Effective Secure Encryption Scheme [One Time Pad] Using Complement Approach

Sharad Patil[1]
Research Student, Bharti Vidyapeeth, Pune, India
sd_patil057@rediffmail.com

Ajay Kumar[2]
Modern College of Engineering, Pune(MS), India
ajay19_61@rediffmail.com

Abstract: Today every security expert worries about the security problems and tries to find out the secure solution. Because it is a challenging aspect of communications today which touches many spheres including memory space, processing speed, code development and maintenance issues? The problem, exaggerated when it comes to dealing with lightweight computing devices. This gives access for possible application to developers incase of have multiple encryption technologies through a framework of common interfaces , methods and classes. The one time pad encryption method is a simple and reliable cryptographic algorithm whose characteristics make it attractive for communication with limited computing devices. The major difficulty of the one-time pad is key distribution. In this paper, we present an implementation of one-time pad with **Binary addition with 2's complements approach.** In this article we observed that the one time pad is unbreakable theoretically, hence the scheme useful for things like communicating with high-value spies . In this article we used the approach as a complement that make cipher difficult, ultimately makes attacker life difficult. Permutation techniques can be used in conjunction with other technique includes substitution, encryption function etc. for effective performance.

Keywords :- Cryptography, Cryptosystem, One-time pad, encryption, Decryption, Security.

I. Introduction

Does every body find that there is a real thing such as a perfect encryption algorithm? In the theoretical sense, there is a solution as OTP. One-time pads are unbreakable if used properly. However, such algorithms still are rarely used in practice. In this article, we'll discuss about the one-time pad, its strengths and weaknesses, and how one time pad encryption with 2's complement approach makes attacker life difficult…….

Perfect data secrecy may not be possible as it seems, particularly if potential attackers are given absurd amounts of time (and can run brute force searches in parallel). But, oddly enough, there is an encryption algorithm that can't be broken if used properly: the one-time pad. What even strange is that the algorithm is incredibly simple.

The basic idea behind a one-time pad is that there's as much key material as there is text. The encryption operation can be simple modular addition. In computer-based uses, it is often XOR. In this article we try to put some added implementation after XOR , here by taking the complement of the binary addition generated bit[3,7]. That may produce difficult guessing to the attacker.

Here's the simple algorithm for one-time pads: For each plain text message, generate a random secret key. The key should be of exactly the same length as the plain text message. The cipher text is created simply by adding binary bit and takes complements. The following criteria may be considered that some requirement is crucial for perfect security. 1] Key must be absolute random 2] The length of the key must coincide with length of plain text 3] each key can be used only once . If at least any one of these requirement is broken then the cipher being stop perfectly secure and there are theoretical possibilities to break it [1,2,3,4].

II. Background

So how does it Work**?** The answer is that One Time Key encryption is used, the random key stream does not come from an algorithm or mathematical formula. Instead, it is obtained from a true random noise source and as the key stream is truly random, it cannot be reproduced.

This random key stream is then used for encryption, whereby each character of the plain text is mixed with one character of the random key stream. This results in a truly random cipher text that cannot be broken by any power in the world. The cipher text is then posted to the recipient of the message which reverses the process by using the same random key stream. Once the One Time Key stream is used for encryption or decryption, it is immediately destroyed. This guarantees that the same key cannot

be reapplied even by mistake. One Time Key encryption is a very simple, yet completely unbreakable cipher method. Over the years, today, high level of automation, high capacity storage media, continuous key protection and key sizes of more than 100 megabytes offer outstanding message security without sacrificing convenience[2,3]. The only proven unbreakable encryption method is available practically and theoretically.

We commented that, "one-time pads are theoretically unbreakable, but practically very weak. In contrast, conventional ciphers are theoretically breakable, while they are practically very strong." They are useful for things like communicating with high-value spies. The Moscow-Washington hotline utilized them, too. For ordinary computer usage, they're not generally practical.

