Applied Engineering
2024; Vol. 8, No. 1, pp. 14-30
https://doi.org/10.11648/j.ae.20240801.12

Research Article

Evaluations of Crypto-System AES Using Multiple Bloc Ciphering Mode

Rakotondramanana Radiarisainana Sitraka* , **Ramafiarisona Hajasoa Malalatiana**

Telecommunication-Automatic-Signal-Image-Research Laboratory of Doctoral School in Science and Technology of Engineering and Innovation, University of Antananarivo, Antananarivo, Madagascar

Abstract

Cryptography, a scientific field that has existed even before the beginning of computer science. This article looks at the development symmetric crypto-system, which falls within the framework of image security, by Advanced Encryption Standard (AES- Advanced Encryption Standard) algorithms. AES algorithm is not integrity protected. Cryptanalysis could use modified encrypted image of each ciphered bloc for generating oracle and getting the key. A new crypto-system uses hash function named AES-GCM (Advanced Encryption Standard-) for solving this problem. Using Galois Counter Mode (GCM) combined with Secure Hash Algorithm 256 bits (SHA-256) or BLAKE2s hash function, the old mode of ciphering like: Cipher FeedBack (CFB), Output FeedBack (OFB), Cipher Block Chaining (CBC), Electronic Codebook Block (ECB) and CounTeR (CTR) mode encryption, will increase the security level at confidentiality and integrity. In this article, robustness of the crypto-system will be evaluated by multiple criteria, indeed the statistical analysis, sensitivity measurement and performance measurement. For the statistical analysis, the histogram is flat and the correlation between adjacent pixel is not linear for the encrypted image. The relation between clear image and encrypted image doesn't exist. For the sensitivity analysis, Number of Pixel Changing Rate (NPCR) and Unified Averaged Changed Intensity (UACI) permit to avoid differential attack of the crypto-system. The two values are respectively near 99% and 30%. For performance measurement, the similarity of the decrypted image and original image will be tested. The Peak Signal to Noise Ratio (PSNR) and Mean Square Error (MSE) are equals of infinity and zero. The decrypted image and original image are the same. The integrity of the image on the crypto-system will be evaluated by modifying one bit of the encrypted image. The decryption process doesn't give decrypted image and show that the tag value is incorrect.

Keywords

AES, Cryptography, GCM, Hash, Security, Symmetric

1. Introduction

In the literature review, an image is transmitting on network in plaintext. For protecting the image, cryptography is used. The encryption used is generally permutation of pixel based, scrambling, space filling and chaotic sequence. Later, the AES algorithm will be proposed and uses generally classic mode like CFB, OFB, CBC, ECB, CTR [1-6]. This encryption doesn't integrity protected. The must problem is that the cryptanalysis could recover the key by sending modified

*Corresponding author: radiarisainanasitraka@yahoo.fr (Rakotondramanana Radiarisainana Sitraka)

Received: 9 March 2024; **Accepted:** 21 March 2024; **Published:** 2 April 2024

encrypted image and having dataset for constructing oracle.

This article proposed new method for avoiding this attack by using encryption system and authenticated mode as SHA-256 and BLAKE2s which permits to verify if some attacker modifies the ciphered image. Following the concepts useful for the realization and the explanations all around this system will be developed.

2. Materials and Methods

To protecting the image before sending it, ciphering operation is needed. On the receiver, inverse transformation is done for having the original image. This method could also be improved by using authentication for integrity of the image. In this, the receiver could verify if the encrypted message is modified or not by some attacker. The choice of authentication methods is between SHA-256 and BLAKE algorithm. By using python, an application used for ciphering and deciphering image is created and evaluated [1-6].

2.1. General Encryption Synoptic

For protecting data, mathematical transformation named encryption and decryption could be used. Now, the National Institute of Standards and Technology (NIST) specifies that it's more secure to use AES crypto-system [1-4].

2.1.1. Encryption

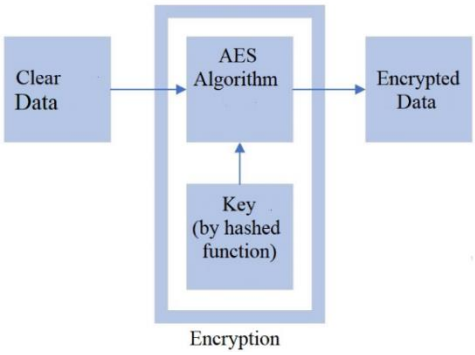


Figure 1. *Bloc diagram of encryption.*

Clear data: this is the original data to be encrypted.

Encryption: this step consists of encrypting the data using the AES algorithm with a key hashed by hash functions.

Encrypted data: this is the result of the data that has been encrypted.

The *Figure 1* illustrates the schema bloc diagram of encryption.

2.1.2. Decryption

The decryption operation is done like the encryption operation except that here the input is the encrypted data, and it's illustrated be the Figure 2.

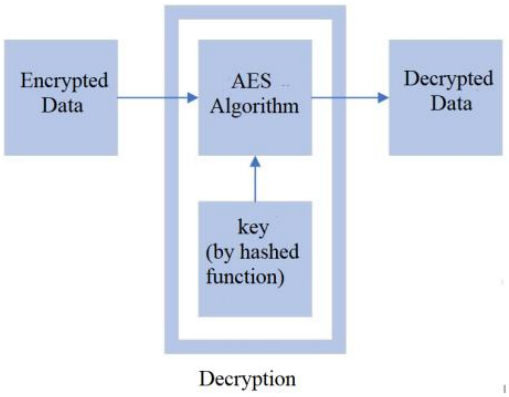


Figure 2. *Bloc diagram of decryption.*

2.2. Choice of Encryption Algorithm

AES algorithm used many modes for each block ciphering mode: GCM; CFB, OFB, CBC, ECB, CTR.

2.2.1. AES Algorithm

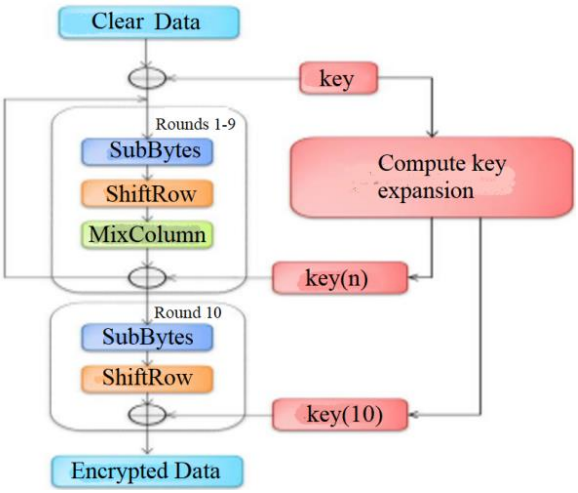


Figure 3. *Bloc diagram of decryption.*

The AES algorithm is a symmetric block cipher algorithm, which operates on a group of fixed-length bits, called blocks. It takes a 128-bits or 256-bits input block and produces a corresponding output block of the same size [5-8].

AES requires a second entry, which is the secret key. AES uses three different key sizes: 128, 192 and 256 bits. For its encryption and inverse encryption, the AES algorithm uses a round function composed of four different byte-oriented layers:

1. Substitution of bytes using a substitution table (S-box)
2. Offset rows of the State array by different offsets.
3. Shuffle the data in each column of the status table.
4. Add a round key to the state. The AES algorithm uses this round function for ten rounds, but the first and last round of the AES algorithm differs from other rounds.

This is illustrated in *Figure 3*.

2.2.2. Encryption Modes Used

Whether for Data encryption standard (DES) or more recent symmetric crypto-systems like International Data Encryption Algorithm (IDEA) or AES or for asymmetric crypto-systems like (Rivest Shamir Adleman (RSA) or El Gamal the keys are of fixed length. Messages can be of arbitrary length. To adapt the size of the message to that of the key, the message is broken down into blocks of fixed size corresponding to the sizes of the keys which are then encrypted one by one and which are sent successively. For this, six block cipher modes are possible: ECB, CBC, CFB, OFB, CTR and GCM [9-17].

