Evaluating AES Performance Using NIST Recommended Block Cipher Modes of Operation

CONFERENCE PAPER · MAY 2015

3 AUTHORS:

Abdelrahman Altitgani
University of Dammam
2 PUBLICATIONS 4 CITATIONS

Muawia Abdelmagid
University of Dammam
1 PUBLICATION 0 CITATIONS

Bazara Barry
University of Khartoum
22 PUBLICATIONS 40 CITATIONS
Evaluating AES Performance Using NIST Recommended Block Cipher Modes of Operation

Abdelrahman Altigani¹, Muawia Abdelmagid² and Bazara Barry³

¹, ²University of Dammam, ³University of Khartoum,
University of Dammam, Dammam, Eastern Province, KSA
aaaltigani@uod.edu.sa, makhlafalla@uod.edu.sa, bazara.barry@outlook.com

Abstract: Encryption plays a central role in providing electronic security and countering cybercrimes. In particular, symmetric encryption algorithms have gained popularity on the Internet and its associated applications. When using a symmetric encryption algorithm, specifically the AES, the Block Cipher Mode of Operation to be used must be specified. Usually this decision is influenced by two main factors: (i) security; and the (ii) performance of the mode. Most of the related literature explores the security of the modes. In contrast, this research paper explores, compares and evaluates the performance of the five modes of operation recommended by the National Institute of Standards and Technology (NIST). Based on the conducted experiments and obtained results, the modes of operation that provide better performance are shown.

Keywords: Block Cipher Modes of Operation, Advanced Encryption Standard, Performance.

1. Introduction

Due to the increasing number of reported computer related security incidents, and according to the statistics maintained by the National Institute for Standards and Technology (NIST) [1] and the Computer Emergency Response Team (CERT) [2], information security has become a vital research area. To avoid such incidents or attacks, we usually need to employ an appropriate security service or services. For instance, to counter eavesdropping attacks we need to enable the confidentiality security service. Providing the confidentiality security service can be achieved in many ways but the most widely used technology to provide confidentiality is Cryptography [3].

Cryptography is the practice of scrambling messages so that even if detected they are very hard to decipher [4]. Cryptographic algorithms are usually classified into two main categories: (i) Symmetric Encryption Algorithms, and (ii) Asymmetric Encryption Algorithms. The key difference between these two categories is that the symmetric algorithms use a single key for both encryption (scrambling the data) and decryption (deciphering the scrambled data to their original plain form), while the asymmetric algorithms use two different keys: one for encryption and the other for decryption. It is worth mentioning that for encrypting bulks of data we usually use a symmetric encryption algorithm because the symmetric algorithms are generally faster than the asymmetric algorithms.

For several concerns, including but not limited to security, efficiency and interoperability, a standard symmetric algorithm is required. In 2001, the NIST announced to the public that rijndael would be the new standard symmetric encryption algorithm or the Advanced Encryption Algorithm (AES) [5].

Most of the well-known symmetric encryption algorithms (including the AES) deal with a fixed size of input (plaintext) at a time. The input size in the case of the AES is 128 bits [6]. This means that even if we wanted to encrypt data less than 128 bits using the AES, we should firstly apply some padding in order to reach the 128 bits. Then we can use the AES. Although the last case might happen, the most occurring scenarios are encrypting a file with a size more than 128 bits. The intuitive solution to this problem is to divide the input into many blocks of 128 bits, encrypt each block, and then concatenate the resulting encrypted blocks. In fact, this is one of the block cipher modes of operation known as Electronic Codebook (ECB) [7]. Although this mode of operation might be satisfactory in some cases, the security of this model is questionable because it might reveal the pattern of the encrypted data, especially if the input data was an image [8].

The NIST recommended five modes of operation that can be used. These modes vary in the technique, but all of them attempt to answer the question “How should the encryption/decryption process of each individual block interact with each other?” [7]. These modes of operation are Electronic Codebook (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB) and Counter (CTR).

