COPYRIGHT © 2011 IJCIT, ISSN 2078-5828 (PRINT), ISSN 2218-5224 (ONLINE), VOLUME 01, ISSUE 02, MANUSCRIPT CODE: 110112

ASCII based Even-Odd Cryptography with Gray code and Image Steganography: A dimension in Data Security

Anupam Kumar Bairagi

Abstract—The message is the composition of some character. Every character of the message can be represented as an ASCII value, which is either even or odd. Depending on this evenness or oddness, the character is encrypted differently. This paper describes how such an even-odd encryption based on ASCII value is applied and how encrypted message converting by using Gray code and embedding with picture can secured the message and thus makes cryptanalyst's job difficult.

Index Terms — ASCII value, Encrytion, Gray code, Cryptanalyst.

1 INTRODUCTION

THIS paper presents an approach of ASCII based cryptography with LSB based image stenography for the security purpose of data transaction in the network and internet. Here encryption is applied to the even or odd ASCII value of the character which represents the data. A character in the plain text is always changed to the ASCII value and adding the key value with it getting the cipher text. This value is then converted to the equivalent binary number and substitutes these bits in the LSB position in each pixel which describe the image. In the opposite side, collect this bits from the image and converting this in equivalent decimal number which is the cipher text and subtracting the key value from it, we get the ASCII value of the plain text. Converting this ASCII value to the equivalent character representation, we get the original text (data). Normally a crypt-analyst can easily find out the key but in this approach a combination of two prime numbers is used for encryption.

2 LITERATURE REVIEW

2.1 Cryptography

Cryptography is the system where encryption and decryption techniques are used to the network and computer for the security of the data. Encryption means the change of original information (plain text) into another form by some operations (algorithm) and decryption means the techniques of getting the original information by some operations (algorithm) from the encrypted data (cipher text).

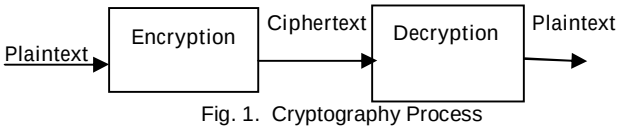

Fig. 1. Cryptography Process

- ***Anupam Kumar Bairagi*** *is with the Discipline of Computer Science and Engineering, Khulna Universty (www.ku.ac.bd), Khulna-9208, Bangladesh. E-mail: cse9620@gmail.com*

2.1.1 Private-key cryptography

In the private-key cryptography, the encryption and decryption on plaintext is done with the same key. As it is done with the same key, it is known both to the sender and receiver.

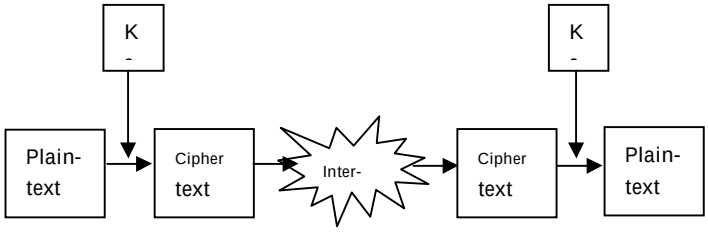

Fig. 2. The process of Private-Key Cryptography

2.1.2 Public-key cryptography

In the public-key cryptography, two different keys: a public key and a private key are used. The public key is known to all authorized users, but the private is known to one person- its owner. Here encryption is performed by public key and decryption is performed by private key.

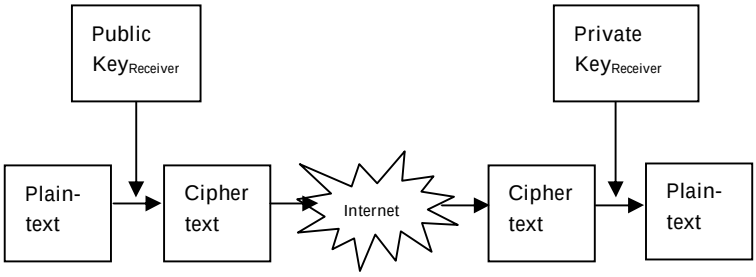

Fig. 3. The process of Public-Key Cryptography

2.1.3 Merkle-Hellman Knapsacks

Merkle and Hellman [1] developed an encryption algorithm based on the knapsack problem. The knapsack problem posed a set of positive integers and a target sum, with the

goal of finding a subset of the integers that summed to the target. The knapsack problem is NP-complete, implying that to solve it probably requires time exponential to the size of the problem: in this case, the number of integers.

