Content Based Double Encryption Algorithm Using Symmetric Key Cryptography

4 authors:

Sourabh Chandra
Calcutta Institute of Technology, Kolkata
40 PUBLICATIONS **383** CITATIONS

SEE PROFILE

Bidisha Mandal
Centre for Development of Telematics (C-DOT)
7 PUBLICATIONS **55** CITATIONS

SEE PROFILE

Sk Safikul Alam
Jubilant Life Sciences
37 PUBLICATIONS **381** CITATIONS

SEE PROFILE

Siddhartha Bhattacharyya
VŠB-Technical University of Ostrava
467 PUBLICATIONS **4,061** CITATIONS

SEE PROFILE

International Conference on Recent Trends in Computing (ICRTC 2015)

Content based double encryption algorithm using symmetric key cryptography

Sourabh Chandra[a*], Bidisha Mandal[b], Sk. safikul Alam[c], Siddhartha Bhattacharyya[d]

[a,b,c] CSE Department, Calcutta Institute of technology, Kolkata, India

[D] IT Department, RCCIIT, Kolkata, India

Abstract-

With the crucial growth of technology, data security over the network and internet has achieved immense of prominence today and achieving good security is always a talk of a good security method being in place. Therefore, there is need of better security method with better efficiency in order to increase the security and authenticity and to efficiently decrease computational complexity. Although there are many symmetric key algorithms, we proposed a content-based algorithm, which follows the Symmetric key cryptography method. This is an algorithm implementing binary addition operation, a circular bit shifting operation and folding method and as symmetric key cryptography needs the transmission of the secret key along with the ciphered text through the network, a deep concern has given to make the key secure.

1. Introduction

Sending or sharing many commercial and confidential data through the network and uses of internet, social network and online storage are the common part of today's civilization. Cryptography[3] is a technique to secure data from unauthorised access when they transmitted over the network. Cryptography achieves the security goals- confidentiality, integrity, authenticity and non-repudiation by two operation called encryption and decryption[4]. The original text, which is to be send over the network, first transferred into a codified non-readable text, called cipher text using some key[1,2] (generally a set of values or string of symbols). This technique is known as encryption. Decryption is just reverse operation of encryption where we retrieved the original text from the cipher text. Key can be of two types' private key and public key. Public key is accessible by anyone who want to encrypt the text and it need not to be secret. Private Key, a strictly secure key, is use to decrypt the cipher text and only known to the intended receiver. Cryptography is explicitly classified into Symmetric key cryptography and Asymmetric key cryptography depending on what types of keys are being used to encrypt or decrypt the data.
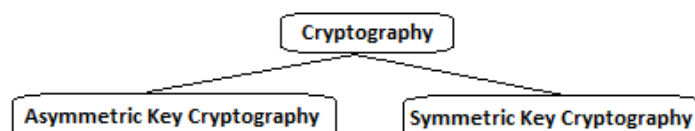
Fig. 1.Classification of cryptography

_____

*Corresponding author. Tel.: +91- 833-506-9527

Email-ID: sourabh.chandara@gmail.com

Asymmetric key cryptography use two different keys[2] i.e. public key and private key, which are complementary in function. The text, which is encrypted using public key, can only be decrypted using the corresponding private key[6]. Symmetric key cryptography uses a trivially related, identical key[5] instead of two key i.e. public key and private key for encryption and decryption. In Symmetric key cryptography sender encrypts the plain text using a secret key and receiver decrypt the cipher text using the same key. So there is a requirement to send the guarded key to the receiver along with the cipher text. Secrecy of information in symmetric key cryptography depends on the secrecy and size of secret key[1]. RSA, DSA, ECC are the example of Asymmetric key cryptography. The major application of Asymmetric cryptography is in the field of message authentication, digital signature scheme etc. The commonly used algorithms of Symmetric key cryptography are DES, 3DES, AES, BLOWFISH etc. Symmetric key cryptography is used by the security protocols for secure online communication. As example Transport level security (TLS) uses HMAC algorithm and Internet protocol security (IPSec) uses HMAC and DES algorithms[7]. Symmetric key cryptography is faster than the Asymmetric cryptography, often by 100 to 1000 times[6] and requirement of storage memory is less as compared to the Asymmetric Key Cryptography[1, 2].
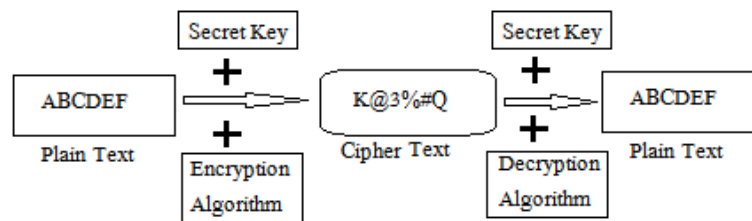