The ECB mode, Electronic Code Book, is the simplest mode. The message, M, is split into blocks, (Mi) with i≥1, and each block is encrypted separately by:

$$Ci = E(Mi) \quad (1)$$

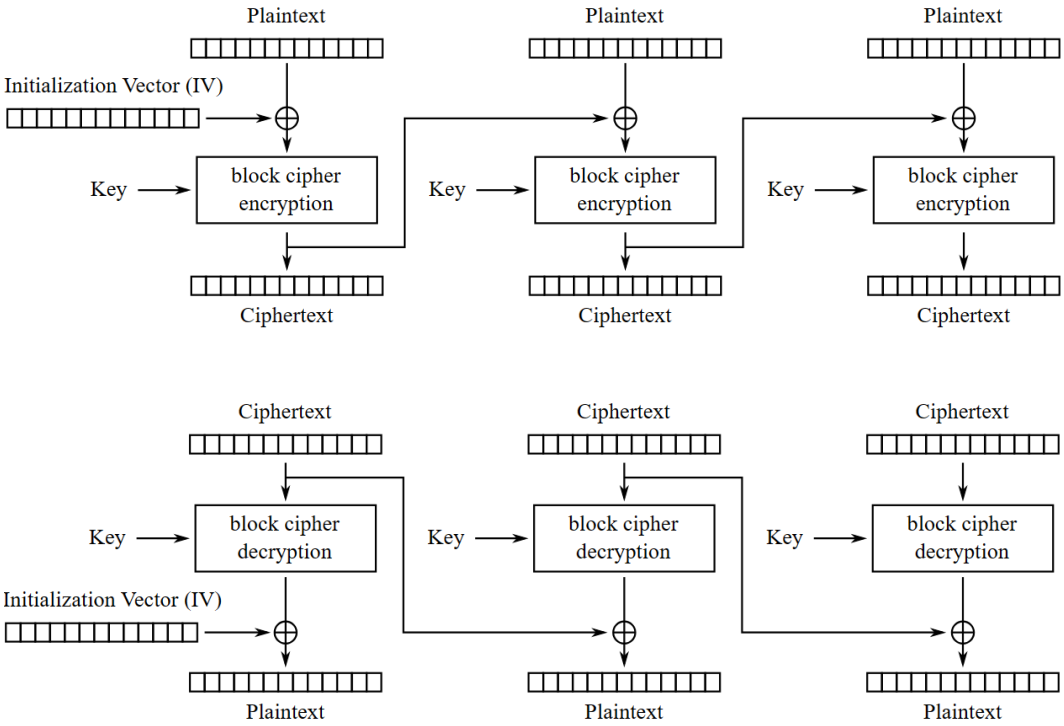
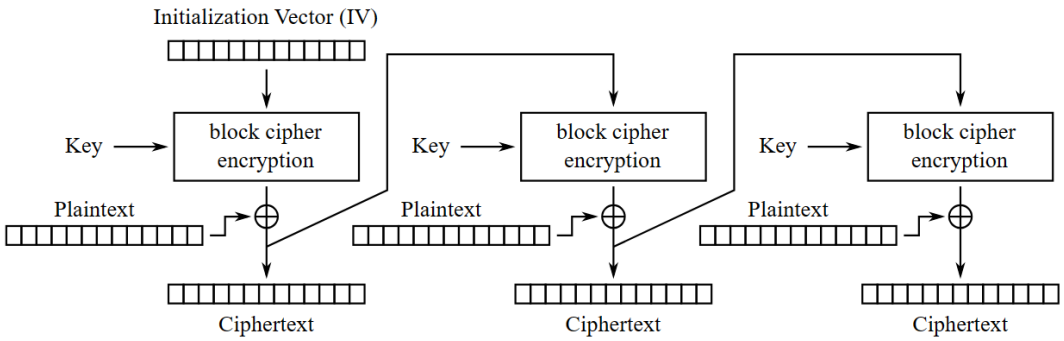


Figure 4. *Encryption and decryption by CBC mode.*

The *Figure 4* illustrates the encryption process using CBC mode.

The CFB mode was introduced so as not to have to calculate the inverse function, Dk, of the cipher function E k. The principle is the same as that of the CBC mode. The message, M, is divided into blocks, (Mi with i ≥1, and each block is encrypted as follows. Beginning an initial block m0, chosen according to the same principles as block C0 in CBC mode. Each clear block mi is XORed with the ciphertext of the previous output block, Ci-1, then the result obtained by the equation 2 for having encrypted with the key by:

$$C1 = Ek\ (M1 \oplus C0);\ C2 = Ek\ (M2 \oplus C1)..... Ci = Ek\ (Mi \oplus Ci\text{-}1) \quad (2)$$



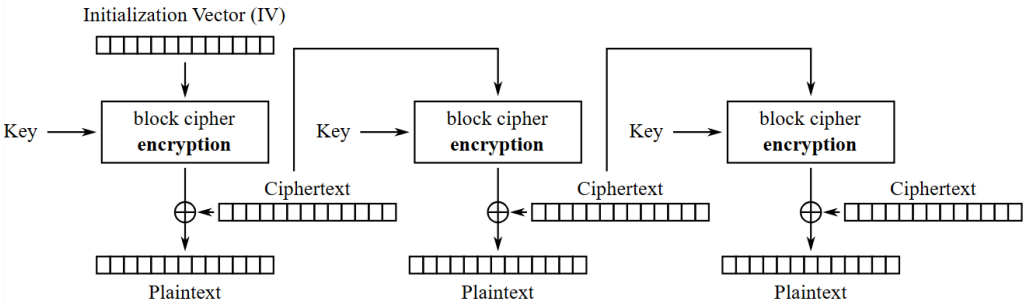


Figure 5. *Encryption and decryption by CFB mode.*

The OFB mode, Output FeedBack, is a variant of CFB which allows having a completely symmetrical encryption and decryption:

$$Z=Ek\ (Zi\text{-}1);\ Ci\ =\ Mi\ \oplus\ Zi \tag{3}$$

This mode is used for example to secure satellite data. The *Figure 6* illustrates how this OFB mode works.

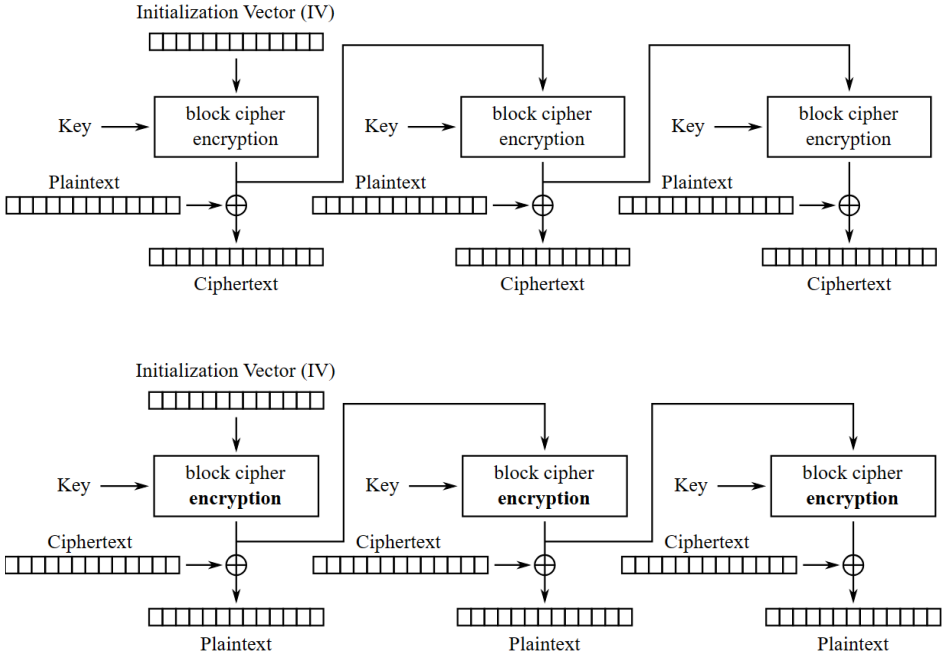
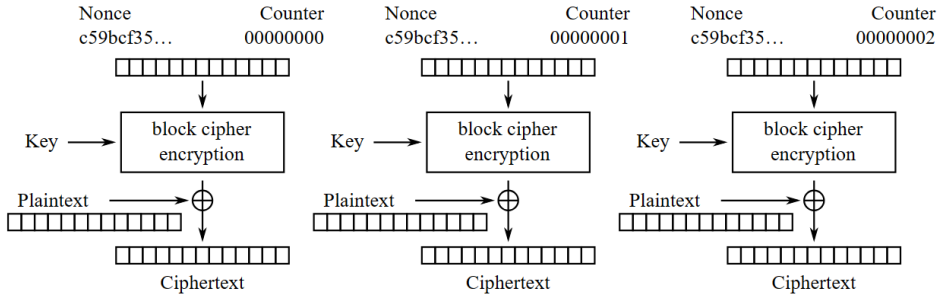


Figure 6. *Encryption and decryption by OFB mode.*

CTR mode, COUNTER is also totally symmetric, but moreover easily parallelized. It uses for encryption an initial value counter T.

$$Ci\ =\ mi\ \oplus\ Ek\ (T\ +\ i) \tag{4}$$



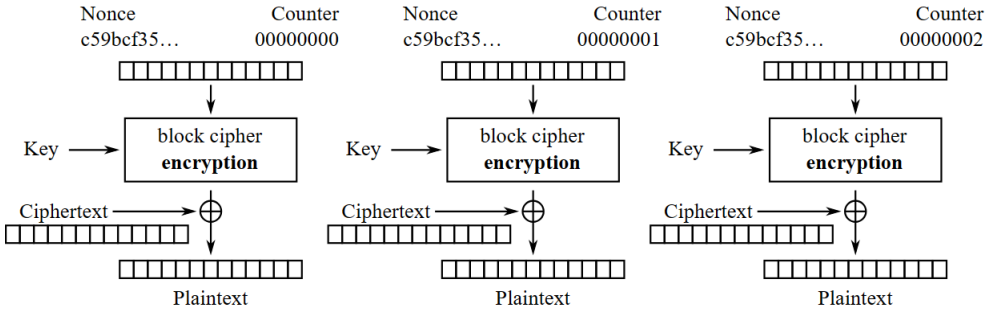


Figure 7. *Encryption and decryption by CTR mode.*

GCM (Galois Counter Mode) is a mode of block cipher in symmetric cryptography. It is an encryption algorithm authenticated designed to ensure the integrity and authenticity of data as well as the confidentiality and is relatively common due to its efficiency and performance. Its functioning and particularities are:

1. GCM encryption is a royalty-free standard. Therefore, it can be used freely, without the need for a patent.
2. AES GCM encryption supports different key sizes: 128, 192 or 256 bits. In depending on these key sizes, the number of operations needed per sequence for the block encryption differs:

o 10 for 128 bits,
o 12 for 192 bits,
o and 14 for 256 bits.

