IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol. 8, Issue.4, Apr 2020 E-ISSN: 2347-2693

A New Bit Level Positional Encryption Algorithm (NBLPEA ver-1)

Asoke Nath[1*], Sankar Das[2], Oishi Mazumder[3], Adrija Saha[4], Monimoy Ghosh[5]

[1,2,3,45] Department of Computer Science, St. Xavier's College (Autonomous), Kolkata, India

**Corresponding Author: asokejoy1@gmail.com*

DOI: https://doi.org/10.26438/ijcse/v8i4.167172 | Available online at: www.ijcseonline.org

Received: 21/Jan/2020, Accepted: 19/Feb/2020, Published: 30/Apr/2020

Abstract- In the world of information we live in, cyber security is of paramount importance. Despite communicating through secure channels, there might be breaches that might be targeted by attackers to extract valuable data. Designing and betterment of these symmetric and asymmetric algorithms remain an open challenge to all. The Bitwise Positional Encryption Algorithm is a completely new idea in this field. This simple encryption technique provides strong protection against any kind of attack, be it brute force or statistical attacks. This multilevel encryption algorithm employs several bit-level encryption procedures and matrix transposition operations. It takes into consideration positional parameters of bits, which is further shuffled randomly and hence is impossible for the intruder to decrypt without knowing the key as well as the exact method. The key here should be a shared parameter, for this is an instance of symmetric encryption. The algorithm works on bit-level so it is not possible to break easily. By this algorithm, one can encrypt any kind of file with extensions like .txt, .doc, .jpg, .png and the like. The present method can be used for encryption of any confidential messages like OTP (One Time Password), ATM transactions etc. It is not possible to get back the original plain text file if there is one change in bits in encrypted text. The testing is done on almost all types of files and it was found that the method is working satisfactorily.

Keywords: Plain text, ciphertext, encryption, decryption, key, transposition, positional extraction.

I. INTRODUCTION

In simple terms, encryption is the process of making information unreadable by unauthorized persons. The process may be manual, mechanical, or electronic. Encryption is essentially important because it secures data and information from unauthorized access and thus maintains confidentiality. The confidentiality of data is an issue of growing concern over the past decade. The rapid evolution of Internet technology has led to millions of transactions per second across the world. In this scenario, the study and enhancement of cryptographic algorithms are quite indispensable. Apart from providing network security, cryptography also assists in disk encryption, anonymous communication, digital signatures, private auctions, implementation of cryptocurrency – just to mention a few. Businesses make use of these techniques to secure their corporate secrets, governments to protect sensitive information, and many individuals use it to safeguard personal information and stop identity theft. Cryptographic systems are an integral part of standard protocols, most notably the Transport Layer Security (TLS) protocol, making it relatively easy to incorporate encryption mechanisms into a wide range of applications. A good cryptographic algorithm might also provide for message integrity, which tells the receiver if the original message from the sender has been tampered with. A secure system should maintain the integrity, availability, and privacy of data. Therefore, algorithms that help prevent interception, modification, penetration, disclosure and enhance data/information security are now of primary importance. Encryption is intrinsic to security in the networked world. An understanding of encryption technologies will assist the security professional in understanding and implementing solutions to security concerns in distributed systems.

The encryption process, which transforms the plain text into ciphertext, may be thought of as a "black box", which takes inputs (the plain text and key) and produces output (the ciphertext). The messages may be handwritten characters, electromechanical representations as in a teletype, strings of 1s and 0s as in a computer or computer network, or even analog speech. The black box will be provided with whatever input/output devices it needs to operate; the insides, or cryptographic algorithm, will operate independently of the external representation of the information. The key, which is more properly called the crypto variable, is used to select a specific instance of the encryption process embodied in the machine. The same black box produces different ciphertexts from the same plain text. In typical operation, a key is inserted prior to encrypting a message and the same key is used for some period of time. This period of time is known as a cryptoperiod. For reasons associated with cryptanalysis, the key should be changed on a regular basis. The most important fact about the key is that it embodies the security of the encryption system. This means the system is designed so that complete knowledge of all system details, including specific plain and ciphertext messages, is not sufficient to derive the cryptovariable.