Fig.2. Symmetric key Cryptography

2. Proposed symmetric key cryptography method

We have proposed a content-based algorithm which implements bitwise circular shift operation and folding method. As we know in Symmetric key cryptography the secret key used for encryption needs to be shared to the authorised recipient to proceed for deciphering of the ciphered text. Therefore, the confidentiality of data highly depends on the secrecy of the key being used. We have given a closure look to this feature of Symmetric key cryptography by ensuring the secrecy of key using a Circular bit shifting operation and folding method. This algorithm encrypt the plain text two times to generate the secured ciphered text using bitwise binary addition operation.

2.1. Encryption Algorithm

Step 1.   Read the plain text.
Step 2.   Repeat the following step to encrypt the plain text to first encrypted text.
   a.   Count the length of each word of the plain text and Find the corresponding ASCII value of each letter.
   b.   Add each letter's ASCII value with the corresponding word length excluding spaces to generate the first encrypted text.
Step 3.   Take user input of any random number.
Step 4.   Repeat the following step to generate the cipher key.
   a.   Consider MSB of input number and generate its corresponding 8-bit binary number and count the number of 1 present in it, say n.
   b.   n bit Left circular shift the digit.
   c.   Store the shifted decimal value in an array.
   d.   Read the next bit and follow step 4 until LSB is being read.
   e.   Insert a negative number at the last of the array.
Step 5.   Do the following steps to generate the cipher text.
   a.   Add all digits of the input number.

      b. If produced result is not a single digit then repeat until a single digit formed.
         i. Add all the digits of produced result.
      c. Add the single digit value with each letter of the first encrypted text including spaces to generate the final cipher text.

Step 6. Send the cipher text, the Cipher key and the negative number to the receiver.

2.2. Decryption algorithm

Step 1. Read the shared negative number and cipher key.
Step 2. Do the following steps to generate the key from cipher key.
      a. Initially key=0.
      b. Repeat until the shared negative number, indicating end of array is accessed.
         i. Consider the first digit from cipher key; generate its corresponding 8bit binary number and count number of 1 present in the binary number, say n.
         ii. n bit Right circular shift the digit.
         iii. Add the shifted decimal value with key and store the result in key.
      c. If generated key is not a single digit then repeat until a single digit formed.
         i. Add all the digits of key and store it in key.
Step 3. Do the following steps to generate the plain text from cipher text.
      a. Read the cipher text and generate the ASCII value of each letter.
      b. Subtract the single digit key from the ASCII value of each letter including spaces to generate the first decrypted text.
      c. Count the length of each word formed in the first decrypted text.
      d. Generate the ASCII value of each letter of the first decrypted text and subtract the corresponding word length from it.
Step 4. Plain text generated.

3. Illustration of proposed method

The proposed encryption and decryption algorithms are illustrated with an example.

3.1. Encryption

Suppose "Hello! Bob use my ID: Alice@202." is the input plain text. To obtain the secrecy we encrypt the plain text twice. First we encrypted the text (excluding the spaces) using a simple addition method on the ASCII Value of each character with the length of corresponding word. Then encryption has been done by taking a random number as input which produce the encryption key using folding method as follows-
Let's the random number 462 taken from the user.

