A NEW TECHNIQUE TO ENCRYPT A CODE WORD

Arnab Chowdhury
Department of Electronics and Communication Engineering
Kalyani Government Engineering College
Kalyani, Nadia, West Bengal, India
chowdhury.arnab008@gmail.com

Abstract

The principal goal of designing any encryption algorithm is to hide the original message and send the non-readable text message to the receiver so that secret message communication can take place over the web. The strength of an encryption algorithm depends on the difficulty of cracking the original message. A number of symmetric and asymmetric key encryption algorithms like DES, TRIPLE DES, AES, BLOWFISH, RSA has been developed to provide greater security affects one over the other. I have described a protocol in this paper, which fits the technique to realistic communication environments, and extend the security and the range of applications of the technique. The basic motive behind the new approach is not to use a 'key'. Encryption is done by the use of the logic operations amongst the bits of the code word. This can be said as "security through obscurity" where the shared secret is not an encryption key, but the encryption algorithm itself. This algorithm can prove to be an efficient method of encryption and can reduce the complexity of a system that encrypts or decrypts a message. Analysis has been done which shows the advantages and the limitations of this algorithm. With this new approach we are implementing a technique to enhance the security level and to further reduce the time for encryption and decryption.

Keywords – Cryptography, Symmetric key cryptography, Asymmetric key cryptography, DES, AES, Triple DES, Blowfish, RSA, DBNS and TBNS

1. INTRODUCTION

Security is the main concern in today's world and securing data from unauthorized access is very important. Different techniques should be used to protect confidential data from unauthorized access as each type of data has its own features.

Cryptography refers to the science and art of transforming messages to make them secure and immune to attacks. Cryptography plays a very vital role in keeping the message safe as the data is in transit. It ensures that the message being sent at one end remains confidential and should be received only by the intended receiver at the other end. Cryptography converts the original message in to non-readable format and sends the message over an insecure channel. The people who are unauthorized to read the message try to break the non-readable message but it is hard to do it so. The authorized person has the capability to convert the non-readable message to readable one.

The original message or the actual message that the person wishes to communicate with the other is defined as Plain Text. The message that cannot be understood by anyone or meaningless message is what we call as Cipher Text. Encryption is the process of converting plaintext into cipher text with a key. A Key is a numeric or alpha numeric text or may be a special symbol. A decryption is a reverse process of encryption in which original message is retrieved from the cipher text. Encryption takes place at the sender end and Decryption takes place at the receiver end.

Figure 1 shows the encryption/decryption process of a plaintext message. The input to the encryption process is plain text and that of decryption process is cipher text. First the plaintext is passed through the encryption algorithm which encrypts the plaintext using a key and then the produced cipher text is transmitted.

At the end of decryption, the input cipher text is passed through the decryption algorithm which decrypts the cipher text using the same key as that of encryption. Finally we get the original plaintext message.

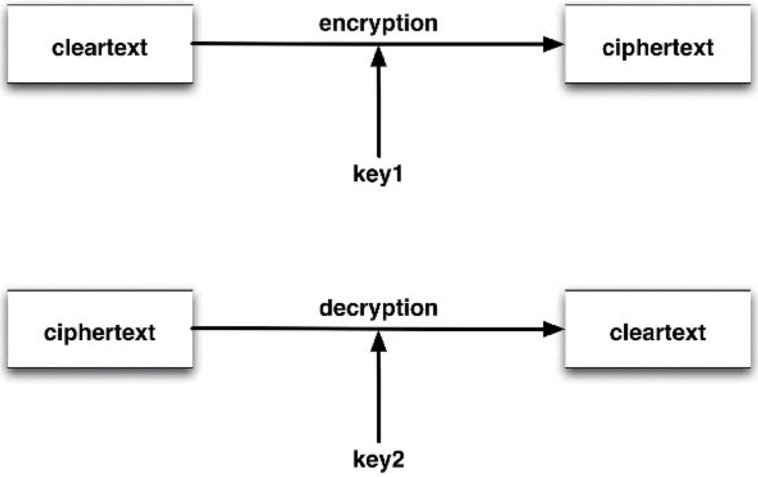

Figure 1: Encryption and Decryption Process

Goals of Cryptography

Cryptography provides a number of security goals to ensure the privacy of data, non-alteration of data and so on. Due to the great security advantages of cryptography it is widely used today. Following are the various goals of cryptography [1].