It is important that the system is designed in this fashion because the encryption process itself is seldom secret. The

details of the data encryption standard (DES), for example, are widely published so that anyone may implement a DES compliant system. In order to provide the intended secrecy in the ciphertext, there has to be some piece of information that is not available to those not authorized to receive the message; this piece of information is the cryptovariable or key[3].

Computer data often travels from one computer to another, leaving the safety of its protected physical surroundings. Once the data is out of hand, people with bad intentions could modify or forge your data, either for amusement or for their own benefit.

Cryptography can reformat and transform our data, making it safer on its trip between computers. The technology is based on the essentials of secret codes, augmented by modern mathematics that protects our data in powerful ways.

A. Computer Security - generic name for the collection of tools designed to protect data and to thwart hackers
B. Network Security - measures to protect data during their transmission
C. Internet Security - measures to protect data during their transmission over a collection of interconnected networks

To assess the security needs of an organization effectively, the manager responsible for security needs some systematic way of defining the requirements for security and characterization of approaches to satisfy those requirements. One approach is to consider three aspects of information security:

- Security attack – Any action that compromises the security of information owned by an organization.
- Security mechanism – A mechanism that is designed to detect, prevent or recover from a security attack.
- Security service – A service that enhances the security of the data processing systems and the information transfers of an organization. The services are intended to counter security attacks and they make use of one or more security mechanisms to provide the service.

The world of cyber security is constantly evolving, and researchers are always striving to design better algorithms and mend the existing loopholes.

II. LITERATURE SURVEY

Some bit-level encryption algorithms are DNA based cryptography and BLSKEA version-1 and 2 that were already developed by Nath et al. Now the authors discuss the mentioned versions below.

A. Bit Level Symmetric Key Encryption Algorithm (BLSKEA): This method deals with bit-level encryption and decryption method. Nath et al(2014) already introduced bit-level encryption method using feedback. But in the present paper, the authors have used some simple but very effective bit level encryption method. The plain text is initially converted to bits and after that bit-wise complement is done on some random prime positions. The entire bit stream is reversed and again applied bit complement operation in some random prime position. The bit complement is followed by bit-wise XOR operation and then the modified bit-streams placed in a 2-dimensional array and perform some bit operations such as left-shift, up-shift, diagonal shift, cycling, right-shift number of times to make the bit patterns random. The bit operations are performed a number of times and finally, bits were converted to bytes and transferred to some output file. The results show that the present method is very much effective to encrypt passwords, SMS or any other confidential message. This algorithm has also been improvised.

B. Bit Level Multi Way Feedback Encryption Standard (BLMWFES): In the present method (BLMWFES-1), the encryption-decryption process was done in bit-level. The entire plaintext was broken up into blocks. For every block, the first plaintext bit, the first key bit, the initial forward feedback bit (FF) and the backward feedback bit (BF) are added and then taken modulo with 2. The bit obtained is the ciphertext bit. This ciphertext bit was then propagated right to the n1-th bit's forward feedback value. 'n1' is equal to the forward skip (FS) value specific to that block. Next, the last plain text bit of that particular block was added (modulo 2)with the key bit, FF, initial BF and then the ciphertext bit obtained was then propagated left, to the n2-th bit's backward feedback value. 'n2' is equal to the backward skip (BS) value of that block. This process of adding the bits (modulo 2) and then propagating it to the appropriate position was done alternately from the left to right and from right to left on the entire block of bits. This completes one round of processing on a single block. After every round of processing, the four important variables such as FF, BF, FS and BS changes dynamically due to changing block size and this makes the encryption very strong. The total number of rounds (encryption number), the block size for every round, initial FF, initial BF, initial FS and initial BS were taken as a function of the keypad. The keypad was generated from the user entered key (seed) using the key-expansion algorithm explained in section-II. This method was then tested on standard plain texts such as ASCII '0', ASCII '1', ASCII '2', ASCII '3' and

the results obtained were quite satisfactory. This method is immune to any classical form of attacks.