This recommendation has been made relying on various factors including the mode popularity and frequency of use. As we are going to see through this research, in recent researches new modes has been suggested such as GCM, CCM and EAX. These newly proposed modes might support additional features; however no standardization institute such as the NIST had standardized them yet.

According to the mode of operation that will be used, the performance of the encryption/decryption process might be affected. In some cases the performance variation might be tolerable but in other scenarios such as real time applications (VoIP, Videoconferencing etc.) any delay might be intolerable. In such cases, a considered decision must be made in order to determine which mode should be used in order to achieve the least possible delay.

On the other hand, some applications are not concerned with slight performance degradation, but are highly
interested in achieving the most possible security. In these cases, many things differ including the mode of operation which will be used.

This research paper evaluates the performance of the AES algorithm – the standard symmetric encryption algorithm – with five block cipher modes of operation recommended by the NIST (i.e. ECB, CBC, CFB, OFB and CTR). The topic of the block cipher modes of operation is well studied in the literature [5, 6, 7, 8 and 9], but no clear results have been obtained regarding which mode is better in terms of performance. Another key issue is that most of materials that the researchers managed to come across do not try to bind the modes of operation to any specific encryption algorithm, and even if they did, the performance issue is usually neglected. The investigators of this research argue that since AES has become the standard algorithm, it will be handier if the AES performance is computed over all the modes of operation recommended by the NIST.

This research sheds light on the performance of the five modes of operation recommended by the NIST when used with the AES, and draws conclusions with regard to rating of these modes based on experimental results.

The next section of this paper provides a review on the modes of operation design pros and cons. Then the related work is going to be briefly discussed. Subsequently, the paper illustrates implementation details and the methodology used to perform the modes of operation performance evaluation. Then obtained results are going to be revealed and discussed. Finally, a brief conclusion is presented to summarize all the research outcomes and the paper highlights some considerations for future work.

2. Block Cipher Modes of Operation Recommended by the NIST

All the modes of operation require breaking the input into blocks of a given size. That size should be equal to the algorithm input block size. However, this research is interested only in the AES algorithm, which has an input block of 128 bits size. Accordingly, this research will always assume that the block size is 128 bits.

As described in section I; the (ECB) Mode of operation is the most straightforward mode of operation (see Figure 1).

![Figure 1. Electronic Codebook Mode of Operation [8]](image)

The plaintext data is simply divided into blocks of 128 bits. If the last block is not equal to 128 bits, a padding scheme can be used, such as PKCS7, ANSI X.923 or any other padding scheme. Those blocks are then encrypted one by one using the chosen encryption algorithm. The resulting encrypted blocks are concatenated to form the encrypted text.

One of the advantages of this mode is that the error is not propagated. That means if an error occurs during the encryption time in any block, it does not affect the other blocks, and the remaining blocks can still be decrypted correctly. On the other hand, this model is deterministic which means identical input blocks (i.e. before encryption) will result in identical encrypted blocks. Also, the order of the encrypted blocks can be changed without the receiver noticing. In general, this mode is not recommended for encrypting data that is more than one block [1, 11].

The second mode recommended by the NIST is the Cipher Block Chaining (CBC) (see Figure 2).

![Figure 2. Cipher Block Chaining Mode of Operation [8]](image)

This mode initially generates an Initialization Vector (IV) of 128 bits. According to the design specifications of the CBC mode; there is no need to keep the IV secret, but it should be unpredictable [7]. The input is then divided into 128 bits blocks and an X-OR operation is applied between the IV and the first input block. The result is encrypted, and the encryption output will be used as an IV with the second plain block.

Intuitively, CBC mode of operation addresses the main problem in the ECB mode, so even if there are identical input blocks, the resulting ciphered blocks will be different due to the X-OR operation with the IV or the previous ciphered block. The main disadvantage is that if an error occurs in one of the encrypted blocks, it will propagate to all the remaining blocks. However, using an incorrect IV in the decryption only affects the first block. In [11], it is claimed that some complex attacks such as Chosen Plain Text Attacks (CPA) or Chosen Ciphertexts Attacks (CCA) can compromise the CBC unless the IV has a random value, andAuthenticated Encryption is used (in which besides the encryption, authentication is also performed) [12].

