International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-2 Issue-6, May 2013

An Approach Of Bitwise Private-Key Encryption Technique Based On Multiple Operators And Numbers Of 0 And 1 Counted From Binary Representation Of Plain Text's Single Character

Ramkrishna Das, Saurabh Dutta

Abstract*— Private-key cryptography is a cryptographic system that uses the same secret key to encrypt and decrypt messages. The Existing Private-key cryptography systems are complex and not full proof in respect of security concern as the distribution of the private-key without interpretation is very hard to achieve. Here, we have proposed an idea to increase the security of Private-key encryption technique. 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 making arithmetic operation between the user defined base value and a decimal value. That decimal value is derived by performing arithmetic operation between number of '0' and number of '1'counted from 8 bit representation of a plain text's character. The operators are being supplied by the user. As the numbers of '0' and '1' are different for several numbers of character, so we get different secret private-key value for several numbers of character in the plain text. Thus the security is increased.

Index Terms - Arithmetic ***operators, Counting of '0' and '1' from binary representation of a character, Private-key Encryption, Stream Cipher.***

I. INTRODUCTION

Private-key cryptography refers to an encryption method where both the sender and the receiver share the same secret key or their keys are different, but they are related in an easily computable way. Fig-1 represents a private-key encryption method.

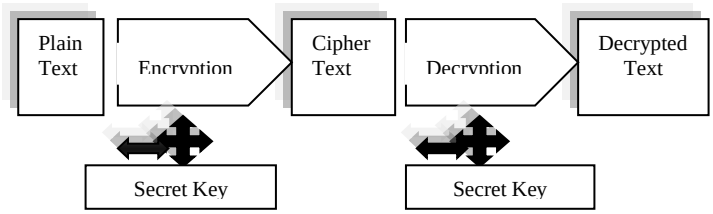

Figure 1: A Private-key Encryption Scheme

Manuscript published on 30 May 2013.
*Correspondence Author(s)
Ramkrishna Das, Department of Computer Applications, Haldia Institute of Technology, Haldia, India.
Saurabh Dutta Department of Computer Applications, Dr B.C.Roy engineering College, Durgapur, India.

The main problem of private-key encryption technique is to distribute the private-key securely. All the existing private-key encryption systems suffer from lack of security. Here, we have proposed a scheme to increase the security of the existing private-key encryption system. We have developed a secret procedure which is responsible to retrieve the secret value from the private-key. The value is being used for encryption and decryption. The secret value is being generated by making different arithmetic operations (operations for base values, operations between N (0) and N(1)) on user defined base value and on decimal values, derived by performing arithmetic operation between number of '0' and number of '1'counted from 8 bit representation of a plain text's character. The operators are being supplied by the user. As the number of '0' and '1' are different for several number of character, so we get different secret private-key value for several numbers of character in the plain text. Combination of user defined operators and derived secret value from plain text construct the private-key. Hence the security is increased as we focus on securing the retrieving procedure rather than the private-key value. Secret value can't be retrieved without the knowledge of the retrieving procedure.

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

II. ENCRYPTION PROCESS

A. *Plain Text Formation: -*

A unit of one character (8 bits) is being encrypted by the secret value at a time. Let 'C' is a character which is being present in the plain text. ASCII value of 'C' is 67 so corresponding 8 bit binary representation is stored in the an array PLAINTEXT with dimension 8.Fig-2 represents the entire procedure.

Plaintext1----------Plaintext8

0	1	0	0	0	0	1	1
1st bit	2nd bit	3rd bit	4th bit	5th bit	6th bit	7th bit	8th bit

Figure 2: Formation of Plain Text

Step-A.1 Read the input file as plain text from the keyboard.
Step-A.2 Read one character at a time from the input file till we reached to the end of the file.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Step-A.3 Convert each character into 8 bit binary representation and store the value into an array PLAINTEXT with dimension 8.

