International Journal of
Emerging Research in Management &Technology
ISSN: 2278-9359 (Volume-6, Issue-1)

Research Article | **January 2017**

Constructing Stream Cipher Package Algorithms

Salim Ali Abbas*, Ali Jumah Hashim
Department of Computer Science, College of Education/ Al-Mustansiriya University,
Baghdad, Iraq

DOI: 10.23956/ijermt/V6N1/128

Abstract—

Constructing stream cipher algorithm in fast and comfortable way is an important requirement for those people specialists in designing stream cipher algorithms to secure their information. The proposed package provides the designer the ability to construct stream algorithm in two ways, the first is building stream algorithm visually by providing a graphical interface and a number of stream cipher algorithm components, and the ability to make connections among selected components to produce the key stream of the any desired length. The second way is constructing stream cipher through enter the C# programming code of the stream algorithm, the package executes the entered C# code and produce output as sequence of binary digits that represent the keystream, also the package will display messages to the designer if the entered code has any syntax errors. The parallel programming is used in construct the proposed package to speed up the operation of generating the key stream that produced in the visual way.

Keywords— Design, Visually, C# code, Package, Stream cipher algorithm

I. INTRODUCTION

As we enter the information age and the emergence of the means of digital transmission and storage of data in digital form, there was a need to develop ways to protect information from theft. [1]. Cryptography includes design and analysis a mathematical technique that provide a secure connection in the presence of a third party that seeking to disclosure of transmitted or stored information [2].

Stream cipher algorithm is one ways that used to encrypt the information and convert it into another form. Stream cipher has number of features, it is fast, and considered efficient in hardware because it is not required expensive hardware circuit. Therefore, the construction and design of stream cipher algorithm in fast and comfortable manner is an important need for those people who specialize in building stream cipher algorithm to protect their information from theft [3].

Encryption algorithm can be classified into two types, the first type is symmetric encryption, in this type used one private key that shared between the communication parties (the sender and the recipient). The second type is asymmetric encryption, in this type used two keys, the first public and open for everyone, and the second is kept secret, and in this type one key cannot be inferred from the other keys [4]. Asymmetric encryption is slow because it's based on mathematical theories, which is impractical in applications that require live and direct processing. In symmetric encryption, there are two types of algorithms, block algorithm and stream algorithm, in Block encryption algorithm the plaintext dividing into blocks of bits. Famous Block algorithms are AES and DES [5].

Stream cipher is developed as a solution to the problems of encryption one-time pad system. In one-time pad the key must be completely random and its length is equal to the length of the message to be encrypted and uses one time, because of the impracticability of one-time pad system so stream encryption system consider acceptable solution to these problems. In the stream cipher algorithm, pseudo-random key is generated using one of pseudo-random generator [6].

Encryption in stream cipher is done by dividing the clear text into bits and then every bit is encrypted individually. Stream algorithm relies entirely on security of pseudorandom key stream. To design stream algorithm, the designer can use several components and methods, the LFSR is most popular components used in the constructing of stream algorithm [7].

II. THE PROPOSED PACKAGE

The proposed package consists of two major parts. The first part is constricting any stream cipher algorithms (whose components are available in the proposed package) visually or by enter C# code of the stream algorithm. The second part is encrypting and/or decrypting any text message. This paper describes the steps which are taken to design the proposed package.

III. BUILDING STREAM CIPHER ALGORITHM

To build stream cipher algorithms, there are two approach. The first one is building stream cipher visually through provide the required components and provide the potentiality to link these components to generate sequence of bits. The second is building stream cipher algorithm through C# programming language. The proposed package provides the two approach. The following sections will discuss how provide these two approaches for constructing stream cipher algorithm in the proposed package.

IV. BUILDING STREAM CIPHER VISUALLY

To build stream cipher algorithm there are number of components that package provides. The common component is a LSFR, and also there are number of other components such as logical gates (XOR, AND, OR, NOR, NOT, NAND, JK), Table, Inner product, police and other components, to constructing stream cipher algorithm the designer has to drag the required components from the stream component list and drop it on area that allocated to building stream cipher algorithm. Fig1 shown the structure of proposed package.

