ISSN 2356-5608

Implementation of Blum Blum Shub Generator for Message Encryption

Belmeguenaï Aïssa
Laboratoire de Recherche en Electronique de Skikda
Université 20 Août 1955- Skikda
BP 26 Route d'El-hadaeik
Skikda, Algeria
belmeguenaiaissa@yahoo.fr

Mansouri Khaled
Département d'Electronique
Université Badji Mokhtar
Annaba, Algeria
mansouri.khaled@voila.fr

Grouche Lakhdar
Laboratoire de Recherche en Electronique de Skikda
Université 20 Août 1955- Skikda
BP 26 Rout El-hadeik
Skikda,Algraia
lakhdar.grouche@yahoo.fr

***Abstract*—The cryptography areas of application are traditionally very diverse. The booming industry nowadays is the secure transmission of data (images, text, video, ...). In this research work, a novel encryption scheme for secure transfer message is developed. The system allows to crypt and decrypt personal data . The proposed encryption message scheme is simple and highly efficient. The system has been implemented using stream cipher algorithm based on BBS generator. The system has been tested using visual test, histogram analysis, correlation coefficient analysis and key sensitivity analysis. The testing and validation of the system is also reported.**

***Index Terms*—Keystream Generator, Message Encryption, BBS Generator, Stream Cipher, Security Analysis.**

I. Introduction

Nowadays the message communications have a vital role in our daily, and with the constant evolution of digital networks they may move from one end of the world to the other end in seconds and few clicks, however, transmission of messages via Internet raises a significant number of problems that are not yet solved. We, for example, security, confidentiality, integrity and authenticity.To ensure that a third person cannot access the content of these messages, the data must be placed in place where a mechanism is making incomprehensible transmitted message that nobody can read its contents except those legitimate. This can be done by setting up a cryptographic system.

Many techniques [1], [2], [3] have been developed in recent years. Our objective is to use the stream cipher for protecting personal data for users who desire to protect their confidential data. In this research work the novel encryption scheme using Blum Blum Shub (BBS) generator [4] for message was introduced. The BBS generator appears to be as secure as other encryption technologies tied to the factorization problem, such as RSA [5] encryption. Finally, this algorithm is robust and very sensitive to small changes in key so even with the knowledge of the key approximate values; there is no possibility for the attacker to break the cipher.

II. Stream Cipher

A stream cipher is a symmetric encryption algorithm which encrypts individual digits of plaintext using a time-varying transformation. Figure 1 shows the general structure of a Stream cipher. A stream cipher usually based on a pseudo random generator. The generator takes as input a secret key and then generates a pseudorandom sequence $x_1, x_2, ..., x_i$ of digits known as the running key stream.

The keystream digits are XORed with the plaintext digits $m_1, m_2, ..., m_i$ to obtain ciphertext digits $c_1, c_2, ..., c_i$ as follows:

$$c_i = m_i \oplus x_i \,. \tag{1}$$

The cipher message at the receiver is decrypted by producing the same keystream and adding it to the cipher message such as:

$$m_i = c_i \oplus x_i = m_i \oplus x_i \oplus x_i \,. \tag{2}$$

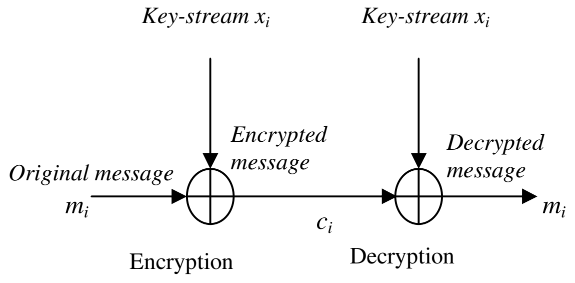


Fig. 1. General structure for stream cipher.

The stream cipher algorithm has several advantages which make it suitable for some applications. Most notably, it is usually faster and has a lower hardware complexity than block ciphers (e.g. [6], [7]) and RSA algorithms. It is also appropriate when buffering is limited, since the digits are individually encrypted and decrypted. In term of application, it is still the type of encryption preferentially and quasi-exclusively used in the industrial world (in particular in telecommunications and governmental). This algorithm is thus used in a privileged way in the case of communications likely to be strongly disturbed because they have the advantage of not propagating the errors [8].