***B: key Generation*: -**

The size of the private-key is 32 bits having 6 numbers of blocks. 1st block having size of 6 bits represent the operator for the user defined base value.2nd block having size of 2 bits represent that how many times the base operator is being used for calculations. 3rd block holds the number of '0' that have been present in single character's binary representation of plain text. 4th block holds the number of '1' that have been present in single character's binary representation of plain text. Both the 3rd and the 4th block are of 3 bits. 5th block holds the user defined operator that is being used between the values of number of '0' and the number of '1'.Size of the 5th block is 6 bits. 6th block holds the user defined base value. The size of the 6th block is 12 bits. Fig-3 represents block diagram of the 32 bits private-key.

1st block	2nd block	3rd block
Operator for base value o(B)	No of time o(B) is performed	Number of 0 in a plain text block N(0)
6 bit	2 bit	3 bit
4th block	**5th block**	**6th block**
Number of 1 in a plain text block N(1)	Operator o(N) between N(0) and N(1)	User defined Base Value.
3bit	6 bit	12 bit

Figure-3: Block diagram of 32 bit Private-key

Step B.1- Read the user inputs for 1st, 2nd, 5th and 6th block from the user.

Step B.2- The value for the 3rd and 4th block are calculated from the plain text. Convert the input values into corresponding bit size of their respective blocks and store the values in an array KEY with dimension 32.

Step-C: Formation of Secret Value from Private-key for encryption:

Two Decimal values are being derived from the plain text. Decimal value for number of '0' is the value which is generated by counting number of '0' from an 8 bit binary representation of a character in the plain text. Decimal value for number of '1' is the value which will be generated by counting number of '1' from an 8 bit binary representation of a character in the plain text.

Step C.1- The secret value is calculated by using the following formula. At first Result1 is calculated. Result= [N (1)-N (0)] o (N) N(1) Then the final result is calculated by using the following formula. Final result= (base value) o(B)[No of time o(B) is performed] Result1.

Step C.2- Make 8 bit binary representation of that final derived value and store that value in the array DERIVEDVALUE with dimension 8.

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.

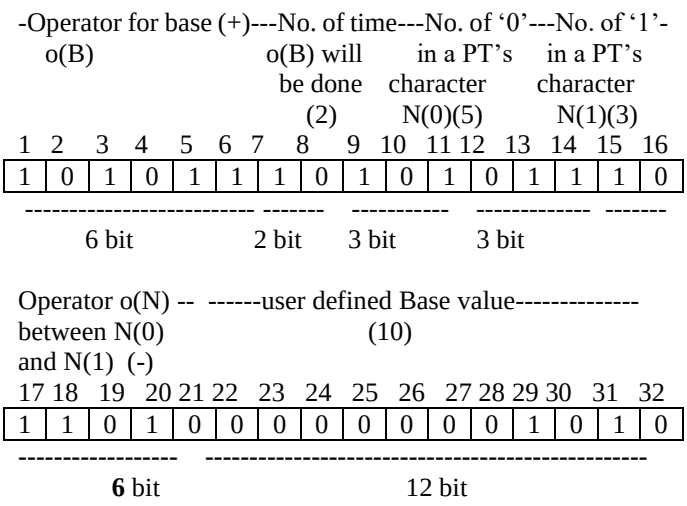

Figure 4: 32 bit representation of private-key for specific value.

Operator for base, operator between N(0) and N(1) and base value are taken from user which are +, – and 10 respectively. Convert these operators into corresponding ASCII codes and corresponding binary value is stored in 1st block, 5th block and 6th block respectively. We have read a character from the plain text. Let 'C' is the character. The ASCII code of 'C' is 67. Now we have converted that ASCII code into 8 bit binary representation. This is 01000011.Then we count how many number of '0' and how many number of '1' are being present in 8 bit representation of a plain text's character. Those values are being stored in N (0) and N (1).Now we have to calculate the result by using the following formula. At first [N (1)-N (0)] is performed. So that is IR= ABSOLUTE (3-5) =2. Now we execute the formula IR1=IR o(N) N(1). So that is ABSSOLUTE (2-3) =1. Here IR is the intermediate result and o (N) is '-'in the private-key. Now the final result is calculated by using final result= (base value) o (B) [No of time o (B) will be performed] IR1. So that is FR=10+1+1=12. Where '+' is the base operator. 10 is the base value and we have to perform base operation for 2 times. Thus the final result is 12. We convert it into 8 bit binary representation and use it for encryption and decryption. The secret value is stored in an array DV with dimension 8.

