Secure Image Encryption using BCH Error Correcting Code and Digital Signature

Aye Myat Myat Mon Chit
University of Computer Studies, (Yangon)
ayemonchit@gmail.com

Abstract

The development of communication technology, security is a major concern. The widely used image in communication process is vital to protect important image data from unauthorized access. In this paper, the digital signature is added to the encoded image like additive noise. The encoding of the image is done by using Bose-Chaudhuri Hochquenghem (BCH) code. At the receiver end, after the decryption of the image, the digital signature can be used to verify the authenticity of the image. To generate the digital signature, the well-known Rivest-Shamir-Adleman (RSA) public key algorithm and MD5 Hash algorithm are used.

Key words: BCH error correcting code, Digital signature, RSA public key encryption algorithm, MD5 hash algorithm.

1. Introduction

Image plays an important role in many organizations; especially they are widely used for various kinds of security verification systems. The transmissions of images over the internet are actually not well protected. The traditional well known image encryption algorithms alone are not appropriate for the protection of important image data from forgery, repudiation and unauthorized access. In this paper, the digital signature of the original image is added to the encoded version of the original image.

The encoding of image is done by using BCH error correcting code. The MD5 algorithm and RSA public key algorithm are used for generating digital signature. Thus, the digital signature is treated like additive noise. At the receiver end, the digital signature is extracted from the encoded image (code word). The recovered digital signature can be used to verify the authenticity of transmitted image.

This paper is organized with seven sections. The first section is introduction of the paper. After that, the second section describes related work. Section three explains BCH error correcting code. The Digital Signature is in section four. The design of the system is included in section five. Then, the section six is the evaluation of the system. Finally, the conclusion of the system is in section seven.

2. Related Work

Several image encryption techniques have been proposed for authentication of image data.

Zalevsky et al, [1] has presented a novel technique for optical random encoding based on two binary phase mask. Each mask is itself random and the decoded information is obtained when two masks are joined together.

N. Takai and Y. Mifune, [2] have proposed Digital Watermarking of an image, for prevent from copying of an image by unauthorized access.

3. BCH Error Correcting Code

BCH codes are a class of linear and cyclic block codes that can be considered as a generalization of the Hamming codes, as they can be designed for any value of the error correction capability *(t)*. These codes are defined in the binary field *GF(2)*, and also in their non-binary version over the Galois field *GF(q)* [3].

For any positive integer $m \geq 3$ and $t \leq 2^m-1$, there exists a binary BCH code C_{BCH} (n,k) with the following properties:

Code length	: $n=2^m-1$
Number of parity bits	: $n-k \leq mt$
Minimum Hamming distance	: $d_{min} \geq 2t+1$
Error Correction Capability	: t error

These codes are able to correct any error pattern of size t or less, in a code vector of length (n),

$$n=2^m-1.$$

3.1. BCH Encoding

There are two methods to generate the BCH codeword, systematic manner and nonsystematic manner. In the nonsystematic manner, the decoding procedure is quite simple. This system used the nonsystematic manner [4].

BCH encoding can be done by multiplication of two polynomials:

1. message polynomial
2. generating polynomial

Let C be an (n,k) code over an field F with the generator polynomial:

$$g(x) = g_0 + g_1 x + \ldots + g_{r-1} x^{r-1} \text{ of degree } r=n-k.$$

If the message vector m is represented by a polynomial m(x) of degree k and m is represented by

$$m(x) = m_0 + m_1 x + \ldots + m_{k-1} x^{k-1}$$

then the following relation between m(x) and c(x) holds

$$c(x) = m(x)*g(x) \text{ (Code polynomial).}$$

3.2. BCH Decoding

The decoding procedure for BCH codes consists of three steps:

1) Compute the syndrome from the received codeword.
2) Find the error location polynomial from a set of equations derived from the syndrome.
3) Use the error location polynomial to identify error bits and correct them.

If r(x) is the received word and α^i , [i=1, 2, … , 2t], are the 2t consecutive roots of the generator polynomial, then the syndromes are given by

$$S_i = r(\alpha^i) \quad i = 1, 2, \ldots, 2t.$$

The syndromes are calculated either directly by evaluating r(x) at the roots using Horner's rule, or by first dividing r(x) by the minimal polynomials of the roots and then evaluating the remainder polynomials at the roots [6].