III. The Blum-Blum-Shub Generator

In this section, we present the Blum-Blum-Shub (BBS) generator described in [4]. The Blum Blum Shub is a pseudorandom number generator proposed in 1986 by Lenore Blum, Manuel Blum and Michael Shub. The generator BBS works as follow:

Select two *k*-bit random primes p and q that are 3 modulo 4, let $n = pq$, and let $0 \prec S \prec n$ be random integer that is relatively prime to n and compute

$$y_0 = S^2 \bmod n$$

The numbers of the generator given by the following relation

$y_i = y_{i-1}^2 \bmod n$ and the i-*th* pseudo-random bit of BBS is the least significant bit of y_i

$$x_i = y_i \bmod 2$$

The output is $x_1, x_2, \ldots$

IV. Security of the BBS Generator

The security of the BBS generator depends on the difficulty of factoring n; we do not have to compute all the previous $i-1$ bits to obtain the $i-th$ bit. If we know p and q, we can compute directly the $i-th$ bit: x_i the least significant bit of y_i, where $y_i = y_0^{(2^i)\bmod((p-1)(q-1))} \bmod n$. The BBS generator is unpredictable to left and unpredictable to right. This means that, given a sequence generated by the BBS generator, the cryptanalysts cannot predict the next bit or the previous bit of the sequence. This is that not security based on a generator complicated that nobody understands, but on the mathematics underlying the factorization of n. For those who want pseudo-random sequences of bits crypto graphically safe is the best choice. Refer to [9], [10].

V. Approach of Message Encryption

The block diagram of the overall encryption and decryption scheme is shown in figure 2. The steps of approach are work as follow:

- Step1: Sender tacks the message .txt file (original message) and converts it into its ASCII value.
- Step 2: Convert the ASCII value into binary coding.
- Step 3: Read the length of binary coding.
- Step 4: Compute the key stream sequence produced by the BBS generator.
- Step5: Encryption of binary coding using the BBS generator and send the encrypted binary coding to receiver through a non-secure channel.

When the receiver, receive the encrypted binary coding then he make the following steps:

- Step 6: Decryption of encrypted binary coding using the BBS generator and stored it in another file.
- Step 7: Get the decrypted message and comparing with message .txt file.

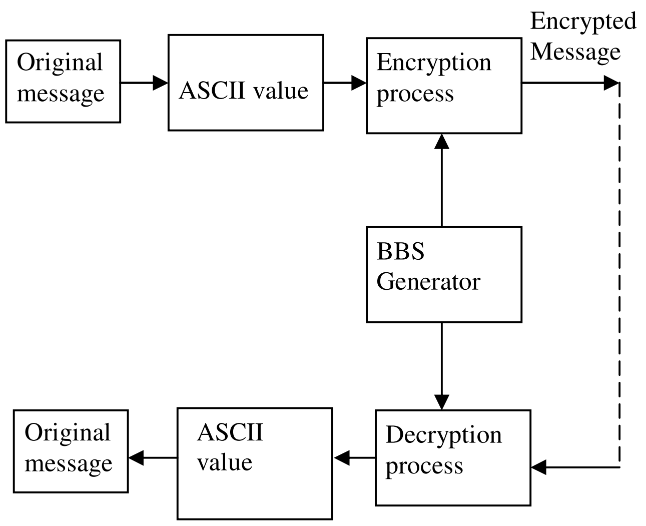


Fig. 2. Block Diagram of the Approach.

A. Encryption and Decryption Algorithm

Encryption

```
1. Load the original message (message.txt file);
2. Convert the message .txt file into integer data;
3. Convert integer data into data digit and stored it in Tab1;
4. $N \leftarrow$ the length of Tab1;
5. for $i = 1$ to $N$ to make ;
6. Compute the BBS bit generator shown in section B;
7. Encrypt the data digit and stored inTab2;
8. End to make ;
9. Sent the encrypt data digit (Tab2).
```

Decryption

```
1. Load the encrypt data digit(Tab2);
2. $N \leftarrow$ of Tab2;
3. for $i = 1$ to $N$ to make ;
4. Compute the BBS bit generator shown in section B;
5. Decrypt the encrypt data digit;
6. End to make ;
7. Get the decrypted message.
```

