Iraqi Journal of Science

ISSN: 0067-2904

Data integrity enhancement for the encryption of color images based on CRC64 technique using multiple look-up tables

Nada Hussein M. Ali, Ruaa A. Abdul-Sattar*
Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq.

Abstract
Communication is one of the vast and rapidly growing fields of engineering, where increasing the efficiency of communication by overcoming the external electromagnetic sources and noise is considered a challenging task. To achieve confidentiality for color image transmission over the noisy communication channels a proposed algorithm is presented for image encryption using AES algorithm. This algorithm combined with error detections using Cyclic Redundancy Check (CRC) to preserve the integrity of the encrypted data. This paper presents an error detection method uses Cyclic Redundancy Check (CRC), the CRC value can be generated by two methods: Serial and Parallel CRC Implementation. The proposed algorithm for the encryption and error detection using parallel CRC64 (Slicing-by-4 algorithm) implementation with multiple look table approach for the encrypted image. The goal of the proposed algorithm optimizes the size of the redundant bits needed to attach to the original data for the purpose of error detection; this reduction is considered necessary to meet the restriction for some computer architectures. Furthermore, it is suitable for implementing in software rather than in hardware. The proposed algorithm uses different tested images by added different noise ratios (1% and 5%) of total images size to study the noise effect on the encrypted images. The noise added on single and multi bits position and study the effect on the output results. The obtained results shown that the small size of the image the large CRC64 affected by noise while the large size of image yields a stable or fixed number of affected CRC64.

Keywords: AES, Cyclic Redundancy Check, Sarwate, Look up table, Encryption

تحسين سلامة البيانات لتشفير الصور الملونة على أساس تقنية CRC64 باستخدام جداول متعددة

ندى حسين محمد علي، رؤى علاء الدين عبد الستار*
قسم علوم الحاسبات، كلية العلوم، جامعة بغداد، بغداد، العراق.

الخلاصة
ان الاتصالات تعتبر من المجالات الواسعة وسريعة النمو في علم الهندسة،وزيادة كفاءة الاتصالات تتم من خلال التغلب على المصادر الكهرومغناطيسية الخارجية والضوضاء والتي تعتبر من المهام صعبة .وللحفاظ على سرية المعلومات لنقل الصور الملونة على قنوات الاتصال ذات الضوضاء تم اقتراح خوارزمية لتشفير الصور باستخدام خوارزمية AES. وبالاشتراك مع الخوارزمية المذكورة أعلاه تم إضافة تقنية الكشف عن الخطأ في البيانات باستخدام طريقة CRC للحفاظ على سلامة البيانات المشفرة .في هذا البحث تم استخدام طريقة الكشف عن الأخطاء بطريقةCRC، حيث ان قيمة CRC يمكن أن يتم توليدها من خلال طريقتين:التنفيذ المتسلسل والمتوازي لطريقة CRC . في حالة التنفيذ المتسلسل فان تقنية CRC تستخدم

*Email: ruaaalqassab@yahoo.com

طريقة Linear Feedback Shift Register وتطبق one bit a time ،اما طريقة التوازي فانه تتم معالجة عدة بايتات في المرة الواحدة وخلال دورة واحدة من دورات CPU . ان تقنيةCRC والتي تنفذ بالطريقة المتوازية حيث انها تستند إلى خوارزميتين؛ Sarwate and Slicing by N. في خوارزمية Sarwate، يتم استخدام جدول بحث واحد (LUT) لتخزين جميع الاحتمالات الممكنة لقيم الإدخال .على سبيل المثال، لنفترض إذا كان لدينا قيمة إدخال 128 بت (ان القيم المدخلة لخوارزمية=AES128 بت) وفي هذه الحالة فان القيم المحتملة سوف تكون 2^{128} احتمال ويتم تخزين كل هذه القيم في جدول واحد، وبالتالي فان حجم هذا الجدول يتطلب استخدام ذاكرة كبيرة. اما في حالة خوارزمية (Slicing by N 32، 8، 16، N) 4 =بدلا من استخدام جدول بحث واحد، يتم استخدام جداول بحث متعددة، لذلك فان هذه الطريقة تكون أقل استخدام الذاكرة مقارنة مع طريقة Sarwate. ان الخوارزمية المقترحة للتشفير والكشف عن الأخطاء تستخدم طريقة CRC64 المتوازية (Slicing−by−4 algorithm) والتي يتم تنفيذها باستخدام عدة جداول مع الصور المشفرة .إن الهدف من الخوارزمية المقترحة هو تحسين وتقليل حجم bits الزائدة عن الحاجة اللازمة والتي يتم ارفاقها مع البيانات الأصلية لغرض الكشف عن الأخطاء، ويعتبر هذا التخفيض ضروريا لتلبية القيود المفروضة على بعض معماريات الحاسوب. وعلاوة على ذلك، فإنه مناسب اكثر للتنفيذ في software افضل من hardware .