The third mode of operation recommended by the NIST is the Cipher Feedback Mode (CFB) (see Figure 3).
The CFB mode also requires an unpredictable IV. The plaintext will be divided into 128 bits blocks, with no need to use any padding scheme even if the last block is not 128 bits. The IV is encrypted. Assume $s$ is an integer value that satisfies: $s \in [1, 128]$, then we select the $s$ Least Significant Bits (LSB) out of the encrypted IV and discard the remaining 128-$s$ bits. An X-OR operation is then applied between the $s$ bits selected from the encrypted IV, and the $s$ LSB bits of the first plain block. The result of this X-OR operation will be the first $s$ bits in the encrypted text. It will also concatenate the $s$ LSB bits for the next input block with the remaining 128-$s$ bits of the first plain block. This mode provides adequate security if a unique IV is used [13].

Decryption operation is performed using the same encryption function [1], and this feature is highly desirable specially when dealing with devices with limited available storage system and memory. However, it has the same disadvantage pointed in the CBC mode, namely, error propagation.

The fourth mode of operation recommended by the NIST is the Output Feedback Mode (OFB) (see Figure 4).

This mode also requires breaking the message into 128 bits blocks without a need for a padding scheme. The IV is needed to be a nonce (i.e. unique for each execution of the mode under the given key) [7]. The mode works by firstly encrypting the IV. The encrypted IV (output) will be the input of the next encryption process, and will also enter in an X-OR operation with the first plaintext block to result in the first ciphered block. In the second step we encrypt the output of the first encryption process (i.e. the encrypted IV), then apply the X-OR operation between the double encrypted IV and second plaintext block to get the second ciphered block, and so on. To get the last ciphered block (e.g. block n) we use the output of the last encryption process (i.e. the n-1 times encrypted IV) encrypt it one more time, and X-OR it with the last block. If the last block is less than 128 bit (e.g. 100 bits) the 100 Most Significant Bits (MSBs) of the n times encrypted IV and those 100 bits in the last plaintext block will be subject to an X-OR operation to result in the last ciphered block. The main merit of this mode is that if an error occurs in any block during the encryption process, it will not be propagated to the next blocks. However, according to [1] it is vulnerable to attacks performed by modification of bits in the encrypted stream.

The last mode of operation recommended by the NIST is the Counter mode of operation (CTR) (see Figure 5).

Like all the previous modes, the secret data must be divided into blocks of 128 bits. We also need a different IV (counter) for each plaintext block. The mode works by simply encrypting each counter. Then an X-OR operation is performed between the resulting encrypted counter and the corresponding plaintext block. If the last block is not 128 bits (e.g. 100 bits) we use the 100 MSBs in the last counter and apply an X-OR operation to those bits and the 100 bits in the last block.

It can be easily noted that there is no error propagation because each block is treated as a separate message. When using the CTR mode, there is no use for the decryption part of the algorithm. Accordingly, we have to implement less code. As mentioned before, this feature is sometimes highly desirable. It should be borne in mind that, according to machine specifications and the count of Cores, in the CTR mode we can encrypt or decrypt all the blocks in parallel, which positively affects the performance. The CTR mode is at least as secure as all the previous modes of operation [1]. For further discussion about block cipher modes of operation design, advantages and disadvantages the reader is invited to check [1, 7, 8, 9, 10 and 11].

3. Related Work

3.1 Discussing Modes of Operation in a Popular Textbook [10]

In [10] all these modes are discussed in detail. Most advantages and disadvantages for each mode had been properly addressed. In addition; examples had been provided to describe some of the security issues associated with all these modes of operation. Although the performance or the efficiency, as mentioned in this textbook, had been
highlighted; the results are a bit vague. In all modes of operation; they mentioned that “speed is the same as the block cipher”. This can be courageously challenged relying on the design differences of these modes of operation, and the results obtained by this research. It is believed that the textbook authors were trying to say that the performance variation between those modes is not significant and can be neglected. What we are trying to do in this research is to measure this amount of variation more accurately.