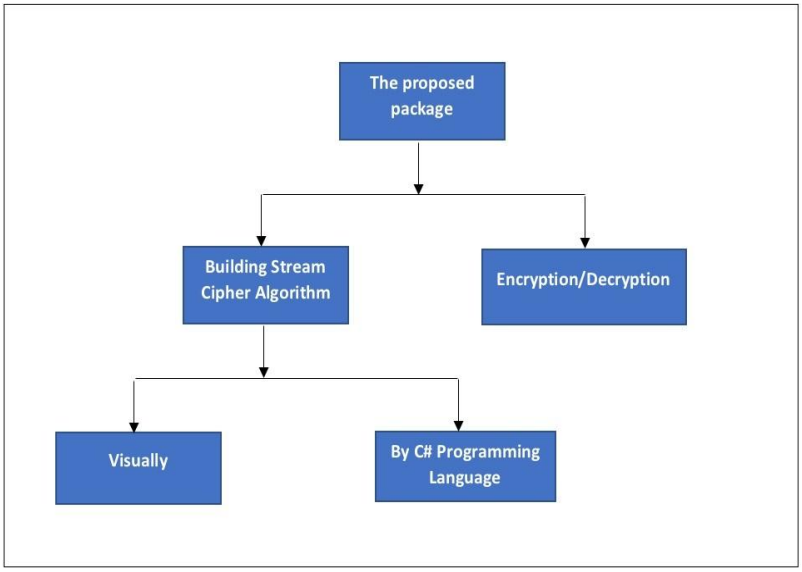

Fig.1 Structure of the proposed package.

A. Design of the Main Interface

The main interface of the proposed package consists of different parts as shown in fig2. The interface consists of four parts, the buttons bar, design area, stream cipher components bar, and information box.

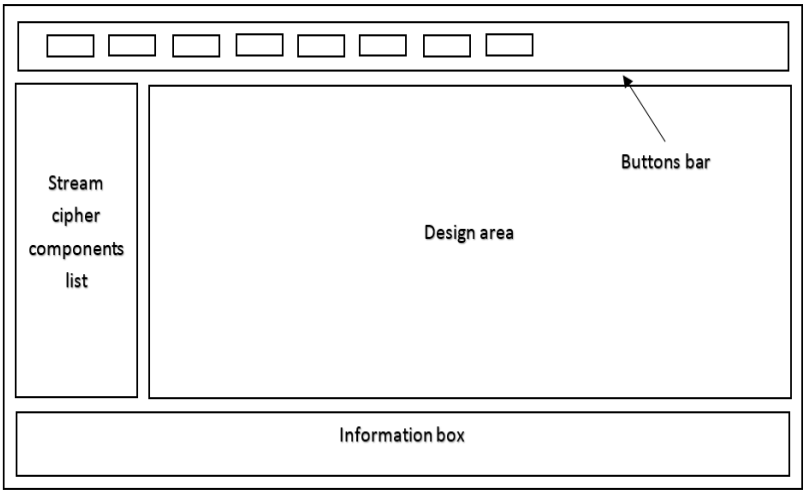

Fig.2 Main interface.

Following is descriptions of each part:

1- Buttons bar has following buttons:
   a) Button for generate one bit from the designed keystream generator in each click.
   b) Button for generate sequence of bits in each click, length of desired sequence enters in the textbox.
   c) Button for open the window of design stream cipher algorithms using C# programming language.
   d) Two buttons to save and load designed stream cipher algorithm.
   e) Button for delete any selected stream cipher components.
   f) Text box used to enter the following:
      1) The length of LFSR and table components.
      2) To enter the content of LFSR and table. Text box also used to enter a number that represent the length of sequence to be generated from the designed keystream generator in the design area.

2- Design area

It is area allocated to design keystream generator visually through drag any desired components from components list and drop them on the design area,

3- Stream cipher components
It is a list of stream cipher components that used in constructed keystream generator.
4- Information box
This text box used to show information about stream cipher components that exist in design area.

B. Design of Stream Cipher Components

To construct stream cipher algorithm there are number of components, the common component is the shift register. Every component further decomposes into three subcomponents, the first is the input, second is processing, the third is output. Each subcomponent will represent in the package by different class. Fig.3 shown the visual representation of the stream cipher component.

