Ibn Al-Haitham Journal for Pure and Applied Science مجلة إبن الهيثم للعلوم الصرفة و التطبيقية
No. 1 Vol. 25 Year 2012 2012 السنة 25 المجلد 1 العدد

Enhancement of Stream Cipher by Using in Length Variant Register

N. A. Taha
Department of Computer Science,College of Education Ibn Al-Haitham, University of Baghdad

Received in: 18 May 2011, Accepted in: 16 June 2011

Abstract

Stream ciphers are an important class of encryption algorithms. There is a vast body of theoretical knowledge on stream ciphers, and various design principles for stream ciphers have been proposed and extensively analyzed. This paper presents a new method of stream cipher, that by segmenting the plaintext into number of register then any of them combined to any other by using combination logic circuit (And, OR, JK, NOT, XOR), then using variant register in length as a key which provides security enhancement against attacks and then compare the strength of this method with RSA by calculaing the time necessary to get the original text by using the genetic algorithm. And the way that has the longest time is the best in encryption. Then it was found that proposed method is stronger in encryption than RSA.

Keyword: stream cipher, variant register, genetic algorithm, complexity.

Introduction

Stream ciphers are an important class of encryption algorithms. They encrypt individual characters (usually binary digits) of a plaintext message one at a time, using an encryption transformation which varies with time. Various design methods where proposed for stream ciphers and the specialists proposed many analysis methods. However, one possible explanation can be the fact that many stream ciphers used in practice tend to be proprietary and confidential. A stream cipher generates what is called a keystream (a sequence of bits used as a key). Encryption is accomplished by a simple operation combining the keystream with the plaintext, usually with the bitwise XOR operation [1].

Types of Stream Cipher

A stream cipher generates successive elements of the keystream based on an internal state. This state is updated in essentially two ways: if the state changes independently of the plaintext or ciphertext messages, the cipher is classified as a *synchronous* stream cipher. By contrast,*self-synchronising* stream ciphers update their state based on previous ciphertext digits.

❖ Synchronous stream ciphers

In a synchronous stream cipher a stream of pseudo-random digits is generated independently of the plaintext and ciphertext messages, and then combined with the plaintext (to encrypt) or the ciphertext (to decrypt). In the most common form, binary digits are used (bits), and the keystream is combined with the plaintext using the exclusive or operation (XOR). This is termed a binary additive stream cipher.

In a synchronous stream cipher, the sender and receiver must be exactly in step for decryption to be successful. If digits are added or removed from the message during transmission, synchronisation is lost. To restore synchronisation, various offsets can be tried systematically to obtain the correct decryption. Another approach is to tag the ciphertext with markers at regular points in the output. If, however, a digit is corrupted in transmission, rather than added or lost, only a single digit in the plaintext is affected and the error does not propagate to other parts of the message. This property is useful when the transmission error rate is high; however, it makes it less likely the error would be detected without further mechanisms. Moreover, because of this property, synchronous stream ciphers are very susceptible to active attacks — if an attacker can change a digit in the ciphertext, he might be able to make predictable changes to the corresponding plaintext bit; for example, flipping a bit in the ciphertext causes the same bit to be flipped in the plaintext. [2]

Binary additive stream cipher is a synchronous stream cipher in which the keystream, plaintext, and ciphertext denoted are sequences of binary digits, and keystream is combined with the plaintext using XOR operation to output ciphertext. For each secret key, the stream cipher generates a different deterministic keystream sequence. Since it is assumed that *K* is a shared secret between the transmitter and receiver, receiver produces the same keystream sequence and obtains plaintext by XOR'ing ciphertext with keystream. In the remaining parts of the study, we suppose the stream cipher is a binary additive stream cipher unless otherwise is stated.[3]

❖ **Self-synchronizing stream ciphers**

Another approach uses several of the previous *N* ciphertext digits to compute the keystream. Such schemes are known as self-synchronizing stream ciphers, asynchronous stream ciphers or ciphertext autokey (CTAK). The idea of self-synchronization was patented in 1946, and has the advantage that the receiver will automatically synchronise with the keystream generator after receiving *N* ciphertext digits, making it easier to recover if digits are dropped or added to the message stream. Single-digit errors are limited in their effect, affecting only up to *N* plaintext digits.