3.2 Advantages of the CTR Mode of Operation [14]
In this paper, the authors introduced how CTR mode of operation works; then they start enumerating the advantages of this mode including the software and the hardware efficiencies. They mentioned that a well-optimized implementation for the CTR mode can result in a substantially faster encryption than the well-optimized implementation for the CBC mode (more than four times faster). Although this claim relies on results obtained by [15], the results obtained by our research differ significantly. This variation may be attributed to many things including the inspection machine specifications used to collect the results in [15], the size of the plain message or the implementation techniques used in both researches.

3.3 Evaluation of Some Block Cipher Modes of Operation [16]
This detailed technical report explores almost all the existing modes of operation, including the five modes that will be explored by this research. It provides a comprehensive evaluation of the security level of all the modes of operation. It also tackles the performance factor, but no clear comparison between the modes of operation (as the one provided by this research) is provided.

3.4 Securing ECB Mode of Operation for Encrypting Images [17]
The assumption of this research paper is that ECB mode of operation is the most suitable mode of operation due to its simplicity. Huang and Ying-Hao analyzed the cause of the pattern revealing. They also proposed some solutions to overcome this drawback in the ECB mode including a technique called “adding number series” as well as using compression before encryption. A good effort had been spent on their research paper, however the security of the ECB mode of operation is challenged in the literature [1, 10]. In addition, the authors did not investigate the performance of any mode of operation as we are trying to do in this research.

3.5 Comparing Block Cipher Modes of Operation on MICAz Sensor Nodes [18]
This study scope is very limited to the MICAz Sensor Nodes, and accordingly the obtained results are very specific to the platform in question. Also, the examined modes of operation differ from the NIST recommendations, which are going to be examined in this research.

3.6 The Security and Performance of the Galois/Counter Mode (GCM) of Operation [19]
This interesting paper focuses on exploring the security and performance of a newly proposed mode which is Galois/Counter Mode (GCM) of operation. The key difference between [19] and this research is that the authors of this research are trying to conduct a comparison between five modes of operation (recommended by the NIST) to come up with recommendations to allow choosing any mode of operation according to requirements. In [19], however, the authors are only interested in exploring the specific GCM of operation which is not among the ones recommended by the NIST. The significance of these five modes of operation has already been highlighted in the introduction.

3.7 Modes of Operation of the AES Algorithm [11]
This paper explores the existing literature in a comprehensive manner. Authors compare all the five NIST modes of operation at the theoretical level without testing the modes of operation in an experimental setup. Furthermore our key research problem, determining the mode that is best in terms of performance and the size of the difference between the modes, is not addressed.

So, after surveying the literature, we can conclude that no research evaluated the performance of all modes recommended by the NIST in an accurate manner. This will be the objective of this research paper.

4. Methodology and Implementation
The most straightforward methodology to evaluate the performance of the five modes of operation with the AES is to implement the AES with all the five modes of operation. We can then run the AES on a fixed input to extract the results. Accordingly, a testing code had been built using VC++ 2010 (Visual Studio 2010). As not all the five modes of operation are supported by the .NET framework [20] we use the standard Crypto++ library which implements the AES with all the five modes considered by this research [21].

The inspection machine specifications are: Octa-Core 3.4 GHz CPU, Front Side Bus 1333 MHz and 16GB DDR3 (1600MHz) with Windows 7 Ultimate (64 bit) as the operating system.

In the testing code, we used the AES with the default key length (128 bits). The testing code includes the five modes of operation ECB, CBC, CFB, OFB and CTR. The encryption key and the Initialization Vector IV (for all the modes of operation except the ECB) are randomly generated.