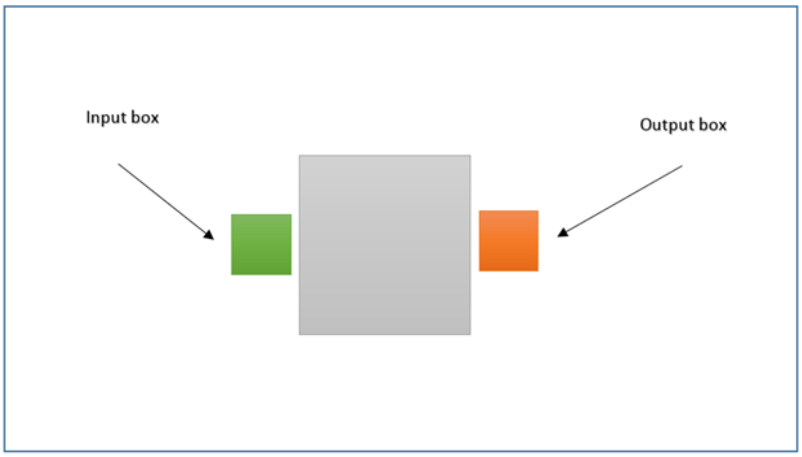

Fig.3 General design of stream cipher component.

Each component either accept one input or more than one input and has one output. Input box and output box will represent in package by separate class, each class has the following proprieties and function:

1- Proprieties called state, store either 1 or 0.
2- A mechanism to inform other component object that associated with when its state value changed.

C. Description of the Mechanism to Connect SCA Components

After putting all the stream cipher components on the design area in the main interface, the designer start to make connections among the components to allow of passing data among connected components. Fig.4 shown the scenario of linking two components (Comp1 and Comp2), each has one input and one output. To make connection between Output1 object that belong to comp1 and Input2 object that belong to Comp2, the designer first clicks on the Output1 of Comp1, then the designer clicks on the Input2 object of Comp2, after that the connection line will appear connecting the two components and it is imply the success of connection.

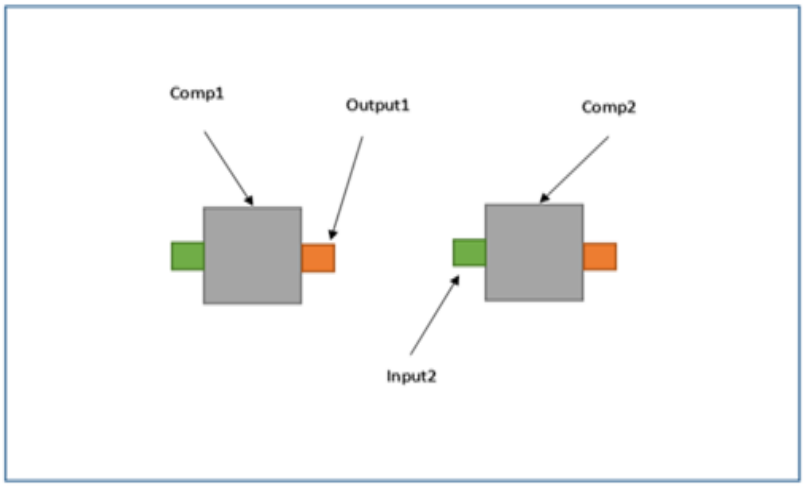

Fig.4 Example of connecting two components

D. Design of Stream Cipher Algorithm Components

There are different stream cipher components that can represent visually in the package. In following sections discussed the design of each component, how processing data that input to it, and how inform other components that linked to of changing in its state.

1) *Shift Register Component:* Fig.5 shows the design of the shift register. LFSR consists of cells that store each one only 1 or 0, each cell has two output box that used in link the cell to other components, except first cell that contain

two output box and one input box, the input box used to feed new bit to the register. The register also has clock box as pointed in fig.5, it is used to control the shifting operation in the LFSR such that if the state of clock box is zero then there is no shifting, otherwise, if the state is equal to one, the shift operation is allowed.