The two functions that make up GCM are called authenticated encryption and authenticated decryption. Their authenticated encryption function encrypts confidential data and calculates an authentication tag to both on the confidential data and on any additional non-confidential data. The function of authenticated decryption decrypts confidential data, subject to tag verification.

Given the selection of a block cipher and a trusted key, there are three input strings for the authenticated encryption function:

1. A clear text noted P,
2. Additional Authenticated Data (AAD), denoted A;
3. And an initialization vector, denoted IV.

Plaintext and AAD are the two categories of data that GCM protects. GCM protects plaintext and authenticity; GCM also protects plaintext privacy, while the AAD is left in the clear. For example, in a network protocol, the AAD may include addresses, ports, sequence numbers, protocol version numbers, and other fields which indicate how the plaintext should be treated.

The IV is essentially a nonce, i.e. a unique value in the specified context, which determines an invocation of the authenticated encryption function on the input data at protect.

The bit lengths of the authenticated cipher function input strings must satisfy the following requirements:

1. len (P) ≤ 239 -256;
2. len (A) ≤ 264 -1;
3. 1 ≤ len (IV) ≤ 264 -1.

Although GCM is set to bit strings, the plaintext, AAD, and IV bit lengths must all be multiples of 8, so these values are byte strings.

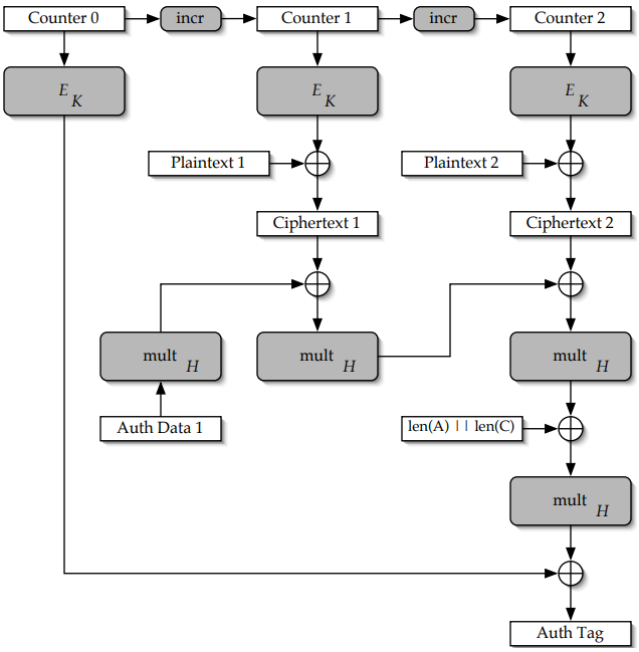


Figure 8. *Authenticated encryption operation.*

Incr: the increment function
Ek: block cipher using the key K
mult H: the multiplication in GF by the hash key H. In GCM, all multiplications take place in the binary 128-bit Galois field GF(2^{128}) with the irreducible polynomial $P(x) = x^{128} + x^{7} + x^{2} + x + 1$.

Given the selection of a trusted block cipher, key, and associated tag length, Inputs to the Authenticated Decryption function are values for IV, A, C, and T. The output is one of the following:

1. the plaintext P which corresponds to the ciphertext C, or an error code.
2. the result P indicates that T is the correct authentication tag for IV, A, and C; otherwise, the output is an error code.

The following two bits strings comprise the output data of

the authenticated encryption function:

1. A ciphertext, denoted C, whose length in bits is the same as that of the plaintext.
2. An authentication tag, or tag, denoted T.

The bit length of the beacon, denoted T, is a security parameter. In Generally, T can be one of five values: 128, 120, 112, 104, or 96 bits. For some applications, T can be 64 or 32 bits.

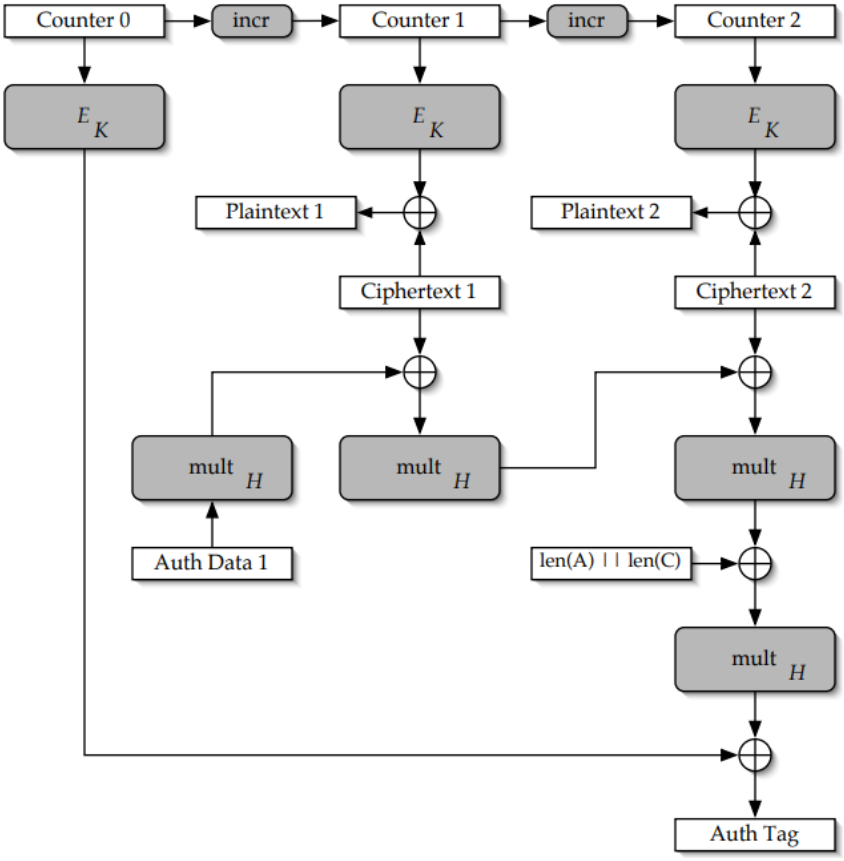


Figure 9. *Authenticated decryption operation.*

2.3. Choice of Authentication Algorithm

The National Security Agency (NSA) announced in 2003 that AES was capable of protecting classified information up to SECRET level, and even TOP SECRET if 256-bits keys are used. So, the solution to have 256-bits keys is to use hash functions, which is the only recourse to meet such a level of security. Because hashing is a process which, at from input data, produces a fixed-length string, the hash. The choice of our password authentication algorithm therefore leans on the two families of hash functions SHA and BLAKE. These two hash functions are:

1. SHA-256: this is the most widely used algorithm due to its balance between security and the computational cost of generation, since it is a very efficient algorithm for the high resistance to collisions it has another peculiarity of this algorithm is that the length of the resulting hash is always the same, regardless of the duration of the content for generating the hash: either a letter or all the words in a book, the result is always a string of 64 letters of letters and numbers (256 bits or 32 coded bytes).
2. BLAKE2s: it works on 32-bit words, but is not susceptible to first-length extension attacks like SHA-256. Thus, BLAKE2s occupies a niche for platforms with limited resources, and the length of the summary produced by this algorithm is between 1 and 32 bytes. It also benefits from a high safety margin, since each of the ten rounds of BLAKE2 is equivalent to two rounds of ChaCha.

For better understanding, the *Table 1* shows the possible combination depending on the key size of AES.

