Journal of Engineering Design and Analysis
Volume 3, Issue 2 - 2020, Pg. No. 92-97
Peer Reviewed Journal

Article

A Novel Aspect of Security Trade-Offs in Image Files

Ranjita Rout[1], Debasis Gountia[2], Neelamani Samal[3], Bijay Srinibas Nag[4]

[1]Gandhi Institute for Education and Technology, Bhubaneshwar, Odisha, India.
[2]Member, IEEE, College of Engineering and Technology, Bhubaneshwar, Odisha, India.
[3]Aryan Institute of Engineering and Technology, Bhubaneshwar, Odisha, India.
[4]College of Engineering and Technology, Bhubaneshwar, Odisha, India.

I N F O

Corresponding Author:
Ranjita Rout, Gandhi Institute for Education and Technology, Bhubaneshwar, Odisha, India.
E-mail Id:
rout.ranjita@gmail.com
Orcid Id:
https://orcid.org/0000-0003-0499-3436
How to cite this article:
Rout R, Gountia D, Samal N et al. A Novel Aspect of Security Trade-Offs in Image Files. *J Engr Desg Anal* 2020; 3(2): 92-97.

Date of Submission: 2020-11-13
Date of Acceptance: 2020-12-06

A B S T R A C T

This paper is a contribution to the ongoing research in the design of security aspects of an image file and important field used to protect the confidentiality of data in the disk. In this paper, we focus on Cipher block chaining as this technique appear to offer the best combination of security and performance. In this paper, we highlight the research to date in the area of security of an image file and propose a novel narrow- block disk encryption mode of operation with compression of data first. This is the Cipher Block Chaining (CBC) mode using Xor-Encrypt-Xor (XEX) to inherit from its security and high performance and use CBC like operations to gain the error propagation property. Here we use "LZW 15-bit Variable Rate Encoder" for the compression of image. We also apply multiplication and exponential in the finite field GF (2128). Here we use Cipher image Stealing when data size is not multiple of 16 bytes. Our hope is to generate an image file scheme that will provide high throughput, faster, memory saving and better resistant to the attacks.

Keywords: Tweak, Image, CBC, Galois Field, LZW Compression, Security

Introduction

The image files require huge space for storage. The issues involved in this paper are giving a look into the security aspects of image file. For example the signature or thumb impression used for authentication of a pen should be stored in secure manner and it should take less space, so that while image file is transferred over communication network or while in client computer should not be accessed unauthorized.[1] So various aspects of providing security of an image file is discussed and our new approaches are also discussed Images are one of the most common 'containers' for hidden messages. A 24-bit image contains 24 million possible colors. Each color is made up of varying amounts of red, blue, and green. The amount of red, blue, and green is represented by 1byte, which is 8 bits per pixel (a small unit of area for a digital image), a 24 bit picture uses 3 bytes, or 24 bits. Special steganography[4] software can hide data within the 3 'least important' bits of the 24 which make up one pixel without affecting image quality to any noticeable degree. Thus, 1/8 of the size of a 24 bit image could be hiding secret data.[3] 24 bit images are very large and therefore fairly rare on the internet where speed is a premium though. Sending a 24 bit through email would probably draw more red flags than just sending a personal message. This 'Least Significant Bit' (LSB) method is also useful for other types of images that are found on the net, including bitmaps and gifs.

JPEGs work under a different process from the previously mentioned files because of the 'lossy' file compression schemes that are used. Gif images use 'loss-less' image compression in which all information for the picture is retained. The compression that is applied to gifs removes possible colors to be used in the image. JPEG compression, however, Bod selectively 'losses' (hence the 'lossy' term) parts of information of the picture. In JPEG picture compression, picture data is converted from the RGB values that describe individual pixels to the 'luminance' and 'chrominance', otherwise known as brightness and hue. The image is then divided in 8 by 8 pixel squares and each square run through a 'Discrete Cosine Transform' (DCT)[7] whose data is quantified into DCT coefficients, which are later compressed by a 'Variable Length Code'. All of this is done using special algorithms that remove as much data possible while maintaining the highest image quality. During the DCT step, information about the JPEG image is discarded. This is a major roadblock for steganography[10] because data cannot be placed into the image before JPEG encoding or it will be deleted by the DCT. Furthermore, data cannot just be appended to the image after it's encoded because of the lack of security.