- **Confidentiality -** Information in computer is transmitted and has to be accessed only by the authorized party.

- **Authentication -** The information received by any system has to check the identity of the sender that whether the information is arriving from a authorized person or a false identity.
- **Integrity -** Only the authorized party is allowed to modify the transmitted information. No one in between the sender and receiver are allowed to alter the given message.
- **Non Repudiation -** Ensures neither the sender, nor the receiver of message can deny the transmission.
- **Access Control -** Only the authorized parties are able to access the given information.

2. PRESENT ALGORITHMS ON CRYPTOGRAPHY

Before discussing about the algorithms, let us define some terms related to cryptography.

Plaintext and Ciphertext – The original message before being transformed is called plaintext. After the message is transformed, it is called ciphertext. An encryption algorithm transforms the plaintext into ciphertext; a decryption algorithm transforms the ciphertext back into plaintext. The sender (Alice) uses an encryption algorithm, and the receiver uses a decryption algorithm.

Key – A key is a number (or a set of numbers) that the cipher, as an algorithm, operates on.

Encryption algorithms can be classified into two broad categories - Symmetric key Cryptography and Asymmetric Key Cryptography.

• SYMMETRIC KEY CRYPTOGRAPHY

In symmetric Cryptography the key used for encryption is similar to the key used in decryption. Thus the key distribution has to be made prior to the transmission of information. The key plays a very important role in symmetric cryptography since their security directly depends on the nature of key i.e. the key length etc. There are various symmetric key algorithms such as DES, Triple DES (TDES), AES, RC4, RC6, and Blowfish [2].

❖ DATA ENCRYPTION STANDARD (DES)

DES (Data Encryption Standard) was the first encryption standard to be recommended by NIST (National Institute of Standards and Technology). It was developed by an IBM team around 1974 and adopted as a national standard in 1997.

DES is a 64-bit block cipher under 56-bit key. The algorithm processes with an initial permutation, sixteen rounds block cipher and a final permutation. DES application is very popular in commercial, military, and other domains in the last decades [4]. Although the DES standard is public, the design criteria used are classified. There has been considerable controversy over the design, particularly in the choice of a 56-bit key [5].

❖ TRIPLE DES (TDES)

The triple DES (3DES) algorithm was needed as a replacement for DES due to advances in key searching [6]. TDES uses three round message. This provides TDES as a strongest encryption algorithm since it is extremely hard to break 2^168 possible combinations. Another option is to use two different keys for the encryption algorithm. This reduces the memory requirement of keys in TDES. The disadvantage of this algorithm is that it is too time consuming.

❖ ADVANCED ENCRYPTION STANDARD (AES)

AES was developed by two scientists Joan and Vincent Rijmen in 2000. AES uses the Rijndael block cipher. Rijndael key and block length can be 128, 192 or 256-bits. If both the key-length and block length are 128-bit, Rijndael will perform 9 processing rounds. If the block or key is 192-bit, it performs 11 processing rounds. If either is 256-bit, Rijndael performs 13 processing rounds [7].

