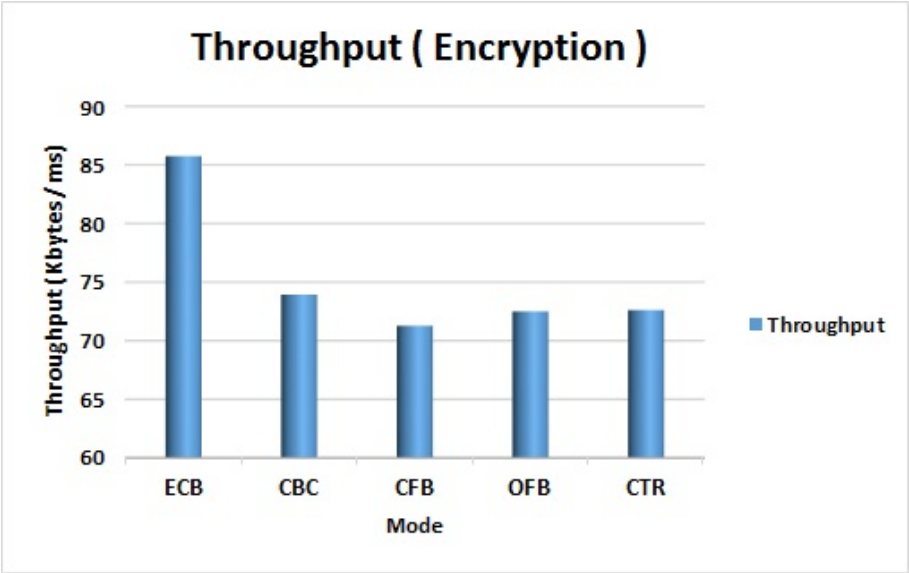

Figure 8: Encryption Throughput of Block Cipher Modes

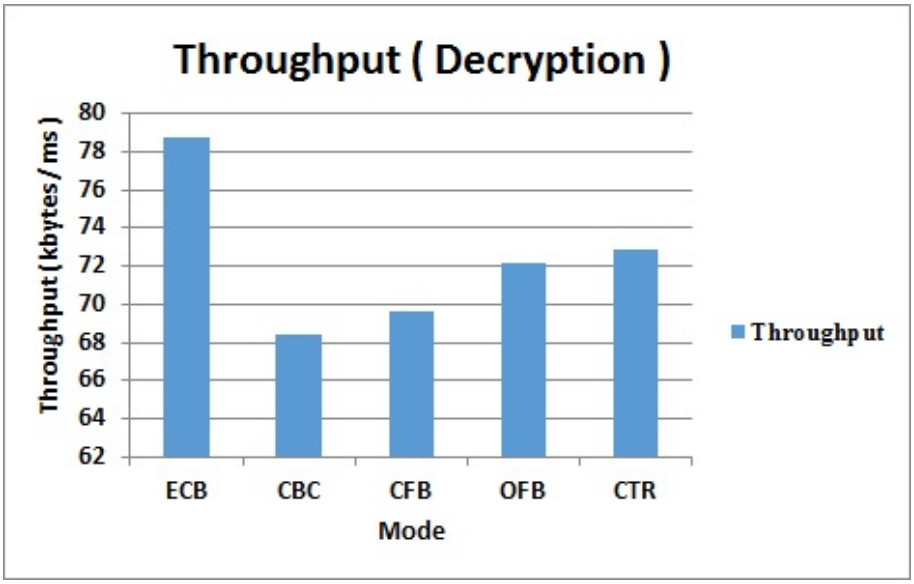

Figure 9: Decryption Throughput of Block Cipher Modes

Table 6: Comparative Throughput (Decryption)

Mode	Throughput (kbytes/ms)
ECB	78.71683
CBC	68.42743
CFB	69.61033
OFB	72.1219
CTR	72.8811

6. Conclusion

In this paper, a brief background on the most common block cipher modes of operation is given, with a comparison between them in terms of encryption time, decryption time and throughput. We presented two comparative studies between popular symmetric-key algorithms. We showed that ECB takes less time to encrypt and decrypt than other modes. The difference between the modes is relatively small, especially in small size files. However, with big size files ($\geq$ 200MB), there is a noticeable difference between the modes.

As a future work, we suggest to perform this analysis using other algorithms, and add other performance metrics such as memory utilization, CPU clock cycle and power consumption. To improve the results, the analysis can be done with simulation software.