Table 1. *Possible combination depending on the AES key size.*

key size	Possible combination
1 bit	2
2 bits	4
4 bits	16
8 bits	256

key size	Possible combination
16 bits	65536
32 bits	4.2×10^{9}
56 bits (DES)	7.2×10^{16}
64 bits	1.8×10^{19}
128 bits (AES)	3.4×10^{38}
192 bits (AES)	6.2×10^{57}
256 bits (AES)	1.1×10^{77}

So, AES-256 which has a key length of 256 bits, supports the largest bit size and is virtually unbreakable by brute force based on today's computing standards, making it, to date, the strongest encryption standard. Even using a host of super-powered computers, cracking a 256-bit AES key would take longer than the assumed age of the universe [18-26].

2.4. Choice of Programming Language

The Python language which is a programming language created by Dutch developer Guido Van Rossum, will be chosen for creating this application which permits encryption, and decryption of images [27]. It is one of the most used languages by developers and constantly evolving. Its main features can be found below:

1. Python is an interpreted programming language: applications built with this language do not need to be compiled. Their source codes are transformed into machine language at the time of their execution.
2. Python is free and open-source: it can be downloaded for free from its official website. Any user can also freely modify the source code of Python according to his will.
3. Python is portable: any application implemented with this language works on most computer platforms such as Windows, Mac OS, GNU/Linux. Admittedly, this requires that the platforms be equipped with an interpreter intended for python.
4. Python is an object-oriented language: it considers everything as objects, even integers.
5. Python is dynamically typed: Python automatically manages variable types.

2.5. Use of the UML Method for Design

To program an application, it is not advisable to launch headlong into writing the code: first organizing the ideas, document them, then organizing the realization by defining the modules and stages of the realization. It is this process prior to writing that is called modelling; its product is a model. As a modeling language, the UML is used for the conception throughout this project. The modeling is limited by the use case diagram and the class diagram to describe the operation of the system [28-30].

2.5.1. Use Case Diagram

In this section, the main user use cases will be detailed. *Figure 10* shows the user's "crypto system" use case.

1. Precondition: the user must perform each task (open a file, enter the password, select a hash function, select an AES mode, encrypt or decrypt) for the crypto system to work.

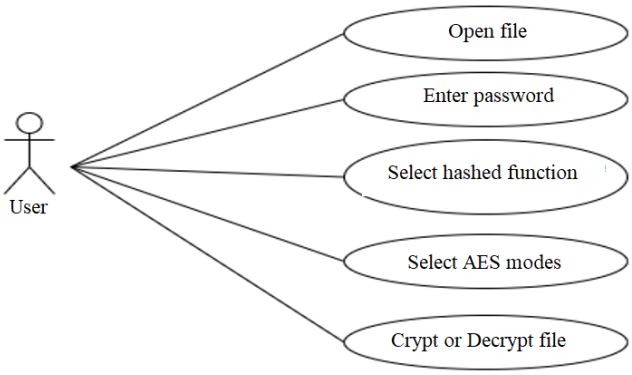


Figure 10. *Crypto system uses case diagram.*

2. Scenarios: *Table 2* below encompasses the crypto system scenarios.

Table 2. *"Crypto system" use case.*

Uses case	crypto system
Use case 1	User must insert file to encrypt or decrypt.
Use case 2	The user must enter the password for authentication.
Use case 3	The user must choose a hash function (SHA-256, BLAKE2s).
Use case 4	The user must choose the AES mode (GCM, CFB, OFB, CBC, ECB, CTR).
Use case 5	The user must choose between encrypting or decrypting.

2.5.2. Class Diagram

The class diagram of the crypto-system is represented in *Figure 11*.

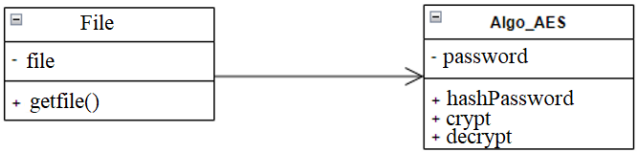


Figure 11. *Crypto system uses case diagram.*

The class "File" herits the class "Algo_AES". The methods of the class "Algo_AES" are:

1. "hashPassword" for hashing the password
2. "crypt" for encrypting
3. "decrypt" for decrypting

The attribute of this class is "Password"

The method of the class 'File" is "getfile" for getting the file to encrypt.

The attribute of this class is "file" which contains the logical value of the file on memory.

3. Results

The crypto system is an application that encrypts and decrypts image files [27]. *Figure 12* shows the user interface of the crypto system after clicking the start button on the first window of the application which is in French language.

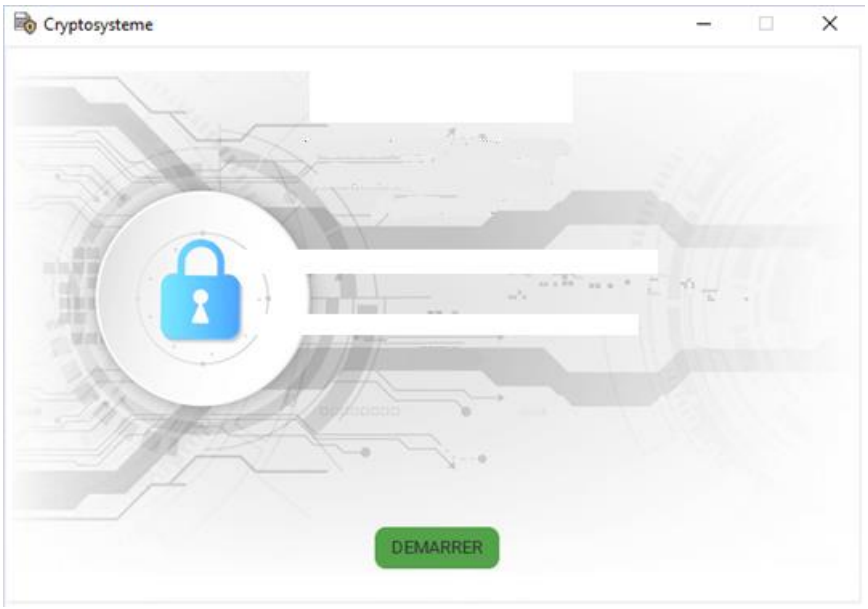


Figure 12. *First window.*

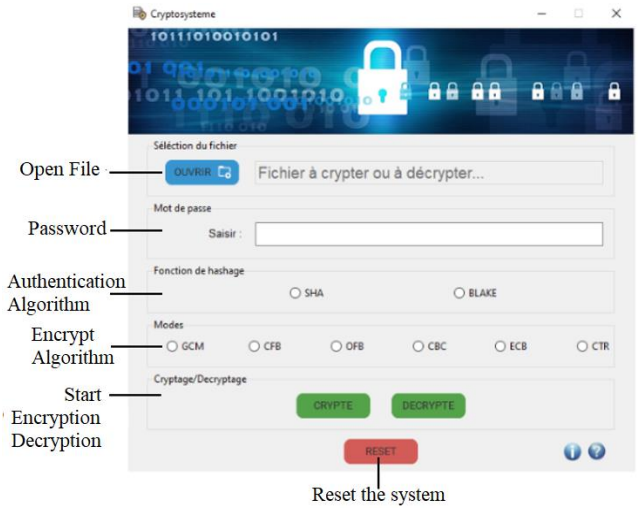


Figure 13. *Crypto system uses case diagram.*

The *Figure 14* represents an example of an unencrypted image.

Figure 14. *Crypto system uses case diagram.*

The encryption mechanism works as follows:

1. In the " File selection" area, click on the "OPEN" button to search for the image to be encrypted. When it has been found, the path of the file to encrypt is displayed in the “File to encrypt or decrypt…” area. The destination of the encrypted file is the same as the unencrypted file.