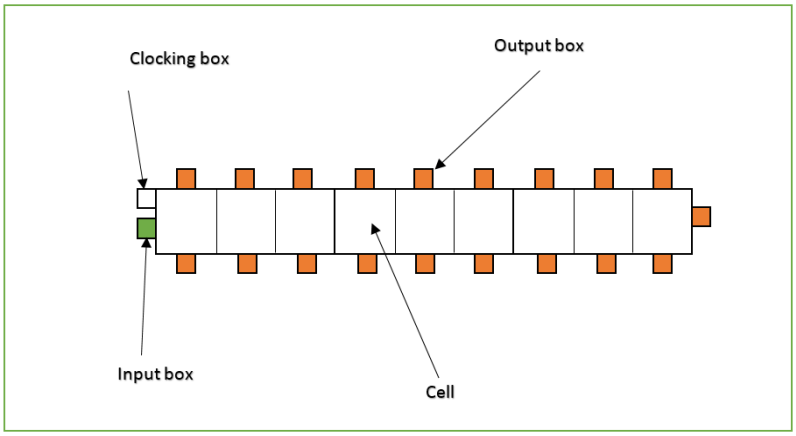

Fig.5 Design of LFSR

2) *Police Component:* Police component consists of three input and one output components. Fig.6 shown the visual representation of the police in the package:

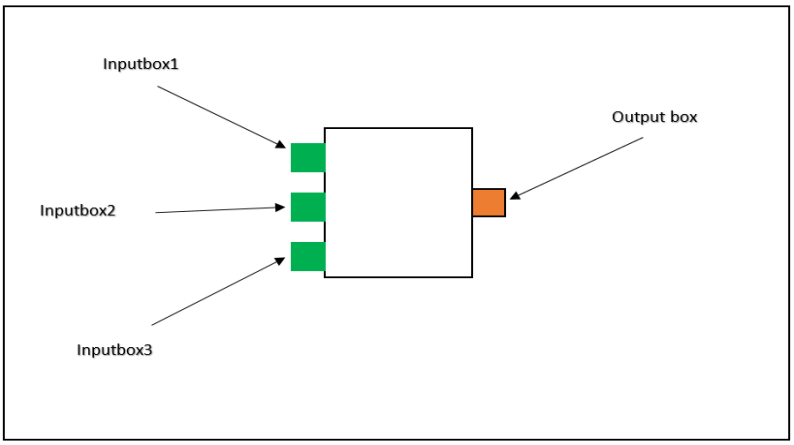

Fig.6: Design of police component

As in fig.6, if the state of Inputbox2 is equal to 0, the police output will be the state of the Inputbox1, otherwise if the Inputbox2 is equal to 1, the police output will be the state of the Inputbox3.

3) *The logical gates:* All logical gates consist of two input and one output except "NOT gate" which contain only one input. The proposed package provides gates: XOR, OR, AND, NOR, NAND, NOT. Fig.7 shows visual representation of the logical gate.

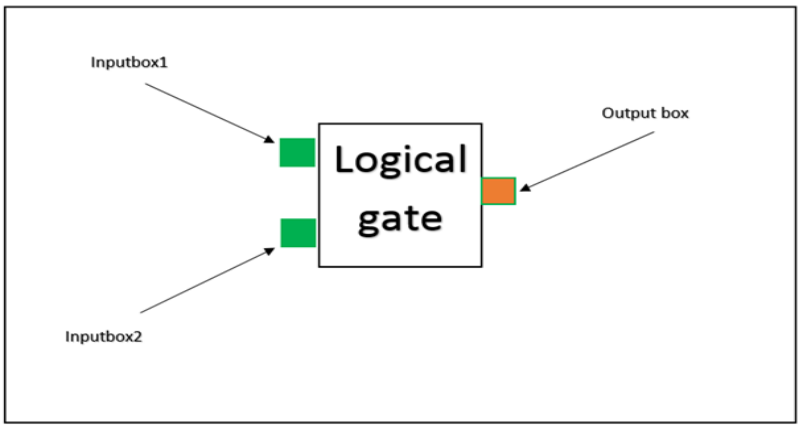

Fig.7 Design of logical gate component

The logical gate as shown in fig7. For example, if the logical gate is XOR, if the state of Inputbox1 and the state of Inputbox2 are both equal 1 or 0, then the XOR gate will output 0, otherwise, if the state of Inputbox1 and the state of Inputbox2 are different value, then the XOR gate will output 1.