Introduction

Many multimedia an applications, such as video surveillance, satellite communications and web cameras, have been derived from the rapid growth of multimedia technologies and networks. As a result, securing the transfer of multimedia has become a difficult issue. The fact that the quality of service (QoS) must be met to provide high-quality media and low latency even when securing media transfer. To provide a secure area as much as possible, it shouldbe chosen appropriate security algorithm in order to send multimedia in real time due to the unique characteristics of real-time multimedia data such as large data size, high bandwidth and real-time requirements. In order to maintain high-quality service QoS, the security mechanism should provide three viable characteristics: high-speed processing, high compression rate and adequate security level [1]. The implementation of AES for data security provides the benefits of less memory consumption and less time calculation compared to other algorithms [2].

Noise Means, the pixel in the image shows different intensity values instead of the actual pixel values, the noise removal algorithm is the process of removing or reducing noise from the image. This algorithm reduce or eliminate the visibility of noise by smoothing the entire image, leaving areas near the contrast limits [3].To verify any change in the data, the checksum that can be visualized as a unique representative value for a given set of data values is calculated using a particular methodology. After calculating the total checksum interval repeatedly and comparing them with the stored test. If the values are different, it means that the data is changed. This methodology assumes that the algorithm that computes the checksum will not produce the same checksum for two different data values [4].

The data transmitted over a noisy channel is almost altered and corrupted, for this reason; the aim of the error detection methods is to help the receiver to discover the corrupted data and even corrected it. The detection operation is performed by computing a checksum value by the sender, which is a function of the message, and it is appended with the source data as a first stage. The receiver uses the same function to compute the checksum value to determine if the message was correctly received [5].

Related work

There are many studies discussed how to solve the problem for error detection and implementation of Advanced Encryption Standard, some of them are:

In [6]:proposed an improved scheme for AES algorithm to enhanced the error that may occur in each layout of this algorithm. The suggested scheme is symmetric and simple to implement where several error detection methods were used in each transformation of the AES algorithm. The error detection scheme based on $(n+1, n)$ cyclic redundancy check(CRC) over$GF(2^8)$, where n ∈ {4,8,16}. The proposed scheme can be applied in the implementation of AES against differential fault attacks and can be easily implemented in a variety of structures, such as the 8-bit, 32-bit, or 128-bit structures.

In[7]: The most significant contribution that the proposed algorithm optimize the size of the redundant bits needed to attached with the source data of huge amount for the purpose of error detection. This

technique is considered necessary to meet the restriction for some computer architectures. The creation of the arbitrary number by slices (slicing-by-4" and slicing-by-8) in the proposed algorithm are processed in a parallel manner where the slices implemented on different processors to obtain a full time usages of the CPUs in the system.

In [8]: propose an improvement for Sarwate algorithm by grouping the input message into chunks each of eight block length. The enhancement algorithm, which also called Slicing-by-8 algorithm, read an 64 bit block and compute its CRC of 32 bit length in parallel manner which increase the performance of Sarwate algorithm.

In [9]:a fast cyclic redundancy check algorithm using lookup tables instead of linear feedback shift registers was proposed and implemented in hardware architecture. The algorithm can calculate different CRC algorithms of any length of the message in parallel. The architecture is configurable and can support CRC algorithms such as CRC32, CRC24, CRC-CCITT, CRC16, and CRC8. CRC value of 128-bit input data can be generated in one cycle.

In [10]: proposeda new method which use Cyclic Redundancy Check (CRC) for the error detection and the generated CRC error is corrected by using Hybrid Matrix Code (HMC) and they are coded in VirologyHDL and simulated using Xilinx ISE Design Suite 14.2. The proposed method can provide maximum error detection and correction capability withareduction in delay.

Color image encrypted by Advanced Encryption Standard (AES) algorithm

in this paper, the color images is encrypted using the Advanced Encryption Standard (AES) algorithm, this algorithm is symmetric block cipher algorithm [11].It can be processed in different modes (e.g. CBC, CRT, ECB……etc.). Each input block of data consists of 128 bits and it's encrypted many times (or rounds) depending on key length to increase the strength of confusion and diffusion properties. The number of rounds are 10, 12 and 14 for the key length 128,192 and 256 respectively. A single Rijndael round is composed of four layout transformation as follow [12]:

1. Sub Byte: provides nonlinearity substitution for blocks and confusion.
2. Shift row: it is a permutation operation and provides diffusion.
3. Mix Column: linear combination provides inter-Byte diffusion.
4. Add Round Key: each byte in the block cipher is XOR with corresponding bytes in key block to provide confusion.

