A comparative analysis of AES common modes of operation

Conference Paper · April 2017

DOI: 10.1109/CCECE.2017.7946655

2 authors:

Sultan Almuhammadi
King Fahd University of Petroleum and Minerals
16 PUBLICATIONS **79** CITATIONS
SEE PROFILE

Ibraheem Alhejri
King Fahd University of Petroleum and Minerals
3 PUBLICATIONS **19** CITATIONS
SEE PROFILE

A Comparative Analysis of AES Common Modes of Operation

Sultan Almuhammadi and 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*—With the rapid increase of the Internet users, network security becomes very essential. Cryptography plays a major role in network security. However, cryptographic systems consume considerable amounts of resources, like memory, CPU time, encryption and decryption time. In this paper, we compared the most common block cipher modes of operation on AES according to the recommendations of the National Institute of Standards and Technology (NIST). The comparison is done in terms of encryption time, decryption time, and throughput with variable data packet sizes. The results of the comparison 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.**

***Index Terms*—Advanced Encryption Standard (AES), Mode of operation, Performance metrics, Encryption, Decryption.**

I. Introduction

Cryptographic algorithms are very important in information security where data is encrypted at the sender side and decrypted at the receiver side. The Advanced Encryption Standard (AES) is a symmetric block cipher adopted by the National Institute of Standards and Technology (NIST) in 2001. A block cipher processes the plaintext of a fixed length, knows as the *block size*. If the length of the plaintext is larger than the block size, it must be divided into several blocks. Typically, the last block of the plaintext must be padded to match the block size. Modes of operation are formal descriptions of the way that the block encryption on a plaintext with size larger than a block size is handled. They also provide desired services such as authenticity, integrity, and confidentiality.

In this paper, we compare different modes of operation on AES for best performance. The modes of operations considered in this paper are those recommended by the NIST as discussed in Section II. The performance evaluation of these modes is based on the following metrics: encryption time, decryption time, and throughput with variable data packet size. We presented a new study that fills the gaps in previous studies regarding the modes of operation on AES.

II. 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 [1]. This section explains these common modes of operation and discusses the results obtained from previous work related to our study.

A. Electronic Code Book (ECB)

ECB describes the use of a symmetric cipher in its raw form, where each block is encrypted independently [2]. The main drawback of ECB mode is that identical plaintext blocks are encrypted into the identical ciphertext blocks [3]. Thus, data patterns in ECB mode are not hidden enough, which may lead to some vulnerabilities [4]. ECB is the easiest and fastest mode to implement. It is the most common mode of AES algorithm used in commercial applications [5]. The two formulae used by the ECB mode for encryption and decryption are as follows:

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

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

B. Cipher Block Chaining (CBC)

The CBC mode solves the problem in ECB. It reduces the likelihood of appearing repeated patterns in the ciphertext [6]. In CBC mode, before encrypting a block of plaintext, it is XORed with the previous ciphertext block. The first plaintext block is XORed with an initialization vector (IV) [4]. The encryption and decryption in CBC are done as follows:

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

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

Large plaintext with possibly repeated patterns can be handled more securely in this mode [7]. Thus, if the same plaintext is encrypted multiple times, the resulting ciphertexts are distinct due to the use of IV. CBC mode requires more processing time than ECB mode due to its chaining mechanism [8]. CBC can be synchronized to avoid channel noise error propagation [2]. Unlike ECB, the CBC mode does not support parallelism, and therefore, it is not recommended for disk encryption [1].

C. Cipher Feedback (CFB)

In CFB mode, a plaintext block of length s bits is XORed with a portion of the output of the encryption module fed with some shift-register R. Figure 1 illustrates an example of the

978-1-5090-5538-8/17/$31.00 ©2017 IEEE

CFB mode with a 64-bit encryption module and $s \leq 64$. The encryption and decryption in CFB are done as follows:

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

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

Where $M_s[E_k(R_i)]$ is the most significant s bits of the encrypted value of R_i. The process is repeated with the next input blocks until the ciphertext is created. Each consecutive input block is encrypted to generate an output block. The shift-register is updated by: $R_i = \text{Shift}(R_{i-1}, C_i)$. Like CBC, in the CFB mode, the block of plaintext is based on the result of the previous ciphertext blocks. Therefore, multiple cipher operations cannot be carried out in parallel. The CFB mode does not require any padding to handle the plaintext that has a variable-length (not necessarily a multiple of the block size) [3]. The CFB mode can be synchronized to avoid channel noise error propagation [2].