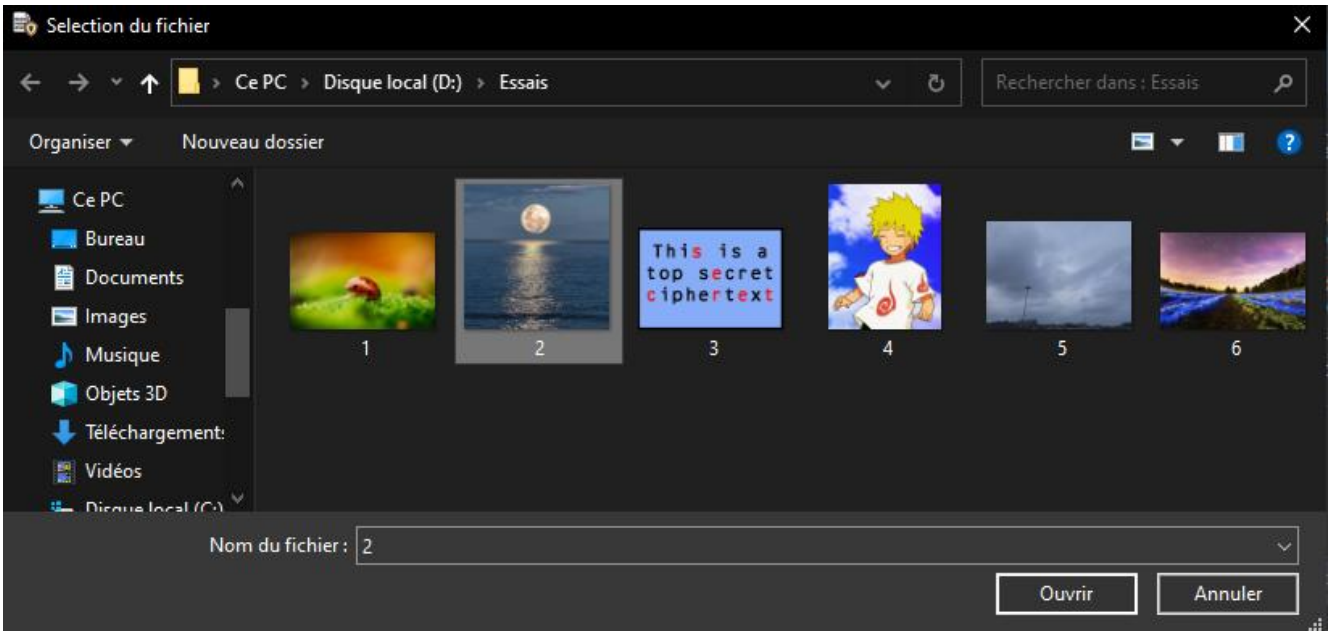


Figure 15. *Open dialog box.*

2. In the "Password" area, you must enter the password for the generation of the key secret of the AES algorithm.
3. In the "Hash function" area, you have the choice between SHA and BLAKE.
4. In the “Modes” area, choose the encryption algorithm mode to use for the data encryption. In the drop-down menu, you have the choice between GCM, CFB, OFB, CBC, ECB and CTR.

After performing these steps, clicking on the button “CRYPT” starts encryption.

The encrypted image therefore looks like the following *Figure 16*:

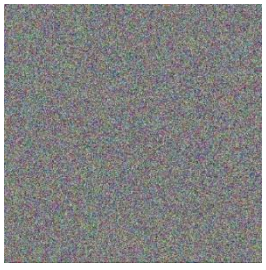


Figure 16. *Example of an encrypted image with system crypto.*

Note: To decrypt the encrypted image, one follows the same processes during encryption. That is, the password, hash function, AES algorithm is the same as encrypting.

4. Discussions

To evaluate the security of this crypto system, the statistical analysis which includes respectively the analysis by histograms and by correlations will be proceeded. Thus, the sensitivity test. So, by using a “bmp” format image of 640 ×426 pixels, analyze the performance of this crypto system will be much evaluated [5-8].

4.1. Statistical Analysis Measurement

The histogram of the image reflects the information about the distribution of the values of its pixels. A spy can make a statistical attack by analyzing the histograms of an encrypted image and this by using attack algorithms to obtain some useful information from the original image. This leads us to say that a well-encrypted image must present a histogram with a uniform distribution.

Table 3. *Histogram comparison of the original image and the encrypted image in AES-SHA.*

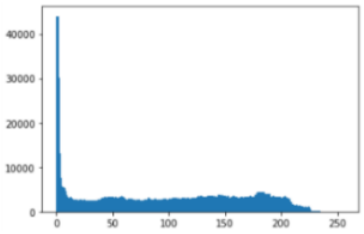
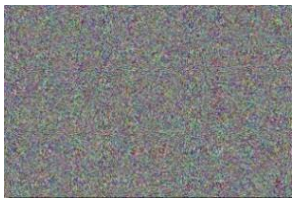
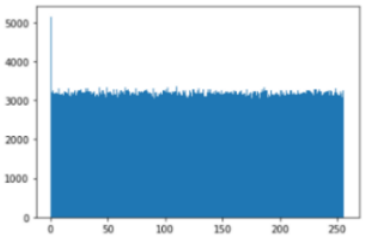
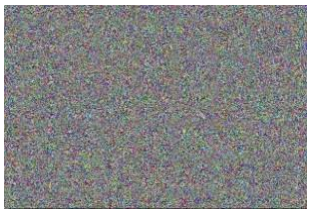
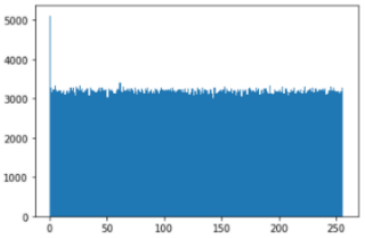
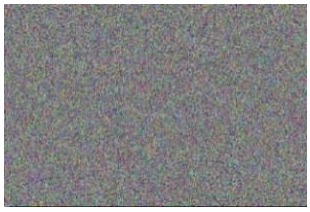
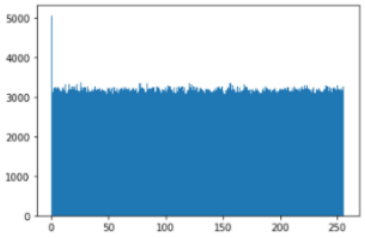
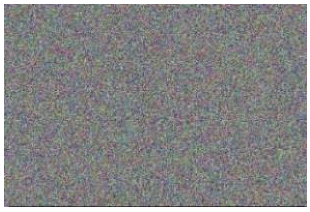
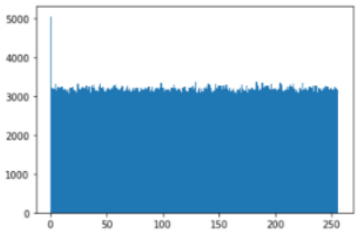
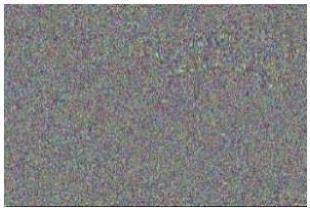
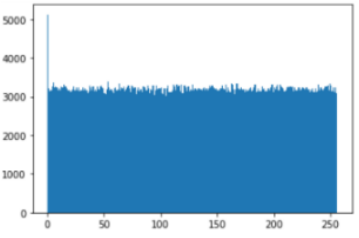
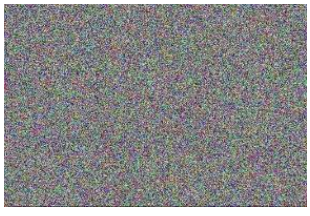
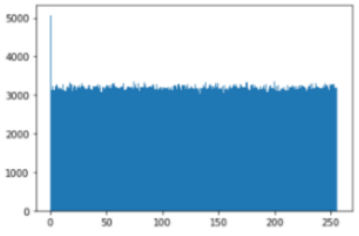
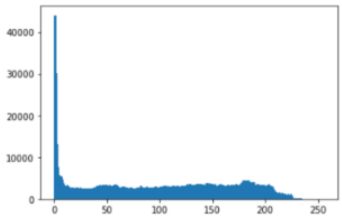
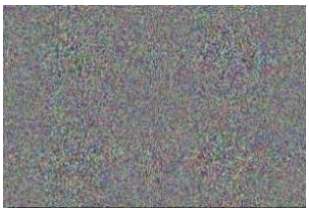
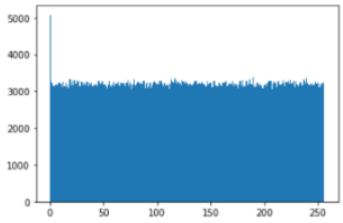
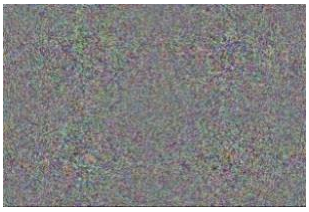
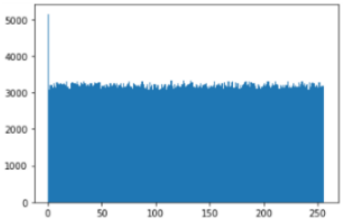
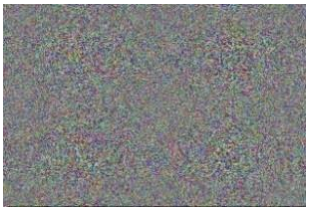
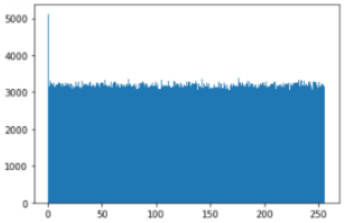
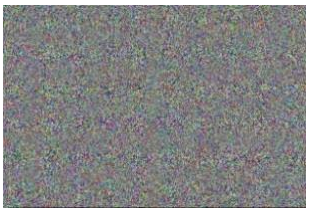
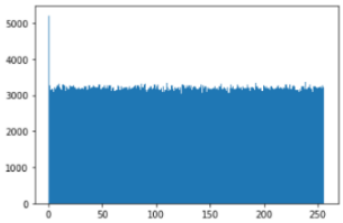
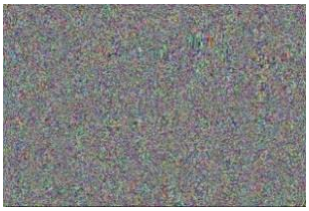
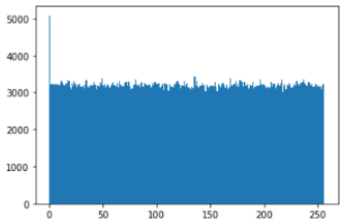
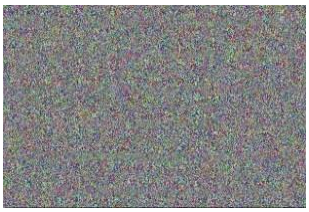
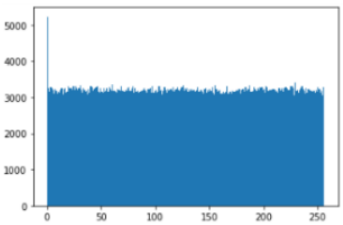
	Pictures	**Histograms**
original		
GCM		
CFB		
OFB		
CBC		
ECB		
CTR		