The one-time pad encryption scheme itself is mathematically unbreakable. Therefore, obviously attacker will focus on breaking the key instead of the cipher text. Therefore a truly random key is essential. If the key is generated by a deterministic algorithm, the attacker could find a method to predict the output of the key generator. For instance a crypto algorithm is used to generate a random key, the security of the one-time pad is lowered to the security of the used algorithm. There is no longer mathematically unbreakable solution. If a one-time pad key, which is truly random, used more than once, simple cryptanalysis can recover the key. Indeed, the cipher text result of a truly random key is a truly random cipher text, by using the same key twice will result in a relation between the two cipher texts and consequently also between the two keys. The keys are no longer truly random and it's possible to recover both cipher texts by heuristic analysis. Another unacceptable danger of using one-time pad keys is more than once is the known-plaintext attack[2,3,4,7,8]. If the plaintext version of a one-time pad encrypted version is known to the user there is no problem for calculating the key. That content of one massages once is known all encrypted messages with the same key are compromised.

For successful cryptanalysis to follow, the slightest mistake in the implementation of one-time pad is sufficient. The negligent use of One Time Pad can be seen in history with many examples, the Venona project can cited as the most significant. The Soviet Intelligence has relied heavily on one-time pad encryption in the past, with good reason and success. Soviet communications record shows to be extremely secured. Moreover, during the Second World War, They had to create and distribute large quantities of one-time pad keys. The obvious compulsions and strategic circumstances lead in some cases to the distribution of more than two copies of certain keys. During 1940's, the United States and Great Britian analyzed and stored very large quantities of encrypted messages, intercepted during the war. Although it was like searching for needles in a hair, the top secret Venona project discovered the double use of some keys, leading to decryption of many messages in the years even after the war. However Venona was very crucial in solving spy cases like Rosenbergs and Cambridge Five. Though Venona is referred to as the project which broke Soviet one-time pads, they never actually broke, but exploited implementation mistakes[2,3,4,7,12]. This shows the importance of following the basic rules of one-time pad.

Due to their deterministic properties, software random number generators will never provide absolute security. The effective way to improve dramatically the security is to combine the generator output with one or more other generators. This makes analysis of the output more difficult. However, the output stream will always depend on the seed or initial state of the generator(s) and a good CSPRNG with a very large truly random seed will be practically impossible to break, which never achieve the theoretical absolute security, as described by Shannon[6].

Mathematically one-time pad encryption scheme itself is unbreakable. So, the attacker will focus on breaking the key and not the cipher text. Therefore truly random key is essential. With the help of key which is generated by deterministic algorithm helps attacker to predict method of output of key generator[4], If for a crypto algorithm is used to generate a random key, the security of the one-time pad will be lowered to the security of the used algorithm and mathematically remains unbreakable. If a one-time pad key, is used more than once, simple cryptanalysis can recover the key even if it is truly random. Indeed, although the cipher text result of a truly random key is a truly random cipher text, using the

same key twice will result in a relation between the two cipher texts and consequently also between the two keys. Here the keys don't remain truly random and it's possible to recover both cipher texts by heuristic analysis. Another danger of using one-time pad keys more than once is inviting the known-plaintext attack. The plaintext version of a one-time pad encrypted version once known, There is no problem to calculate the key. That is, if the content of one message is known, all messages that are encrypted form with the same key are also compromised. So the slightest mistakes in the implementation of one-time pad key give rise to successful cryptanalysis[3,4].

The question remains of mathematical security. Encryption is divided into symmetric and asymmetric. For encrypting and decrypting the traditional symmetric encryption uses the same key. It creates the problem of secure key distribution. While asymmetric public key encryption uses a one public key for encryption and another private for decryption, a public key used to encrypt message from sender and a private key to decrypt massage at the receiver. We can share your public key with everyone except being able to decrypt which they encrypted with that public key. It is magnificent because we no longer have to securely exchange secret keys[4]. Miserably, a public key algorithm is not encrypting our data. Even being quite simple and straightforward, its process is very slow and computational very heavy to do that. For traditional symmetric algorithm, we use a random key to encrypt the actual data. Further, asymmetric algorithm and a public key are encrypted with random key, which is provided by the receiver. Thus, the data and the encrypted key are sent to the recipient[4,7,8]. For the process of decryption at the receiver end, the receiver uses it own private key and uses the retrieved random key to decrypt the actual data.