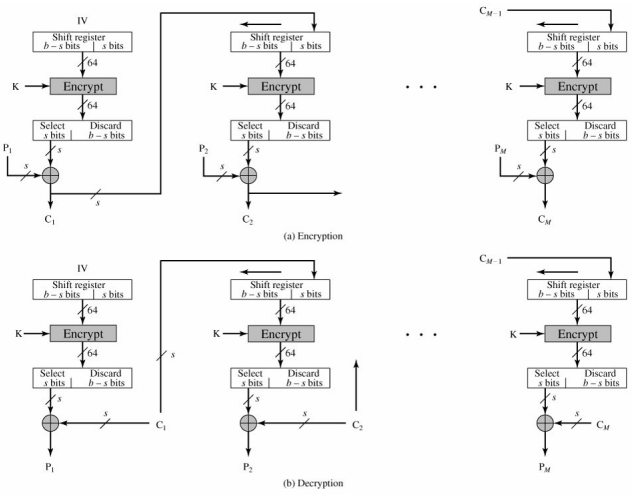

Fig. 1: Cipher Feedback (CFB) [9]

D. Output Feedback (OFB)

This mode is similar to CFB, but instead of using C_i to update the shift-register, we use the selected portion of the output of the encryption module to feedback the shift-register. First, the shift-register is initialized by a random initialization vector, and then it is updated by shifting the $M_s[E_k(R_i)]$ bits for each block. Thus, the shift-register in OFB is updated as follows:

$$R_i = \text{Shift}(R_{i-1}, M_s[E_k(R_{i-1})])$$

The encryption and decryption formulae for the OFB mode are similar to the ones for the CFB mode. Moreover, the OFB mode does not require any padding to handle the plaintext that has a variable-length [3]. In the implementation of CFB and OFB's, only the encryption module is needed for their cryptographic applications [2].

Unlike CFB where the ciphertext is the feedback, the feedback in OFB is taken directly from the encryption module [1]. Thus, the XOR operation of each block of the plaintext in OFB is performed independently of both ciphertext and plaintext. Therefore, bit errors do not propagate in OFB mode during transmission.

E. Counter Mode (CTR)

In the CTR mode, a counter R that starts in an initial value together with a nonce is used instead of a shfit-register to generate a key stream. The key stream is XORed with the plaintext to generate the ciphertext. The counter is updated simply by: $R_i = R_{i-1} + 1$. The encryption and decryption in the CRT mode are as follows:

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

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

Like OFB and CFB modes, the CTR mode does not require any padding to match the block size of the cipher. Moreover, the encryption of a plaintext block 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 [1]. However, the CTR mode does not provide data integrity.

F. Related Work

Many researchers have published comparative analysis on the performance of different modes of operation [5], [8], [10]–[13]. For example, in [10], the authors compared six different symmetric key encryption algorithms using ECB mode, namely: DES, Triple DES, AES, RC2, RC6 and Blowfish. The authors compared the algorithms in terms of CPU clock cycle, CPU time, encryption time, and battery power consumption. Then, in [11], the authors compared AES and RC4 using three modes of operation, namely: ECB, CBC and CFB. The comparison was in terms of throughput, encryption time, decryption time, CPU time and memory utilization. The authors reported that ECB takes less time than CBC and CFB. According to [8], the authors reported that the difference in execution time is negligible when comparing between DES, AES and Blowfish using four modes of operation, namely ECB, CBC, CFB and OFB. In [5], two modes of operation (ECB and CBC) were compared in terms of throughput and data transfer cost. The comparison study in [12] showed that AES with CBC is the fastest approach among all the other algorithms (MIE, VC and N/KC) in image encryption. In [13], the authors compared between three algorithms (DES, AES and RSA) and showed that AES consumes less encryption and decryption time than the other two algorithms.