Each processing round involves four steps:

1. Substitute bytes – Uses an S-box to perform a byte by byte substitution of the block

2. Shift rows – A simple permutation

3. Mix column – A substitution method where data in each column from the shift row step is multiplied by the algorithm's matrix

4. Add round key – The key for the processing round is XORed with the data.

❖ BLOWFISH

Bruce Schneier designed blowfish in 1993 as a fast, free alternative to existing encryption algorithms. Since then it has been analysed considerably, and it is slowly gaining acceptance as a strong encryption algorithm. The Blowfish algorithm has many advantages. It is suitable and efficient for hardware implementation and no license is required [8]. The elementary operators of Blowfish algorithm include table lookup, addition and XOR. The table includes four S-boxes and a P-array. Blowfish is a cipher based on Feistel rounds, and the design of the F-function used amounts to a simplification of the principles used in DES to provide the same security with greater speed and efficiency in software. Blowfish is a 64 bit block cipher and is suggested as a replacement for DES. Blowfish is a fast algorithm and can encrypt data on 32-bit microprocessors at a rate of one byte every 26 clock cycles. The algorithm is compact and can run in less than 5K of memory [9].

• ASYMMETRIC KEY CRYPTOGRAPHY

In Asymmetric Cryptography, two different keys are used for encryption and decryption - Public and Private. The public key is meant for general use so it is available to anyone on the network. Anyone who wants to encrypt the plaintext should know the Public Key of receiver. Only the authorized person can be able to decrypt the cipher text through his own private key. Private Key is kept secret from the outside world.

Symmetric Encryption Algorithm runs faster as compared to Asymmetric key algorithms. Also the memory requirement of Symmetric algorithm is lesser as compared to asymmetric [3].

Many cryptographic algorithms have already been proposed and implemented to provide security to the user that his/her message would remain safe at the time of communication over the web. But now a days hacking has become a common practice in society which made such cryptographic algorithms no longer safe.

❖ RSA ALGORITHM

RSA is an algorithm for public-key cryptography that is based on the presumed difficulty of factoring large integers, the factoring problem. RSA stands for Ron Rivest, Adi Shamir and Leonard Adleman, who first publicly described the algorithm in 1977. Clifford Cocks, an English mathematician, had developed an equivalent system in 1973, but it wasn't declassified until 1997.

A user of RSA creates and then publishes the product of two large prime numbers, along with an auxiliary value, as their public key. The prime factors must be kept secret. Anyone can use the public key to encrypt a message, but with currently published methods, if the public key is large enough, only someone with knowledge of the prime factors can feasibly decode the message. Whether breaking RSA encryption is as hard as factoring is an open question known as the RSA problem.

Let us now discuss about some of the ciphers. We divide traditional ciphers into two broad categories:-

i. Substitution ciphers - A substitution cipher substitutes one symbol with another. If the symbols in the plain text are alphabetic character we replace one character with another.
ii. Transposition ciphers - In transposition ciphers there is no substitution instead their location alters. A character in the first position may appear in the different position in the cipher text.

The modern round ciphers are bit oriented. They are of the following types -

i. XOR cipher - It uses the exclusive-or operation between two data inputs; plaintext as the first and the key as the second.
ii. Rotation cipher - This cipher rotates the input bit left to right. It may be keyed or keyless.
iii. S-box - An S-box (substitution box) parallels the traditional substitution cipher for characters. The S-box is normally keyless and is used as an intermediate stage of encryption and decryption.

iv. P-box - A P-box (transposition box) for bits parallels traditional transposition cipher for characters. It performs a transposition at bit level; it transposes bit.

3. PROPOSED ALGORTIHM

I have discussed about the present algorithms. All the encryption algorithms depend on the concept of 'keys'. Keys can be private or public. Now this paper highlights on a new concept. Encryption of a message can be done without the use of keys. It may seem that this algorithm is nothing but simple encoding of message bits. But analysis of it will prove the efficiency of the algorithm in terms of security. Simple logic operations that are very well known to us can be very efficient tools of encryption. The logic operations that are used in digital circuits are OR, AND, NOR, NAND, XOR and XNOR. This can be said as "security through obscurity" where the shared secret is not an encryption key, but the encryption algorithm itself.

Now let us discuss the algorithm in details.

