International Journal of Innovative Technology & Adaptive Management (IJITAM)
ISSN: 2347-3622, Volume-2, Issue-1, October, 2014

An Approach of Bit-level Private-key Encryption Scheme based on Alphabetic Group and Bit-Shift Operators in Selective Mode

Nilanjan Das[1], Assistant Professor, Department of Computer Applications, Siliguri Institute of Technology, Siliguri , WB, INDIA,

Bidyut Das [2], Assistant Professor, Department of Computer Applications, Siliguri Institute of Technology, Siliguri, WB, INDIA,

Abstract **–** Private-key cryptography is a cryptographic system that uses the same secret key to encrypt and decrypt messages. The problem with this method is transmitting the secret key to the receiver who needs it without being intercepted. Many of the existing private-key cryptography systems are complex and not up to the mark with respect to security, as the distribution of the private-key without interpretation are very hard to achieve. In this paper, we have focused on the secret procedure to retrieve secret value from the private-key rather than securing the actual private-key value. The encryption is done by the secret value derived from the private-key. The secret value is being derived by combining n^{th} (consonant, special character, vowel or semivowel) as per an user defined sequence number. Then we execute a user-defined operator (Right or Left shift in selective mode) on the secret value. That secret value is being used for encryption and decryption. Thus an attempt is made to enhance the security.

Keywords – N^{th} (consonant, vowel or semi vowel, special character), Bit Shift Operator,Private-key encryption, Stream cipher**.**

I. INTRODUCTION

Cryptography is the practice and study of techniques for secure transmission of information between receiver and sender in presence of other parties.

Private-key cryptography refers to an encryption method where both the sender and the receiver use either the identical secret key or two keys derivable from each other. Fig-1 represents a private-key encryption method that uses the same secret key for encryption and decryption.

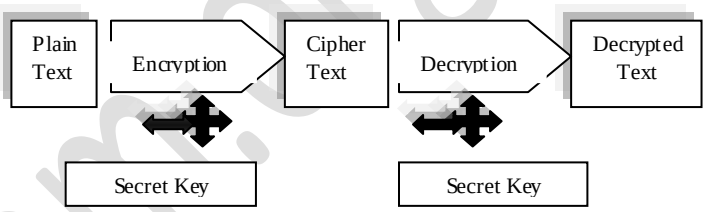


Figure 1: A Private-key Encryption Scheme

The traditional demerit of private-key encryption technique is to distribute the private-key securely. Noticeably, many of the existing private-key encryption systems suffer from lack of security.

Here we have developed a procedure, which is responsible to retrieve the secret value from the private-key. The value is being used both for encryption and decryption. The secret value is being derived by searching and combining n^{th} (consonant, special character, vowel or semivowel) as per an user defined sequence number. Then we execute a user-defined operator (Right or Left shift in selective mode) on the secret value. That secret value is being used for encryption and decryption.

Herein lays the attempt to increase security, as we focus on securing the retrieving procedure rather than directly the private-key value. Secret value can't be retrieved without the knowledge of the retrieving procedure [2] [3] [4]

In this paper, Section-II describes the encryption process; Section-III describes the decryption process. Experimental results are being described in Section-IV and Section-V draws the conclusion.

II. ENCRYPTION PROCESS

Bit Shift Operations- in left bit shift operation we are discarding the extreme left bit from a specific numbers of bit-

representation of a decimal number and inserting the same numbers of '0' from the right end. In right bit shift operation we are discarding the extreme right bit from a specific numbers of bit-representation of a decimal number and inserting the same numbers of '0' from the left end.

We have used the English alphabetic groups like vowel, semi-vowel, consonant and special character. A user defined sequence is being used to combine the results for generating the secret value from the private key. Step A, step B, step C, step D sequentially describes the encryption process [1] [2] [3].

Plain Text Formation

Let 'b' is a character which is present in the inputted file. Fig-2 represents its 8-bit binary representation through an array PLAINTEXT with dimension 8.