In this paper, we targeted in one study the most common five modes of operation (ECB, CBC, CFB, OFB and CRT) in terms of encryption time, decryption time, and throughput. We conduct our study on the most popular encryption algorithm AES. This study fills the gaps in the previous studies on AES. Table I summarizes the encryption algorithms, the modes of operation and the performance metrics of all previous studies, and compares them to our work.

TABLE I: Summary of existing studies on modes of operation

Year and Reference	Algorithm	Mode of Operation	Performance Metric
2010 [10]	DES, 3DES AES, RC2 RC6, Blowfish	ECB	CPU Clock Cycle , CPU Time Encryption Time, Battery Power
2011 [11]	AES, RC4	ECB, CBC CFB	Throughput, Encryption Time Decryption Tim, CPU Time Memory Utilization
2011 [8]	DES, AES Blowfish	ECB, CBC CFB, OFB	Execution Time
2012 [5]	AES	ECB, CBC	Throughput, Data transfer cost
2013 [14]	DES, AES Blowfish	ECB	Execution Time, Throughput Memory Usage
2013 [15]	AES	CBC, ICBC	Encryption Time Decryption Time
2014 [12]	AES	CBC	Encryption Time
This paper	AES	ECB, CBC CFB , OFB CTR	Encryption Time, Decryption Time, Throughput

III. EXPERIMENTAL DESIGN

In our experiment, we used available java classes to do the performance evaluation of the operation modes. The implementation uses managed wrappers for AES-128 available in $java.crypto$ and $java.security$ [CryptoSpec] that wraps unmanaged implementations available in JCE (Java Cryptography Extension) and 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 computer with 2.30 GHz CPU and 4 GB RAM is used to conduct the experiment. The computer encrypts files with different sizes in the ranges 500 KB to 200 MB. For the CFB and OFB modes, we used the value of $s = 16$ bits. The performance evaluation is conducted in terms of encryption time, decryption time and throughput, which are defined as follows:

1) Encryption time: is the time consumed to generate a ciphertext from a plaintext by an encryption algorithm.
2) Decryption time: is the time consumed by a decryption algorithm to reproduce a plaintext from a ciphertext.
3) Throughput: 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 increases, the power consumption decreases [14].

IV. EXPERIMENTAL RESULTS AND ANALYSIS

The following results show the effect of changing file size on the performance in each mode.

A. Performance Evaluation Based on Encryption Time

Figure 2 shows the performance of the modes of operation in terms of encryption time. We plotted the encryption time of these modes for different file sizes. We observed that ECB takes less time than other modes. Moreover, we noticed that the enryption times for CFB, OFB, CBC and CTR are almost the same. Generally, the differences between the modes are negligible in small files (less than 10 MB). Table II summarizes the results of the comparative encryption time analysis among all these modes of operation.

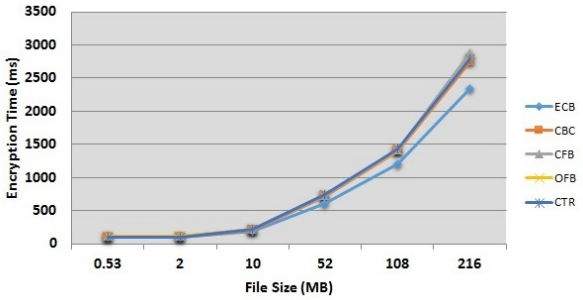

Fig. 2: Encryption Time Analysis

TABLE II: Encryption Time Comparison

File Size (KB)	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

B. Performance Evaluation Based on Decryption Time

We plotted the decryption time in Figure 3 with the same file sizes as we did for encryption time to show the performance of these modes in terms of decryption time. We observed that ECB again takes less time than other modes for files of sizes above 10 MB. We also noticed that OFB takes less time than CBC and CFB (with $s = 16$ bits). While CTR is better than other modes for files of sizes less than 200 MB. The differences between these modes are negligible in small files. The comparative decryption time among block cipher modes is shown in Table III.