Table 4. *Histogram comparison of the original image and the encrypted image in AES-BLAKE.*

	Pictures	**Histograms**
original		
GCM		
CFB		
OFB		
CBC		
ECB		
CTR		

The curves in *Table 3* and *Table 4* respectively represent the original and encrypted histograms of the image in AES-SHA and AES-BLAKE. The ordinate axis indicates the number of occurrences of each pixel on the abscissa axis. Thus, the more the histogram of the encrypted image is smooth and uniform distributed as possible, the less a spy will be able to extract information from the original image and therefore the security of the crypto system is higher.

Another technique to assess the security quality of crypto systems is the visualization of the distribution of the horizontal and vertical correlation of the adjacent pixels of the original image and the encrypted images. By calculating, the correlation coefficient for a sequence of adjacent pixels by the equation (5).

$$r_{xy} = \frac{\operatorname{cov}(x,y)}{\sqrt{D(x)} \times \sqrt{D(y)}} \tag{5}$$

Where x, y are two vectors formed respectively by the values of the pixels of the chosen sequence of the image and the values of their adjacent pixels. The terms cov (x, y), E(x), D(x) are calculated by the following equation (6), (7) and (8).

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{6}$$

$$\operatorname{cov}(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)) \tag{7}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2 \tag{8}$$

Where N is the number of adjacent pixels selected in the image to calculate the correlation coefficient. And are the elements of x and y respectively.

Table 5. *Comparison between the correlation of adjacent pixels of the original and encrypted image in AES-SHA.*

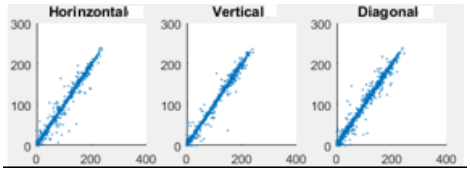
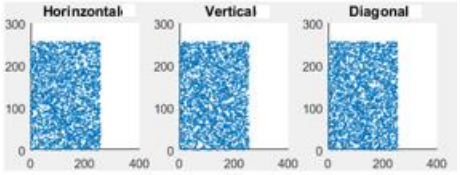
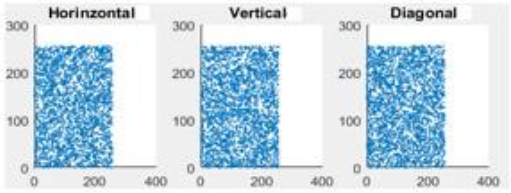
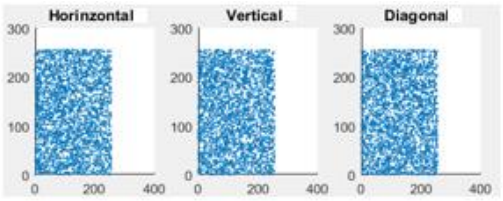
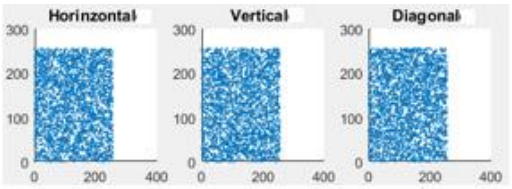
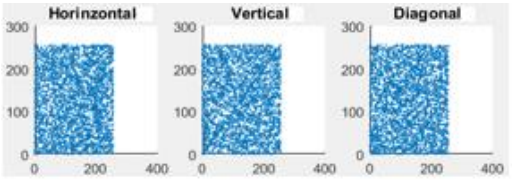
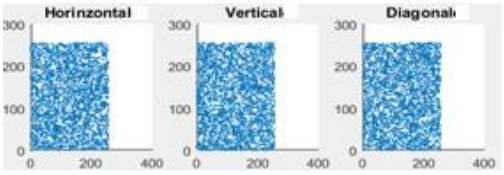
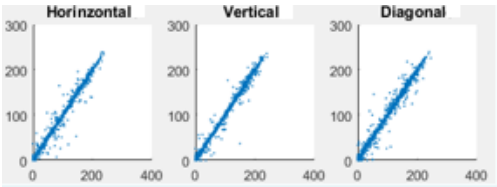
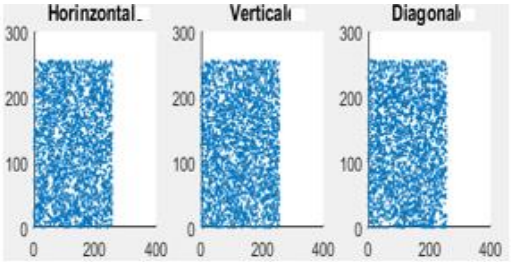
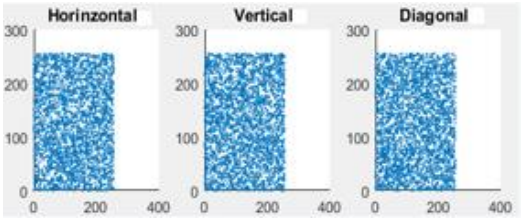
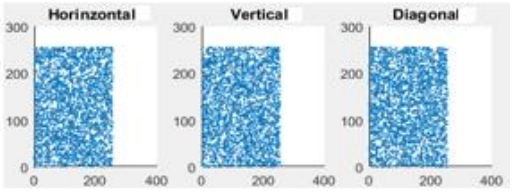
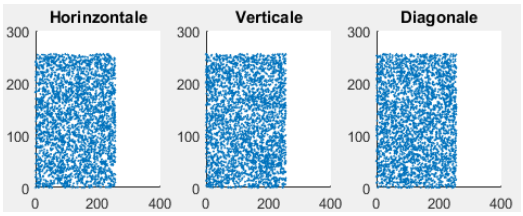
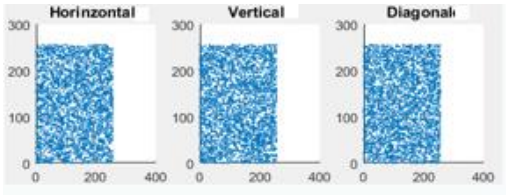
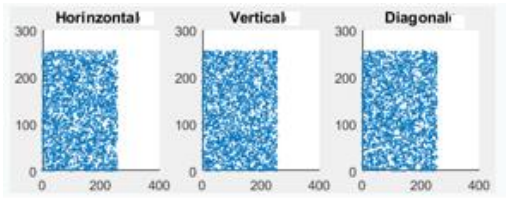
	Correlation (horizontal, vertical, diagonal)
Original	
GCM	

Table 6. *Comparison between the correlation of adjacent pixels of the original and encrypted image in AES-BLAKE.*

	Correlation (horizontal, vertical, diagonal)
CFB	
OFB	
CBC	
ECB	
CTR	
original	
GCM	

Correlation (horizontal, vertical, diagonal)	
CFB	
OFB	
CBC	
ECB	
CTR	

The curves in *Table 5* and *Table 6* represent the comparison of neighboring pixel correlation curves between the original images and the encrypted images, respectively. The points of the correlation figure for an encrypted image should be well scattered and can be seen well in the figures. So, these results show that the statistical attack of this crypto system is almost difficult.

4.2. Sensitivity Test

For the sake of secret key recovery, an attacker could attempt to distinguish any noticeable information between the normal image and its cipher version, by observing the influence of a one-pixel change on the overall system output of encryption. A well-designed crypto-system is one in which a minor modification of its simple image results in a major transformation of its encrypted image, and therefore such attacks are nullified. To carry out the experiment, the following procedure must be adopted:

1. The clear image P1 is encrypted to have an encryption image C1.
2. The simple image P2 here is obtained by applying a minor change to a randomly selected pixel, the altered ordinary image P2 is encrypted using the same secret key to produce the corresponding encryption image C2.