An example of a self-synchronising stream cipher is a block cipher in cipher feedback (CFB) mode.[2]

❖ Genetic algorithm

A genetic algorithm is one of a relatively new class of stochastic search algorithms. Stochastic algorithms are those that use probability to help guide their search.

GAs behaves much like biological genetics. GAs encoded information into strings, just as living organisms encoded characteristic into stands of DNA [4]

Review of Previous Works

1. Martin Boesgaard, at al. in [5], Presented a new stream cipher, Rabbit, based on iterating a set of coupled nonlinear functions. Rabbit is characterized by a high performance in software with a measured encryption/decryption speed of 3.7 clock cycles per byte on a Pentium III processor. We have reformed detailed security analysis, in particular, correlation analysis and algebraic investigations. The cryptanalysis of Rabbit did not reveal an attack better than exhaustive key search.
2. Hongjun Wu,IN. in [6] . It generates keystream from a 256-bit secret key and a 256-bit initialization vector. The encryption speed of the C implementation of HC-256 is about 1.9 bits per clock cycle (4.2 cycle/byte) on the Intel Pentium 4 processor. A variant of HC-256 is also introduced in this paper.

3. Imran Erguler, Orhun KaraI in [7], in this study, propose a new keystream based encryption model ; firstly apply forward error correcting codes (FEC) to plaintext. Next, add a nondeterministic noisy sequence to keystream and obtain a nondeterministic bit sequence which is combined with plaintext to generate cipher text. Even though the receiver does not know the content of the nondeterministic sequence, he can still obtain the original message. These forces a new definition for cryptanalysis of keystream sequences in case of known plaintext.

A General Description of the Proposed Model

Below is a presentation of a new approach for encryption system in which plaintext is XOR'ed with a keystream sequence. For this model at the transmitter site, firstly the plaintext sequence V was encoded to ASCII code A and then encoded to binary code sequence P.

Secondly the binary plaintext P was segmented into to one or more registers denoted by Ri (i=1,2,3…….,9) then any of them combined to any other by using combination logic circuit (And, OR, JK, NOT, XOR), these logic gates can be used randomly and can use the same logic gate more than once , but their use should be limited and thoughtful because if it is used in non-carefully studied and repetitive manner, it may lead to poor results obtained. Then, the final register having primary ciphertext C_p. Various properties of such a logic gates function are important for ensuring the security and can make it extremely difficult to guess plaintext that means it provides more security enhancement against attacks, as shown in figure-1

Next, this primary ciphertext C_p is combined with key stream to generate final ciphertext C_f

Key stream

A key stream sequence is produced by seed and it is mixed with plain text for encryption at the transmitter side note that key stream sequence is deterministic anyone having the seed can reproduce it, and recover the plaintext easily. Therefore, for stream cipher security of the ciphertext relies on the security of the key stream. In this model, a new key stream was proposed in a based encryption model which provides security enhancement against some well known attacks.

First a keystream was created from primary ciphertext Cp. The length of the encryption key is randomly variable from 15-25% percent of the length of primary ciphertext C_p. after that this key is XOR'ed with primary cipher text to obtain the final ciphertext

$$C_p \quad \oplus k = C_f$$

This change in the key length, add further strength to the proposed method which enhanced and increased in the strength of encryption against attack.

Main Algorithm

1- Call stream cipher algorithm creator
2- Call RSA algorithm
3- Call complexity checker

Algorithm of stream cipher

1- Input data file
2- Store data file in memo
3- Do
4- Split input line to register
5- Create key for current register
6- Put key as part of stream

```
 7- Stream=stream + current register
 8- While input line = end of data
 9- Put end of stream
10- Save stream as file
```

Complexity checker

The final stage of this work is to measure the complexity strength of encryption by comparing with one of encryption method known RSA, that by taking a sample from text encrypted in the proposed method denote by (S1s) with the same sample but an encrypted by RSA method denote by ($S1_R$) and then calculate the time necessary to get to the same sample in original text denoted by($S1_O$) by using genetic algorithm.

Genetic algorithm is used to compute the time required to decipher the ciphertext to produce the plaintext.

