Volume 3, Issue 10, October 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Efficient and Secure Encryption Schema based on Random bit's (Rbits)

P.Penchalaiah[1],
[1] Research Scholar,
Department Of Computer Science,
V.S University, Nellore, A.P, India

K. Ramesh Reddy[2]
[2] Assistant Professor,
Department Of Computer Science,
V.S University, Nellore, A.P, India

Abstract: ***The main aim of this paper is to strengthen secured communication by introducing a new cipher, Rbits Cipher (Random Bits Cipher). We can treat this cipher as new approach or an enhancement to One-Time Pad (OTP) encryption scheme, which itself is mathematically unbreakable [0] and more complex in nature of attacker view. The goal of this paper is to develop the One-Time Pad encryption without any secret key overhead while transmission(since key is along as message) by using two algorithms, a Key Exchanging Algorithm and a Random Bit Generation algorithm. Hence forth the new cipher with the above said characteristics called as Rbits Cipher.***

Keywords: Cryptography, Rbits Cipher, One-Time Pad, Key Exchange, Random Bits, RSA, BBS, Encryption, Decryption

I. Introduction

In human's society, people have been very much concerned with the privacy of their communications. With the advent of computers in every field, the need for software tools for protecting files and other information stored on the computer became important. The necessity of information privacy has undergone major changes in the past and present times. There is no doubt that electronic communications have become one of the main pillars of the modern society and their ongoing boom *requires the development of new methods and techniques* to secure data transmission

A. Cryptography:

Cryptography is the practice and study of hiding secret information by encryption. *Encryption* is a process of conversion of data into a unintelligence form, called a cipher-text. *Decryption* is just inverse process, in other words, converting from unintelligible cipher-text back to plaintext. Modern cryptography intersects the disciplines of mathematics and computer science. Cryptographic techniques are needed for confidentiality (privacy) and authentication of digital data. There are two types of cryptic algorithms used in cryptography, namely *Symmetric-Key Encryption* (also known as single-key encryption, one-key encryption) and *Asymmetric Encryption* (Public Key Encryption).

B. Good Cipher Parameters

There is no cipher developed till now which guarantees 100% secure transmission (i.e. without loop holes, where these loop holes gives a possibility to break the code by deep cryptanalysis by a professional cryptanalyst). But we can develop Security algorithms by minimizing the risks and makes lead to optimality.

We can increase of optimality by specifying the robust factors for ***good cipher***, which Depends on the choice of the following parameters and design features[Error! Reference source not found.]:

Block size: The entire message is divided in to a fixed size segments. The size of one block increases/decreases robustness of algorithm.

Key size: The length of the key used to encrypt/decrypt.

Complexity: How much difficult to analyze.

Speed: Fast software encryption/decryption allows the cipher to follow by all the systems.

Ease of analysis: As the easy the algorithm to analyze as mush easy to break the code

C. One Time Pad

One well known realization of perfect secrecy is the One-Time Pad, which was first described by Gillbert Vernam in 1917 for use in automatic encryption and decryption of telegraph messages.

We can only talk about OTP if four important rules are followed. If these rules are applied correctly, the one-time pad can be proven to be unbreakable.

However, if only one of these rules is disregarded, the cipher is no longer unbreakable.

The Four Rules are...[0]

- *The key is as long as the plaintext.*
- *The key is truly random*
- *There should only be two copies of the key: one for the sender and one for the receiver*
- *The keys are used only once, and both sender and receiver must destroy their key after* use.

For practical application, the key used for one time pad cipher is a string of random bits, usually generated by a Crypto-graphically Strong Pseudo-Random Number Generator. How-ever for ultimate security, it is suggested to generate the

key by using the natural randomness. If the key is truly random an XOR operation based one–time pad encryption scheme is perfectly secure against cipher text-only cryptanalysis. We come to the point that if the hackers do not know the sender or receiver key, then the one time pad encryption scheme is 100 % secure. A one-time pad is essentially a pad of paper on which each page has a unique set of random letters. The sender and receiver have identical pads. Each letter on the pad is used to determine a single letter of the enciphered message. Since the letters on the pad are random, there is no formula that can be determined by studying the letters. Assuming that the pad is not compromised, and each page is used only once, then the OTP system is unbreakable. One-time pads are "information-theoretically secure" in that the encrypted message (i.e., the cipher text) provides no information about the original message to a cryptanalyst (except the length of the message).

