Image Encryption / Decryption System based on Permutation and Blowfish Algorithm

May Phyo Thu, Thin Thin Htike
Computer University, Pathein, Myanmar
memoelayOO1@gmail.com

Abstract

Internet and networks applications are growing very fast , so the needs to protect such a applications are increased. Data encryption / decryption is used to securely transmit data in open networks . Each type of data has its own features , therefore, different techniques should be used to protect confidential image data from unauthorized access. This system presents a new permutation technique based on the combination of image permutation and a well known encryption and decryption algorithm called Blowfish ,a symmetric block cipher. Blowfish is a fast cryptographic software algorithm, using the operations of addition, XOR and look-up tables. In addition, Permutation process is also added in this system to be difficult predicting the value of pixel from the values of its neighbors. In this system, the original image is divided into 4 pixel × 4 pixels blocks, which are rearranged into a permuted image using a permutation pro Blowfish algorithm .

Keywords: Image Encryption, Image Correlation, Permutation, symmetric algorithm, Blowfish algorithm.

1. Introduction

The development of information technology and the rapid growth of computer networks allowed large files such as digital images to be easily transmitted in open networks such as the internet [1]. Each type of data has its own aspects and different techniques should be used to protect confidential image data from unauthorized access. Encryption is the process of transforming the information to ensure its security . Due to large data size and real time constrains, algorithms that are good for textual data may not be suitable for multimedia data[2].

In most of the natural images, the values of the neighboring pixels are strongly correlated. Hence, we develop a secure image encryption / decryption system using the combination of permutation process and Blowfish algorithm. The permutation process will be used to divided the original image into a number of blocks (4 pixel × 4 pixels blocks) that are then shuffled their positions within the image. The generated image is then fed to the Blowfish algorithm. We can achieve more security and hide the relationship between the ciphertext and the plaintext by using the permutation process. By using the correlation as a measure of security, this process results in a lower correlation value when compare to using the Blowfish algorithm alone and thus improving the security level of the encrypted images.

The structure of this paper as follow: Section 1 describes Introduction to the system. Section 2 explains cryptography, symmetric algorithm, Blowfish algorithm, straight permutation, permutation algorithm and how it works. Section 3 describes the proposed system. Section 4 presents the experimental results. Finally, section 5 concludes the paper.

2. Background Theory

The parts of Cryptography, Symmetric Algorithm, Blowfish Algorithm, Straight permutation and Permutation Algorithm that they were clearly explained in this section of Background Theory.

2.1. Cryptography

Cryptography is the study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, entity authentication, and data origin authentication [3]. Cryptography is not the only means of providing information security, but rather one set of techniques.

2.2. Symmetric Algorithm

Symmetric cryptography, also referred to as "secret-key" cryptography, is a form of cryptography in which only one encryption key is used; In other words, the sender and receiver of encrypted data use two instances of the same key for both encryption and decryption. It is much faster than asymmetric algorithms. It is hard to break if using a large key size.

2.3. Blowfish Algorithm

Blowfish is a variable-length key block cipher. It does not meet all the requirements for a new cryptographic standard; it is only suitable for applications where the key does not change often, like a communications link or an automatic file encrypts 1]. This encryption process decreases the mutual information among the encrypted image. Blowfish is a symmetric block cipher that encrypts data in 8-byte blocks. It has an advantage over other RC6, AES, 3DES, DES and RC2 in terms of time consumption and through put. It encrypts 64-bits blocks of plaintext into 64-bit blocks of cipher text. It makes use of a key that range from 32 bits to 448 bits. There are two part to this algorithm: a key- expansion part and a data - encryption part . Key expansion converts a variable-length key of at most 56 bytes (448 bites) into several sub key arrays totaling 4168 bytes. Data encryption occurs via a 16-round feistel network. Blowfish has 16 rounds. Each round consists of a key-dependent permutation and a key-and data-dependent substitution. All operations are XORs and additions on 32-bit words. The only additional operations are four indexed array data look ups per round..

Encryption Algorithm

The input data is 64 bit data element: x
Divide x into two 32-bit halves: xL, xR
For i = 1 to 16
    xL = xL XOR Pi
    xR = F (xL) XOR xR
    Swap xL and xR
Swap xL and xR (Undo the last swap)
xR = xR XOR P17
xL = xL XOR P18
Recombine xL and xR
Function F:
Divide xL into four eight-bit quarters: a, b, c, and d
F (xL) = ((S1,a + S2,b mod 2^{32}) XOR S3,c) + S4, d mod 2^{32}.

2.4 Straight Permutation

There are three types of permutation. They are straight, compression and expansion permutation. We use of them named straight permutation. Figure1 shows the straight permutation process.