4) *Table:* Fig.8 shown the table component in the package. The table can accept more than one input. Fig.9 shown the scenario of connecting three logical gates to the table.

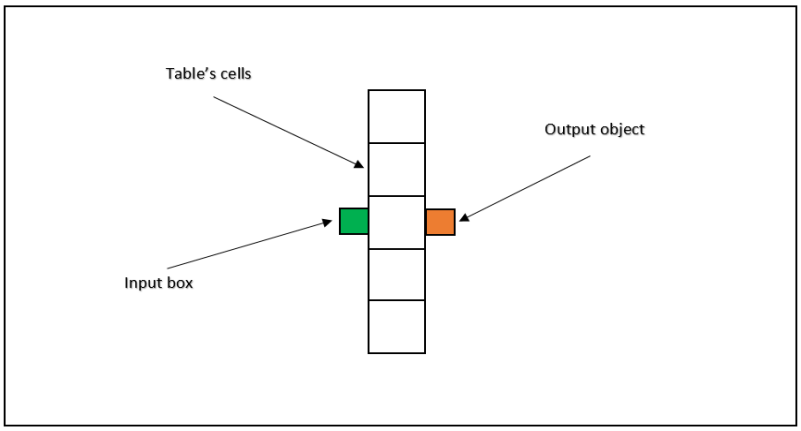

Fig.8 Design of table component

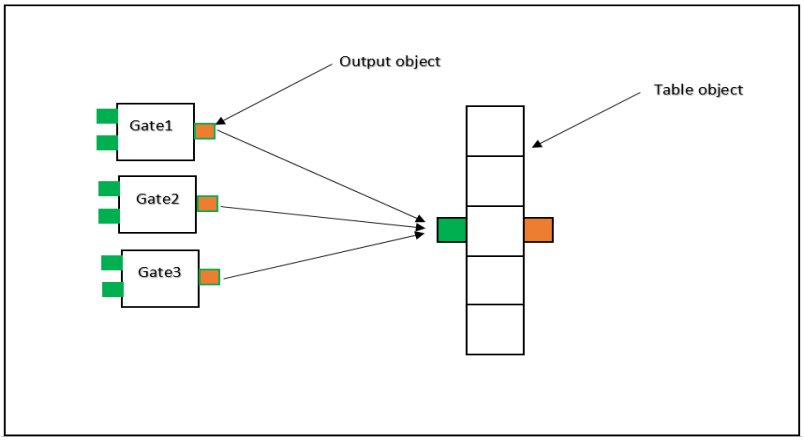

Fig.9 Example of connecting table

Each output object that connected to table has variable called state store the output of gate. If the designer first connects gate1 to table then connect gate2 and last gate3, if states of the of output objects of all gates are 0,1,1 respectively, then the table convert all states of the that gates to the decimal number and output the value of cell that its index is equal to the calculated decimal number.

V. BUILD STREAM CIPHER ALGORITHMS BY C# PROGRAMMING LANGUAGE

Fig.10 shows the package's interface of building stream cipher algorithm by C# programming language.