II. Methodology

There is two basic requirements for developing Rbits Cipher are..

1. *Key Exchanging Algorithm (KE) and*
2. *Random Bits Generation Algorithm. (RG)*

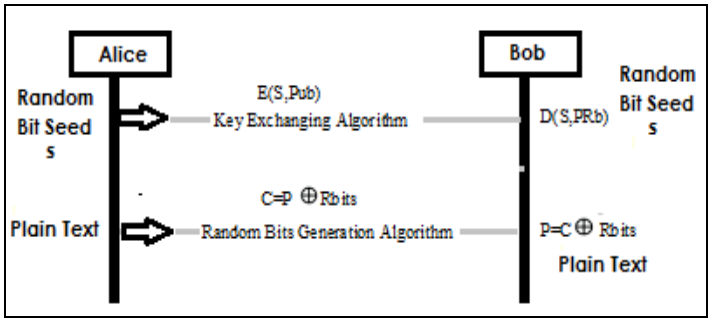

Figure 1

A. *Seed Exchanging and RG*

Alice/Bob selects Random seed(s) "S" which will be used for Random Bits Generation. The Random Bit Generation (RG) algorithm must be such way that it must generate same sequence of random bits for the same seed "S". Both Alice and Bob must follow same RG. Alice/Bob can use any Key Exchanging Algorithm (KE) to send the seed "S" to Bob/Alice.

The Key Exchanging Algorithm must maintain confidentiality and other security policies. Once Bob receives the seed 'S', Bob can able to generate the same Random bits as generated as Alice.

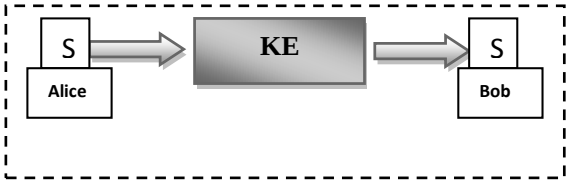

Figure 2

→*KE(S)*→

Where

KE =Selected Key Exchanging Algorithm.
S =Seed (For Random Bit Generation)

Here Random bits are generated at the both ends, but not carried along with message unlike legacy One Time Pad (OTP), which avoids transmission overheads. In Rbits Cipher seed 'S' is only transmitted instead of long key (random bits) which is as large as the message.

This feature frees up the communication parties from carrying large key along with the cipher text and causes a reduced traffic in the channel.

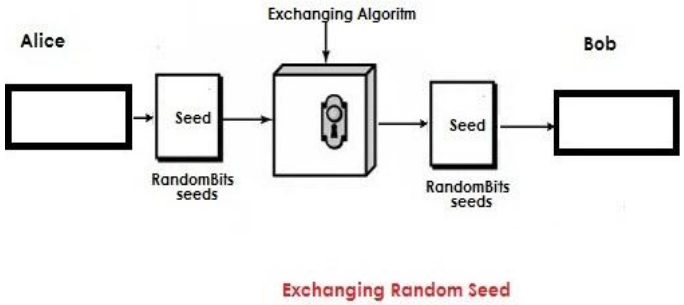

Figure 3

B. Encryption

Once the seed "S" has exchanged, Alice initiates encryption process. Alice converts plain text in to binary stream. The binary stream is divided in to fixed size of chunks for effective system performance. The size of the chunks can be from 8bit to 256bit or more can also allowed. Chunks can be configured by Alice and Bob as their wish.