Genetic algorithm begins with randomly generating an initial population of P size chromosome is evaluated by the chromosome's length equal to the length of the sample $S1_S$. The fitness of each chromosome comparing the chromosome with the $S1_O$ (the sample from original text), after that Crossover and mutation are applied on the population to create a new population. Suppose the time required to the first sample $S1_S$ is T1 and to the sample $S1_R$ is T2, as shown in figure- 2

If T1 is greater than T2 then the proposed method is the best. This is done for 8-15 samples deend on the size of the plaintext.

For example, after plaintext sequence was encoded to ASCII code and then encoded to binary code sequence, the plaintext was segmented into 3 registers and then register 1 was combined with register 2 by XOR gate, register 2 with register 3 combined by AND gate and registers 1& 3 are combined by OR gate to generate primary ciphertext then create a key for current register after that this key is XOR'ed with primary cipher text to obtain the final ciphertext as shown in figure- 3.

For the same example, Table-1 shows the difference between the times required to decipher the sample ciphering by proposed method T1 & Time required to decipher the same sample ciphering by RSA methods T2 that done for 10 samples. It was noted that the proposed method takes time longer than the RSA method to get to the original text and this means that the proposed method is the strongest in terms of encryption

Algorithm of complexity checker

```
1- Read cipher file 1
2- Read cipher file 2
3- Read plaintext
4- Select sample from file 1 and file 2
5- Start analysis with calculated time for two samples to reach to the plaintext
    Note: the analysis depend on genetic algorithm
6- If T file 1 > T file 2 then file 1 is cipher better
```

Result and Discussion

Figures 3, 4, 5 & 6 are graphed by plotting the difference between T1 &T2 (T1-T2) and joining the successive point to a number of samples by a line.

The Sum complex represents the number of sample.

T1 is the time required to decipher the sample is ciphered by proposed method.

T2 is the time required to decipher the sample is ciphered by RSA.

From these figures it can be noticed many points:

1. In the figure -3 and table-1 that the use of key variable in length leads to increase the strength of encryption comparison with RSA method.
2. A comparison between Figure 3 and 4 that the use of logic circuits have helped to increase the strength of encryption where it is noted in table -2 that the time needed to decrypt the code in a RSA which became in 4 samples greater than the time of the proposed method. In other words, the strength of encryption is becoming less with absence of logic circuits
3. The use of logic gates frequently and non-thoughtfully could lead to reverse results as seen in figure -5. This was seemed clear in Table-3, where the time needed to decrypt the sample ciphered by a RSA which became in 7 samples greater than the time of the proposed method.
4. In the figure-6 that the use of large number of registers in input leads to reduce the strength of encryption, especially if the size of text is small. This seems clear from Table-4 where the time needed to decrypt the sample ciphered by a RSA which became in 6 samples greater than the time of the proposed method.

Conclusion

In this proposed method using variant register in length as a key in stream cipher which provides security enhancement against attacks, and can conclude many points as follows:

1- The use of variable key in length and number of logic gates has helped to increase the strength of encryption.

2- The use of logic gates must be limited and thoughtfully, because the frequent use of the same logic gate may lead to reduce the strength of encryption.

3- The use of a suitable number of registers in the input leads to increase the strength of encryption (the number of registers that are used depends on the size of text).

4- This proposed method is suitable for image encryption too that can be done by using pixel value instead of the ASCII code of the letter and convert this pixel to binary code, and then can be used the same algorithm above

Reference

1- Bucerzan, D.; Craciun, M.; Chis, V.and Ratiu, C.(2010), Stream Cipher Analysis Methods, Int. J. of computer, communications& Control, ISSN 1841-9836, E-ISSN 1841-9844 , V (4): 483-489 ,483-489.

2- Matt, J.and Robshaw, B.(1995), Stream Ciphers Technical Report TR-701, version 2.0, RSA Laboratories, 5-8.

3-Menezes, A.; Vanosrschot, P. and vanston, S.(1996), Handbook of applied cryptography, by CRC press LLc, ch 6, 194.

4-Grant, K. (1995), An introduction to gentic algorithms "march , c/c++ users Journal,13 (3): 45-58.

5- Boesgaard M., Vesterager M., Pedersen T., Christiansen J., and Scavenius O.(2010), CRYPTICO A/S, Fruebjergvej, paper ,Rabbit: New High-Performance Stream Cipher, Copenhagen , Denmark, pp.3-4

6-Hongjun, Wu.(2004), Full version of the FSE 2004 paper, “A New Stream Cipher HC-256, Last revised April 15,2-3.

