A Novel Information Communication Security Framework Using Local Binary Pattern and Significant Bit Swapping

B Susheel Kumar[1] Dr Thummaluru Suresh Babu[2] Godina Amruthavani[3] N. Raja Rajeswari[4] Krishnaiah Pulluru[5] Chinthakunta Swetha[6]

[1,3,4,5,6] Dept of Computer Science and Technology, Yogi Vemana University, Kadapa
[2] Dept of Journalism and Communication, Yogi Vemana University, Kadapa

Abstract: Information exchange has been subjected to multiple stages of cryptography testing. The sender, recipient, and communication channel are all part of the fundamental communication system. Providing strong security and authentication for information is the primary goal of cryptography evaluation. Third parties can access data as it is being transmitted. Systems are considered more secure because it takes a long time to figure out the encryption. Many cryptography methods have been devised; however, the combination of these approaches offers great protection for sensitive data. The suggested method uses significant bit swapping and local binary patterns (LBP) to generate keys for encryption and decryption.

Keywords: Encryption, Decryption, Significant Bit Swapping, LBP.

INTRODUCTION

A technique for producing secret communications that are unreadable by others is called cryptography [1]. These days, cryptography is crucial for securely sending and receiving data. There are many algorithms available nowadays to ensure the validity and integrity of information transit. Three separate criteria are used to classify cryptographic systems.

- How ordinary text is transformed into ciphertext. Every encryption relates to either transposition or substitution procedures.
- Each plaintext character element is represented by a different character in replacement techniques, while plain text elements are rearranged in transposition. Here, the main prerequisite is that neither of these methods should result in the loss of any information. Depending on how plaintext is processed, a block cipher is used when all of the plaintext is processed simultaneously as ciphertext. A stream cipher is used if the plaintext is processed element by element.
- Considering how many keys are used. Symmetric, single-key, or secret-key encryption, sometimes known as conventional encryption, is a method in which both the sender and the recipient utilize a single secret key for both encryption and decryption. The system is known as asymmetric, two-key, or public-key encryption if the sender and the recipient utilize distinct keys for encryption.
- Considering how many keys are used. Symmetric, single-key, or secret-key encryption, sometimes known as conventional encryption, is a method in which both the sender and the recipient utilize a single secret key for both encryption and decryption. The system is known as asymmetric, two-key, or public-key encryption if the sender and the recipient utilize distinct keys for encryption.

CONVENTIONAL ENCRYPTION MODEL

Figure 1 illustrates a traditional encryption model. In traditional encryption, a key used for encryption and decryption must be agreed upon by the sending and receiving parties.

The sender encrypts the plaintext, Xp, using a key K, creating the ciphertext Cp. The recipient receives the ciphertext Cp. After receiving the encrypted text Cp, the recipient will use the easily accessible key to decrypt it and retrieve the plaintext Xp. We can enable the fundamental methods for conventional encryption thanks to the numerous classical encryption techniques that are currently available for encryption and decryption. The two fundamental methods of encryption are transposition and substitution. Other strategies incorporate transposition and substitution methods.

The Caesar, Mono-alphabetic, Playfair, and Hill ciphers are a few examples of substitution approaches. Transposition techniques include Rail Fence and Row Transposition, among others. This is how the rest of the paper is structured. The background work for the "Automatic Key Generation of Caesar Cipher" is presented in Section II. The suggested approach is explained with an example in Section III. This paper is finally concluded in Section IV.

BACKGROUND WORK

The plaintext, including alphabets, numerals, and special characters, can be accepted by the current approach described in [2]. There are two stages to the encryption process. Phase one uses the Caesar cipher to generate keys and encrypt plaintext.
The key is created by adding the ASCII values of every plaintext character with a modulus of 256. The generated key is then used to create encrypted text in the manner described below.

K= (1st char+2nd char+…. nth char)

%256 (1)

E = ((P + K) % 256) (2)

P – Plaintext, K – key.

The second stage of the Rail Fence Cipher is then produced by employing the substitution procedure.
In the first stage of the decryption procedure, the Rail Fence cipher is initially decrypted, followed by the Caesar cipher using the sender's previously produced key. The steps involved in decryption are as follows.