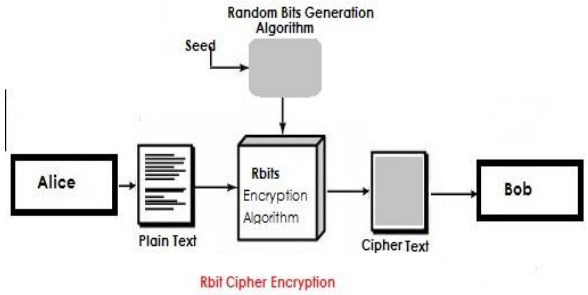

Figure 4

A chunk of binary stream is XOR with the same length of Random Bits, which are generated using seed "S" with a RG. This process is continued until last chunk. As the Random bits are unique for each chunk the generated cipher stream contains no statically relationship with the plain text. So Rbits Cipher does not have an easily analyzed functionality, which makes more difficult to cryptanalysis.

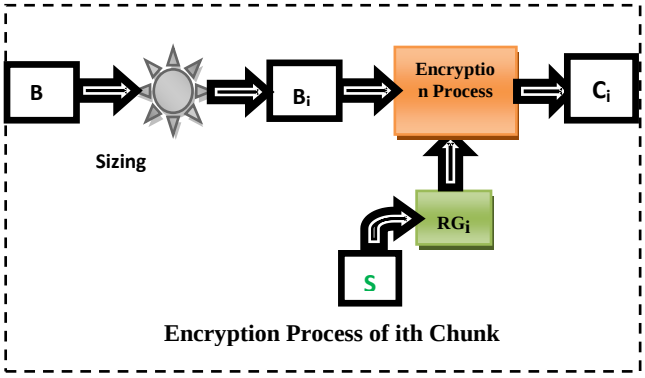

Figure 5

$C_i = E(B_i, RG_i(S))$

Where C_i = Cipher Stream of i^{th} chunk.
E = Encryption Process
B_i =Binary Stream of i^{th} chunk
RG_i =Random Bits Generation for i^{th} chunk
S =Random Bit Generation Seed.
B =Binary Stream of original Plain Text

C. Decryption

As Bob aware of the Random bit generation seed "S', Bob is also able to generate same random bits. Bob decrypts' each chunk of stream by XOR operation with the random bits by using the same seed "S'" and same RG algorithm. Bob continues until the last chunk of cipher stream. Once all chunks are decrypted the resultant binary stream again converted to plain text which is in readable format.

Detailed Rbits Decryption process will be explained in subsequent papers.

III. Algorithm

Algorithm: Rbits Cipher

Alice

P_a -Plaintext

B_a -Binary Stream of P

S_a -large Random Seed

KE_a -Key Exchanging algorithm

RG_a -Random Bit Generation algorithm

$Rbits_a$ -Random Bits generated using RG_a

Bob
KE_b -Key Exchanging algorithm (=$KE_{a)}$
S_b - Large Random Seed Received from Alice (=S_a)
RG_b -Random Bit Generation algorithm (=$RG_{a)}$
Encryption
C= E (Ba, Rbitsa)
Decryption
P= D (C, Rbitsb)

Where E - Encryption Function
D - Decryption Function.

IV. Implementation

Here we consider one simple example for wide understanding of Rbits Cipher algorithm.
In this paper we are going to explain Rbits Cipher Encryption process, by using RSA and BBS. Let us consider RSA as KE and BBS (Blum Blum Shub Algorithm) as RG.

A. RSA[Error! Reference source not found.] *(Rivest, Shamir and Adelman)*

RSA is the well known public key cryptosystem which is based on mathematical function. It is block cipher in which plain text and the cipher text are the integers between 0 and n-1 for some n.
It begins by selecting two prime numbers, p and q and calculating their product n, which is the modulus for encryption and decryption.
The algorithm of Rivest, Shamir and Adelman (RSA) crypto system is as follows:

1. Choose two large prime numbers *p & q*(p!=q)
2. Computer $n = p * q$.
3. $Ø (n) = (p - 1) * (q - 1)$.
4. Select integer e, $gcd (ø (n), e) =1; 1<e<ø (n)$
5. Calculate d such hat $de \bmod ø (n) = 1$
6. Public key PU= {e, n}
7. Private key PR={d,n}

Encryption: Cipher text: $C = M^e \bmod n$, where 0≤M<n
Decryption: Plain text $M = C^d \bmod n$

B.BBS (Blum Blum Shub Generator)

A popular approach to generating secure pseudorandom number is known as the Blum, Blum, and Shub (BBS) generator, named for its developers. It has the strongest public proof of its cryptographic strength. The procedure is as follows.