C. DNA Based Cryptography: DNA cryptography is a relatively new paradigm that has attracted great interest in the field of information security. DNA coding technology is used to convert binary data to DNA strings. Since scientists found that binary computers have many physical limitations, especially in data storage and computation, they have concentrated on DNA computers and tried to implement this new science in the information security field. DNA cryptography is a new concept that needs many improvements. Although there are still problems with DNA cryptography, many scientists are trying to solve them because they believe that, with the characteristics of DNA computers they have more advantages than conventional cryptography. DNA coding technology is another concept in cryptography that is intended to encode binary data to a DNA strand and vice versa. Binary data can be encoded in DNA by using a sequence of the alphabet. It is known that DNA sequences contain four basic letters A, C, G, and T: '00' → A, '01' → C, '10' → G, '11' → T For example, a binary string like'01001011' is converted to 'CAGT'. The cryptosystem is based on the Vigenere cipher, which is a poly-alphabetic cipher. Poly-alphabetic ciphers are multi-substitution ciphers, which means that each letter in the plain text is substituted in different forms. The main achievement of this study is identifying a DNA cryptosystem, which is a new science in information security. But the Vigenere ciphers have some problems. The first problem is that it uses the English alphabet so it is obvious that with frequency analysis we can guess the correct letter of the ciphertext. But in this project all encryption process in bit level. In this project first, the authors have done a 3-dimensional encryption process for n times. Then covert the bit level ciphertext into DNA sequences that are the form of A, C, G and T form. Then perform some randomization operation to randomize the DNA sequence. Then the authors converted the DNA sequence into bits and then the bits are converted into byte form. There are various improved versions of the algorithms.

However, it is apparent that though various bit--level algorithms have been designed to date, none of them incorporates the position of bits into account. The present algorithm is unique and truly innovative in this respect. Additionally, this makes the algorithm truly hard to break using conventional cryptanalysis techniques.

III. ENCRYPTION ALGORITHM

The encryption algorithm is as follows -

Step1: Input plain text and the key. The key can be any random number.

Step2: Convert plain text to bits.

Step3: Find the position of 1s from the bit-stream thus obtained.

Step4: Convert the positions obtained from the previous step into their binary form, and store the positions in a 32-bit format in a one-dimensional array. E.g. If the position of the first 1 in the original bit-stream is 7, then the corresponding stored format should be: 00000000 00000000000000000000000111.

Step5: Complement the bits in non-prime positions in the bit sequence.

Step6: Reverse the entire bit sequence, and complement the bits in the prime positions.

Step7: Perform bitwise XOR operation with the bits at the odd position and even position and substitute the resulting bit at the even position. E.g. Bits at positions 1 and 2 are XOR-ed and the resulting bit is substituted for bit 2. Similarly, bits at positions 3 and 4 are XOR-ed and the result is substituted for bit 4, and so on.

Step8: Reverse the entire bit sequence and perform bitwise XOR operation in the following manner: Bit 1 and bit n(n being the last bit position in the bit sequence) are XOR-ed and resulting bit is substituted for bit n, bit 2 and bit (n-1) are XOR-ed and the resulting bit is substituted for bit (n-2), and so on.

Step9: Take a two-dimensional array of size key*key, and store the first key*key elements from the bit sequence into it. Perform the following shifting operations on the elements:

- Perform bitwise left-shift. This is done by shifting all elements in each row by one unit in the left direction.
- Perform bitwise diagonal shift. This is done by shifting all elements in each diagonal by one unit diagonally along both diagonals.
- Perform bitwise down-shift. This is done by shifting all elements in each column by one unit in the down direction.
- Perform bitwise right-shift. This is done by shifting all elements in each row by one unit in the right direction.