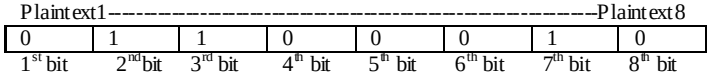


Figure 2: Formation of Plain Text

Step-A.1 Read one character at a time from the inputted file till we reached to the end of the file. Convert each character into 8-bit binary representation and store the value into the array PLAINTEXT with dimension 8.

A. key Generation

The size of the private-key is 24 bits having 6 blocks. 1st 5-bit block represents the choice nth consonant and 2nd 5-bit block represents the choice nth special character. 3rd 3-bit block represents the choice nth vowel or semivowel. 4th 5-bit block the choice nth sequence. 5th 1-bit block represents choice for bit shift(0 for Left shift/1 for right shift).6th 5-bit block represents no of bit shifted to Left or Right. Fig-3 represents block diagram of the 24-bit private-key.

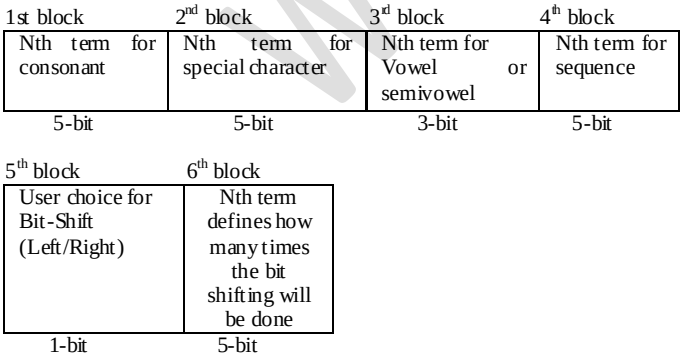


Figure-3: Block Diagram of 24-bit Private-key

Step B.1 Read the user inputs for 6 blocks from the user. Convert those input values into corresponding bit size of their respective blocks and store the values in an array KEY with dimension 40.

C. Formation of Secret Value from Private-key for Encryption

Step C.1 The value of the 1st block, 2nd block, 3rd block is holds the nth term of consonant, special character, vowel and semi-vowel respectively. A user-defined sequence value is stored in the 4th block of the private-key. The choice value for the operator (Left or Right shift) is stored in the 5th block of the private-key. The value of the 6th block is being used to determine that the how many times the bit shift operators will execute..The user defined sequence value stored in 4th block is being used to combine the results (Nth (consonant, special character, vowel and semi-vowel,)) together.

Step C.2 Generate the corresponding nth consonant, special character, vowel or semi-vowel.

Step C.3 Sequence wise combined the 24 bit Value generation.

Example:-The example demonstrates the private-key formation and the secret value generation procedure. Fig-4 represents the bit wise representation of a private-key for some specific value.

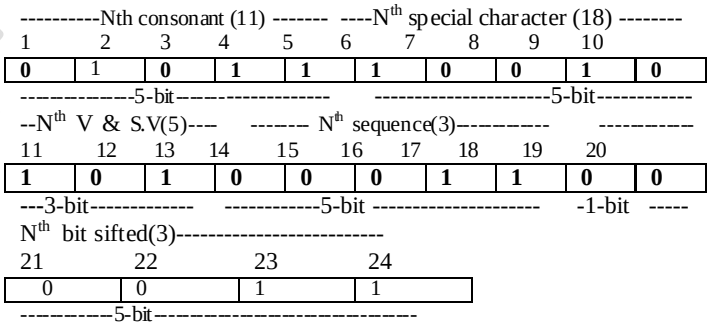


Figure 4: 24- bit Representation of Private-key for a Specific Value.

1st block of the private-key is 11, so the corresponding consonant is "N", 2nd block of the private-key is 18, so the corresponding special character is "=". 3rd block of the private-key is 5, so the corresponding vowel or semi-vowel is "U". We combine all this result as per the combination no which is 3.The combination number is store in 4th block. The definition of the combination is (Nth special character + nth vowel or semi-vowel + Nth consonant). The result is combining as per the combination value. The value is "=UN". Fig-5 represents the 48-bit representation of secret value.