Influence is measured quantitatively using two commonly used metrics:

Number of Pixel Change Rates (NPCR): it calculates the number of pixel differences between two images ciphered below C1 and C2, using the following equation (9).

$$NPCR = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} D(i,j)}{H \times L} \times 100 \tag{9}$$

Where N and M denote the width and height of the image respectively, and D (i; j) is defined as follows:

$$D(i,j) = \begin{cases} 0 \; si \; C_1(i,j) = C_2(i,j) \\ 1 \; si \; C_1(i,j) \neq C_2(i,j) \end{cases} \tag{10}$$

Table 7. *NPCR for Encrypted Images.*

	SHA	BLAKE
GCM	99.61004586260732	99.6089480874317
CFB	99.61553473848556	99.60467896174863
OFB	99.60431303669009	99.6135831381733
CBC	99.6173643637783	99.60260538641687
ECB	99.62163348946136	99.61529078844653
CTR	99.6044350117096	99.61346116315377

NPCR's benchmark for encryption is 99.6094%. *Table 7* shows that the crypto-system pass this test because values are closer to the reference. This means that the crypto-systems is resistant to differential attack.

Unified Average Change Intensity (UACI): it calculates the average intensity of the differences between the original image C1 and the encrypted image C2, using the following equation (11).

$$UACI = \sum_{i=1}^{N}\sum_{j=1}^{M} \frac{|C_1(i,j) - C_2(i,j)|}{H \times L \times 255} \times 100 \tag{11}$$

Table 8. *UACI for Encrypted Images.*

	SHA	BLAKE
GCM	30.415273281456106	30.485258242653657
CFB	30.46358112918564	30.487924473080636

	SHA	BLAKE
OFB	30.44087368555215	30.486615752871348
CBC	30.432582254226148	30.454480357729125
ECB	30.3907606075344	30.419510837134716
CTR	30.418087316906693	30.41796390688642

Table 8 shows the UACI values of the encrypted image. The reference value for this test is 33.4635%. The results are closer to this reference.

4.3. Performance Analysis Measurement

The decrypted image I' is always compared to the original image I or in clear to determine its similarity ratio. This criterion is the most used. It is based on the measurement of the Mean Squared Error (MSE) calculated between the original pixels and the deciphered ones by the equation (12).

$$MSE = \frac{1}{M*N}\sum_{m=1}^{M}\sum_{n=1}^{N}(I(m,n) - I'(m,n))^2 \quad (12)$$

Where (M × N) which denotes the size of the original and decrypted image, and I, I' are respectively the amplitudes of the pixels on the original and decrypted images. It is likely that the eye takes much more account of errors at large amplitudes, which favors quadratic measurement. The lower the value of MSE, the lower the error. If the MSE is zero, it means that the original image and the decrypted image are identical and the Peak Signal to Noise Ratio (PSNR) value will be infinity.

Table 9. *MSE value for the six modes combined with the two hash functions.*

Hash	**SHA-256**						**BLAKE2S**					
Modes	**GCM**	**CFB**	**OFB**	**CBC**	**ECB**	**CTR**	**GCM**	**CFB**	**OFB**	**CBC**	**ECB**	**CTR**
MSE	0						0					

According to *Table 9*, the value of the MSE of the image is equal to zero in all the modes, which implies that the decrypted and original images are identical.

For this evaluation criterion, instead of measuring the distortion, the value PSNR measures the fidelity of the decrypted image compared to the original or in clear, since it is proportional to the quality. The equation 12 express MSE; its definition and usage come from the field of signal processing:

$$PSNR = 10 * \log_{10}\left(\frac{255^2}{MSE}\right) \quad (13)$$

Table 10. *PSNR value for the six modes combined with the two hash functions.*

Hash	**SHA-256**						**BLAKE2S**					
Modes	**GCM**	**CFB**	**OFB**	**CBC**	**ECB**	**CTR**	**GCM**	**CFB**	**OFB**	**CBC**	**ECB**	**CTR**
PSNR	Infinity						Infinity					

An infinite PSNR value corresponds to an ungraded image. And this value decreases according to the degradation. The PSNR thus relates the MSE to the maximum energy of the image. The higher the PSNR, the more similar the decrypted image is to the original. Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) are originally used to compare image compression quality for assessing the quality of decrypted images against the original.

There is an inverse relationship between PSNR and MSE. Thus, a higher PSNR value indicates better image quality. According to the results indicated in *Table 10*, that the PSNR value of the image for the six modes combined with the two hash functions combined is equal to infinity, which means that the decrypted and original images are identical.

The entropy H(m) can be used as a performance indicator of a crypto-system. It indicates the quantity of information contained or delivered by an image. It is linked to the probability distribution of appearance of each pixel and is calculated by equation (14). The value of entropy of image for an ideal crypto-system is 8.

$$H(m) = -\sum_{i=0}^{L-1} p(m_i)\, log(p(m_i)) \quad (14)$$

With p(m_i) the probability of appearance of the symbol m_i

Table 11. *Entropy of encrypted images.*

	SHA	BLAKE
GCM	7.63747834	7.63830899
CFB	7.63881666	7.63738745
OFB	7.63907246	7.64134985
CBC	7.63743805	7.64047588
ECB	7.63787775	7.63747895
CTR	7.63872385	7.6368872

The reference value of entropy is 8. *Table 11* shows that entropy results are not far from this value. This shows the effectiveness of the methods.

4.4. AES GCM Authentication Test

In AES-GCM, there is an Authentication Tag, also called Message Authentication Code or Integrity Check Value. This tag is intended to prove that it is indeed the encrypted image.

The *Table 12* shows the verification of the authentication of images encrypted in AES-GCM.

Table 12. *Verification of authentication of images encrypted in AES-GCM.*

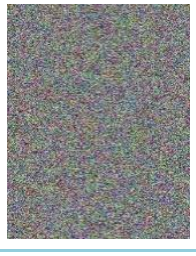
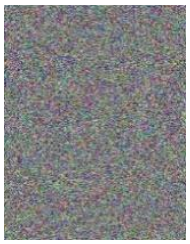
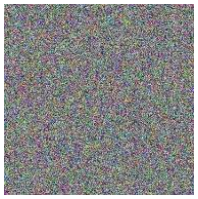
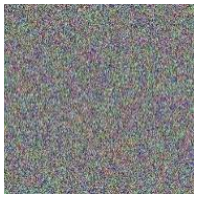
Original image	Encrypted	Tag in hexadecimal
	 (Non-Modified)	390f6927 bd6780d3 09ccbb19 a15265a1
	 (Modified)	Tag Incorrect
		b289b331 343ef8b6 70940578 a3a26b3c
	(Non-modified)  (Modified)	Tag Incorrect
	 (Non-modified)	9552de48 642ecc75 bc64e29e c526ae8a
	 Modified	Tag Incorrect

Table 12 shows that the crypto-system verifies the authenticity of the image encrypted in AES-GCM. When an attacker modifies even one bit of the encrypted data, the application could detect it by informing tag incorrect and not deciphering it. The oracle attack also could be avoiding this. Oracle attack consists of sending multiple modified encrypted data and if the application gives the decrypting modified data, it's possible for the attacker having a key.

5. Conclusions

This article made an encryption application to encrypt different images using graphical interfaces. Developed by python, The AES algorithm will be chosen firstly with classic mode CFB, OFB, CBC, ECB, CTR and secondly with integrity mode protected by using GCM modes with functions of hashes SHA-256 and BLAKE2s to better secure the key. For the evaluation of the results, the histogram, the correlation, NPCR, UACI, MSE, PSNR and the entropy will be studied. The results will be satisfactory: the encrypted image and original image have a strictly low correlation; the crypto-system will be robustness with differential attack; the decrypted message and original are the same; the crypto-system detect the modification of encrypted image and doesn't decrypt it. In the future, the project could be integrated on android phone as apk file for mobility and more

uses.

Abbreviations

AAD: Additional Authenticated Data
AES: Advanced Encryption Standard
CBC: Cipher Block Chaining
CFB: Cipher FeedBack
CTR: CounTeR
DES: Data encryption Standard
ECB: Electronic Codebook Block
GCM: Galois Counter Mode
IDEA: International Data Encryption Algorithm
MSE: Mean Squared Error
NIST: National Institute of Standards and Technology
NPCR: Number of Pixel Change Rates
NSA: National Security Agency
OFB: Output FeedBack
PSNR: Peak Signal to Noise Ratio
RSA: Rivest Shamir Adleman
SHA: Secure Hash Algorithm
SHA-256: Secure Hash Algorithm 256 Bits
UACI: Unified Average Change Intensity

Author Contributions

Rakotondramanana Radiarisainana Sitraka: Conceptualization, Resources, Data curation, Software, Formal Analysis, Supervision, Funding acquisition, Validation, Investigation, Visualization, Methodology, Writing – original draft, Project administration, Writing – review & editing.