7-Erguler, I.and Kara, O.(2007), A New Approach to Keystream Based Cryptosystems, INFORMATION SECURTIY & CREPTOLOGY CONFERENCE WITH

INTERNATIONAL PARTICIPATION Journal, ISC Turkey, Aralik, Ankara,December,13-14 .

Table(1): Time required to decipher the sample that ciphered by proposed &RSA methods for 10 samples

Samples	Time required to decipher the sample ciphered by proposed method in sec T1	Time required to decipher the sample ciphered by RSA in sec T2
S1	15.23	15.21
S2	16.02	4.14
S3	17.12	17.01
S4	14.3	7.72
S5	15.14	13.06
S6	17.27	4.21
S7	12.81	9.13
S8	13.03	4.14
S9	10.13	7.92
S10	16.21	2.31

Table(2): Time required to decipher the sample that ciphered by proposed &RSA methods for 10 samples without using logic gates

Samples	Time required to decipher the sample ciphered by proposed method in sec T1	Time required to decipher the sample ciphered by RSA in sec T2
S1	10.24	13.95
S2	16.02	3.14
S3	16.12	14.01
S4	14.3	10.72
S5	13.14	18.46
S6	15.84	1.64
S7	11.31	12.83
S8	14.03	3.64
S9	6.13	9.22
S10	14.61	1.31

Table(3): Time required to decipher the sample that ciphered by proposed &RSA methods for 10 samples with Using logic gate frequently

Samples	Time required to decipher the sample ciphered by proposed method in sec T1	Time required to decipher the sample ciphered by RSA in sec T2
S1	10.34	13.54
S2	15.02	12.24
S3	11.02	18.81
S4	15.3	17.12
S5	16.71	12.06
S6	13.27	13.71
S7	12.21	20.13
S8	14.03	23.14
S9	10.13	12.92
S10	9.21	15.71

Table(4): Time required to decipher the sample that ciphered by proposed &RSA methods for 10 samples with Using 8 register in input

Samples	Time required to decipher the sample ciphered by proposed method in sec T1	Time required to decipher the sample ciphered by RSA in sec T2
S1	7.23	16.41
S2	17.02	13.86
S3	6.12	19.01
S4	10.3	17.72
S5	8.14	21.75
S6	17.27	19.21
S7	14.81	22.13
S8	13.32	7.14
S9	20.03	9.92
S10	15.31	13.97