To measure accurately the elapsed time when performing the encryption/decryption, we used two functions: QueryPerformanceCounter ( ), and QueryPerformanceFrequency ( ). The first one returns the current value of a system high resolution counter; the later returns that counter frequency (ticks per second). Accordingly, we fetch the value of the counter before and after the encryption (as well as the decryption), subtract the before from the after values, and divide the resulting value by the counter frequency returned by the function QueryPerformanceFrequency ( ). The resulting value will be the time in seconds required to perform those instructions. Regardless of some applications which run at the computer start-up, we tried to keep the inspection machine in its initial state, to get accurate results.

We run this testing code ten times (iterations) for each mode of operation and input size to get the encryption/decryption times, and then summarize those ten encryption/decryption experiments by calculating the average values.
In the first 10 iterations we fixed the input size (plaintext data) to 100 characters. In the next ten iterations we fixed the input size to 1,000 characters. In the last ten iterations we fixed the input size to 10,000 characters. The obtained results are discussed and analyzed in the next section.

5. Results

The Following three figures (Figure 6, 7 and 8) summarize all results obtained by running the five modes of operation with the three predefined input sizes (100 - 1,000 and 10,000 characters). When the input size is 100 characters (See Figure 6), the required encryption time ranges between 42.5 microseconds (42.5 MS) in the CFB mode and 64.3 MS in ECB mode. On the other hand, the decryption time ranges between 33.3 MS in the CTR mode and 45.8 MS in the ECB mode. Results presented in Figure 7 and Figure 8 can be interpreted in the same manner.

It should be noted that performance of all these modes of operations is acceptable, however CTR mode of operation performance results in both encryption and decryption are generally good especially when we increase the input size. For example, when the input size is 10,000 characters; the CTR needs 89.1 MS to perform the encryption process while ECB needs 110.3 MS and the CFB needs 123.3 MS to perform the same encryption process. The remaining modes are slightly slower (126.9 MS for CBC and 128.6 MS for OFB) in encrypting the same input size.

In decryption 10,000 characters, CTR mode of operation results are also superior (65.6 MS), while the remaining modes require more time to perform the decryption process. Another observation is that as we increase the input size (e.g. 10,000 characters), the range between encryption/decryption times in CBC mode increases significantly (54.2 MS). This can be justified when inspecting the design logic of the CBC mode which supports the parallel processing in decrypting only [7].

The justification for the CTR mode performance superiority is the simplicity of generating a new counter for each block using the standard incrementing function, and the support of parallel processing in both encryption/decryption operations.

6. Conclusion and Future Work

After inspecting the performance of all five modes of operation recommended by the NIST, the resulted performance of all modes of operation is convergent; excluding the ECB encryption time in small input size such as 100/1,000 characters. However as we increase the input size to 10,000 characters, CTR got better results than other modes of operation recommended by the NIST.

Although authors of this research tried their best to collect the results and analyze results accurately, the previous results and conclusions may differ due to the following reasons:

- The implementation is delivered by the standard Crypto++ library. This library might not support specific features which may cause a variation in the obtained results.
- A limitation on the hardware/software of the inspection machine and processes running in the background at the time of experimentation may increase/decrease the performance results of one mode over the others.

So these issues should be considered, and verified in a future research to support or challenge the results of this research. On the other hand, this research focuses on using just textual data to keep the implementation and the results’ analysis tasks simple. However performance significantly matters in the cases of multimedia input data (audio, video, etc). Accordingly; the matter of using different input data types can be addressed in a future research.

Finally, it is worth mentioning that a new set of modes (such as GCM, CCM and EAX) are getting significant interest by the public. The key feature provided by those modes is that they support both confidentiality as well as authenticity. Hence, the performance evaluation for such modes can be inspected in a future research.
Acknowledgement

This work was funded by the Deanship of Scientific Research of the University of Dammam (KSA), for the Project Grant Number 2014276.

References


[15] Lipmaa, Helger. "IDEA: A cipher for multimedia architectures?.” Selected areas in cryptography. Springer Berlin Heidelberg,