B. Algorithm BBS Pseudorandom bit Generator

```
A pseudorandom bit sequence $x_1, x_2,...x_N$ of length $N$ is generated
1. Generate two large secret random (and distinct) primes $p$ and $q$, each congruent to 3 modulo 4, and compute $n = pq$.
2. Select a random integer $0 \prec S \prec n$ (the seed) such that $\gcd(S,n)=1$, and compute $y_0 = S^2 \bmod n$
3. For $i$ from 1 to $N$ do the following:
       $y_i = {y_{i-1}}^2 \bmod n$
       $x_i = y_i \bmod 2$ the least significant bit of $y_i$
4. End to make ;
5. The output sequence is $x_1, x_2,...x_N$.
```

VI. Simulation and Experimental Results

This section discusses the obtained results from implementing the proposed algorithm. The implementation for this simulated project is written by MATLAB V 7.5. In the simulation, two messages file 1 .txt and file 2 .txt indicated in fig.3 and fig.4 are used. By comparing the original messages shown in fig. 3.a and fig. 4.a, and their encrypted messages shown in fig. 3.c and fig. 4.b, the encrypted messages was very different than originals messages; there is string of characters random in fig. 3.c and fig. 4.b.

The histograms of difference between originals messages and its corresponding decrypted messages are shown in fig .5are prove that, there is no loss of information, the difference is always 0.

The main objective of this work was to encrypt and decrypt message using stream cipher based on BBS generator. In this work the message tack and was saved as .txt file format. To encrypt this message, at first we extracted ASCII value from the message "txt" file and convert the ASCII value into binary coding and saved into another data binary file. The encryption algorithm taken the data digit file as input, and performed the encryption operation on the file to produce an unreadable encrypted message and saved another encrypted data digit file.

On the other hand, the decryption algorithm taken the encrypted data digit file as input and performed decryption operation on the file to produce the original data digit and convert it into integer data. In this work, $p = 7603$, $q = 7487$ and $S = 7817$ were used to calculate respectively the $n = pq = 56923661$ and $y_0 = S^2 \bmod n$ such that y_0 is the secret key.

```
Stream ciphers encrypt individual digits of plaintext
using a time-varying transformation.
```

a

```
83  116  114  101   97  109   32   99  105  112
104  101  114  115 32  101  110   99  114  121
112  116   32  105  110  100  105  118 105  100
117   97  108   32  100  105  103  105  116  115
32  111 102   32  112  108   97  105  110  116
101  120  116   32  117  115 105  110  103   32
97   32  116  105  109  101   45  118   97  114 121
105  110  103   32  116  114   97  110  115  102
111  114  109 97  116  105  111  110   46   32
```

b

```
õÔZÛY=º&-K¬   äl▯  @O
▯              ▯  ▯  -
gQ  cZ¼Ë`b▯ R!k½▯   ▯ ▯ CØ▯ G"m*;´VÈ.B▯ ³w▯ lão
               ▯ +M▯ Ø▯ xô▯ Æ¨°▯ ebznÈìX▯ ký-Ñy
```

c

```
245 212 90 219 89 61 186   38   45 195    8   75  172
127  228  108   20   64 79   10   24    9   25  158
45 157   103 81   141   99   90  188  203   96   98
22 82 33  107  189  133  144   24  131   67 216    6
71 34  109   42   59  180   86 200   46   66  158
179  119   23   73  227 111    9  158   43   77    3
216   22  120 244  146  198  168  176  148  101
98  122 110  200  236   88  154  107  253   45
209  121
```

d

```
Stream ciphers encrypt individual digits of plaintext
using a time-varying transformation.
```

e

Fig. 3. Encryption and Decryption for file 1 . txt: Frame (a) OriginalMessage, Frame (b) ASCII value of Original Message, Frame (c) Encrypted Message, Frame (d) ASCII value of Encrypted Message, Frame (e) Decrypted Message.