The 8 bit binary representation of 12 is done and we can also convert it into 16 bit or 24 bit depending upon the range of the value. Fig-5 represents the 8 bit representation of the derived secret value.

1bit	2bit	3bit	4bit	5bit	6bit	7 bit	8bit
0	0	0	0	1	1	0	0

DV1----------DV8

Figure 5: 8 bit representation of Secret value derived from key.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Step-D: XOR operation And formation of Cipher Text file

The plain text is being encrypted by the 1 block of the secret value. XOR operation is performed bitwise between the plain text and the secret value. Fig-6 represents the encryption process

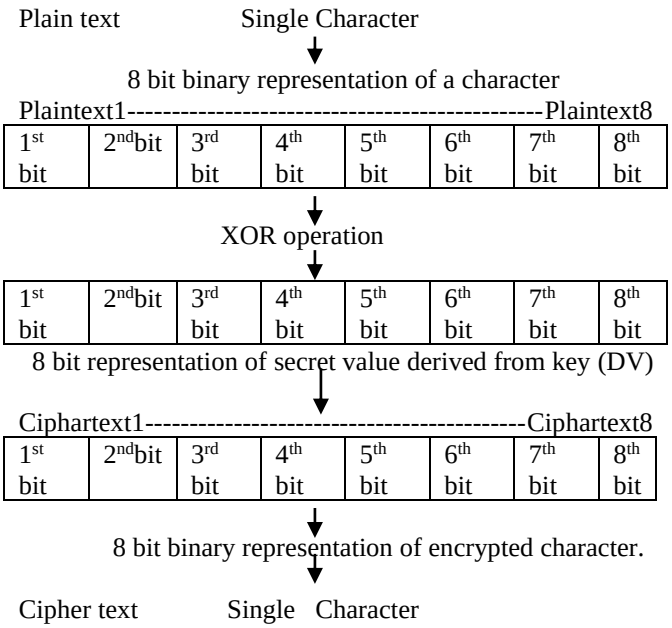

Figure 6: XOR operation between Plain Text and Secret Value.

XOR operation is performed between plain text block (8 bits) and the block (8 bits) of the secret value derived from the private-key (DV). We get the final encrypted single character from the encrypted block. In this way all the characters of the plain text are encrypted and those encrypted characters are stored into the cipher text file. The file is sent to the receiver with the secret private-key file. Fig-7 demonstrate the total XOR procedure between plain text and the secret value derived from the private-key where PT is the plain text, DV is the Secret Value derived from the Private-key, and CT is the cipher text.

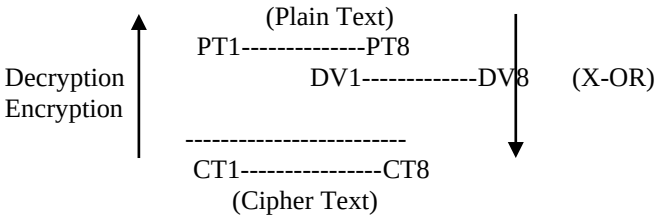

Figure 7: Block wise XOR operation between Plain Text and Secret Value derived from the private-key.

Step D.1- Perform XOR operation between array PLAINTEXT and DERIVEDVALUE and the final encrypted value is stored in array ENCRYPTED with dimension 8.

Step D.2- The binary value of the array ENCRYPTED is converted into corresponding ASCII value. Then the corresponding character is generated from the ASCII code and the character is stored into cipher text file.