Derek Upham recognized that data can be stored in JPEGs by placing the data in the message while it is encoded. His program, JSTEG, adds information into the JPEG after the DCT step, which means that the data won't be deleted. Furthermore, after the DCT step, additional compression is applied to the DCT coefficients so that the information is fairly well hidden within the JPEG and not just added to it.

Contributions of this Paper

This paper is a contribution to the ongoing research in the design of a security model of an image file. We focus on the AES based algorithm for image encryption as these appear to offer the best combined security and performance, so we will motivate in our work. In this work, we implemented image encryption using chaos and block cipher operation with compression. We decided to build the modified AES based algorithm for image encryption and an improved image encryption algorithm based on chaotic system. Encryption using block based transformation algorithm, an approach to larger size data with authenticity and integrity.

This paper is a contribution to the ongoing research in the design of a security model of an image file. We focus on the AES based algorithm for image encryption as these appear to offer the best combined security and performance, so we will motivate in our work. In this work, we implemented image encryption using chaos and block cipher operation with compression. We decided to build the modified AES based algorithm for image encryption and an improved image encryption algorithm based on chaotic system. Encryption using block based transformation algorithm, an approach to larger size data with authenticity and integrity.

The objective of the work is to develop a fast security model of image file system. The objective is actually to achieve security, speed and error propagation with less consumption of space, i.e., the size of hardware implementation and the amount of secure storage space required. Otherwise, encryption and decryption may take so much time that software which run on computers become unacceptably slow. Our contributions to the field are the following:

The paper first presents LZW 15-bit Variable Rate Encoder for effective compression of data in order to achieve speed as LZW encoder reduces files to about half original size on large image which contains a huge amount of data. This work also includes efficient algorithms for exponentiation and multiplication in the finite field GF (2128) that can operate in any common field representations. This paper includes the description of the AES transform in both encryption and decryption modes, as well as how it should be used for encryption of a sector with a length that is not an integral number of 128-bit blocks. The scope is limited on the size of storage data encrypted with a single key.

Encryption with Compression and Error Control

Using a data compression algorithm together with an encryption algorithm makes sense for two following reasons:

- Cryptanalysis relies on exploiting redundancies in the plaintext; compressing a file before encryption reduces redundancies.
- Encryption is time-consuming; compressing a file before encryption speeds up the entire process.

Any type of transmission encoding or error detection and recovery will be added after encryption if needed.

Proposed Scheme

The goals of designing the proposed scheme, Image Encompression with Cipher Block Chaining (IECBC) mode are:

Security

The constraints for image encryption imply that the best achievable security is essentially what can be obtained by using CBC mode with a different key per block.

Performance

IECBC should be at least as fast as the current available solutions.

Parallelization

IECBC should offer some kind of parallelization.

Error Propagation

IECBC should propagate error to further blocks (this may be useful in some applications).

ISSN: 2582-5607

Encryption of a data Unit

The encryption procedure for a 128-bit block having index j is modeled with Equation (1):

$C_i \leftarrow$ IECBC-AES-blockEnc (Key, P_i, I, j) (1)

Where;

Key is the 256-bit AES key Pi is a block of 128 bits (i.e., the plainimage) is the address of 128-bit block inside the data unit j is the logical position or index of the 128-bit block inside the sector C_i is the block of 128 bits of cipherimage resulting from the operation.

The key is parsed as a concatenation of two fields of equal size called Key1 and Key2 such that:

Key = Key1

Key2.