Error detection using linear block method CRC

CRC (Cyclic Redundancy Check)is most common errordetectingcode computed through binary polynomial division, the polynomialmust be selected to maximize error detection capabilities while minimizing the potential for total collision probabilities. Two parties; the sender and receiver; are converting the binary data into a polynomial representation, this operation is performed by dividing the polynomial by another one called generator polynomial$G(x)$; e.g. CRC32. The division operation is implemented by module-2 between the above polynomials, the resulted value which represents the remainder is considered a CRC and it is attached with the original data as a check sum value and is sent over the network. On the other side, the receiver performs the same operations to compute the CRC with the same generator $G(x)$ and compare it with the incoming data for error detection [5].

The probability of a bad frame getting through without being noticed is $(1/2^r)$and r is the G (x), so CRC-16, CRC-32 and CRC-64 have a probability of error equal to 0.000015, 2.33^{-10}, and 5.42^{-20} respectively as shown in equation 1[13]:

$$x^r . M(X) = G(x). Q(x). R(x) \quad \ldots (1)$$

- Where $G(x)$ is the generator polynomial,
- $r$is the degree of $G(x)$,
- Polynomial$M(X)$ corresponding to some frames with m bits length to compute the checksum for it
- $Q(x). R(x)$ represent the division quoting and remainder respectively

x^r Represent the total numbers in G(X) for M(X) to encode. For example x^r in CRC64 equal 64 and the total number in M(X) to be encoded =128 bits. Table- 1 clarifies the error probability of a bad frame getting through unnoticed in different CRC.

Table 1-Error probability using different sizes of CRC

Bits for r	Error probability
CRC8	$\frac{1}{2^8} = 0.004$
CRC16	$\frac{1}{2^{16}}$ =0.000015
CRC32	$\frac{1}{2^{32}}$ =2.33*10^{-10}
CRC64	$\frac{1}{2^{64}}$ =5.42 *10^{-20}

Mean Square Error (MSE) is one of the most widely used quality degradation metrics, besideMSE does not take advantage of the human visual system (HVS) properties. In other words, MSE does not reveal the way our perception work. Let X and Y two plain images of size $N.M$ respectively to be encrypted or decrypted. The mean square error between the two images is thus defined as [14]:

$$MSE = \frac{1}{N \times M} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} [X(i,j) - Y(i,j)]^2 \quad \text{.....(2)}$$

The more Y is similar to X, the more MSE is small. Obviously, the greatest similarity is achieved when MSE equal to 0.

The Proposed algorithm for detection error in encrypted image

This paper presents an error detection method uses Cyclic Redundancy Check (CRC), the CRC value can be generated by two methods: Serial and Parallel CRC Implementation. The Serial CRC implementation uses the technique of Linear Feedback Shift Register one bit a time while in parallel CRC implementation, multiple bytes are considered during one clock cycle. The parallel CRC implementation is based on two algorithms; Sarwate and Slicing by N algorithms. In Sarwate algorithm, a single look-up table (LUT) is used to store all the possible combinations of input value. For example, suppose if we have 128-bit input value (AES input block is 128 bits), it has 2^{128} possible combinations and all these values are stored in a single look-up table, the LUT become large, thus, it requires more memory usage. In case of Slicing by N algorithm (N=4, 8, 16, 32) instead of using a single look-up table, multiple look-up tables are used, so less memory usage compared with Sarwate method[9].

In dissection slicing by 4 algorithm, 128 bits of incoming encrypted image by AES algorithm are taken and are XORed with the primary CRC value. The initial value of CRC-64 used here is$0xFFFFFFFFFFFFFFFF$.Thismodified bit stream is divided into four slices; each of slice consists of 4 bytes of data. These four bytes of data are divided into four single bytes as shown in Figure- 1, and all possible values are recalculated for each byte before using the LFSR existing interpreter method and stored in the corresponding LUT[10]. Algorithm 1 represents the of AES encryption process while Algorithm 2 demonstrates the process of computing the CRC 64 slice 4 for the encrypted images (or even a plaintext images).