1. Select two large prime numbers, ***p*** *and* ***q***.
   (p, q both have a remainder of 'r' when divided by 'm'.)
   $$p \equiv q \equiv r \pmod m.$$
2. Let $n = p \times q$.
3. Choose a random number '***s'***, is relatively prime to '***n'***

Then the BBS generator produces a sequence of bits B_i according to the following algorithm:

$$X_0 = s^2 \bmod n$$

for i = 1 to ∞

$$X_i = (X_{i1})^2 \bmod n$$

$$B_i = X_i \bmod 2$$

The BBS is referred to as a **cryptographically secure pseudorandom bit generator.** *In other words input of the first k bits of an output sequence there is not a practical algorithm that can even allow you to state that the next bit will be 1 (or 0) with probability greater than 1/2*.

For all practical purposes, the ***sequence is unpredictable***

The security of BBS is based on the difficulty of factoring n.

C. Rbits Cipher Encryption Process

Exchanging of BBS Seeds set S={s, n}

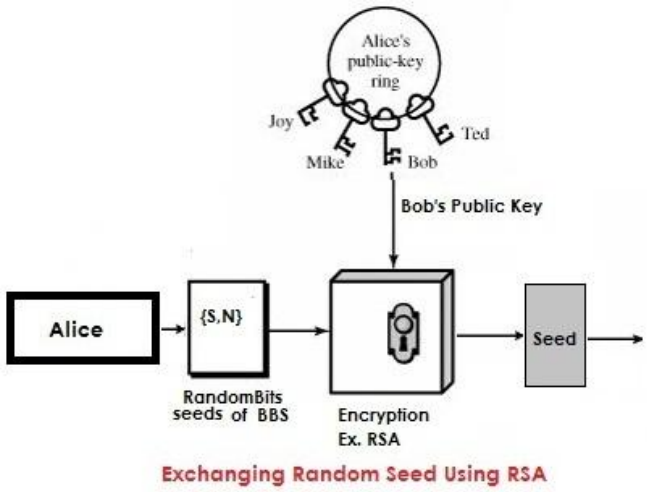

Figure 6

Alice/Bob selects Random seed of BBS, i.e. S=***{s, n}.***

Let Alice selected 'S'.

Alice and Bob generates their own pair of keys A= {PUa, PRa} and B= {PUb, PRb} .Alice and Bob publishes their public keys into say Alice's Public Key Ring (PKRa) and Bob's Public Key Rings (PKRb) and retains their private keys with themselves securely.

Alice and Bob follow RSA encryption/Decryption process. Alice encrypts the seed set 'S' by accessing Bob's public key from Alice PKRa and sends the resultant key to Bob. Bob Decrypts the seed set 'S" and retains it with him for BBS random bit generation.

Now Both Alice and Bob having same seed set S={s, n}.

Encryption

Alice finds equaling binary stream (B) for Plain Text (P).

The B divided into 'L' blocks each block having predefined size (chunks) called sizing process. (Block size of 64 bits has been considered a reasonable).

The ***sizing*** process as follows..

$B=b_1+ b_2+ b_3+\dots b_{L+} b_r$.**Where** $(b_r = b_{L+1})$.

$$B = \Big\| _{i=1}^{L+1} b_i$$

Where i= {1, 2, 3…L}

The symbol || (Pair of pipelines) indicates concatenation operation, i.e. B can regenerate from $\mathbf{b_i}$ by concatenation.