The plain-image data unit is partitioned into m blocks, as follows:

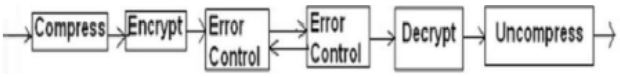

Figure 1.Encryption with compression and error control

For access data from the disk, we have to first decrypt and then uncompressed the decrypted data. For image encryption, there is no requirement of any encoding or error detection and recovery as there is no transmission. Hence, the steps for our scheme are as follows:

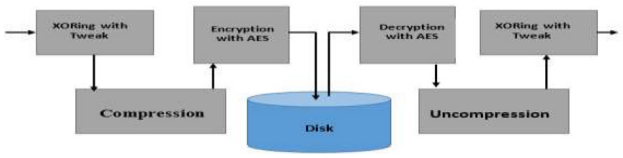

Figure 2.Steps for our image encryption scheme

$$P = P_1 \| \ldots \| P_{m-1} \| P_m$$

where m is the largest integer such that 128(m-1) is no more than the bit-size of P, the first (m -1) blocks $P_1,\ldots,P_{m-1}$ are each exactly 128 bits long, and the last block Pm is between 0 and 127 bits long (Pm could be empty, i.e., 0 bits long).

The cipherimage Ci for the block having index j shall then be computed by the following or an equivalent sequence of steps (see Figure 2):

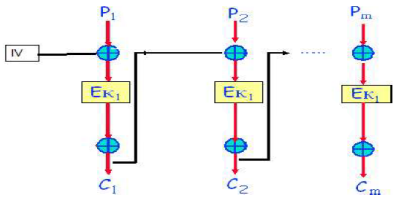

Figure 3.Encryption of image data unit using IECBC

1. $PP_i \leftarrow P_i \oplus IV$
2. $CC_i \leftarrow$ AES-enc(Key_1, PP_i)
3. $C_i \leftarrow CC_i \oplus T_{i-1}$

Case2 (j > 0):

1. $PP_i \leftarrow P_i \oplus IV$
2. $CC_i \leftarrow$ AES-enc (Key_1, PP_i)
3. $PP_{i+1} \leftarrow P_{i+1} \oplus Ci$
4. $C_{i+1} \leftarrow$ AES_enc (key, PP_{i+1})

AES-enc (K, P) is the procedure of encrypting plain image P using AES algorithm with key K, according to FIPS-197. The multiplication and computation of power in step (1) is executed in GF (2^{128}), where *a* is the generated primitive element.

Decryption of a Data Unit

The decryption procedure for a 128-bit image block having index j is modeled with Equation (2):

$$P_i \leftarrow \text{IECBC-AES-block Dec } (Key, C_i, I, j)$$

where;

Key is the 256-bit AES key.

C_i: the 128-bit block of cipher-image.

I: Is the address of the 128-bit block inside the data unit j is the logical position or index of the 128-bit block inside the sector.

P_i: Is the block of 128-bit of plain-image resulting from the operation.

The key is parsed as a concatenation of two fields of equal size called Key1 and Key2 such that:

Key = Key1||Key2

The cipher image first partitioned into m blocks, as follows:

C = C1||...|| C_{m-1}||C_m

Where m is the largest integer such that 128(m-1) is no more than the bit-size of C, the first (m-1) blocks $C_1\ldots,C_{m-1}$ are each exactly 128 bits long, and the last block Cm is between 0 and 127 bits long (C_m could be empty, i.e., 0 bits long).

The plain-image P_i for the block having index *j* shall then be computed by the following or an equivalent sequence of steps (see Figure 4):

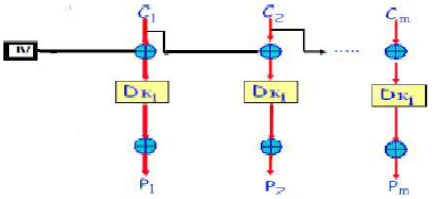

Figure 4.Decryption of cipher image blocks using IECBC

Algorithm

IECBC-AES-blockDec

(Key, C_i, I_i, j) Case1 (j = 0):

1. $CC_i \leftarrow C_i \oplus IV$
2. $PP_i \leftarrow$ AES-dec

(Key_1, CC_i) Case2 (j > 0):

1. $IV \leftarrow$ AES-enc (Key_2, I_i) $\otimes C_{i-1}$
2. $CC_I \leftarrow C_i \oplus IV$
3. $PP_i \leftarrow$ AES-dec (Key_1, CC_i)

AES-dec (K, C) is the procedure of decrypting cipher image C using AES algorithm with key K, according to FIPS-197. The multiplication and computation of power in step (1) is executed in GF (2^{128}), where *a* is the generated primitive element.