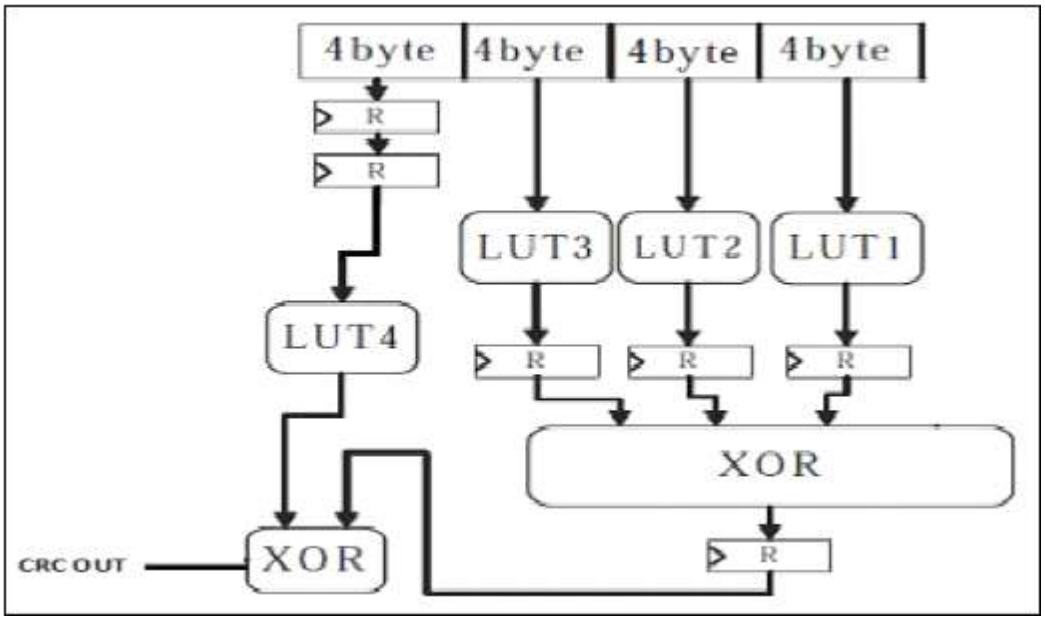


Figure 1-the structure of Slice 4 -CRC64[10]

Algorithm 1- Encryption of the color image using AES Algorithm
Input: BMP color image Color_img[row,colum]as plaintext row=0,1,….,width-1, colum=0,1…,height-1
Output: Encrypted color image Color_img_Enc[row,colum] as a ciphertext row=0,1,….,width-1, colum=0,1…,height-1
step1: repeat step2 to step 5 **step2**: Extract 16 bytes from Color_img and save it in state and Encrypted through step 3 and step 4 **step3**: Encrypt the state using the following three layout of AES algorithm **SubByte** **Shiftrow** **AddRoundKey** **Step4**: for k=0 to 9 **SubByte** **Shiftrow** **MixColumn** **AddRoundKey** **Step5**: save the encrypted blocks state in color_img_Enc

Algorithm 2: Compute CRC 64 for encrypted color images

Input: Color_img_Enc[row,colum]as ciphertext
row=0,1,….,width-1,colum=0,1…,height-1

Output: New file CRC64_check_file

for every block in color_img do steps

step1: for every row in color_img_Enc do
step2: for every columincolor_img_Enc do
step3: for every 16 bytes in color_img_Enc divided in to four slices[i] ,i=0,1,2,3…,etc
step4: each slice[i] consist 4-bytes ,and for every byte compute all possible values
step5: compute Look Up Table LUP_Table[] for the values in step 4
step6: compute the final CRC 64 for each slice using the equation

Results and Discussion

The proposed algorithm has been established using Microsoft Visual Studio C++ programming language, Windows 7 with 64-bit operating system, Intel(R) Core(TM) i5. Four BMP image files of different sizes as shown in Figure- 2 were used to test the proposed algorithm for the encryption quality and tested noise factor for these images. Besides that, different noise ratio was added to these BMP images to study the effect on them. The proposed algorithm uses these tested images by added different noise ratios (1% and 5%) of total images size to study the noise effect on the encrypted images. The noise added on single and multi bitspoistion and study the effect on the output results, Figure- 3 presents the output results of the encrypted images for the aforementioned cases. Tables- 2 demonstrate the MSE for both the source and noise sample images and clarify the probable error that might be occurring in decrypted operation for these images. The number of bytes affected also vary from one image to another due to the different sizes of these images. Figure-4 clarifies graphically the difference in MSE for eachimagesamplesof 5% and 1% added noise, it has been noticed from this figure that the large image size (the Cinderella image) the large MSE. The obtained results from the proposed algorithm compared to [15] are considered better for the encrypted images by AES, since this method is used to detect errors in encryption only or decryption only, while the proposed method detect errors for the images sent over the noisy channels beside the error in images itself during the decryption operation.