If the length of the B is multiple of configured Block size (like 64bit), then there won't exist any chunk like $\mathbf{b_r}$.

The $\mathbf{b_r}$ is the case where the length of the B not multiple of block size. The length of each $\mathbf{b_i}$ is same except $\mathbf{b_r}$ if exist.

L=Len (B)/ Bz.

Len(r) =Len (B) mode Bz.

Where L -Number of Blocks. (In sizing).

Len -Total Length of B.

Bz -Block Size (ex. 64bits)

Example:

B="10100000100010101001110010000110100100001000001010011000100000101001001010000010100100".

Len (B) =87.

BZ=8

L=87 / 8 =>10 Blocks.

Len(r) =87 mode 10=>7

The Above 'B' has the length of 87bits. So B will be sized in to '10' blocks and each block has 8bits (as Bz=8) .

'B' is not a multiple of '8', so it has $\mathbf{b_r}$ having 7bits.

By using BBS, random bits are generated **Rbits$_i$**, as same length of a chunk in question.

$$C_i=E\ (b_i,\ Rbits_i(S))$$

Encryption function can be any encryption technique. For Simplicity, here we are using XOR operation as encryption function.

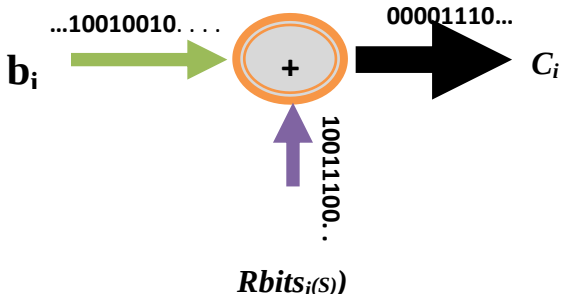

Figure 7

Rbits Cipher -Encryption Run Log.

Machine implementation of Rbits Cipher Encryption produced log during its execution.

----------LOG START----------
Accessing Key(BOB) ....
Accessing rsaN....
STREAM SIZE Init..128bits/block
ALICE :Start Key Exchanging.....
ALICE :Sending Request to BBS.....
BBS Init..
The Value Of BBS P is **4285997803**
The Value Of BBS Q is **3126909677**
The Value Of BBS N is **13401928005801439631**
The Value Of BBS S is **2773107377**
BBS Successfully Initialized
ALICE :BBS SEEDS Generated.
ALICE Is Ready To Encrypt SALT for BBS....
ALICE :BBS SEEDS Encryption Started.
ALICE :Request **To BBS S-SEED Encryption**.
9687748098216470662266302830473425172266302830473425175143369684038853574505345098272231590214039400765366236852266302830473425175143369684038853574226630283047342517226630283047342517
ALICE: BBS S-SEED Encryption Done.

ALICE :Request **To BBS N-SEED Encryption**.
5053450982722315902514336968403885357423292225731354497761403940076536623685505345098272231590243556488732909788849687748098216470661146554043818696174140394007653662368514039400765366236859611850457325430501146554043818696174140394007653662368550534509827223159022329222573135449776514336968403885357445564887329097888413215923209859021051433696840388535744355648873290978884132159232098590210514336968403885357450534509827223159025053450982722315902514336968403885357423292225731354497761403940076536623685505345098272231590243556488732909788849687748098216470661146554043818696174140394007653662368514039400765366236859611850457325430501146554043818696174140394007653662368550534509827223159022329222573135449776514336968403885357445564887329097888413215923209859021051433696840388535744355648873290978884132159232098590210514336968403885357450534509827223159025
ALICE :BBS N-SEED Encryption Done.
Sending S ....
Message Sent....
Sending N ....
Message Sent....
ALICE :End Key Exchanging.....
ALICE Is Ready ....
ALICE :Plain Text :
There Is A Bomb In The College, Explodes @10.30 AM
ALICE :Rbit Cipher Encryption Process Started..
ALICE :Plain **Binary Stream**:
10101000110100001100101011100100110010100100000010010010111001100100000010000010010000001000010011011110110110101100010001000000100100101110111000100000010101000110100001100101001000000100001101101111011011000110001100101011001110110010100101100001000000100010101111000011100000110110001101111011000110010101110011001000000100000000110001001100000001011100011001100110000001000000100000100101001101
ALICE :**Random Bits Generation** ...
10000110100011011101010001000011000001111001010000110101111110111111011110101111001111010000001111100001111100001010001001101101100001101000110111010100010000110000011110010100001101011111101111101111010111001111010000001111100001111100001010001001101101100001101000110111010100010000110000