The Merkle-Hellman encryption technique is a public-key cryptosystem. That is, each user has a public key, which can be distributed to anyone and a private key, which is kept secret. The public key is the set of integers of a knapsack problem (on a super increasing knapsack), the private key is a corresponding super increasing knapsack. The contribution of Merkle and Hellman was to design of a technique for converting a super increasing knapsack into a regular one. The trick is to change the numbers in a non-obvious but reversible way.

2.1.4 RSA Cryptosystem

In 1978 Ronald L. Rivest, A. Shamir and Leonard M. Adleman [2] proposed a method for realizing public key encryption as suggested by Deffie and Hellman [3]. RSA cryptographic system with public keys, based on modular exponentiation, which is considered today as a most reliable cryptographic system in the world. In the RSA public-key cryptosystem, a participant creates the public and private keys with the following procedure:

1. Select at random two larger prime numbers p and q i.e., the primes p and q might be, say, 100 decimal dig its each.
2. Compute n by the equation n = p*q
3. Select a small odd integer e that is relatively prime to where $\phi(n) = (p-1)*(q-1)$
4. Compute d as the multiplicative inverse of e, modulo i.e., e*d mod
5. Publish the pair P = (e,n) as RSA public key.
6. Keep secret the pair S = (d, n) as RSA secret key

The transformation of a message M associated with a public key P=(e, n) is C = E(M) = Me (mod n)

The transformation of a cipher text C associated with a secret key S = (d, n) is D(C) = Cd (mod n) = M

Though RSA technique uses the fact that it is easy to generate large prime numbers and multiply them, but it is extremely difficult to factor the product. The RSA technique is costly and relatively slow and thus limiting the throughput rate.

2.1.5 New Public Key Cryptosystem based upon Diophantine equation

In 1995, C. H. Lin, C. C. Chang and R. C. T. Lee [4] proposed a new public key cipher system based upon the Diophantine equation to solve the key management problem. In this public key cipher system, each user U uses the encryption algorithm E(Kp, P) and decryption algorithm D(Kr, C), where Kp is the public key, Kr is the private key of user U, P and C are plaintext and cipher text respectively. Each user publishes his encryption key by putting it on a public directory, while the decryption key is kept secret by him. Here is how it works for encryption: suppose that user X wants to send a message M to user Y. First, X finds the public encryption key, namely Kpy for Y from public directory. Then X encrypt the message M to C by C = E(Kpy, M) and sends C to Y. On receiving C, Y can decrypt it by computing M = D(Kry, C) and can read it. Since Kry is private for Y, no one else can perform this decryption process. Any one can send an encrypted message to Y but only Y can read it. Clearly, one requirement is that no one can figure out the private key from the corresponding public key. In practical purposes, the encryption and decryption algorithms E and D have to satisfy the following three requirements:

1. D(PRu, E(PKu, M)) = M
2. Neither of algorithm E and D needs much computing
3. To derive the associate PRu from the publicly known PKu is computationally infeasible.

2.2 Steganography

Although steganography is an ancient subject, the modern formulation of it is often given in terms of the prisoner's problem proposed by Simmons [5], where two inmates wish to communicate in secret to hatch an escape plan. All of their communication passes through a warden who will throw them in solitary confinement should she suspect any communication [6].

The warden, who is free to examine all communication exchanged between the inmates, can either be passive or active. A passive warden simply examines the communication to try and determine if it potentially contains secret information. If she suspects a communication to contain hidden information, a passive warden takes node of the detected covert communication, reports this to some outside party and lets the message through without blocking it. An active warden, on the other hand, will try to alter the communication with the suspected hidden information deliberately, in order to remove the information [7].

2.2.1 Different kinds of Steganography

Almost all digital file formats can be used for steganography, but the formats that are more suitable are those with a high degree of redundancy. Redundancy can be defined as the bits of an object that provide accuracy far greater than necessary for the object's use and display[8]. The redundant bits of an object are those bits that can be altered without the alternation being detected easily[7]. Main categories of file formats that can be used for steganography are shown in the Fig. 4.