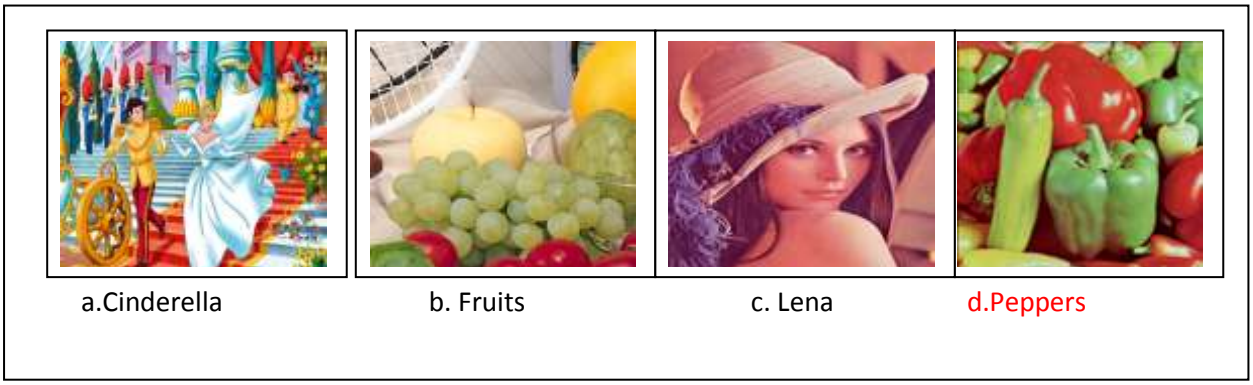


Figure 2-The tested images used by the proposed algorithm to compute the CRC

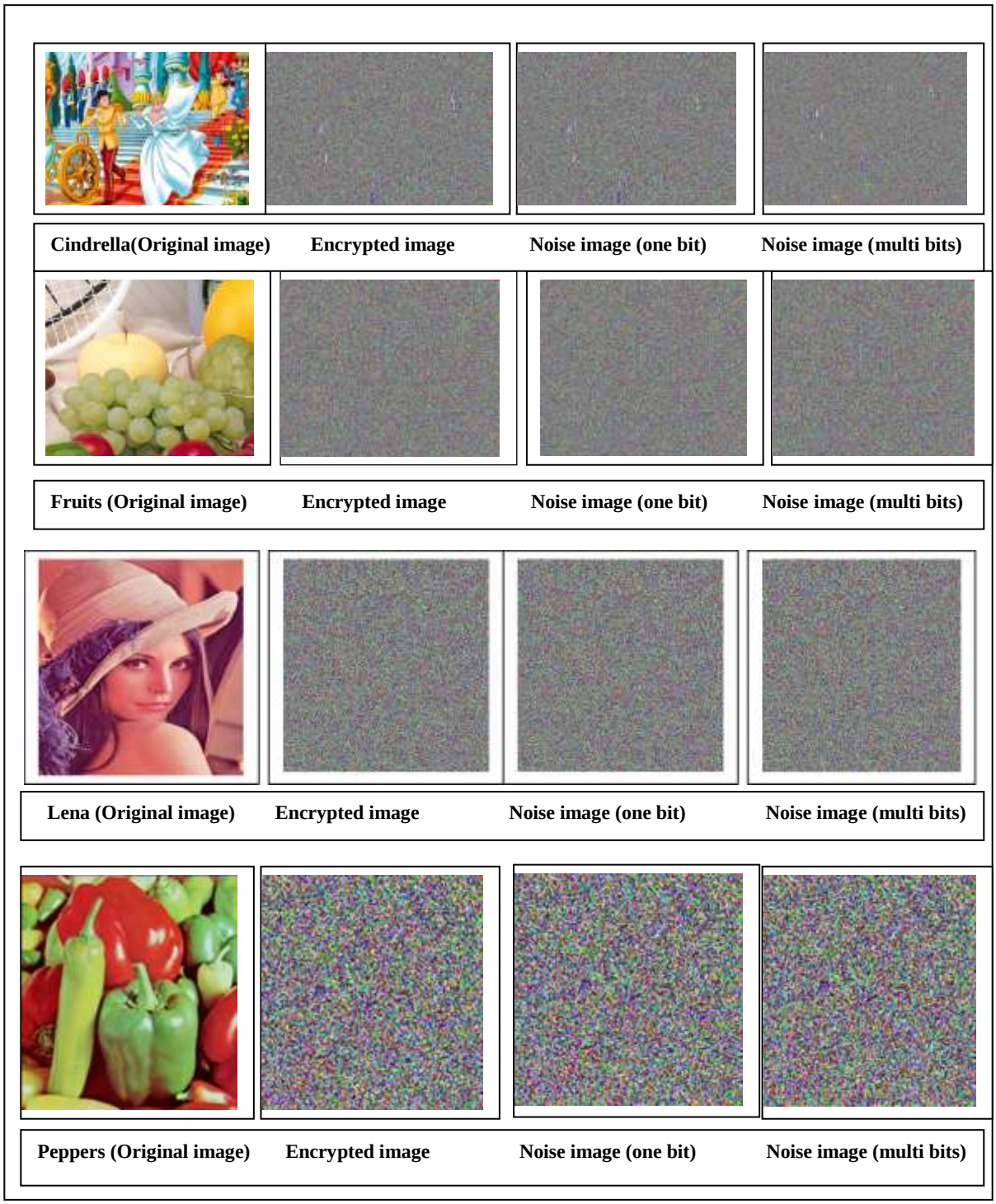


Figure 3-Encrypted images with added noise on single and multi bits.