De tous temps, les services secrets ont utilisé toutes sortes de codages et de moyens cryptographiques pour communiquer entre agents et gouvernements, de telle sorte que les "ennemis" ne puissent pas comprendre les informations échangées. La cryptologie a alors évolué dans ces milieux fermés qu'étaient les gouvernements, les services secrets et les armées. Ainsi, très peu de gens, voire personne n'utilisait la cryptographie à des fins personnelles. C'est pourquoi, pendant tant d'années, la cryptologie est restée une science discrète.

a

```
âÊW%ée0Ö            ^-            äeP              □ O□
□ }□ □ bL¯Â4u_Z&kîÐ□ □ ÏZÇ□
8v:7¥□   .D□ 2□ lócD□ i_□ Í
bº □ü¦ □□qnÃäW "ñ1□)¿¾Âò□1ï,¸ê2□]ýü.□ ¬Þ æy«´
yóZåÑ5Õ□
□4í¸ãÊÜò:□_=G□ÀÑÄ·□¦□ôc ä¯,□±=Í□óãj□£§o1¦0q²-
(Íä□ø¥ sÉyÛ□«fÕt□        à7âs£ðè£h□×□û□      □ÌÆÛç
¸®óß¥OcbR□ `dÑöV6Ó□ * ´ÕÈèCªBZ□M□æl□Æ□È
9kèmô6Y
×□ a ]n¬¼¦T»ælÖë×~}□ ×ì□ R¿ÀÜd"ö{ï-*µZóõÖ®
□ T-mÑïò}Ï□ K  g7ETÑa7_b  4«-ñùèI□ ï®õ□ ´□
        28                   □ zsr[ø²v5□ 8M¢Dd□ &-Ó9
b3□ Û³Fh¬ )^ðÚ3Æ¯òB    Ð□
□Îó@ÞF□L<p\□{B:□nnck¤ê´□\è    _/ p□□JÀr"sy·¨LÓo□¡
□ Ò'4Bf¬Ú1CPvU¦ ¾□KäQÏc£□□□□¦~□'□Ïµäô□      a±}
yé}g!ew¨ÛÉA□ E+EkëP□ ^□ >¬æ□ ÅÍ(TËah?□ Í ¦Îb
□ 6V$ÔØ-□ Ô^□ ?V□ ,Zz±D□ £fec ¾¿ØK¼@O9d5'□
```

b

De tous temps, les services secrets ont utilisé toutes sortes de codages et de moyens cryptographiques pour communiquer entre agents et gouvernements, de telle sorte que les "ennemis" ne puissent pas comprendre les informations échangées. La cryptologie a alors évolué dans ces milieux fermés qu'étaient les gouvernements, les services secrets et les armées. Ainsi, très peu de gens, voire personne n'utilisait la cryptographie à des fins personnelles. C'est pourquoi, pendant tant d'années, la cryptologie est restée une science discrète.

c

Fig. 4. Encryption and Decryption for file 2 . txt: Frame (a) Original Message, Frame (b) Encrypted Message, Frame (c) Decrypted Message.

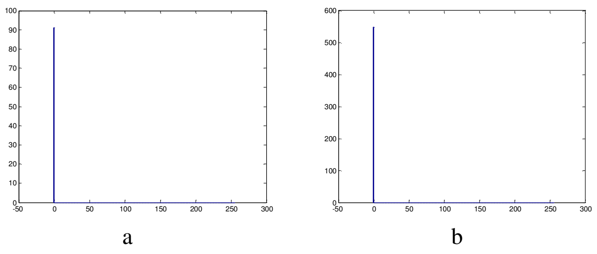


Fig. 5. Frame (a) show the histogram of difference between original message shown in fig. 3.a and its decrypted message shown in fig. 3.e, Frame (b) show the histogram of difference between original message shown in fig.4.a and its decrypted message shown in fig. 4.c.

VII. SECURITY ANALYSIS

In this section, the performance of the proposed message encryption is analyzed in detail. We discuss the security analysis of the proposed encryption scheme including some important ones like key sensitivity analysis, histogram analysis, and correlation coefficient analysis. This is to prove that the proposed cryptosystem is secure against the most common attacks.