Example- We have read a character 'C' from the inputted file and generate the plain text in step-1 and secret value has been derived from the private-key in the step-3. Now we perform the XOR operation between the plain text and the secret value derived from the private-key. Fig-8 represents the XOR procedure

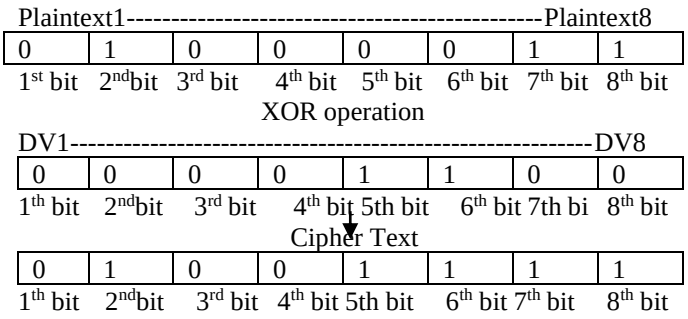

Figure 8: Generation of Cipher Text by block wise cumulative XOR operation between Plain Text and Secret Value.

Corresponding ASCII value which is 79 is generated from this 8 bit binary string (Cipher Text).Then corresponding character 'O' is generated from this ASCII value. In this way all the characters of the plain text are encrypted and the corresponding encrypted character are generated. Those characters are stored into the cipher text file. The encrypted file is sent to the receiver with the secret private-key file.

III. DECRYPTION PROCESS

E. Conversion of Cipher Text into Predefined Format:

Read each of the encrypted character from the cipher text file and store it into 8 bit binary format into an array CIPHARTEXT with dimension 8.

F. Formation of Secret Value from Private-key for decryption:-

Derive the secret value from the private-key by using step-c and store the 8 bit binary value in the array DERIVEDVALUE with dimension 8.

G. XOR operation And formation of Decrypted Text file

Step-G.1 Here array CIPHERTEXT is used for decryption. Perform XOR operation between CIPHERTEXT and DERIVEDVALUE and the final decrypted value is stored in array DECRYPTED.

Step G.2- The binary value of the array DECRYPTED is being converted into corresponding ASCII value. The corresponding characters are generated from the ASCII code and the character is stored into decrypted text file.

Example- we have read a character 'O' from the cipher text file. The ASCII value of 'O' is 79. We convert 79 in 8 bit binary representations and store it into array CT. The secret value has been derived from the private-key in the step-c. Now we are going to perform the XOR operation between the cipher text (CT) and the secret value derived from the Private-key (DV). Fig-9 represents the XOR procedure

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

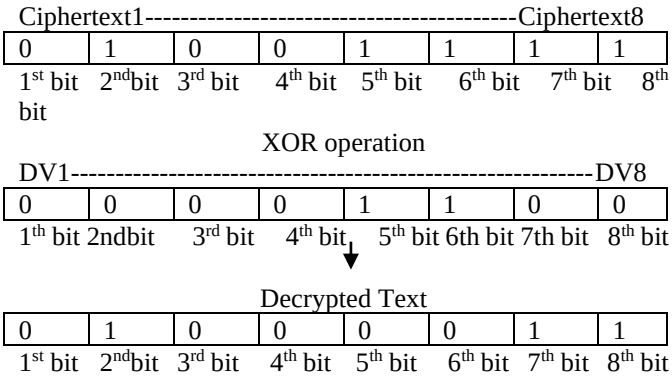

Figure 9: Generation of Plain Text by block wise XOR operation between Cipher Text and Secret Value derived from the key.

Corresponding ASCII value which is 67 is generated from this 8 bit binary string (Decrypted Text).Then corresponding character which is 'C' is generated from this ASCII value. In this way all the characters of the cipher text file are decrypted. Those characters are stored into the decrypted text file and receiver will able to get the plain text.

IV. EXPERIMENTAL RESULT

Here we have displayed the content of the plaintext, the corresponding encrypted content of the cipher text and the corresponding content of the decrypted file. Table-1 demonstrates the entire content of the source files, encrypted file and the decrypted file.