The Berlekamp algorithm is used for finding error location polynomial. The goal of Berlekamp algorithm is to find a polynomial $б^{(i+1)}$ of minimum degree at iteration i+1 that satisfies

$$\sum_{j=0}^{l_{i+1}} S_{k-j}\, \sigma_j^{(i+1)} = 0 \quad l_i < k < i+1$$

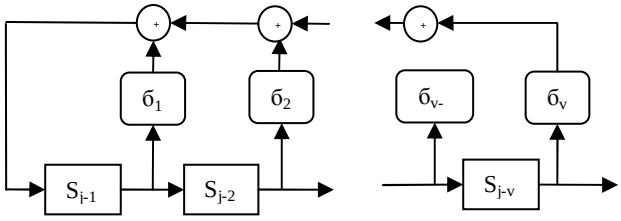
Denote the error locator polynomial at the i^{th} iteration as:

$$\sigma^{(i)}(x) = 1 + \sigma_1^{(i)} x + \cdots + \sigma_{l_i}^{(i)} x^{l_i}.$$

Define the discrepancy at iteration i as

$$d_i = S_{i+1} + S_i \sigma_1^{(i)} + \cdots + S_{i-l_i} \sigma_{l_i}^{(i)}.$$

Where $\{S_i\}$ are the syndromes of the received codeword. Figure 1 shows the iteration procedure.



j = v+1, v+2, … ,2v

Figure 1. Structure of linear feedback shift register

If $d_i=0$, then

$$\sigma^{(i+1)}(x) = \sigma^{(i)}(x) \qquad (l_{i+1} = l_i).$$

If $d_i \neq 0$, Let $б^{(m)}(x)$ be the solution at iteration m, such that $-1 < m < i$, $d_m \neq 0$, and $(m-\ell_m)$ is maximal. Then

$$\sigma^{(i+1)}(x) = \sigma^{(i)}(x) + d_i\, d_m^{-1}\, x^{i-m} \sigma^{(m)}(x)$$

$$l_{i+1} = \max(l_i, l_m + i - m)$$

With an initial value of i=0, the iterative procedure stops when either $i \leq \ell_{i+1} + t_d - 1$ or $i = 2t_d - 1$ is satisfied. The initial conditions of this algorithm are

$б^{-1}(x) = 1$, $\ell_{-1}=0$, $d_{-1}=0$
$б^{0}(x) = 1$, $\ell_0=0$, $d_0=S_1$ [5].

To correct the error r(x) xor e(x) = c(x). If there are no errors, then the syndromes all work out to zero. If the number of error is greater than t, then the decoding will be erroneous. These errors are called undetectable errors.

4. Digital Signature

The process of digitally signing starts by taking a mathematical summary (called a hash code) of the check. This hash code is a uniquely-identifying digital fingerprint of the check. If even a single bit of check changes, the hash code will dramatically change. The next step in creating a digital signature is to sign the hash code with private key. The signed hash code is then appended to the check.

The recipient of check can verify the hash code sent by the sender. At the same time, a new hash code can be created from the received check and compared with the original signed hash code. If the hash codes match, then the recipient has verified that the check has not been altered. The recipient also knows that only the sender could have sent the check because only the sender have the private key that signed the original hash code.

4.1. MD5 Hash Algorithm

MD5 is a message digest algorithm developed by Ron Rivest at MIT. The algorithm takes as input a message of arbitrary length and produces as output a 128-bit message digest of the input. This is mainly intended for digital signature applications where a large file must be compressed in a secure manner before being encrypted with a private key under a public key cryptosystem. The processing involves the following steps.

(1) Padding

The message is padded to ensure that its length in bits plus 64 is divisible by 512. That is, its length is congruent to 448 modulo 512. Padding is always performed even if the length of the message is already congruent to 448 modulo 512.

(2) Appending length

A 64-bit binary representation of the original length of the message is concatenated to the result of step(1). The extended message at this level will exactly be a multiple of 512-bits.

(3) Initialize the MD buffer

A four word buffer (A,B,C,D) used to compute the message digest. Here each A, B, C, D is a 32-bit register and they are initialized as the following values in hexadecimal. Low-order bytes are put first.

Word A: 01 23 45 67
Word B: 89 AB CD EF
Word C: FE DC BA 98
Word D: 76 54 32 10

(4) Process message in 16-word blocks

This is the heart of the algorithm, which includes four rounds of processing. The four rounds have similar structure but each uses different auxiliary functions F, G, H and I.

F(X, Y, Z) = (X ^ Y) v ((Not X) ^ Y)
G(X, Y, Z) = (X ^ Z) v (Y ^ (Not Z))
H(X, Y, Z) = (X xor Y xor Z)
I(X, Y, Z) = (Y xor (X v (Not Z)))

(5) Output

After all 512-bits blocks have been processed, the output is the 128-bit message digest. The generation of message digest is as shown in Figure 2. In this figure, IV and CV represent initial value and chaining variable respectively Let the expanded message be represented as a sequence of L 512-bit blocks $Y_0, Y_1, \ldots, Y_{L-1}$ [7].