Ramafiarisona Hajasoa Malalatiana: Conceptualization, Resources, Formal Analysis, Supervision, Funding acquisition, Validation, Investigation, Visualization.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] Flevina Jonese D'souza and Dakshata Panchal, (2017). *Advanced encryption standard (AES) security enhancement using hybrid approach,* in IEEE International Conference on Computing, Communication and Automation (ICCCA), 21 December 2017, Greater Noida, India, https://doi.org/10.1109/CCAA.2017.8229881

[2] Megha Chakole and S. S. Dorle (2023). *Design of Advanced Encryption Standard Algorithm,* in IEEE International Conference On Computing, Communication, Control And Automation (ICCUBEA), 16 January 2023, Pune, India, https://doi.org/10.1109/ICCUBEA54992.2022.10011065

[3] Simon Heron (2009). *Advanced Encryption Standard (AES), in Network Security,* Volume 2009, Issue 12, December 2009, pp. 8-12, https://doi.org/10.1016/S1353-4858(10)70006-4

[4] Yan Qing. Zhong, Jian Ming. Wang, Z. F. Zhao, D. Y. Yu and L. Li (2007). *A Low-cost and High Efficiency Architecture of AES Crypto-engine,* in IEEE International Conference on Communications and Networking in China, 07 March 2008, Shanghai, China, https://doi.org/10.1109/CHINACOM.2007.4469389

[5] Salim M. Wadia, b and Nasharuddin Zainala, (2013). Rapid Encryption Method Based on AES Algorithm for Grey Scale HD Image Encryption, in Procedia Technology, Iraq, 2013, pp. 51–56, https://doi.org/10.1016/j.protcy.2013.12.161

[6] Venkata Krishna Pavan Kalubandi, Hemanth Vaddi, Vishnu Ramineni and Agilandeeswari Loganathan, (2017). *A novel image encryption algorithm using AES and visual cryptography,* in IEEE International Conference on Next Generation Computing Technologies (NGCT), 16 March 2017, Dehradun, India, https://doi.org/10.1109/NGCT.2016.7877521

[7] Amandeep Singh, Praveen Agarwal and Mehar Chand, (2019), *Image Encryption and Analysis using Dynamic AES,* in IEEE International Conference on Optimization and Applications (ICOA), 03 june 2019, Kenitra, Morocco, https://doi.org/10.1109/ICOA.2019.8727711

[8] Ijaz Khalid, Tariq Shah, Sayed M. Eldin, Dawood Shah, Muhammad Asif and Imran Saddique (2022). *An Integrated Image Encryption Scheme Based on Elliptic Curve,* in IEEE Access, vol. 11, 16 December 2022, pp. 5483–5501, https://doi.org/10.1109/ACCESS.2022.3230096

[9] Miguel León Chávez, Francisco Rodríguez Henríquez and Emmanuel López Trej (2007). *AES-CCM Implementations for the IEEE 802.15.4 Devices,* in IFAC Proceedings, vol. 40, issue 22, 2007, pp. 223-229, https://doi.org/10.3182/20071107-3-FR-3907.00030

[10] Arshad Aziz and Nassar Ikram (2007). *An FPGA-based AES-CCM Crypto core for IEEE 802.11i architecture,* in International Journal of Network Security, vol. 5, issue 2, September 2007, pp. 224–232, http://ijns.jalaxy.com.tw/contents/ijns-v5-n2/ijns-2007-v5-n2-p224-232.pdf

[11] Rei Ueno, Sumio Morioka, Naofumi Homma and Takafumi Aoki, (2016). *A High Throughput/Gate AES Hardware Architecture by Compressing Encryption and Decryption Datapaths - Toward Efficient CBC-Mode Implementation,* https://eprint.iacr.org/2016/595.pdf

[12] Suhyeon Jeon and Jihwan P. Choi (2019). *CFB-AES-TURBO: Joint Encryption and Channel Coding for Secure Satellite Data Transmission,* in IEEE International Conference on Communications (ICC), 15 July 2019, Shanghai, China, https://doi.org/10.1109/ICC.2019.8762018

[13] Suhyeon Jeon, Jeongho Kwak, Jihwan and P. Choi (2022), *Cross-Layer Encryption of CFB-AES-TURBO for Advanced Satellite Data Transmission Security, in IEEE Transactions on Aerospace and Electronic Systems,* vol. 58, issue. 3, June 2022, pp. 2192 – 2205, https://doi.org/10.1109/TAES.2021.3134988

[14] Mahmoud Alfadel, El-Sayed M. El-Alfy and Khaleque Md Aashiq Kamal (2017). *Evaluating time and throughput at different modes of operation in AES algorithm*, in International Conference on Information Technology (ICIT), 23 octobre 2017, Amman, Jordan, https://doi.org/10.1109/ICITECH.2017.8079948

[15] Mohamed Boussif (2022). On The Security of Advanced Encryption Standard (AES), in IEEE International Conference on Engineering, Applied Sciences, and Technology (ICEAST), 20 July 2022, Chiang Mai, Thailand, https://doi.org/10.1109/ICEAST55249.2022.9826324

[16] Razvi Doomun, Jayramsingh Doma and Sundeep Tengur (2008), *AES-CBC software execution optimization*, in IEEE International Symposium on Information Technology, 26 September 2008, Kuala Lumpur, Malaysia, https://doi.org/10.1109/ITSIM.2008.4631586

[17] Shashank Srivastava, Avinash Kumar Singh and G. C. Nandi (2012). Inter Cipher Block Diffusion: *A Novel Transformation for Proposed Parallel AES*, in Procedia Technology vol. 6, 2012, pp 872-879, https://doi.org/10.1016/j.protcy.2012.10.106

[18] Tomoya Hikida, Yasuyuki Nogami, Md Arshad Ali and Yuta Kodera (2022), *Comparison of conversion matrices for a compact AES-CTR defined over an isomorphic field*, in IEEE International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), 03 October 2022, https://doi.org/10.1109/ITC-CSCC55581.2022.9895089

[19] Tom St Denis and Simon Johnson (2007). *Advanced Encryption Standard*, in Cryptography for Developers, 2007.

[20] Youngbeom Kim and Seog Chung Seo (2021), *Efficient Implementation of AES and CTR_DRBG on 8-Bit AVR-Based Sensor Nodes*, in IEEE Access vol. 9, 16 February 2021, pp. 30496 – 30510, https://doi.org/10.1109/ACCESS.2021.3059623

[21] Ignacio Algredo-Badillo, Claudia Feregrino-Uribe, René Cumplido and Miguel Morales-Sandoval (2010), *Efficient hardware architecture for the AES-CCM protocol of the IEEE 802.11i standard*, in Computers & Electrical Engineering vol. 36, issue 3, May 2010, pp. 565-577, https://doi.org/10.1016/j.compeleceng.2009.12.011

[22] Benjamin Buhro, Karl Fritz, Barry Gilbert and Erik Daniel (2016). *A highly parallel AES-GCM core for authenticated encryption of 400 Gb/s network protocols*, in IEEE International Conference on ReConFigurable Computing and FPGAs (ReConFig), 01 February 2016, Riviera Maya, Mexico, https://doi.org/10.1109/ReConFig.2015.7393321

[23] Byung-Yoon Sung; Ki-Bbeum Kim; Kyung-Wook Shin (2018). *An AES-GCM authenticated encryption crypto-core for IoT security*, in IEEE International Conference on Electronics, Information, and Communication (ICEIC), 05 April 2018, Honolulu, HI, USA, https://doi.org/10.23919/ELINFOCOM.2018.8330586

[24] Eduardo Mobilon and Dalton Soares Arantes (2021). *100 Gbit/s AES-GCM Cryptography Engine for Optical Transport Network Systems: Architecture, Design and 40 nm Silicon Prototyping*, in Microelectronics Journal vol. 116, October 2021, https://doi.org/10.1016/j.mejo.2021.105229

[25] Nabihah Ahmad, Lim Mei Wei, and M. Hairol Jabbar (2017), *Advanced Encryption Standard with Galois Counter Mode using Field Programmable Gate Array*, in International Conference on Green and Sustainable Computing (ICoGeS), 2017, https://doi.org/10.1088/1742-6596/1019/1/012008

[26] Jack Zhao, *Performance analysis of cryptographic functions on programmable NICS*, in Dalhousie University, August 2022, Halifax, Nova Scotia, https://dalspace.library.dal.ca/bitstream/handle/10222/81823/JackZhao2022.pdf

[27] Sitraka Rakotondramanana, https://github.com/SitrakaResearchAndPOC/aes_bloc_ciphering_modes

[28] Hatice Koç, Ali Mert Erdoğan, Yousef Barjakly and Serhat Peker, *UML Diagrams in Software Engineering Research: A Systematic Literature Review*, in MDPI International Management Information Systems Conference, 10 March 2021, https://doi.org/10.3390/proceedings2021074013

[29] Mert Ozkaya and Ferhat Erata, *A survey on the practical use of UML for different software architecture viewpoints*, in Elsevier Information and Software Technology Volume 121, May 2020, 106275, https://doi.org/10.1016/j.infsof.2020.106275

[30] Mohammad N. Alanazi, *Basic Rules to Build Correct UML Diagrams*, in IEEE International Conference on New Trends in Information and Service Science, 25 September 2009, Beijing, China, https://doi.org/10.1109/NISS.2009.