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

Content of the source file (f.txt)	Content of the Encrypted file (f_en.txt)	Content of the Decrypted file (f_de.txt)
12345678 tfeeygewhfwej BHGDFRJH KMH =]['/.,@#$% ^&*()_+|{} ":?><	=>!8'$'4 ftwwiw wydtywx44LFUJJ^ FFY_EF ?-MK5 +?< T/*)N*&&%& %Q9lus,(1..	12345678 tfeeygewhfwej BHGDFRJHK MH =]['/.,@#$%^ &*()_+|{}":? ><

We have used ,+,- 10,2 as base operator, operator between N(0) and N(1), base value and number of time base operator is being executed. The encryption is done by using the secret derived value from secret private-key which is dedicated and may or may not distinct for each character of the plain text. The private-key value is depend on the number of 'o' and '1' of a single character in the plain text. There are N numbers of may or may not be distinct private-key values if a plain text have N number of characters. The encryption or decryption has taken 14954 milliseconds.

We have executed our program on 15 number of files of different types (*.com,*.txt,*.exe,*.sys and *.dll). We have taken 3 numbers of files of each type. The Execution results are being displayed in the following Tables.

Table II: - Execution Result for *.com files

Source File name	Source File size (Byte)	Encrypted file size (Byte)	Encryption / decryption time (Mille seconds)
loadfix.com	1131	1131	**113547**
README.COM	4217	4217	**224047**
diskcomp.com	9216	9216	**391781**

Table III: - Execution Result for *.txt files

Source File name	Source File size (Byte)	Encrypted file size (Byte)	Encryption/ decryption time (Mille seconds)
ReadMe.txt	286	286	**113906**
LICENSE.TXT	4829	4829	**323500**
TechNote.txt	9232	9232	**579831**

Table IV: - Execution Result for *.exe files

Source File name	Source File size (Byte)	Encrypted file size (Byte)	Encryption/ decryption time (Mille seconds)
WINSTUB.EXE	578	578	**62359**
mqsvc.exe	4608	4608	**213578**
label.exe	9728	9728	**410656**

Table V: - Execution Result for *.sys files

Source File name	Source File size (Byte)	Encrypted file size (Byte)	Encryption/ decryption time (Mille seconds)
VIAPCI.SYS	2712	2712	**148297**
rootmdm.sys	5888	5888	**273968**
sffp_mmc.sys	10240	10240	**480703**

Table VI:- Execution Result for *.DLL files

Source File name	Source File size (Byte)	Encrypted file size (Byte)	Encryption/ decryption time (Mille seconds)
iconlib.dll	2560	2560	**130156**
KBDAL.DLL	6656	6656	**286015**
panmap.dll	10240	10240	**592031**

If we are looking to the relation between encryption or decryption time and the source file size, then we get almost a straight line in figure 10. This is a graphical representation of the relation between the encryption time and the source file size. That means the relation is linier. As the source file size increase the encryption time increases and vice versa. The encryption or decryption time do not depend on the type of the file.Fig-10 represents the relation between the encryption time or decryption time and the source file size.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

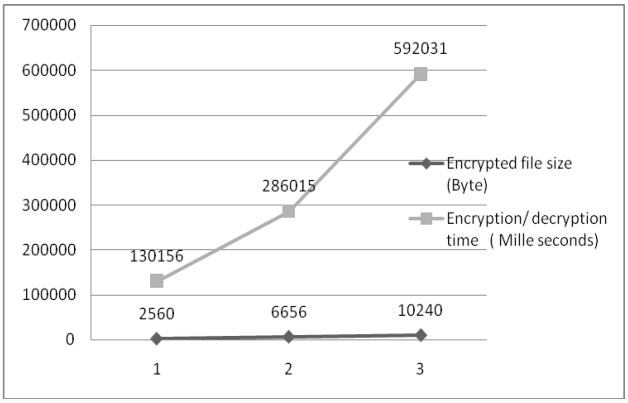

Figure 10: Relationship between Encryption time and source file size

The Pearsonian Chi-square test has been performed here to decide whether the sample of encrypted files may be supposed to have arisen from a specified population. The Pearsonian Chi-square is defined as follows: $X2 = \Sigma \{(f\,0 - f\,e)2 / f\,e\}$ Here $f\,e$ and $f\,0$ respectively stand for the frequency of a character in the source file and that of the same character in the corresponding encrypted file.[5]