- Perform bitwise up-shift. This is done by shifting all elements in each column by one unit in the down direction.

The elements of the modified two-dimensional array are stored in their respective positions in the original array.

Step10: Take each successive (key*key) number of bits from the bit sequence and perform step9 until the number of residual bits is less than (key*key).

Step11: Bring the residual bits to the beginning of the two-dimensional array and perform step9.

Step12: Repeat Step5.

Step13: Repeat Step6.

IV. DECRYPTION ALGORITHM

The decryption algorithm is as follows:

Step1: Enter the ciphertext and the key.

Step2: Convert ciphertext to bits.

Step3: Complement the bits in prime positions and reverse the bit sequence.

Step4: Complement the bits in the non-prime positions.

Step5: Take a two-dimensional array of size (key*key), and store the first (key*key) elements from the bit sequence into it. Perform the following shifting operations on the elements:

- Perform bitwise down-shift. This is done by shifting all elements in each column by one unit in the down direction.
- Perform bitwise left-shift. This is done by shifting all elements in each row by one unit in the left direction.
- Perform bitwise up-shift. This is done by shifting all elements in each column by one unit in the down direction.
- Perform bitwise diagonal shift. This is done by shifting all elements in each diagonal by one unit diagonally along both diagonals.
- Perform bitwise right-shift. This is done by shifting all elements in each row by one unit in the right direction.

The elements of the modified two-dimensional array are stored in their respective positions in the original array.

Step6: Take each successive (key*key) number of bits from the bit sequence and perform Step5 until the number of residual bits is less than (key*key).

Step7: Bring the residual bits to the beginning of the two-dimensional array and perform step5.

Step8: Perform bitwise XOR operation in the following manner: Bit 1 and bit n(n being the last bit position in the bit sequence) are XOR-ed and resulting bit is substituted for bit n, bit 2 and bit (n-1) are XOR-ed and the resulting bit is substituted for bit (n-2), and so on. Then the entire bit sequence is reversed.

Step9: Perform bitwise XOR operation with the bits at the odd position and the even position and substitute the resulting bit at the even position. E.g. Bits at positions 1 and 2 are XOR-ed and the resulting bit is substituted for bit 2. Similarly, bits at positions 3 and 4 are XOR-ed and the result is substituted for bit 4, and so on.

Step10: Complement the bits in the prime positions and reverse the entire bit sequence.

Step11: Complement the bits in the non-prime positions.

Step12: Find the position of 1s from the bit-stream.

Step13: Construct a new bit sequence and place 1s at the positions obtained from Step12.

Step14: Take up each block of 8 bits and convert them back into bytes. This gives us the deciphered text.

V. RESULTS

Test Case 1: (Key - 22)

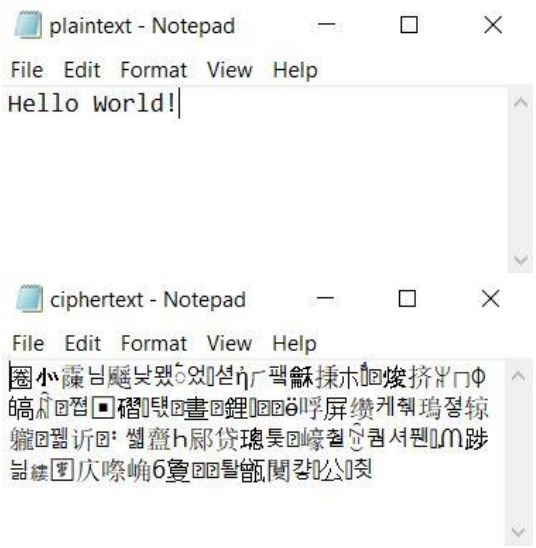

Figure 1. A plain text with corresponding cipher-text using key: 22.