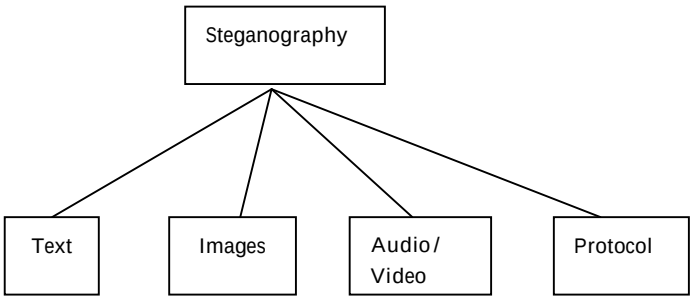

Fig. 4. The Categories of Steganography

Hiding information in text is historically the most impor-

tant method of steganography. An obvious method was to hide a secret message in every nth letter of every word of a text message. It is only since the beginning of the Internet and all the different digital file formats that is has decreased in importance Text steganography using digital files is not used very often since text files have a very small amount of redundant data.

Digital representation of image has large amount of redundant bits and for that images are the most popular cover objects for steganography.

To hide information in audio files similar techniques are used as for image files. One different technique unique to audio steganography is masking, which exploits the properties of the human ear to hide information unnoticeably. Although nearly equal to images in steganographic potential, the larger size of meaningful audio files makes them less popular to use than images [9].

Protocol steganography refers to the technique of embedding information within messages and network control protocols used in network transmission [10]. In the layers of the OSI network model there exist covert channels where steganography can be used [11].

2.2.2 Image Steganography

Image steganography technique can be divided into two groups: those in the Image Domain and those in the Transform Domain [12]. Image domain technique embed message in the intensity of the pixels directly, while for transform domain, images are first transformed and then the message is embedded in the image [13].

Image domain techniques encompasses bit-wise methods that apply bit insertion and noise manipulation and are sometimes characterized as "simple system" but in the transform domain involves the manipulation of algorithms and image transforms [14]. The image formats that are most suitable for image domain steganography are lossless and the techniques are typically dependent on the image format but many transform domain methods are independent of the image format and the embedded message may survive conversion between lossy and lossless compression [15]. Categories of images steganography are shown in the Fig. 5.

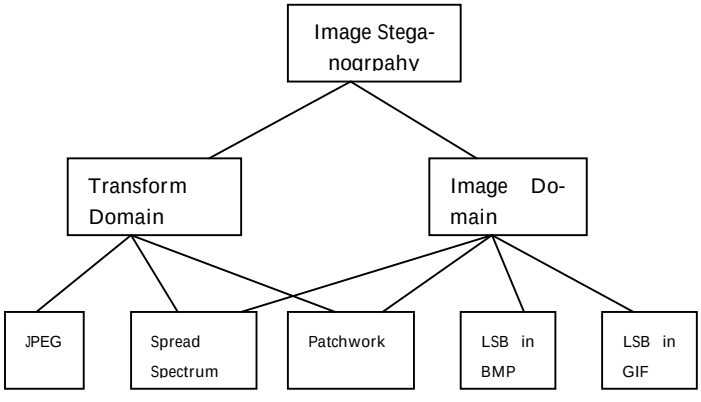

Fig. 5. Categories of image steganography

According to T. Morkel, J.H.P. Eloff, M.S. Olivier [16], the comparison of various image steganographic technique are shown in Table 1.

TABLE 1: COMPARISON OF IMAGE STEGANOGRAPHIC ALGORITHM

Characteristics	LSB in BMP	LSB in GIF	JPEG compression	Patchwork	Spread Spectrum
Invisibility	High	Medium	High	High	High
Payload capacity	High	Medium	Medium	Low	Medium
Robustness against statistical attacks	Low	Low	Medium	High	High
Robustness against image manipulation	Low	Low	Medium	High	Medium
Independent of file format	Low	Low	Low	High	High
Unsuspicious files	Low	Low	High	High	High

3 BACKGROUND

In the proposed approach the even and odd numbers representing the plain text (which are the ASCII values) are treated differently. We know that the sum of an even number and an odd number is odd and sum of two odd numbers is even. Considering this axioms, the two key values chosen should be odd and relatively prime.