00111100101000011010111111011111101111010111100111101000000111110000111110000101000100110110000110100011011101010001000011000001111001010000110101111110111111011110101110011110100000011111000011110000101000100110110000110100011011101010001000011000001111001010000110101111110111111011110101110011110100000011111000011110000101000100110110100111100101000111101010011111001101110101001010011100011101101110010110101111101100001110111111001010011001100010110100010100010100011001010011101011110101110101111001110101011111010000101111100101011100111111110001110110110111001011110101111101100011101111011001011011000010001011010000101000001011

ALICE :Rbits Cipher Encryption Process Finished..

ALICE :**Rbits Cipher Binary Stream**:

00101110010111010001111010100111110011011101010010100111000111011011011100101101011111011000011101111100101010011001100010110100010100010100011001010011101011110101110101111001110101011111010010100101110111111100101110010010010111100111010111110100010111010000110001111101001101001001101111011000101011100111111111000111011011011100101111010111110110001110111101100101101100001000101101000010100000101100

Sending Cipher Text....

Cipher Text Sent....

----------LOG END----------

BUILD SUCCESSFUL

V. Conclusion

Rbits Cipher is more simple for user point of view but very difficult for attacker point of view. Here we are following four rules of OTP [0], If these rules are applied correctly, the one-time pad can be proven to be unbreakable [0] . There are no heavy computational requirements to encrypt/decrypt (like RSA, DES, and AES where major part of the algorithms are involved in generation of 'n' keys for 'n' rounds and requires heavy exponential computations which reduces the performance by consuming more time). The entire strength of this Rbits cipher depends up on the selection of KE and RG

References

[1]. Sharad Patil Ajay Kumar "Implemented Encryption Scheme (One Time Pad) using 9'S Complement" International Journal of Advanced Research in Computer Science Volume 1, No. 2, July-August 2010 PP 48-50

[2]. Md. Mizanur Rahman "Any File Encryption by Translating ASCII Value of Characters" International Journal of Advanced Research in Computer Science Volume 3, No. 2, March-April 2012 PP41-43

[3]. Dr. S. Udaya Kumar , Ravindra Babu Kallam "An Enhanced RSA Public key Cryptographic Algorithm" International Journal of Advanced Research in Computer Science Volume 2, No. 5, Sept-Oct 2011 PP 497-499

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

[5]. Neal R. Wagner "The Laws of Cryptography: Perfect Cryptography: The One-Time Pad".

[6]. Avanish Kumar Singh , Amit Kumar Pathak "Encryption Techniques as Security Tools: A Technical Review" International Journal of Advanced Research in Computer Science Volume 3, No. 7, Nov-Dec 2012 PP 269-2073

[7]. Ayushi, (2010), A Symmetric Key Cryptographic Algorithm, International Journal of Computer Applications (0975 - 8887) Volume 1. No. 15.

[8]. Bruce Schneier, "Beyond Fear", Springer-Verlag, New York, 2006

[9]. Saikat Ghosh Sukalyan Som A Survey of Traditional or Character Oriented Symmetric Key Cryptography Volume 2, No. 4, July-August 2011 International Journal of Advanced Research in Computer Science PP 147-151

[10] Neal R. Wagner "The Laws of Cryptography: Perfect Cryptography: The One-Time Pad".

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