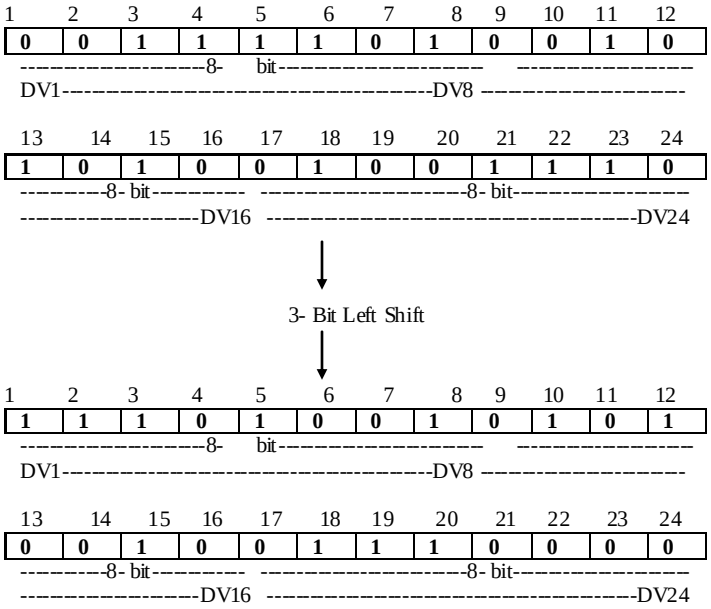


Figure 5: 24- bit Representation of Secret Value derived from Private-key.

D. XOR operation and Formation of Cipher Text

Plain text is being encrypted by the 6 blocks of the secret value cumulatively where the block size is 8 bits. Bitwise XOR operation is being performed between the plain text and the secret value Fig-6 represents the encryption process

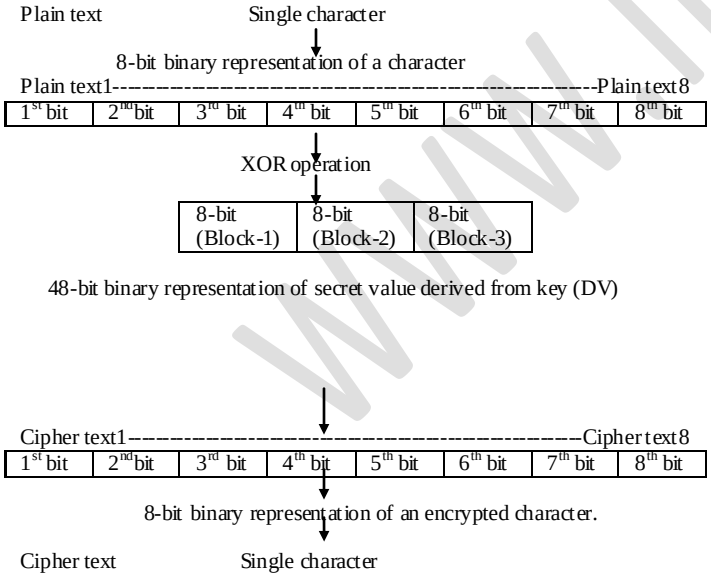


Figure 6: XOR operation between Plain Text and Secret Value.

Cumulative XOR operation is performed between plain text and the secret value. 3rd block, 2nd block and the 1st block of the secret value (DV) are used for XOR respectively After 3 times, we get the binary value of corresponding ASCII code of a final encrypted character. In this way cipher text file is generated and sent to the receiver with the secret private-key file. Fig-7 demonstrate the total XOR procedure between plain text and the secret value where IET means intermediate encrypted text