If two odd values are chosen, the even and odd numbers representing the plaintext gets converted to odd and even respectively. So at the time of decryption little bit opposite task should be performed. If gn……..g2g1g0 denote a code word in the (n+1)st-bit Gray code and bn……b2b1b0 designate the corresponding binary number, where the subscripts 0 and n denote the least significant and most significant digits, respectively. Then the ith digit gi can be obtained from the corresponding binary number as follows:

$$g_n = b_n$$
$$g_i = b_i \oplus b_{i+1}, \quad 0 \leq i \leq n-1$$

To convert the Gray codes to binary number follow the process:

$$b_n = g_n$$
$$b_i = g_i \oplus b_{i+1}, \quad 0 \leq i \leq n-1$$

Least significant bit (LSB) insertion is to embedding information in a cover image. The least significant bit of some or all of the bytes inside an image is changed to a bit of the secret message. When using a 24-bit image, a bit of each of the red, green and blue color components can be used, since they are each represented by a byte. In other words, one can store 3 bits in each pixel. On average, only half of the bits in an image will need to be modified to hide a secret message using the maximum cover size [17]. Since there are 256 possible intensities of each primary color, changing the LSB

of a pixel results in small changes in the intensity of the colors. These changes cannot be perceived by the human eye. So message is successfully hidden.

4 PROPOSED METHOD

In this method a shared key pair (Ke, Ko) which are odd and prime are chosen and the ASCII value representing the character is tested for evenness. If it is even then Ke is added to the number and if it is not even then Ko is added to it.

When we add odd number with an even number results to an odd number and in the same fashion an odd number added with an odd number results an even number.

On the decrypting side each number is tested for evenness and if it is even then Ko is subtracted but if it in not even then Ke is subtracted.

4.1 Forwarding process

1. Choose a pair of key value (Ke, Ko) which are primes
2. M = ASCII value of a character in the message
3. If M mod 2 = 0 Then
   C = M + Ke
   Else
   C = M + Ko
4. Convert C into equivalent binary number
5. Convert the binary to the Gray code
6. Substitute this bits in the LSB position of the image pixel
7. Send the image to the receiver

In this algorithm, M is the ASCII value of plaintext and C is the number representing the cipher text. The two numbers Ke and Ko are the components of the shared key. The term "mod" indicates that the remainder obtained by dividing M by 2 is used for comparing evenness.

4.2 Backwarding Process

1. Extract the bits from the image and group by 8 bits
2. Convert the Gray code to the equivalent binary num ber separately
3. Convert binary number to equivalent decimal number which is C
4. If C mod 2 = 0 then
   M = C – Ko
   Else
   M = C - Ke
5. Convert M (which is the ASCII value) to the equivalent Character
6. Putting this character we will get the original message

In this algorithm M and C are the same as before. The two numbers Ke and Ko are the components of the same shared key.

4.3 Example

Suppose that the shared key is the pair (11, 19). The sender needs to send the message "abc". The ASCII values of a, b, c are 97, 98, 99 respectively. Then encryption process calculates 97 mod 2 as 1 so C = 97+ 19 = 116. Similarly 98 mod 2 as 0 so C = 98+11 = 109 and 99 mod 2 as 1 so C = 99+19=118. This numbers are the cipher text. Now converting this numbers to the equivalent binary numbers, we get 01110100, 01101101 and 01110110 respectively. Now converting this numbers to the gray code we get 01001110, 01011011 and 01001101 respectively. Now a grid of 8 pixels of a 24-bit image can hold all this gray code in the LSB position. That can be like:

(00101000	01100111	01001101)
(01101001	01100010	01000101)
(00111000	01101110	01010100)
(00101011	01110010	01101001)
(00101001	01000010	01101111)
(00111011	01000110	01001101)
(01101000	01100010	01100101)
(01111011	01101010	01011101)

This image is send to the receiver. In the receiving end, collect the LSB bits from the image and group it by 8 bits i.e. 01110100 01011011 01001101. Now convert this in the binary number and that is 01110100, 01101101 and 01110110 respectively. Convert this binary number to equivalent decimal number which are 116, 109 and 118 respectively and it is the cipher text. The decryption process calculates 116 mod 2 as 0 so M = 116 – 19 = 97. Similarly 109 mod 2 as 1 so M = 109 – 11 = 98 and 118 mod 2 as 0 so M = 118 – 19 = 99. This numbers are the ASCII value of the message. Now converting this values to the equivalent character we get the message "abc" that is the original message.