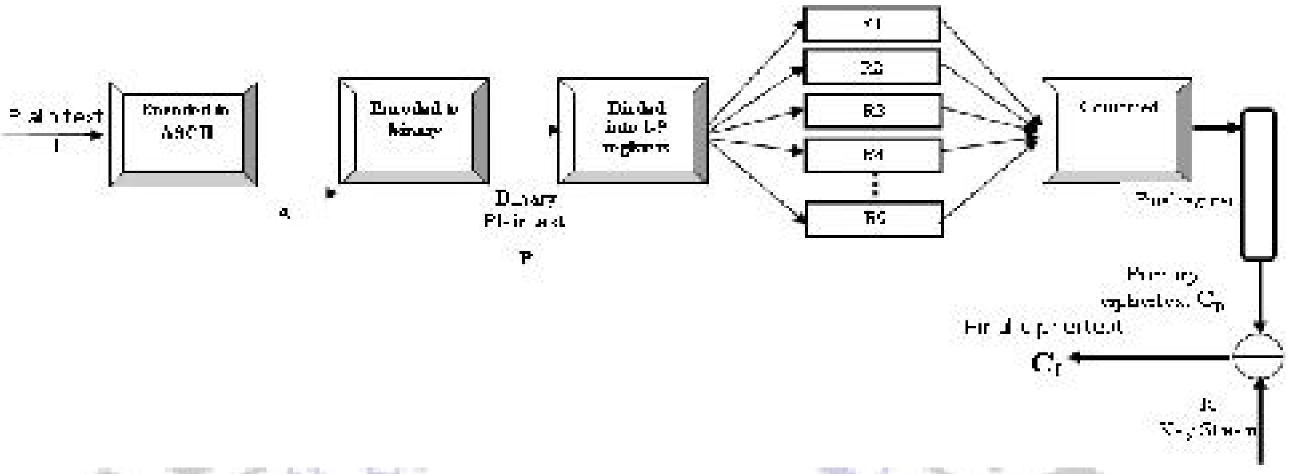


Fig.(1): Describes the proposed method

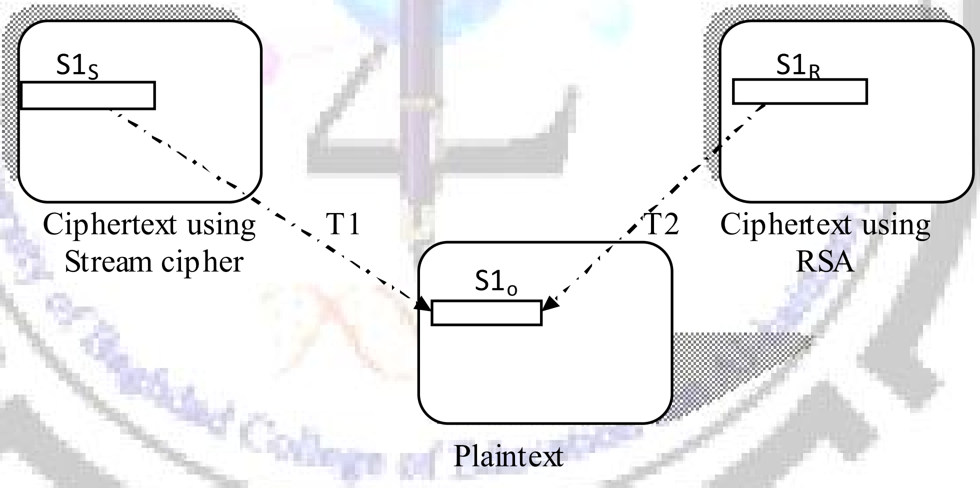
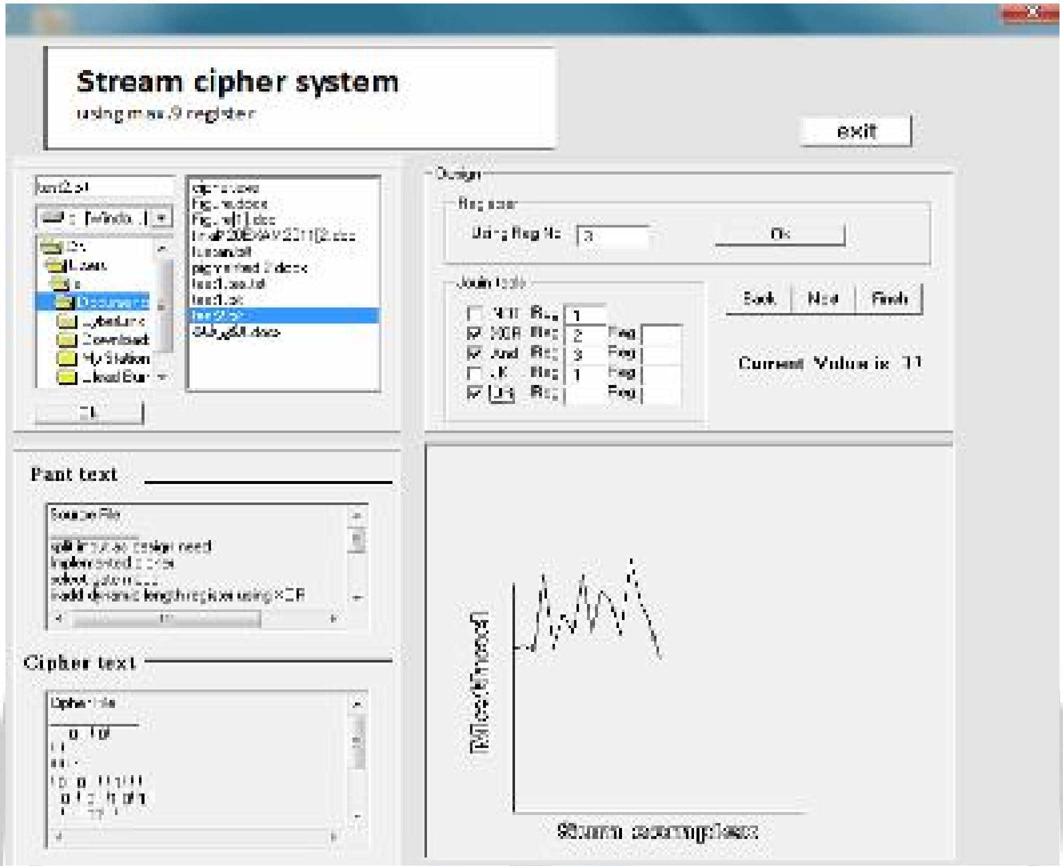