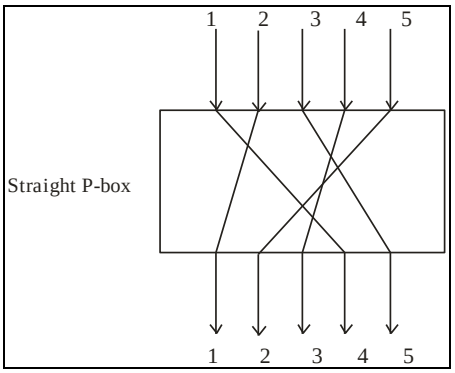

Figure 1.Straight permutation

The straight permutation box with n inputs and n outputs is a permutation. It is keyless transposition cipher. This permutation maps each input bit to an output positions ; no bits are used twice and no bits are ignored . We can use a straight permutation box in the encryption cipher and its inverse in the decryption cipher[7].

2.5 Permutation Algorithm

The permutation technique works as follows ; The plain image can be decomposed into blocks; each one contains a specific number of pixels (4 pixels × 4 pixels blocks) The blocks are transformed into new locations[5]. The permuted image is fed to the Blowfish encryption algorithm. The intelligible information present in an image is due to the correlation among the image elements in a given arrangement[6].This perceivable information can be reduced by decreasing the correlation among the image elements using certain permutation techniques. For better permutation, the block size should be small because fewer pixels keep their neighbors. As a result the correlation will be decreased and thus it becomes difficult to predict the value of any given pixel from the values of its neighbors . This process of dividing and shuffling the positions of image blocks will confuse the relationship between original image and permuted one. The original image can be obtained by the inverse permutation of the blocks.

ALGORITHM CREATE_PERMUTATION_TABLE

1: Load the plain Image
2: Input secret key
3: Get the Width and Height of the image
4:
  4.1: Lower Horizontal Number of Blocks =
    Integer (Image Width / 4)
  4.2: Lower Vertical Number of Blocks =
    Integer (Image Height /4)
5: Number of Blocks = Horizontal Number of Blocks
    × Vertical Number of Blocks
6: Seed = | Hash value (Key) |
7: Randomize ( )
8: For I = 0 to Number of Blocks -1
  8.1: Get the new location of block I from the permutation table
  8.2: Set block I in its new location

END PERFORM_PERMUTATION

Input: plain Image and permutation table
Output: permuted Image.

3. Proposed System

In this paper, we propose a permutation process and Blowfish encryption Algorithm. First, we input the image and key .Then we get the width and height of the image .Next, we get the horizontal number of blocks by dividing the image width by 4. Also, we get the vertical number of blocks by dividing the image height by 4. Next, we get the number of blocks by multiplying the horizontal number of blocks and the vertical number of blocks. Next, we calculate the Seed by hashing the key. Next, we shuffle the image with the key by block by block. Next, we move the blocks into the new location from the existing ones. By the way, we get the permuted image. After we getting the permuted image, we divide the image into left and right halves. In the iteration 1 to 16 rounds, first the left halve is XORed with the first row of the P array. Then, the XORed result is put into the left halve again. Next, Function of the left halve is XORed with the right halve. Then, swap the left halve and right halve. In this way, we calculate the 16 times. After calculating the 16 times, we again swap the left halve and right halve. Next, the right halve is XORed with the 17^{th} row of the P array. Next, the result is put into the right halve again. Next, the left halve is XORed with the 18^{th} row of the P array. Next, the result is put into the left halve again. Finally, we get the encrypted image by recombining the left halve and right halve. On the other hand, image decryption retrieves the original image from the encrypted one in this way. In this system, we can only input the key length from 32 bits to 448 bits. The encryption key and the decryption key must be the same. We propose this system for the security storage of the image. This figure 2 is the system flow diagram.

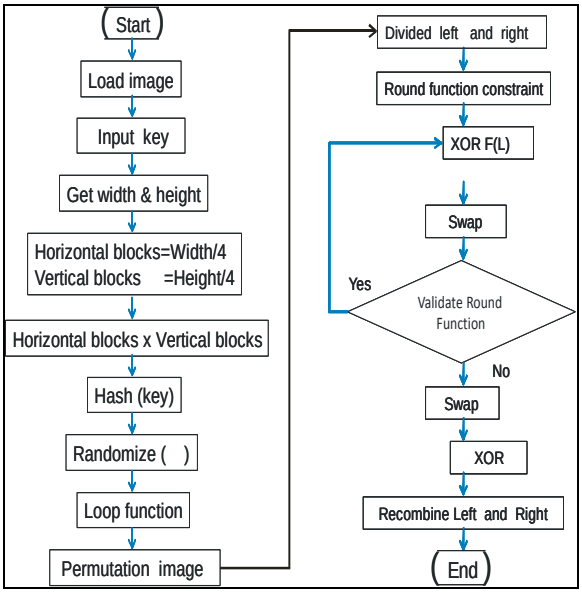

Figure 2. System flow diagram

4. Experimental Results

This system is implemented using Java programming language and it was tested in computer with processor Intel (R) Pentium (R) Dual CPU 2.20 GHz and 2GB of RAM. This system cannot process every type of image that Windows support. In this system we emphasize on bmp, jpg and gif images. We experiment the different sizes of the image with different key sizes. Let F and F' denote the original image (plainimage) and the encrypted image (cipherimage) respectively each of size M*N pixels with L grey levels. The encryption quality (EQ) represents the average number of changes to each grey level L and is expressed mathematically as