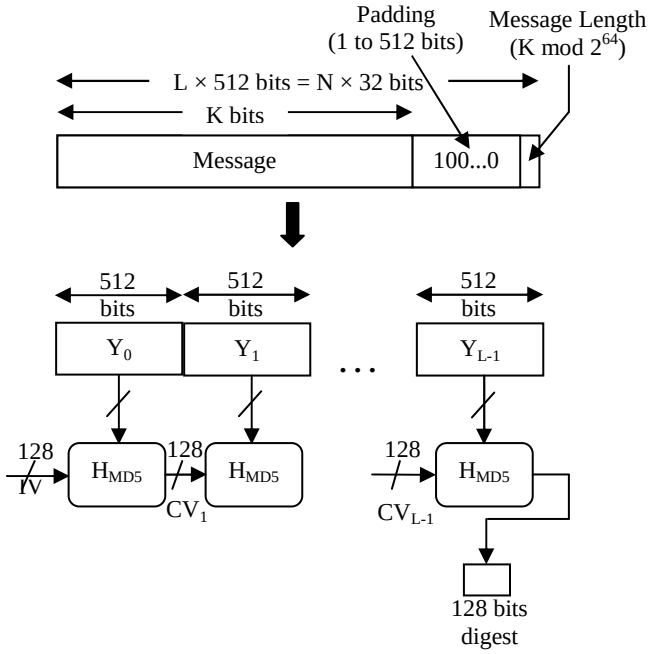


Figure 2. Generation of Message digest

4.2. RSA Cryptography

An RSA public-key and private-key pair can be generated by the following steps:

1. Generate a pair of large, random prime p and q.
2. Compute the modulus n as n=pq.
3. Select an odd public exponent e between 3 and and n-1 that is relatively prime to p-1 and q-1.
4. Compute the private exponent d from e, p and q.
5. Output (n, e) as the public key and (n, d) as the private key [8].

The encryption operation in the RSA cryptosystem is exponentiation to the e^{th} power modulo n:

$$c = \text{ENCRYPT}(m) = m^e \bmod n$$

The decryption operation is exponentiation to the d^{th} power modulo n:

$$m = \text{DECRYPT}(c) = c^d \bmod n.$$

5. System Design

The system design includes two parts: encryption part and decryption part.

At encryption part, the sender gives an input image to the system. The input image enters two parts. At encoding part, the image is encoded by using BCH encoding. At digital signature part, the MD5 algorithm takes image and generates hash code. The output hash code is encrypted by well-known RSA public key algorithm. Finally, the digital signature is XORed with the encoded version of image. Then, the encrypted image is sent to the receiver. The encryption system is shown in Figure 2.

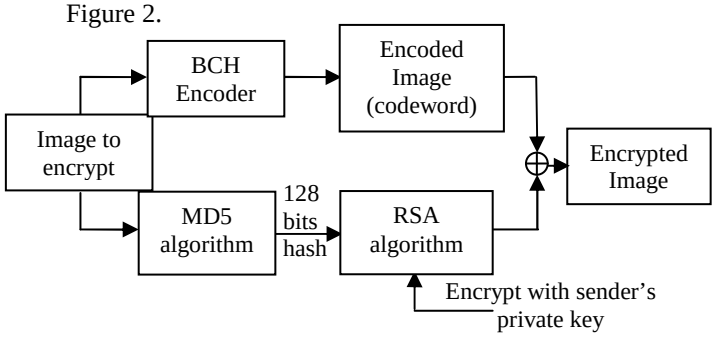


Figure 3. The block diagram of encryption system

For decryption part, the receiver gets the encrypted image from the sender. This image is passed through the BCH decoding to recover the original image (code word). The MD5 algorithm takes recover original image to get the receiving hash code. The original image is XORed with the encrypted image to recover image's signature. The recover signature is decrypted by RSA algorithm to get the signature hash code. At the receiver end, the digital signature can be used to verify the authenticity of the transmitted image. To verify for authentication, the receiving hash code is compared to the signature hash code. Figure 3 shows the decryption system.