Fig.(2): Describes the way to measure the complexity strength of encryption



Stream cipher system
using max.9 register

Design

Reg size
Using Reg No 3

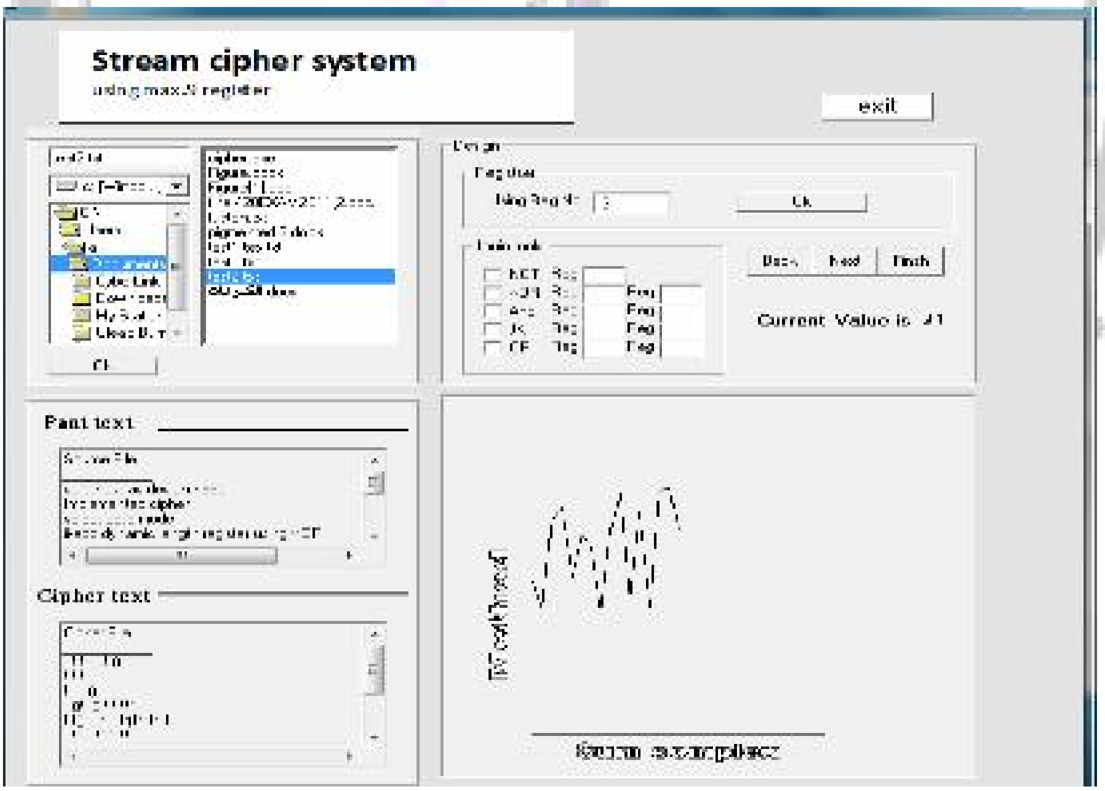
Join tools
☐ NOT Reg 1
☑ XOR Reg 2 Reg
☑ And Reg 3 Reg
☐ JK Reg 1 Reg
☑ DF Reg Reg

Current Value is 11

Pant text
Source File
split into ... design need
implemented cipher
select gates ...
build dynamic length register using XOR