A. *Correlation Coefficient Analysis*

Table 1 gives the correlation coefficient results. By Cor1, Cor2, and Cor3 we called respectively correlation coefficient between original message and encrypted message, correlation coefficient between original message and decrypted message and correlation coefficient between original message and decrypted message with wrong key K_2 . It is observed that the values of Cor1 shown in the tab 1 are quite close to the value of zero, which implies that the original message and its encryption are totally different i.e. the encrypted message has no features and highly independent on the original message. It is also clear that the values of Cor2 shown in the tab1 are equal to the value 1, which is implying that the encrypted message is the same as the original message.

Thus, values of Cor3 shown in the tab 1 are quite close to the value of zero, which implies the original message and decrypted message with wrong key:K_2 are totally different.

TABLE I. CORRELATION COEFFICIENTS ANALYSIS

Case	Cor_1	Cor_2	Cor_3
File 1 .txt	-0.0663	1.0000	-0.2405
File 2 .txt	0.0102	1.0000	0.312

B. *Key Sensitivity*

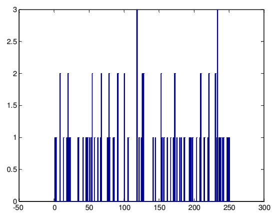
A good cryptosystem should be sensitive to the secret keys, which means change of a single bit in the secret key should produce a completely different encrypted message. The cryptosystem was tested to the sensitivity of keys, we encrypt the messages shown in fig. 3.a and fig. 4.a with the secret key $K_1 = 4181528$ and, we decrypt their encrypted messages shown fig. 3.c and fig. 4.b with another different key; $K_2 = 4557620$. The result is given by fig. 6. So it can be concluded that the proposed cryptosystem is highly sensitive to the key.

```
▯ ?3#OD▯ ▯ .▯ Þz0jÌÜ▯    ó-\▯ ▯ w-▯ íÒ5▯ ▯ Mú«*Vº
D
´w▯ ïé▯ û▯ ê7/Òø§Þ»U¬PÑ¶ÂÃç\}[▯ êw7CêÄì[Æ¢e×
▯          : ▯   Oeñ°ç
```

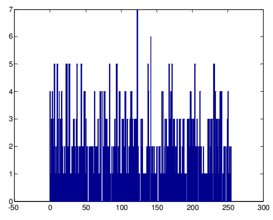
a

```
üödFÀð ÷.£ò▯¹L▯ÙÎè       O       Øjõ?k  /▯ ▯ ▯ \©$wþ
Kz«▯ ³Zí ▯$ÿ-:0▯▯s÷°Xû
Æ8* koSõ²Ù ©▯-'▯▯y´úzn▯PÉö▯¬·ó▯▯▯  z▯¦Þ▯~
´ô▯¾'{$▯▯L ·§¡K.hç▯:|_Ï)¶:½▯©l÷×ã ô▯Á¿£ÖÛ!-¿▯
Ô▯ E®--85ßbÀ▯ ▯ Ô▯ y_;lõqco©qÑÜú9*▯ ÄR³▯ Ü×▯ ]
^c▯ jBZê▯ ▯ ¤BHÝ~J'▯    Êç/ì¢▯ ▯ Î▯ ▯ l\K®E
          U8ç8K<°¢
/Ó%>¼ù¬þ¬,~*G
±²'▯ û¤k*æ▯ t▯  ▯úé¤åþ!▯n`fÁ▯Oº▯-▯¡§i¯µX▯ %[s9-bî(°
¿▯ ▯ ▯ W¤;c^0▯ '▯ ▯ i▯ <¬Õ>ý▯ ÉnÉ▯ ▯ ¤ÖÎh6ÑR8i-
¿6+WÛ-"þ_<¬N▯ ©▯ QdÃmÈ▯ -
▯ Æ=  áë®▯  ▯¤▯ß^▯ À63DÜ<h▯µßTlG4F▯p®q▯ýR1[éÛ
▯▯5Ç¿FÏ▯Ã;ö▯c▯X`▯▯ù¡▯▯Ïl▯á/2£])  tR.µ-
▯'Ä▯   Æv_▯ ·▯ ëÏ È▯▯×Kv¬Äõ▯ð nü¦æ▯%%«@Õ▯ _Ì
£▯ TüG«Î
¶ðäÐ▯1#h  |▯(Á~`lbB▯▯▯▯ò>▯jÏl▯▯▯_'GÓ▯▯i$▯ÈF9▯¾ÇY
[illegible]
```