So, the question is whether the one-time encryption still in use or not? The answer is yes! Because It's the only proven system which is mathematically unbreakable. This system also provides the real long-term protection[4]. Thus one-time pad is so basic and transparent, so that it can be trusted and useful. There are different more practical systems such as computational security, absolute security and unconditional security, but in some specific conditions, absolute security and unconditional security have got the prime importance. That is the reason why one-time encryption was used very often, there were other cryptographic solutions available and still it will be used in future also. The practical security and the reasonable privacy which is needed to our economy and daily life is provided by modern crypto algorithms. In some cases you need everlasting absolute security and privacy which is only possible with one time pad. In today's world of electronic and software one-time key encryption solutions have capacity of processing large amount of data. Hence the manual one-time pad has a future as a low tech, which can be used easily and cheaply enough to encrypt small text messages. Therefore commoner has brought alive one time pad a absolute security and privacy in his reach, this being his basic right[4,10,11]. So one cannot say one-time pads are thing of the past.

III. Methodology

We used here simulation methodology to check the encrypted text for alphabets. Here we first create the table of alphabet ATO Z with number 1 to 26 shown in Table-1 and then write the plain text alphabet with equivalent decimal number given in the table , then decimal number converted into 6 bit binary format and supply the random number key with same length of plain text then perform binary addition , and get the sum , then After that take the 1's complement of the sum , write the binary to decimal equivalent of result in term of Alphabet and send the alphabet [cipher text] to the recipient . then recipient receive it convert the equivalent 6 bit binary, take the complement and add with same key will get the plain text . Here we add the decimal number as random key at the first , that treat as the key , recipient convert it into the binary format actually cipher text includes decimal number with alphabet i.e. Decimal number indicate the key while alphabet indicate actual message .which can be helpful for the recipient to identify the actual string or message. Following Figure shows Encryption and Decryption process at both end.

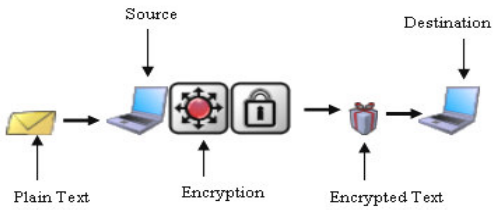

Fig.1-Encryption Process by Random Key generation (At source)

Fig.2-Decryption Process at destination

Here the practical information is used that can be used to setup one time pad encryption system. For easy adoption the steps is given 1] create the key 2] format the massage 3] Encrypt the massage 4] Decrypt the massage . The Figure 1 and Figure 2 shows the easiest map for the process[9]. The proposed method of encryption is shown in Figure -1. This process is done at source host. The part of decryption process will be performed at the destination end. that shown in Figure -2

IV. Algorithm

Step-1 Consider the plain text [massage] , write the decimal value of plain text character from table.

Step-2 Convert decimal value into equivalent Binary

Setp-3 Generate random key not less than 6 bit

Step 4 Perform binary addition

Step 5 Take the 1's complement of binary sum

Step 6 if complement is greater than 26 , then subtract 26 from complement .

Step 7 Convert decimal [which is less than 26] into alphabet from table and then add decimal equivalent of random key and Send as cipher text .