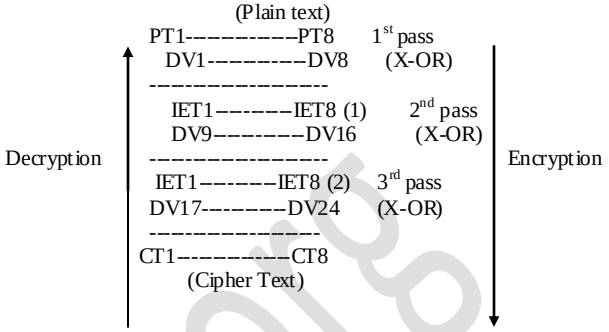
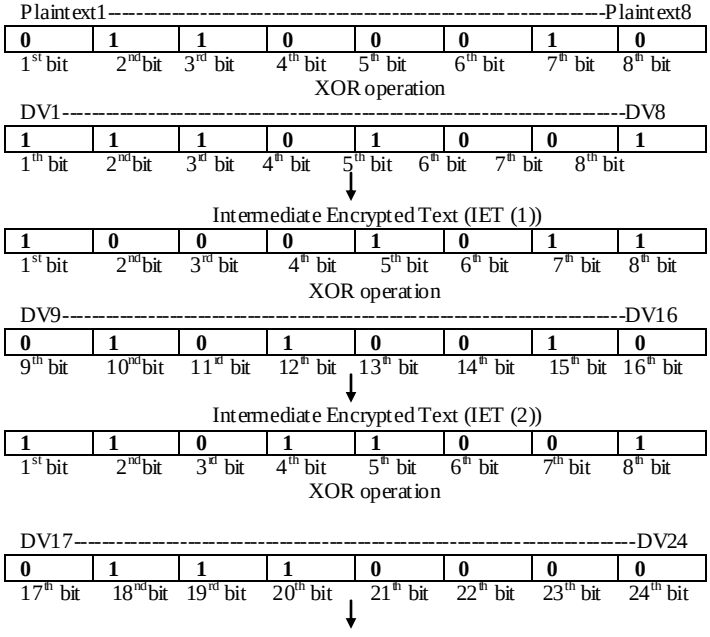


Figure 7: Block wise Cumulative XOR operation between Plain Text and Secret Value derived from the Private-key.

Step D.1 Perform XOR operation between the array PLAINTEXT and DERIVEDVALUE. We consider 1st block, 2nd block and 3rd block of the array DERIVEDVALUE for XOR operation in 1st pass, 2nd pass, 3rd pass, respectively. Intermediate result is stored into an array IET with dimension 8. And final result is stored into array ENCRYPTED with dimension 8.Corresponding ASCII code is generated from of the array ENCRYPTED and from there we get the encrypted character 'b' from plain text whose ASCII value is 98. We generate the plain text in step-A and secret value in step-C. Now we perform the XOR operation between the plain text and the secret value. Fig-8 represents the XOR procedure



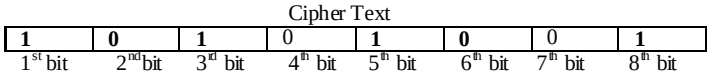


Figure 8: Generation of Cipher Text by Block Wise Cumulative XOR operation between Plain Text and Secret Value.

ASCII value corresponding to the cipher text is 338 and the character corresponding to that ASCII is 'R' In this way all the characters are encrypted and stored into the cipher text file. The encrypted file is sent to the receiver with the secret private-key file.

III. DECRYPTION PROCESS

A Conversion of Cipher Text into Predefined Format

The receiver read one character at a time from the cipher text file till he/she reach to the end of the file and converts the character into 8 bit binary format and store it into an array CIPHERTEXT with dimension 8.

B. Formation of Secret Value from Private-key for Decryption

Derive the secret value from the private-key by using step- C and store that 24-bit binary value into an array DERIVEDVALUE with dimension 24.

C. XOR operation and Formation of Decrypted Text

Step C.1 Perform XOR operation between the array CIPHERTEXT and DERIVEDVALUE. We consider 1st block, 2nd block and 3rd block of the array DERIVEDVALUE for XOR operation in 1st pass, 2nd pass, and 3rd pass respectively. Intermediate result is stored into an array IET with dimension 8. And final result is stored into array DECRYPTED with dimension 8. Corresponding ASCII code is generated from of the array DECRYPTED and from there we get the decrypted character which is stored into decrypted text file.