Cipher text
Cipher file

Fig .(3): Example for proposed method



Stream cipher system
using max.9 register

Design

Reg size
Using Reg No 3

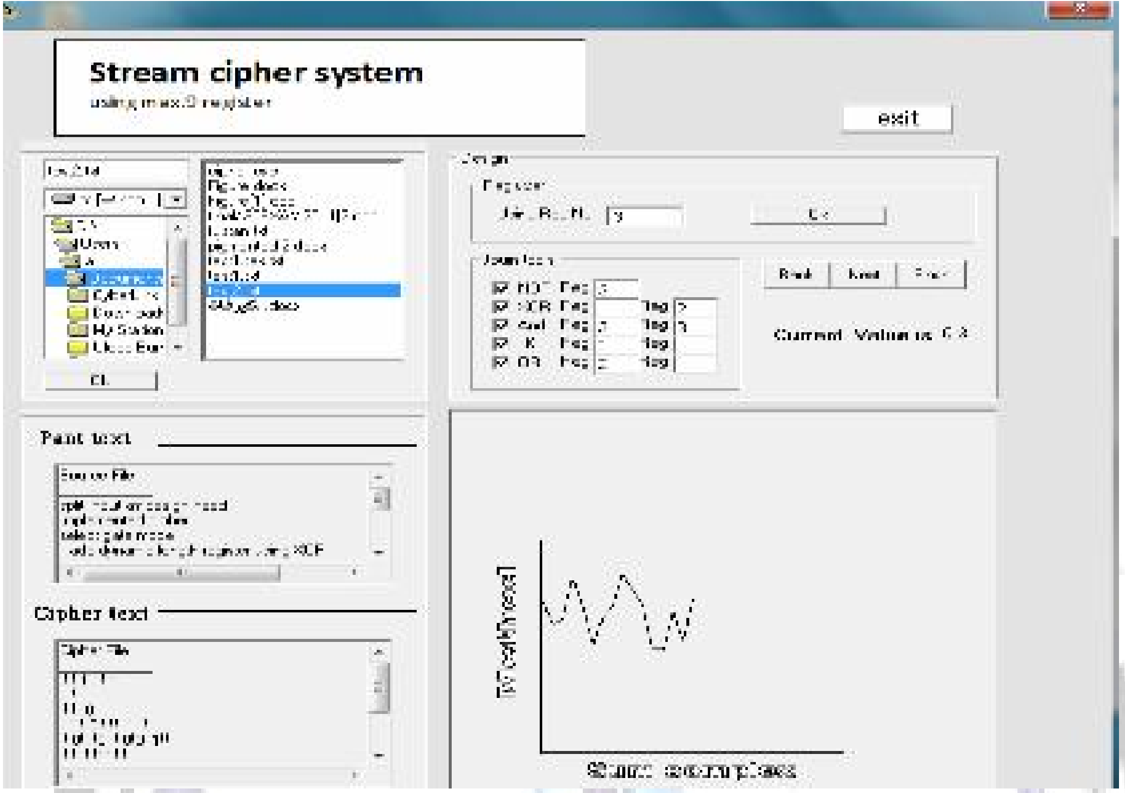
Join tools
☐ NOT Reg
☐ XOR Reg Reg
☐ And Reg Reg
☐ JK Reg Reg
☐ DF Reg Reg