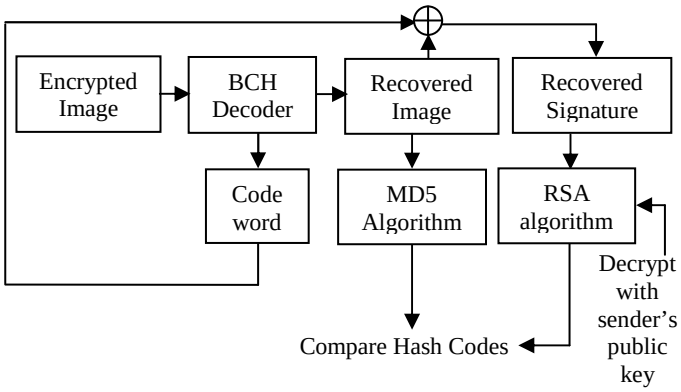


Figure 4. The block diagram of decryption system

6. Evaluation of the system

The class of BCH codes has the property that the number of correctable error can be specified. The system added the digital signature like additive noise. In this system, the BCH(255,215,5) class is used. If the error is greater than t=5, that error will not be corrected. In table 1, every valid value n has various k and t values. Figure 5 shows the GUI of encryption system. Figure 6 shows the GUI of decryption system.

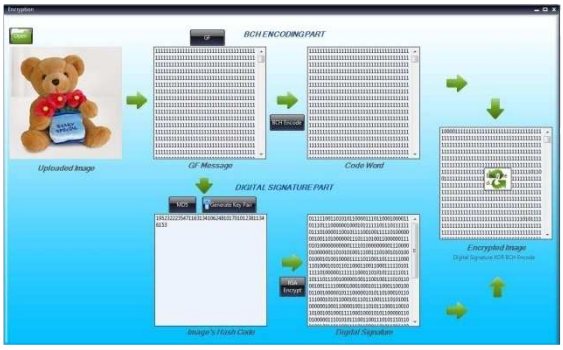


Figure 5. GUI of encryption system

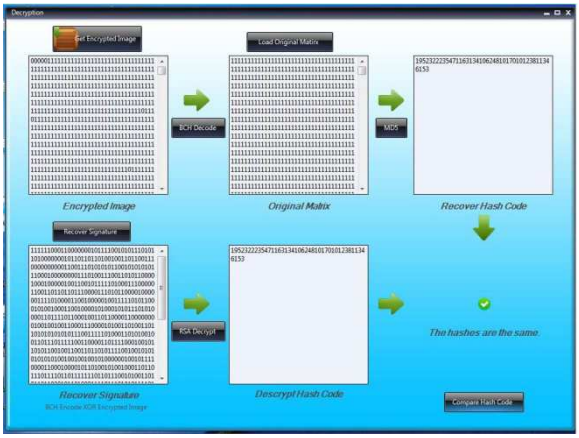


Figure 6. GUI of decryption system

When the message length k is decreased, the number of correctable error t is increased. The error control codes of different strengths are chosen for images of different sizes. The valid BCH code standards with correctable errors (t) are shown in Table 1.

Table 1. The valid BCH code standards with correctable errors (t)

Codeword Length (n)	Message Length (k)	Correctable Errors (t)
7	4	1
14	11,7,5	1,2,3
31	26,21,16,…,6	1,2,3,…,7
63	57,51,45,…,7	1,2,3,…15
127	120,113,106,…,8	1,2,3,…,31
255	247,239,231,…,9	1,2,3,…,63
511	502,493,484,…,10	1,2,3,…,121

7. Conclusion

This system demonstrated an image encryption system used BCH error correcting code and Digital Signature. The choice of the error control code is always such that the original image can be recovered. The digital signature is added to the encoded image in specific manner. This information can be protected to make the system more secure. The digital signature can be used to verify the authenticity of the transmitted image. The added advantage is that there is no need to transmit the keys separately. In this system, BCH(255,215,5) code is used. So, this system can correct at most t=5 errors. If n=255 and k<215 BCH code standard, it can correct at most 63 errors.

8. References

[1] Z. Zalevsky, D. Mendlovic, U. Kevy, G. Shabtay Novel technique for optical random encoding based on two binary phase mask.

[2] N. Takai, Y. Mifune, Digital Watermarking of an image.

[3] Jorge Castineira Moreira, Patrick Guy Farrell, Essential of Error Control Coding by John Wiley & Sons, Ltd.

[4] Zuzana Kuklova, Coding theory, cryptography and cryptographic protocols-exercises with solutions, Masaryk University Faculty of Informatics, given in 2006.

[5]Lei Zhou, Implementation of the Berlekamp-Massey algorithm and Peterson's algorithm in C Programming Language, University of Toronto.

[6] Jonathan Hong and Martin Vetterli, Senior Member, IEEE, Simple Algorithm for BCH Decoding.

[7] Janaka Deepakumara, Howard M. Heys and R, Venkatesan, FPGA Implementation of MD5 Hash Algorithm, by Faculty of Engineering and Applied Science (Memorial University of Newfouondland).

[8] Burt Kaliski, The Mathematics of the RSA Public-Key Cryptography, at RSA Laboratories.