PKCS5 Padding is a padding scheme described in: RSA Laboratories, "PKCS #5: Password-Based Encryption Standard," version 1.5, November 1993.

PKCS5Padding schema is actually very simple. It follows the following rules:

- The number of bytes to be padded equals to "8 number of Bytes (clearText) mod 8". So 1 to 8 bytes will be padded to the clear text data depending on the length of the clear text data
- All padded bytes have the same value - the number of bytes padded

PKCS5 Padding schema can also be explained with the diagram below, if M is the original clear text and PM is the padded clear text:

If number of Bytes (clearText) mod 8 = 7, PM = M + 0x01

If number of Bytes (clearText) mod 8 = 6, PM = M + 0x0202

If number of Bytes (clearText) mod 8 = 5, PM = M + 0x030303

...If number Of Bytes (clearText) mod 8 == 0, PM = M + 0x0808080808080808 PKCS#5 and PKCS#7 specifies the same padding scheme: to repeatedly appends bytes each of them containing total amount of padded bytes. i.e., if you need 5 bytes to pad your message, then padding will be {5,5,5,5,5}.

Security and Performance Analysis

Security in general is the degree of protection against attack, danger, loss, and criminals. Security has to be compared and contrasted with other related concepts: Safety, continuity, reliability. The key difference between security and reliability is that security must take into account the actions against people attempting to cause destruction. Different algorithms offer different degrees of security; it depends on how hard they are to break.

Security

Each block is encrypted with a different tweak T, which is the result of a nonlinear function (multiplication) of encrypted file address and previous cipherimage (a^{Js} for 1st block); due to this step the value of the cipher is neither known nor controlled by the attacker. By introducing the cipher, the attacker cannot perform the mix- and-match attack[19] among blocks of different sectors, as each sector has a unique secret cipher. Any difference between two tweaks result full diffusion in both the encryption and decryption directions. These enhance the security.

Here we also give option for the value of *a to* the user; it reduces the probability of getting plaintext from cipherimage. This is so because same plaintext produces different cipherimage if we choose different value for *a*. This also increases confusion.

Complexity

IECBC possesses high performance as it uses only simple and fast operations as standard simple shift and add (xor) operators are used in the multiplication in the finite field GF (2128) having O(1) time complexity. Compression before encryption also enhances the speed and hence performance.

Parallelization

IECBC can be parallelized on the sector level as each sector is encrypted independently to other sectors as in [20]. Also a plaintext can be recovered from just two adjacent blocks of cipher text. As a consequence, decryption can be parallelized.

Error Propagation

As each block depends on its previous block, a one-bit change in a plain image affects all following cipher image blocks. Hence, error propagation is met.

Conclusion

Contributions of the Paper

In this paper, a highly secure AES-based Cipher Block Chaining with Cipherimage Stealing has been proposed for security of image file. The important features of CBC are the use of Cipher block chaining mode like operations to gain the error propagation property.

A one-bit change in a plaintext affects all the following ciphertext blocks in a sector. In the cipher-block chaining (CBC) mode, each block of plaintext is XORed with the previous ciphertext block before being encrypted. This way, each ciphertext block is dependent on all plaintext blocks processed up to that point. Also, to make each

ISSN: 2582-5607

message unique, an initialization vector must be used in the first block. CBC has been the most commonly used mode of operation. Its main drawbacks are that encryption is sequential (i.e., it cannot be parallelized) and that the message must be padded to a multiple of the cipher block size. One way to handle this last issue is through the method known as cipherimage stealing. It is important to note that a onebit change in a plaintext affects all following cipher image blocks. A plaintext can be recovered from just two adjacent blocks of cipher image. As a consequence, decryption can be parallelized, and a one-bit change to the cipher image causes complete corruption of the corresponding block of plaintext and inverts the corresponding bit in the following block of plaintext.