If the Chi-square value is a higher one that means there is a higher frequency in source file and a lower frequency in encrypted file for a specific character. So, this proposed encryption procedure generate highly secured encrypted file. Table VII shows the Chi-square value for different types of file.

Table VII: Chi Square test Result on different files.

Source File name	Source File size (Byte)	Encrypted File size (Byte)	Chi-square Test Value
loadfix.com	1131	1131	31305486336.0
ReadMe.txt	286	286	276128064.00
WINSTUB.EXE	578	578	16863553536.0
VIAPCI.SYS	2712	2712	15576862720.0
iconlib.dll	2560	2560	18683279360.0

V. CONCLUSION

In this paper a new private-key encryption scheme is proposed which is based on bitwise XOR operation between the plain text and the secret value derived from the private-key.

The secret value is being calculated depending upon the number of '0' and '1' of a single character in the plain text and the different operators inputted by the user. There is several numbers of secret key values for a plain text file as the number of '0' and'1' are different for several numbers of distinct characters in the plain text file. Thus provide a great security.

Besides this, a user can also do the encryption or decryption by using different types of operators in the private-keys allotted for specific numbers of user defined blocks in a plain text file. So the security is increased as different keys are being used for encryption for different portion of plain text file, The size of the encrypted or decrypted file is same as of the plain text file. So we don't need any additional memory for storing the encrypted or decrypted file.

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

REFERENCES

1. J. K. Mandal, S. Dutta, "A 256-bit recursive pair parity encoder for encryption ", Advances D -2004, Vol. 9 nº1, Association for the Advancement of Modeling and Simulation Techniques in Enterprises (AMSE, France), www. AMSE-Modeling.org, pp. 1-14
2. William Stallings, Cryptography and Network security: Principles and practice (Second Edition), Pearson Education Asia, Sixth Indian Reprint 2002.
3. Atul Kahate (Manager, i-flex solution limited, Pune, India), Cryptography and Network security, Tata McGraw-Hill Publishing Company Limited.
4. Mark Nelson, Jean-Loup Gailly, The Data Compression Book. BPB Publication
5. Saurabh Dutta, "An Approach towards Development of Efficient Encryption technique", A thesis submitted to the University of North Bengal for the Degree of Ph.D., 2004.

AUTHOR PROFILE

Ramkrishna Das, M.Tech (Computer Science of Engineering) MCA, BCA, Research experience 1 year, Seminar Attend-2, Foreign Journal Publications – 2. Specialization: Visual cryptography, Information Security and Cryptology

SAURABH DUTTA, PhD. (computer science), MCA, B.Sc. (Mathematics), Publication of book-2, Research Papers publications: 45, Foreign Journal Publications - 13 Peer-Reviewed International Conference Papers - 8, Indian Journal Publications -2, Published/Accepted in International Conference -7, others -3.National Conference Publications -12. Research Experience- 12 years Specialization: Information Security and Cryptology Member-IACSIT, IJIST, IEDRC, , IJISMRD, IJMCAR, IJCSS, IJMER, ISOC, IAENG, IJCSI. Reviewer-(IJNS), Taiwan AMSE, France, IJACSA, ICCIA, ACCT12, Award- Enlisted in the directory of Marquis Who's Who in the World in its 2010 edition. Biography selected for inclusion by ABI (American Biographical Institute, Inc.) in the list of International Profiles of Accomplished Leaders in its 2011 edition

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)