ENCRYPTION

The encryption process is shown in the following steps –

1. The message that is to be transmitted is provided to the system in binary code word.
2. Now the sender (Alice) provides the order or sets the preference of the logic operations. The program that I have designed denotes OR, AND, NOR, NAND, XOR and XNOR as 1, 2, 3, 4, 5 and 6. Thus Alice may simply give an input as 2, 1, 4, 5, 3 and 6.
3. Since there are 6 logic operations, two case may arise. They are –
    a. The length of the message can be less than 6. In that case, Alice provides only that many orders as that of the length of the code word.
    b. The second case is much more critical. The length of the code word is greater than 6 in this case. Alice sets the preference of the operations. After the counter reaches 6, the system may ask for a new set of preference or simply continue with the older one.

    In both the cases, the order is stored in an array.
4. Now the system performs the encryption process in the following manner – The logic operations are performed among the consecutive bits of the message.

    For an order of operation as 2, 1, 4, 5, 3 and 6 and message as 101110; AND operation is performed between the 0th bit and 1st bit, OR is performed between 1st bit and 2nd bit and likewise the process goes on until XNOR operation is performed between the 5th bit and 0th bit.
5. Now Alice can transmit the ciphertext. The algorithm demands two more things. The transmitter must send the array of order of operation and the difference of the sum of the weights of the

message bits. For example, for the message 1011, the weights of the 1's are 1, 2 and 4. Since the message can be predicted from the sum of the weights and thus encryption may fail, it is transmitted in a different manner. The sum of the 1's are calculated at both the even and odd positions. Then the difference is found out between them. For the above shown message, the number is 5 (6-1=5). Thus an outsider (Eve) can never understand why such a number is transmitted.

The reason behind the transmission of the array and the difference of the sum of the weights is explained in the later part of the paper.

The above steps explain the encryption process.

Let me explain the encryption process by taking the help of an example.

1. Let the message be 1010111. The sum of the weights by using the algorithm is 14. Let Alice set the preference as 3, 4, 6, 1, 5 and 2, that means that NOR will be the first operation that will be performed and AND will be last operation. Again let us consider that the same order is repeated. The operation of the bits is shown in the figure below.

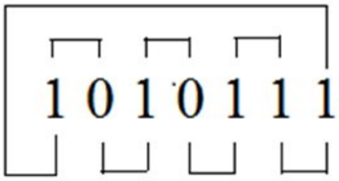

Figure 2: Operation between the Bits of the Message

2. The ciphertext that will be obtained is 0011000. This ciphertext can be transmitted along with the array of the order of operation as [3, 4, 6, 1, 5, 2, 3].

DECRYPTION

The decryption process is much more complex with respect to the encryption process, since a straight algorithm cannot decrypt or decode the ciphertext to the original message.

Before discussing about the process, let us discuss about the need of transmission of the array and the sum of the weights.

Without knowing the order, it is not possible for the receiver (Bob) to obtain the plaintext. For each encryption of a message, there can be other code words for which the same encrypted code word is obtained. Thus it will be difficult to obtain the original code word. For this reason, if the system knows the difference of the sum of the weights, it can predict the correct code word.

The decryption algorithm mainly depends on finding the cases for each and every bit. Let us consider the ciphertext that is obtained in the above example (0011000). From the array, Bob knows about the order of operation. In this case, the order is NOR, NAND, XNOR, OR, XOR, AND and NOR. The process is described in the following steps –