D = ((p – k) % 256) (3)

c – ciphertext, k – key

The plaintext is provided by the ASCII character that matches the decimal value.

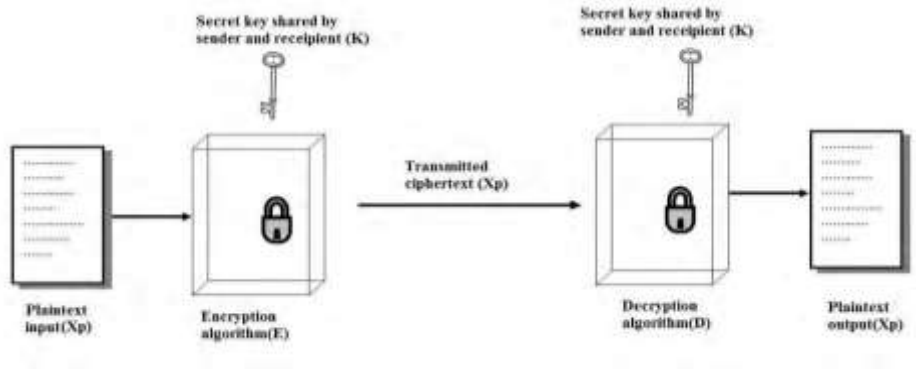

Figure 1: Simplified model of conventional encryption

LIMITATIONS OF THE EXISTING METHOD

Rule Sequence for generating the LBP code is:

- The key is created by adding up all of the plain text's ASCII characters mod
- 256. Each plaintext character is encrypted using the Caesar cipher approach, which makes it simple to break out.
- The generated key range is limited to 1 to 255.
- It is restricted to simple text with just ASCII-compatible characters.

PROPOSED METHOD

The Local Binary Pattern [3], [4], & [5] (LBP) operator is used in the key generation process of the suggested technique. LBP is a non-parametric technique that uses a decimal number's bit locations to create LBP codes, or Local Binary Patterns, that encode the decimal value.

Local Binary Pattern for the decimal value of key is represented as

2^7	2^6	2^5
2^0	Key	2^4
2^1	2^2	2^3

Figure 2: LBP Pattern

Table 1: Rule Sequence

Rule/Base Pairs	Bits
i<j (01, 02, 12)	a, b, c
i>j (10,20,21)	d, e, f
i=j (00,22)	g, h

The following is binary code based on the sequence of rules.

Bit No.	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bits obtained from Rule Sequence	g	a	b	c	f	e	d	h

The key is provided by the decimal representation of the bits (g a b c f e d h). The suggested approach accepts plain text with UNICODE characters. The suggested method's encryption and decryption procedures are explained in the sections that follow.

A. ENCRYPTION ALGORITHM

STEP 1: The random seed and plaintext median will be combined to create the key.
STEP 2: Use LBP to create the local binary pattern.

2^7	2^6	2^5
2^0	Key	2^4
2^1	2^2	2^3

Figure 3: LBP Pattern

The following rule sequence is used to create the local binary pattern:

Table 2: Rule Sequence

Rule/Base Pairs	**Bits**
i<j (01, 02, 12)	a, b, c
i>j (10,20,21)	d, e, f
i=j (00,22)	g, h

Local Binary pattern is:

Bit No.	7	6	5	4	3	2	1	0
Bits obtained from Rule Sequence	g	a	b	c	f	e	d	h

Table 1. Local Binary Pattern Using Rule Sequence

- **STEP-3:** Every plain text character should be XORed with the local binary pattern found in STEP-2
- **STEP-4:** Using the result from STEP-3, create the encrypted text character by significantly switching the bits.

B. DECRYPTION ALGORITHM

STEP-1: Swap important bits for every character in the ciphertext.
STEP-2: Using LBP, create the local binary pattern for the key that was sent by the sender in accordance with the encryption procedure.
STEP-3: To generate the plain text, XOR each encrypted text character with the local binary pattern. The encryption and decryption process of the suggested method is demonstrated in the following example.