Content of the Source File (z.txt)	Content of the Encrypted File (en.txt)	Content of the Decrypted File (de.txt)
abcdefghijklmnopqrstuvwxyz	WTURSPQ^_\]Z[XYFGDEBC@ANOL	abcdefghijklmnopqrstuvwxyz

Example- we read a character 'R' from the cipher text whose ASCII value is 338. We convert 338 in 8-bit binary representations and store it into an array CT. The secret value is derived in the step-C. Now we perform XOR operation between the cipher text and the secret value. Fig-9 represents the XOR procedure

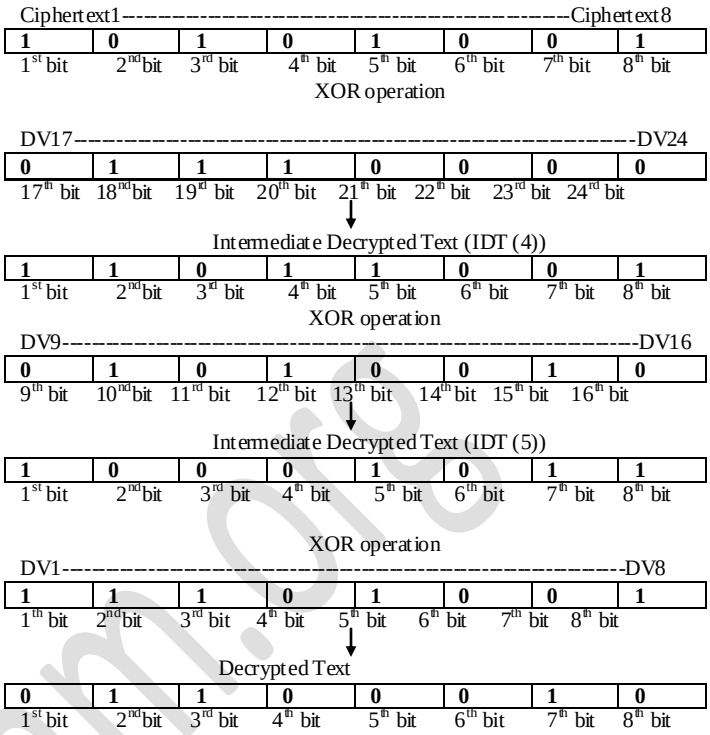


Figure 9: Generation of Decrypted text by Block Wise Cumulative XOR operation between Cipher Text and Secret Value.

ASCII value corresponding to the decrypted text is 98 and the character corresponding to that ASCII is 'b'. In this way all the characters are decrypted and stored into the decrypted file and receiver is able to get the plain text.

IV. EXPERIMENTAL RESULT & DISCUSSION

The encryption of a plain text is done by using the 8th Palindrome number with a forward movement from the user-defined base value 10. The encryption or decryption has taken 31871 milliseconds. Table-I demonstrates the content of the source files, encrypted file and the decrypted file.

Table I: Corresponding content of source, encrypted, decrypted file

V. CONCLUSION

Here we proposed a private-key encryption scheme based on bitwise XOR operation between the plain text and the secret value. The secret value is the Nth palindrome number, counted by making a forward or backward movement from the user-defined base value. Where N is a positive integer in the range of (1<=N<=16).So as the base value or the Nth term is changed then the corresponding palindrome number is also being changed. Thus the security is increased.

The process of retrieving the secret value from the private-key is only being known by the receiver and the sender. So it is not possible for an unauthorised person to derive the secret value only with the presence of private-key. Thus the security is increased in a great entrance.

Besides this, a user can also do the encryption of an inputted file by using several numbers of distinct private-keys where each key is allotted for a specific block among several numbers of user-defined blocks in a plain text file. So the security is increased.

The size of the encrypted file is same as of the plain text file. So we don't need any additional memory for encryption.

The execution time is depends on the file size not on the type of the file as we have done the encryption in bit level.

The only drawback is that if the value of the N or the base value is very higher then it will take very much time to generate the palindrome number. Thus the encryption or decryption time will be increased.

So, the proposed scheme is better in respect of providing security for encryption, encrypted file size management, encryption or decryption time requirement.

REFERENCES

[1] William Stallings, Cryptography and Network security: Principles and practice (Second Edition), Pearson Education Asia, Sixth Indian Reprint 2002.

[2] Atul Kahate (Manager, i-flex solution limited, Pune, India), Cryptography and Network security, Tata McGraw-Hill Publishing Company Limited.

[3] Mark Nelson, Jean-Loup Gailly, The Data Compression Book. BPB Publication