1. Considering the 0th bit, we can conclude that that the 1st and 0th bit of the original message can be 01, 10 or 11.
2. Considering the 1st bit, we can conclude that that the 2nd and 1st bit of the original message can be 11. Now comparing between the above cases, the 2nd, 1st and 0th bit of the original message is either 110 or 111.
3. Considering the 2nd bit, we can conclude that that the 3rd and 2nd bit of the original message can be 10 or 01. Now comparing between the above cases, the 3rd, 2nd, 1st and 0th bit of the original message is either 0110 or 0111.
4. Considering the 3rd bit, we can conclude that that the 4th and 3rd bit of the original message can be 10, 11 or 01. Now comparing between the above cases, the 4th, 3rd, 2nd, 1st and 0th bit of the original message is either 10110 or 10111.
5. Likewise if we continue with the same procedure, the two probable outputs will be 1010110 or 1010111.
6. Now to obtain the correct code word, Bob compares the difference of the sum of the weights of both the probable code words using the prescribed format with the transmitted sum. The correct message thus obtained is 10101111 (difference of the sum of the weights is 14).

4. LIMITATIONS OF THE ALGORITHM AND THEIR POSSIBLE SOLUTIONS

I have already discussed in details about the encryption and decryption process. Now improving the security and reducing the complexity of the algorithms is the main deal in cryptography. If we analyse the above algorithm, some flaws can be found out. They are -

- Although the encryption process is simple and the time complexity depends on the length of the message, the decryption process is much more complex since it depends on finding the cases for which we can obtain a particular bit in the ciphertext. The complexity increases as the length of the message and thus the ciphertext increases.
- The probability that an intruder (Eve) finds out the weight of the message is very less, but Eve may be able to decode the array by trial and error process.
- For small length of the message, there is a chance of outsider attack since Eve can decode the message by trial and error process.
- A case may arise when the encrypted message is exactly the same as the original message.

The second, third and fourth limitation can be removed very easily. The message can be encrypted through a round of same operations, but that may be prove to be complex or the ciphertext obtained after encryption can be again operated by any of the logic operations (selected by Alice) with alternating bits of 0s and 1s, 0th bit starting as 0. The logic operation is again stored in the array as the MSB. The receiver reads the MSB, and again predicts the code word. In this case, the difference of the sum of the weights is to be transmitted. Then it follows the same decryption process. Thus the probability of decoding the ciphertext is reduced.

To nullify the effect of the third limitation, a number of zeros can be appended in the message for a good encryption. For example, if the length of the code word is 2, 10 zeros can be appended to it to make the length as 12. While decryption, the receiver considers the last two bits by the help of the difference of the sum of the weights.

5. ADVANTAGES

Comparing with the other present algorithms, the proposed algorithm can prove to be much more efficient. In case of DES algorithm, the keys are too short. In case of RSA algorithm, the value of P (message) must be less than the value of n (public key). If P is a large number, the plaintext needs to be divided into blocks to make P less than n. The proposed algorithm does not have these restrictions since it does not use any key to encrypt a message.

For a 6 bit message, the time taken to encrypt is only 0.031s and the execution time does not increase linearly with the increase in the length of the message.

The advantages of the proposed algorithm are –

- Simplicity
- Security
- Efficiency
- Robustness
- Availability
- Integration
- Distribution
- Time efficiency
- Flexibility

If we analyse DES, RSA and the proposed algorithm with respect to some parameters, we can observe the following differences as shown in the table below.

Table 1: Performance Analysis of DES, RSA & the Algorithm Proposed

Features	**DES**	**RSA**	**Proposed Algorithm**
Key Used	Same key is used for encryption and decryption purpose	Different keys are used for encryption and decryption purpose	No such key is required for encryption and decryption purpose
Scalability	It is scalable algorithm due to varying the key size and block size.	No scalability occurs	Generally not scalable but can be made with a little modification
Avalanche Effect	Not much effected	More effected	Not much effected
Power Consumption	Low	High	High
Throughput	Very high	Low	Low
Confidentiality	High	Low	Moderate

6. APPLICATIONS

It is very well known to us how cryptography is important in terms of security. In the modern world, we cannot even think of transmitting data or image or any other information without encrypting the message. To explain the application of the algorithm, let us illustrate with the help of a case study.