5 SECURITY CONCERN

There is nothing common in between two numbers rather than both of them are odd prime number. If one number is known to the adversary, he cannot deduce the other number. In case of a 32 bit machine (long integer of 32 bits), each number can be 32 bits long. If one number is fixed, the other number can be any one of 232 possibilities and the first number can be one of 232 possibilities. So the number of possible alternatives becomes 232*232 = 264. Trying possible alternatives are not so easy.

Further the steganography itself with hide the converted information in a secured way so that human eye cannot easily detect it.

6 CONCLUSION

Though the cipher text can be broken, this even-odd based cryptography differentiates the encryption scheme to be applied based on the evenness or oddness of the ASCII value of the character. Further this encrypted message converted by using Gray code embedding with picture can reduce the security tension. Further works including the development of the embedding channel of converted data which can secured the process greatly.

REFERENCES

[1] R.C. Merkle and M. Hellman, 1978. Hiding information and signa-

tures in trap door knapsacks, IEEE Trans. Inform. Theory, vol 24, pp 525-530.

[2] R. L. Rivest, A Shamir and L. Adleman, 1978. A method for obtaining digital signatures and public-key cryptosystems, Communications of the Association for Computing Machinery, vol 21, no. 2, pp 120-126

[3] W. Diffie and M. E. Hellman, Nov 1976. New direction in cryptography, IEEE Transactions on Information Teory, vol. IT-22, no. 6, pp. 644-654.K. Elissa, "An Overview of Decision Theory", unpublished. (Unplublished manuscript)

[4] C. H. Lin, C. C. Chang and R. C. T. Lee, January 1995. A new Public-Key Cipher System Based Upon Diophantine Equations, IEEE Trans. On Computers, vol. 44, no. 1

[5] G. Simmons, The prisoners problem and the subliminal channel, CRYPTO, 1983

[6] R. Chandramouli, M. Kharrazi and N. Memon, Image steganography and steganalysis: Concepts and Practice, Proceedings of the 2nd International Workshop on digital Watermarking, October 2003.

[7] R. J. Anderson and F. A. P. Peticolas, On the limits of steganography, IEEE Journal of selected Areas in Communications, May 1998

[8] H 8. D. L. Currie and C.E. Irvine, Surmounting the effects of lossy compression on Steganography, 19th National Information Systems Security Conference, 1996

[9] D. Artz, Digital Steganography: Hiding Data within Data, IEEE Internet Computing Journal, June 2001

[10] K. Ahsan and D. Kundur, Practical Data hiding in TCP/IP, Proceedings of the workshop on Multimedia security at ACM Multimedia, 2002

[11] T. Handel and M. Sandford, Hiding data in the OSI network model, Proceedings of the 1st International Workshop on Information Hiding, June 1996

[12] J. Silman, Steganography and Steganalysis: An Overview, SANS Institute, 2001

[13] Y.K. Lee and L.H. Chen, High capacity image steganographic model, Visual Image Signal Processing, 147: 03, June 2000

[14] N.F. Johnson and S. Jajodia, Steganalysis of Images Created Using current Steganography Software, Proceedings of the 2nd Information Hiding Workshop, April 1998

[15] S. Vendatraman, A. Abraham and M. Paprzycki, Significance of Steganogrpahy on Data Security, Proceedings of the International Conference on Information Technology: Coding and Computing, 2004

[16] T. Morkel, J.H.P. Eloff, M.S. Olivier, An Overview of Image Steganography, ICSA Research Group, Department of Computer Science, University of Pretoria

[17] R. Krenn, Steganography and Steganalysis, www.krenn.nl /univ/cry/steg/article.pdf

Anupam Kumar Bairagi has been serving as a Lecturer in the Discipline of Computer Science and Engineering (CSE), Khulna University, Khulna-9208, Bangladesh. He joined at KU in November 2009. Prior to his joining Khulna University he taught in Khulna Polytechnic Institute, Khulna as an instructor in the department of computer technology for about five years. He has five published books for diploma level students on computer technology.