- 1st round: 4+6+2=12
- 2nd round: 1+2=3

Encryption Key: 3

How the plain text is encoded using the encryption key is illustrated as follows-

Table 1. String Character Table

Character fetched	Length of word	ASCII value	ASCII after addition with word length [Excluding Spaces]	1st encoded String	ASCII After addition with Encryption key [Including Spaces]	Ciphered text
H		72	78	N	81	T
e		101	107	k	110	n
l		108	114	r	117	u
l		108	114	r	117	u
o	6	111	117	u	120	x

!		33	39	'	42	*
		32	32		35	#
B		66	69	E	72	h
o		111	114	r	117	u
b	3	98	101	e	104	h
		32	32		35	#
u		117	120	x	123	{
s		115	118	v	121	y
e	3	101	104	h	107	k
		32	32		35	#
m		109	111	o	114	r
y	2	121	123	{	126	~
		32	32		35	#
I		73	76	L	79	O
D		68	71	G	74	J
:	3	58	61	=	64	@
		32	32		35	#
A		65	75	K	78	N
l		108	118	v	121	y
i		105	115	s	118	v
c		99	109	m	112	p
e	10	101	111	o	114	r
@		64	74	J	77	M
2		50	60	<	63	?
0		48	58	:	61	=
2		50	60	<	63	?
.		46	56	8	59	;
\0		0	0	\0	0	\0

Generated Ciphered Text: Tnuux*#Huh#{yk#r~#OJ@#NyvprM?=?;

3.2. Key encryption

Now to generate the shared link separate each digit of the random number and convert each digit to their corresponding binary number. Rotate left each digit according to the number of 1 present in their corresponding binary number and store the number in an array.

Input No: 462

Table 2. Key decryption table

Digit	Binary Number	No of 1 in binary number (n)	Binry value after n bit left rotation	Decimal value after n bit left rotation
4	00000100	1	00001000	8
6	00000110	2	00001100	12
2	00000010	1	00000100	4

Resulting array:

8	12	4

1st Shared link:

8	12	4	-94

2nd Shared link: -94

Receiver recieved the ciphered text " Tnuux*#Huh#{yk#r~#OJ@#NyvprM?=?;" and two shared link, one array of the circular shifted digits of the taken input number used for 2nd encyption and another one is a non-ciphered number to spcecify the end of the array.

3.3. Decryption

Decryption is the reverse process of encryption to deciphered the ciphered text. Reciver 1st fetch the non-ciphered number indicating the end of array and then fetch each digit from array and decipher them. Then receiver procced for the decryption of the ciphered text as following process.

Table 3. Key Decryption table

Decimal value after n bit left rotation	Binry value	No of 1 in binary number (n)	Binary Number after n bit right rotation	Decoded digit
8	00001000	1	00000100	4
12	00001100	2	00000110	6
4	00000100	1	00000010	2

Decoded number : 462

Applying folding method generate the decryption key as follows-

- 1st round: 4+6+2=12
- 2nd round: 1+2=3

Decryption key: 3

The process of decoding the fetched cipherd text using the decryption key is illustrated as follows-

Table 4. String charecter table

Fetched charecter	ASCII value	ASCII after subtraction with the decryption key [Including Spaces]	1st decrypted string	Length of word	ASCII after subtraction with the word length [Excluding Spaces]	Decipherd text
T	81	78	N		72	H
u	110	107	k		101	e
u	117	114	r		108	l
u	117	114	r		108	l
x	120	117	u	6	111	o
*	42	39	'		33	!
#	35	32			32	
H	72	69	E		66	B
u	117	114	r		111	o
h	104	101	e	3	98	b
#	35	32			32	
{	123	120	x		117	u
y	121	118	v		115	s
k	107	104	h	3	101	e
#	35	32			32	
r	114	111	o		109	m
~	126	123	{	2	121	y
#	35	32			32	
O	79	76	L		73	I
J	74	71	G		68	D
@	64	61	=	3	58	:

#	35	32			32	
N	78	75	K		65	A
y	121	118	v		108	l
v	118	115	s		105	i
p	112	109	m		99	c
r	114	111	o		101	e
M	77	74	J		64	@
?	63	60	<	10	50	2
=	61	58	:		48	0
?	63	60	<		50	2
;	59	56	8		46	.
\0	0	0	\0	0	0	\0

Deciphered Text after decryption:

"Hello! Bob use my ID: Alice@202."