Table2-The MSE for 1% and 5% added noise in images

Image name	ImageSize(bytes)	No. of noise bytes	MSE	No. of noise bytes	MSE
		1%(noise)		5%(noise)	
Cinderella	1,218,870	12188	11854.268	76175	11854.467
Fruits	786,486	7864	10047.267	39321	10049.079
Lena	451,142	4510	9059.456	22554	9058.019
Peppers	76,854	767	10048.684	3839	10040.052

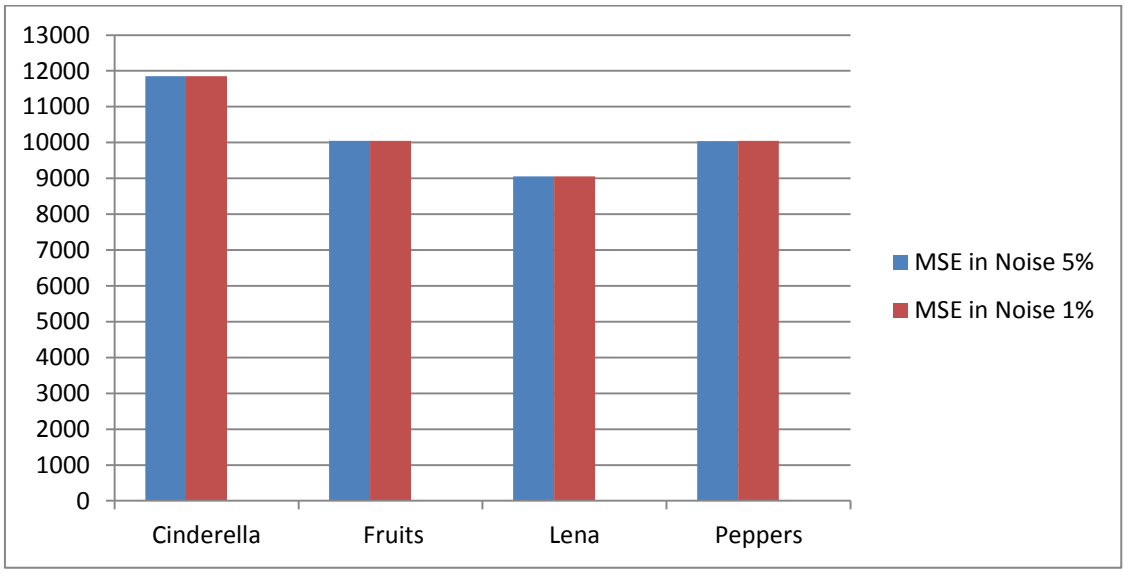


Figure 4-comparison Performance graph of MSE for 5% noise and MSE for 1% noise of total size encrypted images.

For the study purpose, the noise was added randomly on single bit position (the fourth bit is chosen) and multi-bits position in different bytes of the tested images. Table-3 shows the computed results implemented by Algorithm2 to compute CRC64for both the encrypted andnoisyimagesof 1% added noise affected on single and multi-bits position. In this table, one can notice that the 1% noise will change approximately (14.6 to15) % of the total number of CRC in the tested images, also the small image size the large numbers of affected CRC. Table-4 demonstrates the obtained results for 5% added noise ratio affected on single and multi-bitpoitions. Figure- 4 graphically demonstrate the comparision between 1% and 5% noise ratio for each tested image.

Table 3-The differences in CRC64 between encrypted and noise images, noise = 1% for single bitandmulti bitsposition.

Image file name	Size in bytes	Total number of CRC64 blocks	No. of effected Bytes by noise	No. of CRC64 affected by added noise	CRC64 difference ratio in %	No. of effected Bytes by noise	No. of CRC64 affected by added noise	CRC64 difference ratio in %
			Singe bit poition (bit position= 4)			Multi bits position		
Cinderella	1218870	76175	12188	11142	14.62 %	12188	11197	14.69%
Fruits	786431	49151	7864	7253	14.75 %	7864	7288	14.82%
Lena	451087	28192	4510	4192	14.86 %	4510	4203	14.90 %
Peppers	76799	4799	767	720	15 %	767	721	15.02%

Table 4-The differences in CRC64 between encrypted and noise images, noise = 5% for single and multi bitpositions.