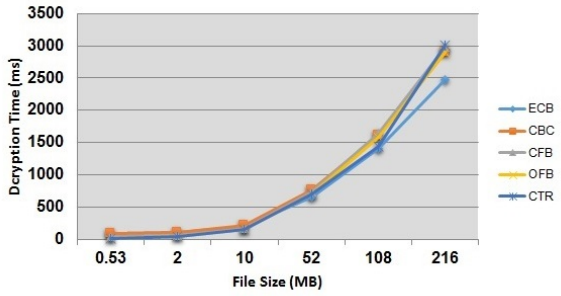

Fig. 3: Decryption Time Analysis

C. Performance Evaluation Based on Throughput

In our experiment, we use AES with a fixed key size of 128 bit. The difference of the throughput among the modes of operation is relatively small in general. The encryption and decryption throughput of these modes are shown in Figure 4

TABLE III: Decryption Time Comparison

File Size (KB)	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

and 5 respectively. We noticed that the ECB mode is the fastest among the other modes of operation, and therefore, consumes less power than the other modes. Table IV summarizes the numerical results of the comparative analysis of the encryption and decryption throughput. We observed that both CBC and CFB are slightly faster in decryption than in encryption. While the OFB and CTR modes have quite the same throughput for encryption and decryption.

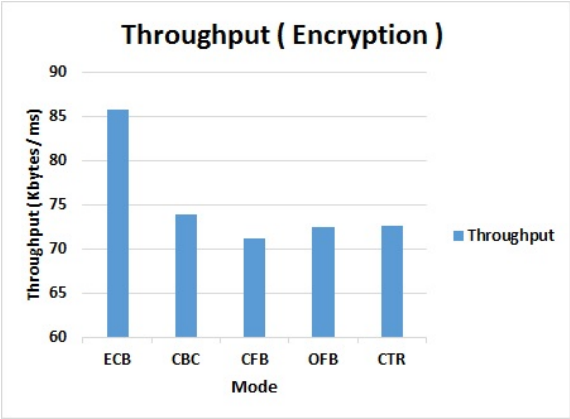

Fig. 4: Comparison of Encryption Throughput

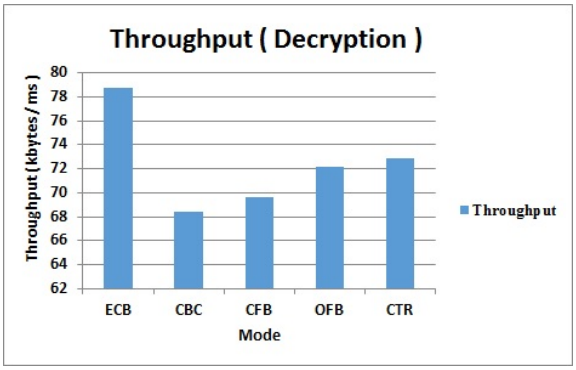

Fig. 5: Comparison of Decryption Throughput

V. Conclusion

In this paper, we presented a detailed comparison of the most common block cipher modes of operation on AES in terms of encryption time, decryption time and throughput. The modes of operation we targeted in this study are ECB, CBC, CFB, OFB and CTR as recommended by NIST for bolck ciphers. Our study shows that ECB takes less time to encrypt and decrypt than the other modes. The difference between the modes is relatively small for small files. However, with big size files, there is a noticeable difference in the performance among these modes.

TABLE IV: Throughput Numerical Results (KB/ms)

Mode	Encryption Throughput	Decryption Throughput
ECB	85.75	78.72
CBC	73.92	68.43
CFB	71.28	69.61
OFB	72.49	72.12
CTR	72.60	72.88

Acknowledgment

The authors would like to thank King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, for supporting this research. Figures and descriptions in this paper were provided by the authors and are used with permission.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

[12] 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.

[13] P. Mahajan and A. Sachdeva, "A study of encryption algorithms aes, des and rsa for security," *Global Journal of Computer Science and Technology*, vol. 13, no. 15, 2013.

[14] 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.

[15] 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.