$$EQ = \frac{\sum_{L=0}^{255} | H_L(F') - H_L(F) |}{256}$$

Following tables (1) and (2) show encryption time and decryption time for different image sizes of different key sizes. The encryption quality (EQ) is computed and the results obtained for the image is shown in Figure (3).

Table1. Encryption time

Image Size	Encryption Time in milliseconds

		.bmp	.jpg	.gif
96 × 96	8 bytes	165	181	192
	16 bytes	180	216	260
	32 bytes	199	231	240
300 × 250	8 bytes	1175	842	1980
	16 bytes	2491	741	1955
	32 bytes	2447	1106	1373

Table2. Decryption time

Image Size		Decryption Time in milliseconds		
		.bmp	.jpg	.gif
96 × 96	8 bytes	164	145	200
	16 bytes	174	210	258
	32 bytes	200	230	242
300 × 250	8 bytes	1170	845	1978
	16 bytes	2490	745	1950
	32 bytes	2450	1100	1973

Following figures (3) and (4) show the sample runs of the image encryption / decryption program.

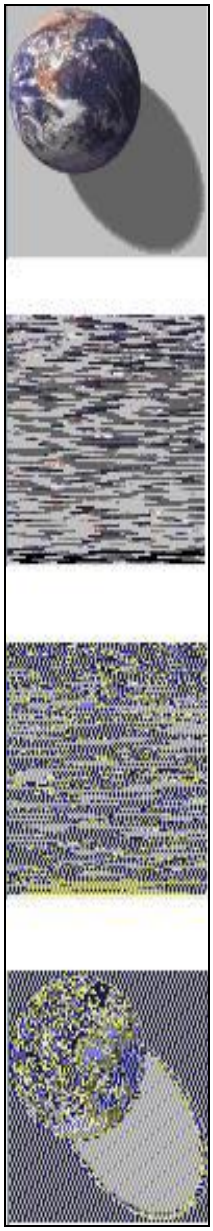

Figure3. Encryption process

Figure 4. Decryption process

5. Conclusion

Image data has strong correlation among adjacent pixels. However, it is very important to disturb the high correlation among image pixels to increase the security level of the encrypted images. This system proposes a simple and strong method for image security using a combination of image permutation and encryption techniques. From the theory results, the correlation was decreased when the proposed algorithm was applied to them before the Blowfish algorithm. The process of dividing and replacing an arrangement of original

image into 4pixels×4 pixels blocks reduced the correlation between image elements and then achieved to confuse the relationship between the original image and the permuted one. This system cannot process every type of images. As a result, the security level of the encrypted images is increased by reducing the correlation among image elements.

6 REFERENCES

[1] B.Schneier, The Blowfish Encryption Algorithm Retrieved Oct. 25,2008. (http://www.schneier,com/blowfish.html)

[2] I.Ozturk, and I.Sogukpinar, "Analysis and comparison of image encryption algorithm," International Journal of Information Technology, Vol.1, no.2, pp.64-67. http://www.waset.org/

[3] J.Daemen, and V.Rijmen, "Rijndael: The advanced encryption standard" Dr.Dobb's Journal, pp.137-139, Mar 2001

[4] Maniccam., G.Nikolaos, and Bourbokis, " Lossless image compression and encryption using SCAN" Journal of : Pattern Recognition, vol-34, no.6:,2001, pp.1229-1245.

[5] Mitra, Y,V,Subba Rao, and S.R.M. Prasnna, "A new image encryption approach using combinational permutation technique " Journal of computer Science, vol.1, no.1, P.127, 2006, Available: .http://www.enformatika.org

[6] Shujun, Li. Chengqing, C.Guanrong, Fellow, IEEE., Dan Zhang., and Nikolaos, G.,Bourbakis Fellow., IEEE. "A general cryptanalysis of permutation – only multimedia encryption algorithms" 2004, http://eprint. Iacr.Org/2004/374.pof.

[7] W.Lee, T.Chen and C.Chieh Lee, "Improvement of an encryption scheme for binary images," Pakiston Journal of Information and Technology. Vol.2, no.2, 2003, pp.191-200, http://www.ansinet.org/