Image file name	Size in bytes	Total number of CRC64 blocks	No. of effected Bytes by noise	No. of CRC64 affected by added noise	CRC64 difference ratio in %	No. of effected Bytes by noise	No. of CRC64 affected by added noise	CRC64 difference ratio in %
			Single bit position (bit position =4)			Multi bits position		
Cinderella	1218870	76175	60940	41223	54.11 %	60940	41847	54.93%
Fruits	786431	49151	39321	26544	54 %	39321	26963	54.85 %
Lena	451087	28192	22554	15372	54.52 %	22554	15608	55.36 %
Peppers	76799	4799	3839	2624	54.67 %	3839	2659	55.40 %

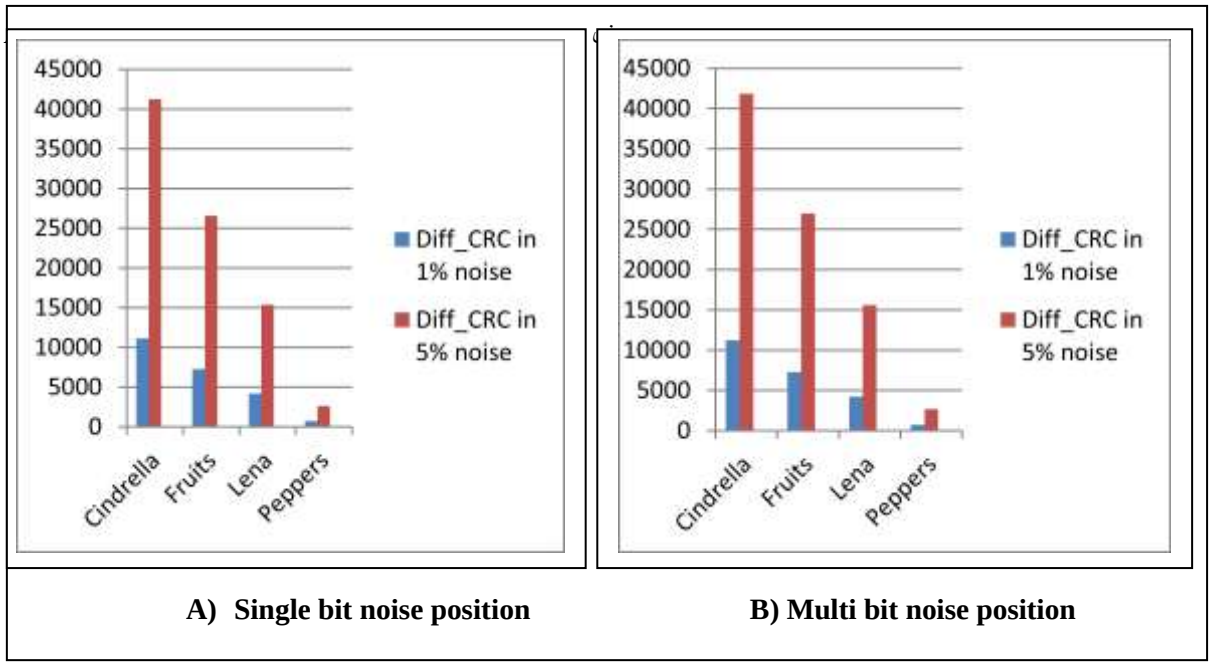
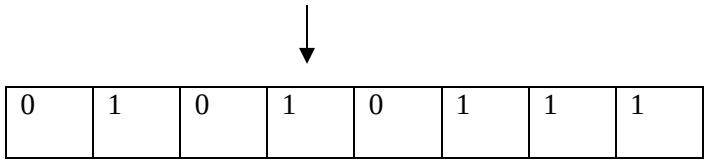


Figure 5-comparison performance graph between different CRC64 computed for noise = 5% and 1% of total size for different images

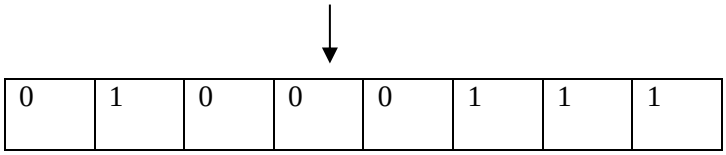
Some of the notes from the above table could be concluded as follows:

- The large size of the image the large CRC affected as in Cinderella image compares to Pappers image for both single and multi-bits position alteration.
- The added noise is random, so the probability of change the same bit in the same byte may occur many times, this leads either the change of specific bit or retrieve its original value as shown in following:
  The byte number 4000 has this value in binary (01010111); bit 5 is change from 1 to 0 when affected by added noise

Bit effect noise

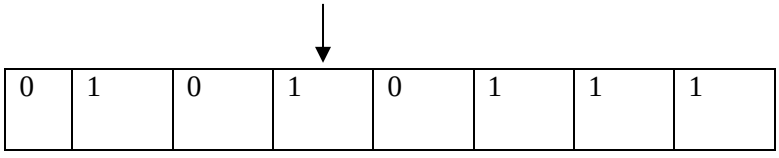


Bit effect noise



Moreover, the same bit is exposed to other noise, which yields to change to its original value.