EXAMPLE:

The plain text is "NETWORK"

ENCRYPTION:

PLAIN TEXT: N E T W O R K
STEP 1: Key: 179(by combined random and median of plain text)
STEP 2: Local Binary Pattern for key (=179) as shown in figure-6.

1	0	1
1	**Key(179)**	1
1	0	0

Table 3: Rule Sequence

LOCAL BINARY PATTERN IS:

Bit No.	7	6	5	4	3	2	1	0
Bits obtained from Rule Sequence	1	0	1	1	0	1	1	0

Figure 6: Local Binary Pattern Using Rule Sequence Table 2.

STEP 3: XOR between binary pattern and each character in the plain text.
UNICODEs for plaint text characters are

N → (004E) 16
E → (0045)16
T → (0054)16
W → (0057)16
O → (004F) 16
R → (0052)16
K → (004B) 16

For Plain Text Character 'N'								
Local Binary Pattern	1	0	1	1	0	1	1	0
Plain Text character	0	1	0	0	1	1	1	0
Cipher Text Character	1	1	1	1	1	0	0	0

Figure 7: Showing XOR Operation between Local Binary Pattern and Plain Text Letter 'N'

After significant bit swapping, the ciphertext character for „N‟ is „y.‟

Similarly for „E‟: ≤
Similarly for „T‟: c
Similarly for „W‟: ß
Similarly for „O‟: ·
Similarly for „R‟: e
Similarly for „K‟: ²

The ciphertext is: y ≤ c ß · e ²

DECRYPTION:

CIPHER TEXT: y ≤ c ß · e ²
STEP1: For every character in plain text, swap relevant bits.
STEP2: As illustrated in figure 8, create the local binary pattern for the given key (=179).

Bit No.	7	6	5	4	3	2	1	0
Bits obtained from Rule Sequence	1	0	1	1	0	1	1	0

Figure 8: Local Binary Pattern Using Rule Sequence Table 2**.**

STEP 3: XOR between the local binary pattern and each cipher text character.

UNICODEs for cipher text characters are y
→ (0079)16
≤ → (2264)16
c →
(0063)16 ß
→ (00DF)16
· → (2219)16
e → (0065)16
² → (00B2)16

For Cipher Text Character 'y'								
Local Binary Pattern	1	1	1	1	1	0	0	0
Cipher Text character	1	0	1	1	0	1	1	0

Plain Text Character	**0**	**1**	**0**	**0**	**1**	**1**	**1**	**0**

Figure 9: Showing XOR Operation between Local Binary Pattern and Cipher Text Letter 'y'

Plaintext Character for ciphertext 'y' is N.
Similarly, for „≤": E
Similarly, for „c": T
Similarly for „ß": W
Similarly for „·": O
Similarly, for „e": R Similarly for „²": K

ADVANTAGES OF THE PROPOSED METHOD

- The suggested approach is compatible with NICODE.
- Generation of random keys.
- Considerable bit switching was used to provide high security.

CONCLUSION

An essential part of encryption and decoding is key generation. The suggested solution uses bit-significant switching and the dynamic key produced by the LBP to encrypt and decrypt text. With multiples of 8-bit chunks, the key (currently 8-bit) can be any length.

REFERENCES

1. William Stallings, "Cryptography and Network Security: Principles and Practices", 4th Edition, Prentice Hall, 2006, page numbers 30-39.
2. Bazith Mohammed, "Automatic Key Generation of Caesar Cipher", International Journal of Engineering Trends and Technology (2231-5381), Volume 6, No.6.
3. Ms. Drashti H. Bhatt1, Mr. Kirit R. Rathod, Mr. Shardul J. Agravat, "A Study of Local Binary Pattern Method for Facial Expression Detection", International Journal of Computer Trends and Technology (IJCTT) – volume 7 number 3– Jan 2014, page numbers 152-153.
4. Pradeep Kumar Reddy. R, Dr. Nagaraju .C "Text Encryption Through Level Based Privacy using DNA Steganography", International Journal of Engineering Trends and Technology in Computer Science (2278-6856), Volume 3, Issue3, May- June 2014.
5. Hans Delfs and Helmut Knebl, "Introduction to Cryptography: Principles and Applications", Springer, first edition, 2002, page numbers 11-14.