4. Reults and discussion

4.1.Plain text

Fig.3. Plain text by sender

4.2. 1st encrypted text without space encryption

Fig.4. Encrypted text after first encryption
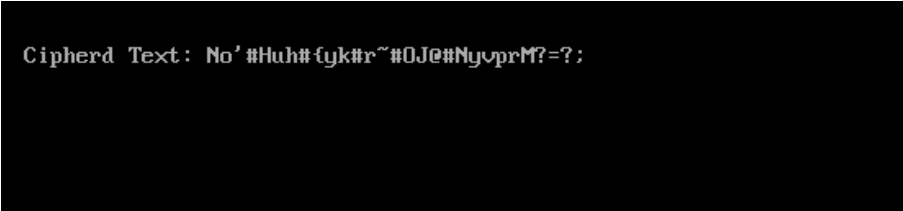
4.3. User input of any number

Fig.5.User input any number
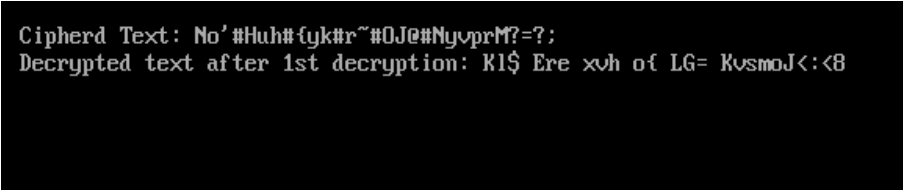
4. 4. Ciphered Text after 2nd encryption

Fig.6. Final cipher text
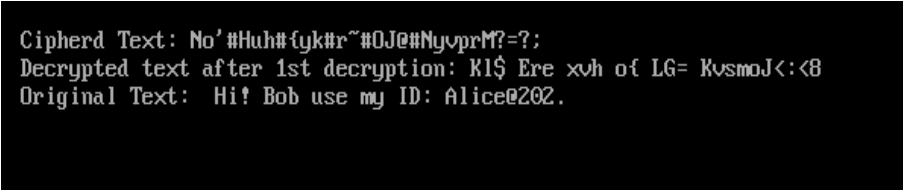
4.5. Decrypted text after 1st decryption

Fig.7. First decrypted text

4.6. Plain Text

Fig.8. Original text after second decryption

5. Conclusion

Our proposed algorithm follows a simple technique to encrypt the plain text, applying a binary addition operation on the content of the plain text twice. The integral part of this algorithm is its key encryption and decryption method. A special attention has given to secure the secret key better to say to secure the shared link using a circular bit shifting operation and folding method. It is hard to decipher the key and to decrypt the ciphered text without knowing the proper key. Here we work with string and 8bit binary data and focussed on symmetric key cryptography technique. Our future work will be focused to worked with Doc file, Text File and secure those files of data using Image Steganography technique.

6. References

1. Behrouz A. Forouzan,. *Cryptography & Network Security*. Special Indian Edition, Tata McGraw-Hill,2007.
2. Atul Kahate. *Cryptography and Network Security*. Second Edition, Tata McGraw-Hill,2009.
3. W. Stallings. *Cryptography and Network Security Principles and Practices*. Fourth Edition, Pearson Education, Prentice Hall,2009.
4. E Surya, C. Diviya. "*A Survey on Symmetric Key Encryption Algorith*". International Journal of Computer Science & Communication Networks, Volume 2(4), 475-477
5. https://www.princeton.edu/~achaney/tmve/wiki100k/docs/Symmetric-key_algorithm.html
6. http://technet.microsoft.com/en-us/library/cc962028.aspx
7. http://icourse.cuc.edu.cn/computernetworks/references/Internetworking%20with%20TCP-IP%20(Principles,%20protocols%20and%20architecture)%20vol1%204ed%20-%20Comer.pdf