Step 8 The recipient perform process for decryption

Step 9 if the value is greater than 26, the subtract and subtracted number less than 26 , then write the equivalent decimal and then related alphabet i.e. the Plain text will be recover at the destination

Step 10 end

Table-1 Sample Decimal Number chart

Alphabet	Number	Binary 6 bit Equi. Decimal	Alphabet	Number	Binary 6 bit Equi. Decimal
A	1	000001	N	14	001110
B	2	000010	O	15	001111
C	3	000011	P	16	010000
D	4	000100	Q	17	010001
E	5	000101	R	18	010010
F	6	000110	S	19	010011
G	7	000111	T	20	010100
H	8	001000	U	21	010101
I	9	001001	V	22	010110
J	10	001010	W	23	010111
K	11	001011	X	24	011000
L	12	001100	Y	25	011001
M	13	001101	Z	26	011010

V. Implementation

In this scheme we consider text like " GOD" and same random key for all bit i.e 17 [010001] and then access equivalent binary and decimal value of text G,O,D from the table above.

Encryption Process

G =7 [Decimal] and Binary 000111
O = 15 001111
D =4 000100

and Random Key = 010001 , Here we consider same random number key

	000111	001111	000100
	010001	010001	010001
Perform Binary Addition	011000	100000	010101

Take 1's Complement 101000 (39) 100000 (31) 101011 (42)

Here sum is more than 26 so then subtract it [39-26= 13 i.e m from table , so similarly next alphabet e,p respectively so cipher text is **"17mep"** Send this cipher text to recipient , [here **17** is the random key value] write equivalent decimal and its binary value of cipher text

m= 001101 e=000101 p= 010000

Perform Decryption

	001101	000101	010000
	010001	010001	010001
Perform Binary	011110	010110	100001

Addition

Take 1's 100001 101001 011110
Complement 33 41 30

[if complented value more than 26 so that subtract it and write the decimal value and respective alphabet hence here we subtract 33-26= 07 i.e g form table in this way recipient get the plain text “god”

V. Analysis and Results

In this type of encryption system ,we took Alphabet and their Decimal Value and then converted the decimal into 6 bit equivalent binary number and same length of random generated key , then take the Binary Addition and take 1 ‘s complement of the sum , the result is more complex and it’s analysis is difficult for attackers if we consider 1’s complements. If we can use similar 6 bit random key for all bit ,instead of diff. random key, even the scheme work smoothly and Hence ,We come to conclusion that by designing encryption method as one time pad with Binary addition with 1’s complement is more difficult for cracking . and not easily guessable to the attacker . In further research ,we would like to design the algorithm on modular arithmetic with 2’s complements in binary addition, multiplication and division concepts.

V. Conclusion

Future Work : Presently our country India is advancing technically in the field of computer and communication . The need arise to develop software related to security of data and information. This algorithm has a full scope to enhance the security by using combining the different approaches such as EX-OR with 2’s complement, multiplication and division with complement approach and modular arithmetic function are also common. instead of using ASCII and other Code .

REFERENCES

[1] Ritter, Terry 1991. The Efficient Generation of Cryptographic Confusion Sequences. Cryptologia

[2] Douglas R, Stinson “ CRYPTOGRPHY Theory and Practice “ Second Edition .

[3] Charlie Kaufman st al. “ Network Security “ PRIVATE Communication in a PUBLIC World. , Prentice Hall of India Private Limited. 2003.

[4] Dirk Rijmenants, “Cipher machines and cryptology, the one-time pad” , “http://users.telenet.be/d.rijmenants/en/onetimepad.htm”

[5] Information Technology Journal 4(3) : 204-221, 2005

[6] Claude Shannon's " Communication Theory of Secrecy Systems" .

[7] Neal R. Wagner “The Laws of Cryptography: *Perfect Cryptography: The One-Time Pad* “

[8] Ritter, Terry 1991. The Efficient Generation of Cryptographic Confusion Sequences. Cryptologia “15: 81-139.

[9] “Modified One Time Pad Data Security Scheme: Random Key Generation Approach “ International Journal of Computer and Security Volume 3 issue 2 March/April 2009 Malaysia (Published) by Sharad Patil, Dr. Ajay Kumar:

[10] www.EFYMAG.com - February-2007

[11] www.zdnetindia.com

[12] www.sans.org