7. References

[1] "Symmetric-key cryptography algorithms," https://en.wikipedia.org/wiki/Symmetric-key_algorithm.

[2] A. M. El-Semary and M. M. A. Azim, "Counter Chain: A New Block Cipher Mode of Operation," *Journal of Information Processing Systems*, vol. 11, no. 2, 2015.

[3] D. Hook, *Beginning cryptography with Java.* John Wiley & Sons, 2005.

[4] K.-T. Huang, Y.-N. Lin, and J.-H. Chiu, "Real-time mode hopping of block cipher algorithms for mobile streaming," *International Journal of Wireless & Mobile Networks*, vol. 5, no. 2, pp. 127–142, 2013.

[5] K.-T. Huang, J.-H. Chiu, and S.-S. Shen, "A Novel Structure with Dynamic Operation Mode for Symmetric-Key Block Ciphers," *International Journal of Network Security & Its Applications (IJNSA)*, vol. 5, no. 1, p. 19, 2013.

[6] M. Vaidehi and B. J. Rabi, "Design and analysis of AES-CBC mode for high security applications," in *2nd International Conference on Current Trends in Engineering and Technology (ICCTET), 2014.* IEEE, 2014, pp. 499–502.

[7] Q. Li, C. Zhong, K. Zhao, X. Mei, and X. Chu, "Implementation and analysis of AES encryption on GPU," in *2012 IEEE 14th International Conference on High Performance Computing and Communications ,.* IEEE, 2012, pp. 843–848.

[8] A. Ramesh and A. Suruliandi, "Performance analysis of encryption algorithms for Information Security," in *2013 International Conference on Circuits , Power and Computing Technologies (ICCPCT),.* IEEE, 2013, pp. 840–844.

[9] W. Stallings, *Cryptography and network security: principles and practices.* Pearson Education India, 2006.

[10] G. Kumar, M. Rai, and G.-s. Lee, "Implementation of Cipher Block Chaining in Wireless Sensor Networks for Security Enhancement," *International Journal of Security and Its Applications*, vol. 6, no. 1, pp. 57–72, 2012.

[11] J. Thakur and N. Kumar, "DES, AES and Blowfish: Symmetric key cryptography algorithms simulation based performance analysis," *International journal of emerging technology and advanced engineering*, vol. 1, no. 2, pp. 6–12, 2011.

[12] A. M. Alabaichi, R. Mahmood, F. Ahmad, and M. S. Mechee, "Randomness analysis on Blowfish block cipher using ECB and CBC modes," *Journal of Applied Sciences*, vol. 13, no. 6, p. 768, 2013.

[13] D. Jayasinghe, R. Ragel, J. A. Ambrose, A. Ignjatovic, and S. Parameswaran, "Advanced modes in AES: Are they safe from power analysis based side channel attacks?" in *32nd IEEE International Conference on Computer Design (ICCD), 2014*. IEEE, 2014, pp. 173–180.

[14] A. Desai, K. Ankalgi, H. Yamanur, and S. S. Navalgund, "Parallelization of AES algorithm for disk encryption using CBC and ICBC modes," in *Fourth International Conference on Computing , Communications and Networking Technologies (ICCCNT), 2013*. IEEE, 2013, pp. 1–7.

[15] K. R. Saraf, V. P. Jagtap, and A. K. Mishra, "Text and Image Encryption Decryption Using Advanced Encryption Standard," *International Journal of Emerging Trends & Technology in Computer Science*, 2014.

[16] S. Ahmad, M. R. Beg, and Q. Abbas, "Energy efficient sensor network security using Stream cipher mode of operation," in *International Conference on Computer and Communication Technology (ICCCT), 2010*. IEEE, 2010, pp. 348–354.

[17] N. Singhal and J. Raina, "Comparative Analysis of AES and RC4 Algorithms for Better Utilization," *International Journal of Computer Trends and Technology*, vol. 2, no. 6, pp. 177–181, 2011.

[18] D. S. A. Elminaam, H. M. Abdual-Kader, and M. M. Hadhoud, "Evaluating The Performance of Symmetric Encryption Algorithms." *IJ Network Security*, vol. 10, no. 3, pp. 216–222, 2010.

[19] S. B.L, A. Shanbhag, and A. S. D'Souza, "A Comparative Performance Analysis of DES and BLOWFISH Symmetric Algorithm," *International Journal of Innovative Research in Computer and Communication Engineering*, 2014.