Test Case 2: (Key - 23)

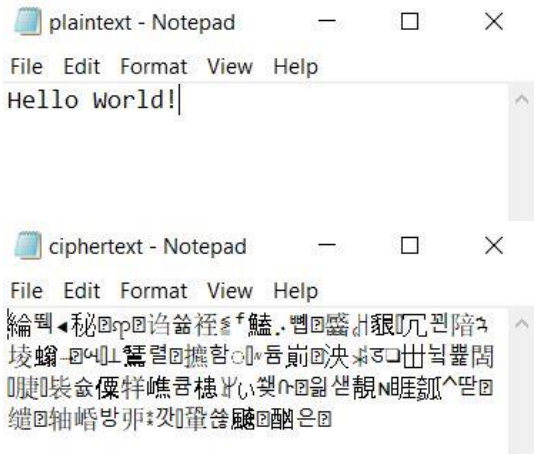

Figure 2. A plain text with corresponding cipher-text using key: 23.

Test Case 3: (Key - 23)

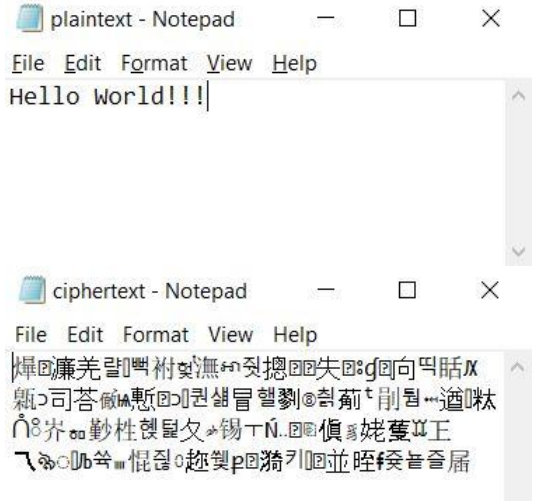

Figure 3. A plain text with corresponding cipher-text using key: 23.

Test Case 4: (Key - 25)

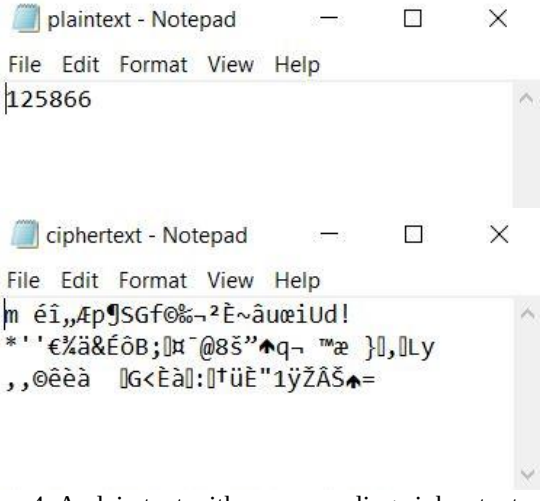

Figure 4. A plain text with corresponding cipher-text using key: 25.

Test case 1 and test case 2 shows two cases with the same plain text but different keys. We observe that the ciphertexts generated in both cases are strikingly different (Fig 1 and Fig 2).

Test case 2 and test case 3 presents different plain text with the same keys. Here, too, the cipher texts generated in both cases differ substantially (Fig 2 and Fig 3).

Test case 4 presents a case where the plain text is a sequence of digits (like in an OTP). (Fig 4).

VI. CONCLUSION AND FUTURE SCOPE

The present algorithm has been tested on various types of files like .txt, .doc, .pdf, .png, .jpg and other media files. It can be verified that the encrypted files cannot be decrypted without the initial knowledge of the key. The algorithm is secure from brute force attack or any kind of statistical attack, since it deals with bit positions, rather than encryption of individual characters as is done in conventional cryptographic algorithms.