Any difference between two tweaks result full diffusion in both the encryption and decryption directions. All these factors improve security. It has been shown that the proposed mode possesses a high throughput as compression is done before enciphering scheme. Only standard shift and add (xor) operators have been used for the non-linear multiplication function in the finite field GF(2128) having O(1) time complexity, therefore gives better resistance against linear cryptanalysis without degradation in performance speed. This proposed mode has ability to encrypt arbitrary length messages due to the use of cipherimage stealing technique. Although, it was designed based on the CBC mode, it does not suffer from the bit flipping attack.

IECBC can be parallelized on the sector level as each sector is encrypted independently to other sectors. But encryption of the blocks of a sector is sequential (i.e., it cannot be parallelized) as each block depends on its previous block in a sector. A plaintext can be recovered from just two adjacent blocks of cipherimage.

As a consequence, decryption can be parallelized and a one-bit change to the cipherimage causes complete corruption of the corresponding block of plaintext, and inverts the corresponding bit in the following block of plaintext.

To the best of my knowledge, the proposed scheme IECBC with ciphertext stealing will provide a designer of a practical image encryption algorithm with attractive alternatives.

Future Scope

There still remain many open problems in the search for efficient and secure security of an image file. In the past few years the interest of the research community in image files can be solved in the coming years as the following interesting open problems:

- There is a lack of good Boolean functions for which image files are efficient and also resist the cryptanalytic attacks, in particular algebraic and fast algebraic attacks
- Extend the current work to audio, and video files encryption with compression first
- Implement the proposed scheme by using AES with a 256-bit key
- Introducing the image encryption/ decryption hardware implementation of the entire work.

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

References

1. Schneier B. Applied Cryptography, Wiley Press, Second Edition.
2. Stinson DR. Cryptography Theory and Practice, CRC Press, Second Edition.
3. Nelson M, Gailly JL. The Data Compression Book, M&T Press, Second Edition.
4. Daemen J, Sand B, Rijmen V. The Design of Rijndael: AES The Advanced Encryption Standard, Springer-Verlag, Berlin, 2002.
5. Welch LZ. Available: http://en.wikipedia.org/wiki/Lempel Ziv Welch.
6. Bourbakis N, Dollas A. Scan-based compression encryption hiding for video on demand. *IEEE Multimedia Mag* 2003; 10: 79-87.
7. Jolfaei A, Mirghadri A. Federal Information Processing Standards Publications (FIPS 197). A Novel Image Encryption Scheme Using Pixel Shuffler and A5/1. In Proceedings of the 2010 International Conference on Artificial Intelligence and Computational Intelligence (AICI 2010), Sanya, China. 2010a.
8. Jolfaei A. Mirghadri A. An Image Encryption Approach Using Chaos and Stream Cipher. *Journal of Theoretical and Applied Information Technology* 2010b; 19(2), 117-125.
9. Jolfaei A, Mirghadri A. Survey: Image Encryption Using Salsa 20. *International Journal of Computer Science* 2010c; 7(5).
10. Jolfaei A, Mirghadri A. A New Approach to Measure Quality of Image Encryption. *International Journal of Computer and Network Security* 2010d; 2(8): 38-44.
11. Kocarev L, Stczepanski J, Amigo JM. Discrete Chaos I: Theory. IEEE Encryption Standard (AES), 26 Nov. 2001 transaction on circuit system. 2006.
12. Bruice Schneier, Applied Cryptography, Wiley Press, Second Edition.
13. Stinson DR. Cryptography Theory and Practice, CRC Press, Second Edition.
14. Nelson M, Gailly JL. The Data Compression Book, M&T Press, Second Edition.
15. Stallings W. Cryptography and Network Security, Pearson Education, Fourth Edition.

16. http://en.wikipedia.org/wiki/Image_processing
17. Gonzalez RC, Woods RE. Digital Image processing.
18. Guwalani P, Chandrashekar R, Kala M et al. Image File Security using Base-64 Algorithm, International Journal of Computer Technology and Applications, 2010.
19. Kumar D, Nishad et al. Review Paper on Image Steganography and Security Using Cryptography. *International Journal for Research in Applied Science & Engineering Technology* 2016.
20. Gounita D, Chowdhury DR. A new narrow-block made of operation for disk encompression wiath tweaked block chaining. *International Journal of Computer Science & Engineering Technologies* 2011; 2(1): 71-76.

ISSN: 2582-5607