CASE STUDY

We all know the utilisation of multi base number system like Double Base Number System and Triple Base Number System. Two of the most important advantages of DBNS are –

[1] In binary number representation, each bit has approximately 0.5 probability of being 1.But in DBNS, the number of bits that are logic 1 in the tabular representation could be much less. Effectively, we can reduce the number of 0→1 and 1→0 transitions, thus saving power.

[2] It can be shown that expected number of bits that are 'turned on' in a DBNS representation of integer is O[log x/(log log x)], which is significantly lower than the corresponding number in the positional binary system, O(log x).

As an example, consider the integer $2^{215;}$ in binary system, number of '1's ≈ 100 & in DBNS, number of '1's ≈ 30.

Now let us consider a number (suppose 124416) that is to be transmitted. This number is first represented in DBNS form using any of the algorithms that are present (124416 can be represented in DBNS as $2^9 3^5$). Now the indices of the bases 2 and 3 can be encrypted using the above stated encryption algorithm and then transmitted.

This can prove to be extremely advantageous in the field of Digital Signal Processing, Cryptography, and Communication etc.

7. CONCLUSION

This paper gives a detailed study of the present encryption algorithms and a new approach to cryptography which has its own advantages. The proposed algorithm can achieve better results in terms of parameters such as Encryption time, Decryption time and Throughput. The striking feature of the proposed encryption algorithm is that no key is used for encryption and for the same input plaintext the cipher text generated at each time will be different. This is because every time Alice can set his own preference of logic operations. The advantage of different cipher text generated for the same input is it will greatly enhance the security aspect of the algorithm.

The second biggest advantage of this approach is that it is less time consuming as compared to other encryption algorithms. I am still working on the programs that can implement the above proposed algorithms.

8. ACKNOWLEDGEMENT

Intellectual guidance is the royal road to great success. This project owes its existence to many people without whose co-operation and efforts the realizations would not have been possible.

I would like to articulate our profound gratitude and indebtedness to all the professors of my department who have always been a constant motivation and guiding factor throughout the project time in and out as well.

I am grateful to my parent who always supported me throughout the entire work. I would also like to mention the help of my friends, for their cooperation and constantly rendered assistance.

REFERENCES

1) O.P Verma, Ritu Agarwal, Dhiraj Dafouti and Shobha Tyagi, “Performance Analysis of Data Encryption Algorithms”, IEEE Delhi Technological University India, 2011.
2) Diaa Salama, Abdul. Elminaam, Hatem Mohamed, Abdul Kader and Mohie Mohamed Hadhoud, “Performance Evaluation of Symmetric Encryption Algorithms”, International Journal of Computer Science and Network Security, vol.8 No.12, December 2008.

3) Ketu File white papers, “Symmetric vs. Asymmetric Encryption”, a division of Midwest Research Corporation.

4) “Data Encryption Standard,” Federal Information Processing Standards Publication No. 46, National Bureau of Standards, January 15, 1977.

5) Tingyuan Nie and Teng Zhang, “A Study of DES and Blowfish Encryption Algorithm”, IEEE, 2009.

6) Aamer Nadeem and Dr M. Younus Javed, “A Performance Comparison of Data Encryption Algorithms”, IEEE, 2005.

7) Vishwa Gupta, “International Journal of Advanced Research in Computer Science and Software Engineering”, Vol. 2, Issue 1, January 2012, pp. 1-3.

8) Dripto Chatterjee, Joyshree Nath, Suvadeep Dasgupta, Asoke Nath “A new Symmetric key Cryptography Algorithm using extended MSA method: DJSA symmetric key algorithm” published in 2011 International Conference on Communication Systems and Network Technologies, 978-0-7695-4437-3/11.

9) Symmetric key cryptography using random key generator, A. Nath, S. Ghosh, M. A. Mallik, Proceedings of International conference on SAM-2010 held at Las Vegas (USA) 12-15 July, 2010, Vol-2, P-239-244.