If the ciphertext is altered by even just a single character, the original plain text cannot be recovered, as the algorithm is sensitive.

As the algorithm is highly secure, it may be used for the security of highly confidential texts of small lengths such as OTP or password encryption.

VII. ACKNOWLEDGEMENT

The authors are indebted to the Dept. of Computer Science, St. Xavier's College, Kolkata, for providing them an opportunity to work on cryptographic algorithms. One of the authors AN expresses his sincere gratitude to Rev. Dominic Savio, Principal of St. Xavier's College, Kolkata for allowing the author to do research work in the field of Network Security.

REFERENCES

[1] William Stallings, "Cryptography and Network Security: Principles and Practice", Tata Mc-Graw Hill Publishing LTD.
[2] Behrouz A. Forouzan, "Cryptography and Network Security", Special Indian edition 2007, Tata Mc-Graw Hill Publishing LTD.
[3] Ronald A. Gove, "Introduction to Encryption Technology."
[4] Dan Boneh, Victor Shoup, "A Graduate Course in Applied Cryptography." from Stanford University.
[5] Sachin Sharma, Jeevan Singh Bisht, "Performance Analysis of Data Encryption Algorithms", Volume-**3**, Issue-**1**, pp. **1-5, 2015.**
[6] M. Arora, S. Sharma, "Synthesis of Cryptography and Security Attacks", Volume-**5**, Issue-**5**, Oct **2017**.
[7] Sreeparna Chakrabarti, Dr. G.N.K. Suresh Babu, "A Literature Survey on the Cryptographic Encryption Algorithms for Secured Data Communication." in

International Journal on Future Revolution in Computer Science & Communication Engineering Volume: **4** Issue: **10**.

[8] Asoke Nath, Soumyadip Ray, Salil Anthony Dhara, Sourav Hazra, "3-Dimensional Bit Level Encryption Algorithm Version-3 (3DBLEA-3)", International Journal of Latest Trends in Engineering and Technology Vol.(**10**)Issue**(2**), pp.**347-353**

[9] Bit Level Encryption Standard(BLES) : Version-I, Neeraj Khanna, Dripto Chatterjee, Joyshree Nath and Asoke Nath, International Journal of Computer Applications(IJCA)(0975-8887) USA Volume **52**-No.2.,Aug, Page.**41-46**(2012).

[10] Multi Way Feedback Encryption Standard Ver1, Purnendu Mukherjee, Prabal Banerjee, Asoke Nath, IJACR, published in September **2013** issue.

Authors Profile

Dr. Asoke Nath is working as Associate Professor in the Department of Computer Science, St. Xavier's College (Autonomous), Kolkata. He is engaged in research work in the field of Cryptography and Network Security, Steganography, Green Computing, Big data analytics, Li-Fi Technology, Mathematical modelling of Social Area Networks, MOOCs etc. He has published more than **248** research articles in different Journals and conference proceedings.

Mr. Sankar Das is an Associate Professor in the Deartment of Computer Science, St. Xavier's College (Autonomous), Kolkata. He has done projects on Cryptography and Network Security. He is currently doing research work in field of Image processing, Digital Steganography and extensive research work on Ethical Hacking.

Miss Oishi Mazumder is currently a third year undergraduate student in the Department of Computer Science, St.Xavier's College. She is interested to work in the domains of Artificial Intelligence and Cryptography with their implementation in real life applications for solving the problems and easing the world of the common people.

Ms. Adrija Saha is a student of Dept. of Computer Science, St. Xavier's College (Autonomous), Kolkata. She is interested in DBMS, Visual Basic, SQL and Automata Theory.

Mr. Monimoy Ghosh is a student of the Dept. of Computer Science, St. Xavier's College, Kolkata. He is enthusiastic about Cryptography, Network Security, DBMS, Automata Theory, Compiler Design and Operating Systems. He is also a member of various online learning communities like Codechef, Coursera, edX, etc.