b



c

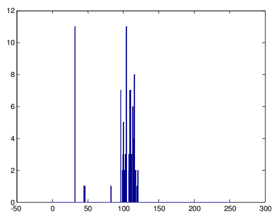


d

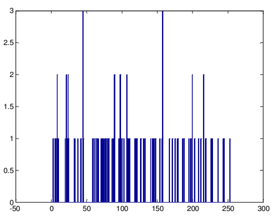
Fig. 6. Sensitivity analysis: Frame (a) and (b) respectively, show decrypted message with wrong key of the encryption messages shown in fig. 3.c and fig.4.b. Frame (c), and (d) respectively, show histogram of messages (a) and (b).

C. Histogram Analysis

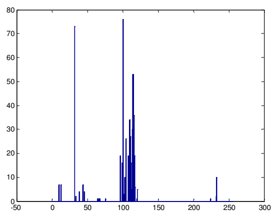
The histograms of original messages and their corresponding encrypted messages are shown in fig. 7. It is clear that the histograms of the encrypted messages are almost uniformly distributed, and significantly different from the respective histograms of the original messages. So, the encrypted messages do not provide any clue to employ any statistical attack on the proposed encryption message procedure, which makes statistical attacks difficult.



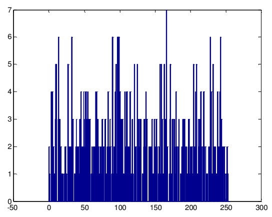
a



b



c



d

Fig. 7. Histogram analysis: Frame (a) and (c) respectively, show histograms of originals messages shown in fig .3.a and fig. 4.a. Frame (b) and (d) respectively; show the histograms of the encrypted messages shown in fig .3.c, and fig. 4.b.

VIII. CONCLUSION

In this Work, an implementation of new approach using stream cipher based on the BBS generator for encryption data message was developed. Simulations were carried out two different messages. The test indicates that the encrypted message was very different than original message. This method is very simple, fast to implementation, the message encryption and decryption is an easy task.

Here the security aspects such as sensitivity analysis, histogram analysis, and correlation coefficient analysis, are discussed with examples.

REFERENCES

[1] Z. Philip, "*PGP Source Code and Internals*". MIT Press.ISBN 0-262-24039-4, 1995.

[2] H. Mathkour, G. Assassa, A. A. Muharib, A. Juma'h, "*A Secured Cryptographic Messaging System*", International Conference on Machine Learning and Computing, IPCSIT vol.3 (2011) © (2011) IACSIT Press, Singapore, 2009.

[3] J. Singh and J.S. Sodhi, "*Secure Data Transmission using Encrypted Secret Message*", International Journal of Computer Science and Information Technologies, Vol. 4 (3) , 522-525, 2013.

[4] L. Blum, M. Blum and M. Shub, Mike, *"A Simple Unpredictable Pseudo-Random Number Generator". SIAM Journal on Computing* **15** (2): 364–383. doi:10.1137/0215025, 1986.

[5] R.L Rivest, A. Shamir et L.M. Adeleman., "*A method for obtaining digital signatures and publique- key cryptosystems"*, Communication of the A.CM, Vol 21, N° 3, 1978, pp 120-126.

[6] H. Feistel., "*Cryptography and computer privacy"*, Scientific American, Vol 228, N° 5, pp 15-23, 1973.

[7] htpp://www.nist.org/aes/

[8] C. Carlet, "*On the cost weight divisibility and non linearity of resilient and correlation immune functions*", Proceeding of SETA'01 (Sequences

and their applications 2001), Discrete Mathematics, Theoretical Computer Science, Springer p 131-144, 2001.

[9] P. Junod, "*Cryptographic Secure Pseudo-Random Bits Generation* ", The Blum-Blum-Shub Generator", 1999.

[10] A. Sidorenko and B. Schoenmakers, "*Concrete Security of the Blum-Blum-Shub Pseudorandom Generator Andrey Sidorenko and Berry Schoenmakers*", Cryptography and Coding: 10th IMA International Conference, Lecture Notes in Computer Science 3796, 355-375. Springer-Verlag, 2005.