Current Value is 41

Pant text
Source File
implemented cipher
build dynamic length register using XOR

Cipher text
Cipher file

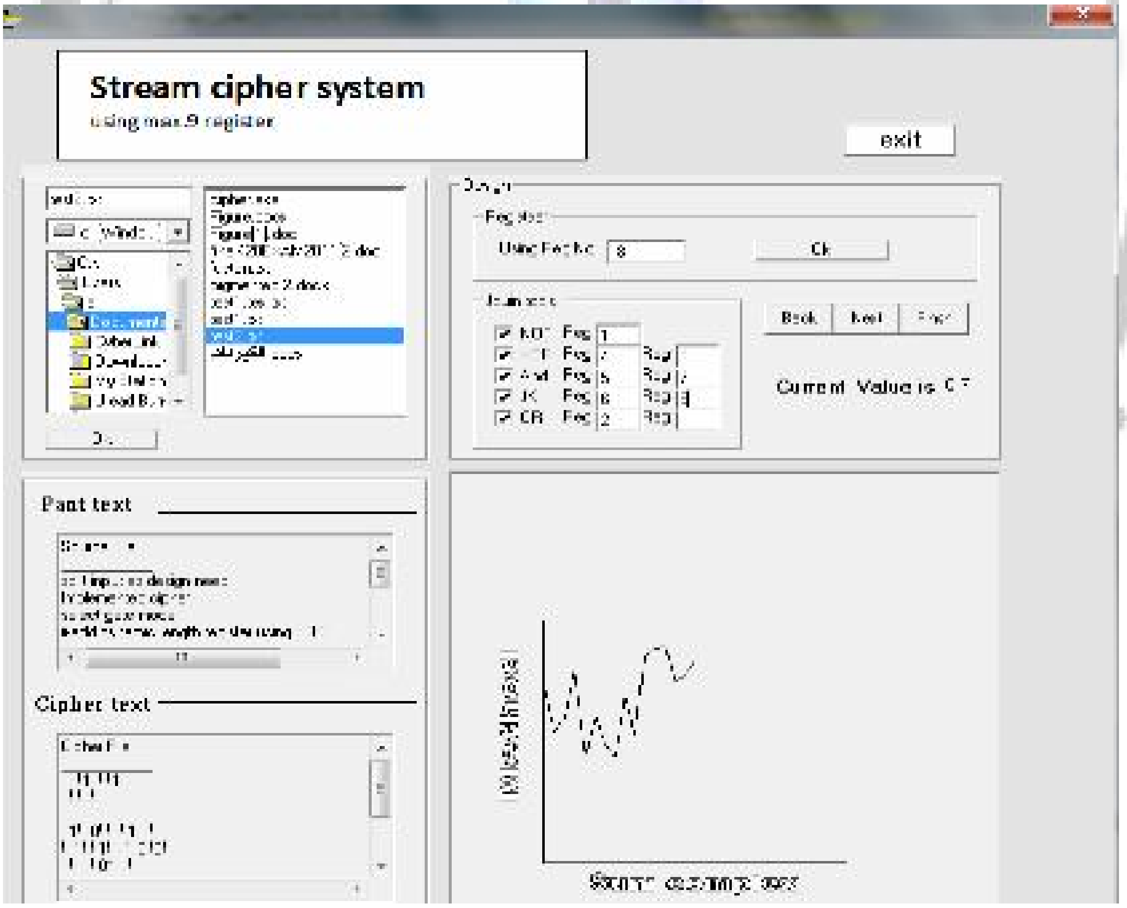
Fig .(4): THE proposed method without using logic gates



NOT Reg 3
XOR Reg 2
And Reg 7 Reg 5
OR Reg 2

Current Value is 0.8

Fig.(5): Using logic gate frequently



Using Reg No 8

NOT Reg 1
Reg 7
And Reg 5 Reg 7
JK Reg 6 Reg 8
OR Reg 2

Current Value is 0.7

Fig.(6): Using 8 register in input

تعزيز طريقة التشفير الانسيابي باستعمال مفتاح تشفير متغير في الطول

نمار عبدالملك طه

قسم علوم الحاسبات،كلية التربية– ابن الهيثم ، جامعة بغداد

استلم البحث في: 18 ايار 2011، قبل البحث في : 16 حزيران 2011

الخلاصة

التشفير الانسيابي هو نوع مهم من خوارزميات التشفير وهناك مجموعة واسعة من المعرفة النظرية، و اقترحت تصاميم مختلفة للتشفير الانسيابي وحللت على نطاق واسع، وفي هذا البحث اقترحت طريقة جديدة للتشفير الانسيابي وذلك بتقسيم النص الأصلي الى عدد من المسجلات وبعدها تجمع هذه المسجلات مع بعضها باستعمال البوابات المنطقية وهذا يعد تشفيرا إبتدائيا ومن ثم يجمع مع مفتاح تشفير هو مسجل ذو طول متغير الذي يعزز من قوة التشفير ضد الهجمات المعروفة. وبعد ذلك تقارن قوة هذه الطريقة مع طريقة معروفة و هي (RSA) و ذلك بحساب الزمن اللازم للحصول على النص الاصلي باستعمال خوارزمية الجينية. و الطريقة التي تمتلك زمن اطول هي الافضل بالتشفير و قد وجدت ان الطريقة المقترحة اقوى في التشفير من الطريقة الثانية .

الكلمات المفتاحية : التشفير الانسيابي، مسجل متغير،الخوارزمية الجينية، قوة التعقيد