Bit effect noise



So in this case the no change in original value.

- The number of the affected block (16-bytes) is about (2048 blocks) of total CRC64 numbers for a single bit for Cinderella image and about (2048 blocks) for multi bits position.

6- Conclusions

The color image is encrypted with the AES algorithm to achieve confidentiality, and these images were used as input for parallel 64 bit CRC implementation to achieve integrity. The CRC is used only for the error detection; the large size of the image the large CRC64 affected by noise while the small size of images yields a stable or fixed number of affected CRC64. The reason for that because the probability of change the same bit in the same byte may occur many times, this lead either the change of specific bit or retrieve its original value.

References

1. Nada, M., Ali Abdul Monem, S. and Abdul Mohsen, J.**2013**. Encryption using Dual Key Transformation based on Creation of Multi S- Boxes in AES Algorithm.*International Journal of Computer Applications*,**83**(10):1-6.
2. Nada, M., Ali, Abdul Monem S, Abdul Mohsen, J. and Sufian, Y. **2014**. A Byte-Oriented Multi Keys Shift Rows Encryption and Decryption Cipher Processes in Modified AES. *International Journal of Scientific & Engineering Research*,**5**(4): 953-955.
3. Verma, R. and Ali, J. **2013**. A Comparative Study of Various Types of Image Noise and Efficient Noise Removal Techniques, *International Journal of Advanced Research in Computer Science and Software Engineering*.**3**(10): 617-622.
4. Carey, G.,Smoak,MarioWidel and Sy Truong.**2012**.The use of checksums to ensure data integrityin the healthcare industry. *Pharmaceutical Users Software Exchange*, **5**(1&2): 38-42.
5. Daniel, N. Owunwanne.**2010**. Analysis Of The Effectiveness Of Error Detection In Data Transmission Using Polynomial Code Method. *International Journal of Management &Information Systems – Second Quarter*,**14**(2):105-112.
6. Yen,C. and Wu,B. **2016**. Simple error detection methods for hardware implementation of Advanced Encryption Standard, IEEE Transactions on Computers,**55**(6): 720-731.
7. Kounavis M. and Berry F. **2008**. Novel Table Lookup-Based Algorithms for High-Performance CRC Generation.*IEEE Transactions on Computers*,**57**(11): 1550-1560.
8. InduI , Manu T S.**2012**.Cyclic Redundancy Check Generation Using Multiple Lookup TableAlgorithms. *International Journal of Modern Engineering Research*,**2**(4): 2445-2451.
9. HuoY.m Li X. and Liu W. **2015**. High Performance Table-Based Architecture for Parallel CRC Calculation.*International conference on Local and Metropolitan Area Networks (LANMAN), IEEE.* 22-24 April.
10. Mathewa, N. P. and Mohanb, A.**2016**. Matrix Code Based Error Correction For LUT Based CyclicRedundancyCheck.*Published by Elsevier Ltd, Procedia Technology*,**25**: 590 – 597.
11. Nada, M. Ali, Abdul Monem S, Sufian Y and Abdul Mohsen J. **2014** . A Novel Multi Modification in AES Block Cipher Algorithm for Comlexity, *International Review on Computers and Software (I.RE.CO.S.)*, **9**(6):01-905.
12. Nada, M. Ali, Abdul Monem S, Abdul Mohsen, J. and Sufian, Y.**2014**. Random Key Permutation Stream Algorithm Based on Modified Functions in AES Algorithm.*International Journal of Engineering and Technology*,**4**(6): 367-373.
13. Sun,Y. and Kim, M. S. **2010**. Table-Based Algorithm for Pipelined CRC Calculation. *IEEE International Conference onCommunications(ICC)*, pp: 1-5.
14. Nada, M. Ali and Suaad Ali Abead.**2016**.Modified Blowfish Algorithm for Image Encryption using Multi Keys based on five Sboxes, *Iraqi Journal of Science*, **57**(4C): 2968-2978.
15. Chih-Hsu Yen and Bing-Fei Wu.**2006**. Simple Error Detection Methods for HardwareImplementation ofAdvanced Encryption Standard.*IEEE TRANSACTIONS ON COMPUTERS*, **55**(6):720-731.