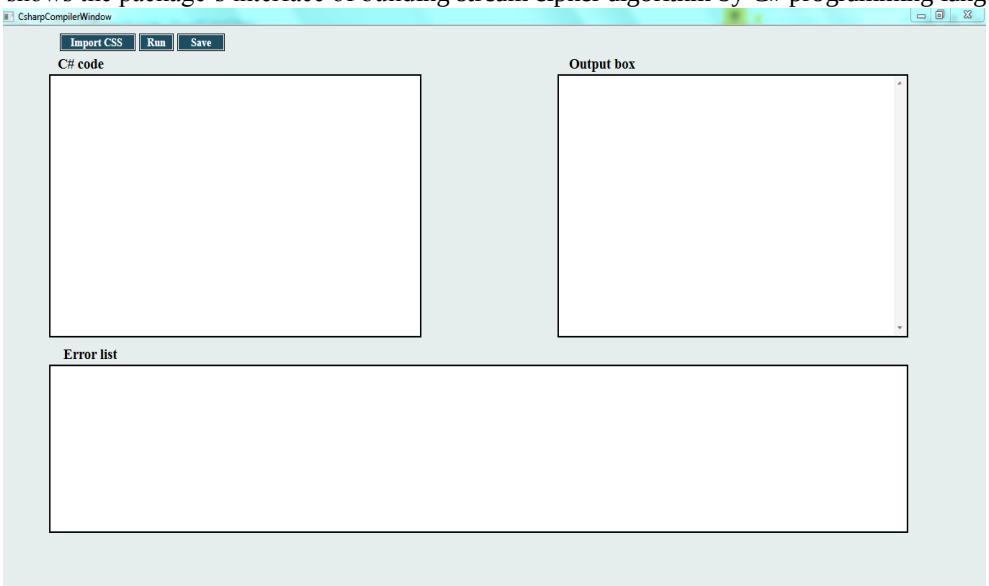

Fig.10 C# compiler interface

The designer can enter the C# code of any stream cipher algorithm in C# code box or can import the C# code by click on the import button in button bar. After enter C# code, the designer has to click on the Run button in button bar to start compiling and execute of entered code. After execute the entered C# code, the output of C# code will appear in output box. Any syntax error in the entered code this will show error message in the error box as pointing in fig.10.

VI. DESIGN ENCRYPTION/DECRYPTION INTERFACE

Fig.11 shows the package's interface of encryption and decryption. In the interface, there are number of buttons that perform following actions:

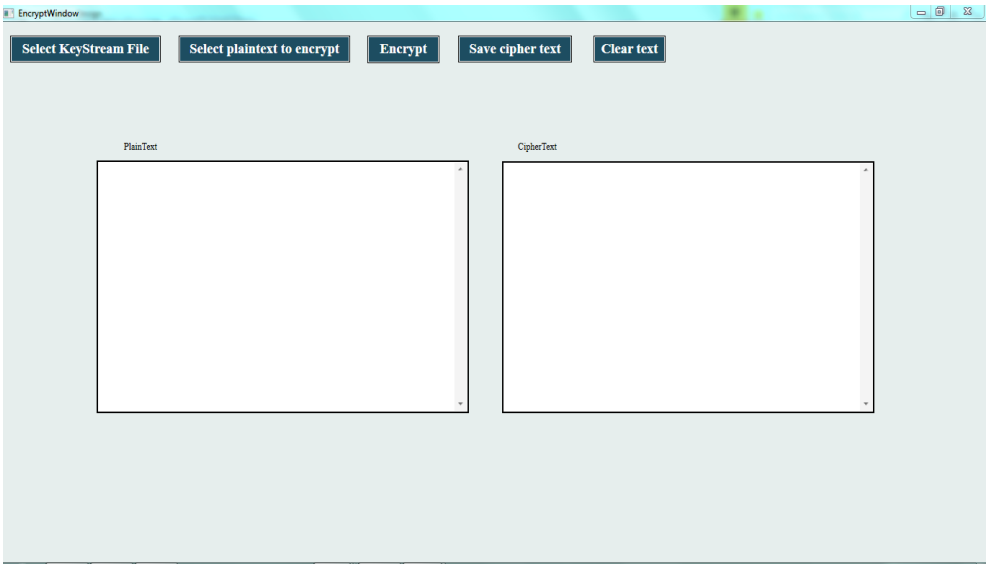

Fig.11 Encryption/decryption interface

1- Import any text to encrypt it or decrypt.
2- Import any key file to use it in encryption/decryption process.
3- Clear all text in text boxes.

When the designer clicks on the encrypt button, the following actions will perform:

1- Convert the plain text and key stream characters to a sequence of byte.
2- Mix the key bytes with plaintext bytes by using XOR function, the result represents the cipher text. The cipher text stored in new array.
3- The new sequence that produce from mix plaintext with key stream will appear in Textbox2 in the encryption /decryption interface.

To decrypt the cipher text, the designer has to do the following

1- Import the cipher text file through click on import Plain/cipher text button.
2- Import the key stream used in produce the cipher text.
3- Click on the Encrypt/Decrypt button to start mixing the cipher bytes with key bytes, the result is the plaintext and will appear in the Textbox2.

VII. DESIGN NUMBER OF KEYSTREAM GENERATORS BY PROPOSED PACKAGE

Following is implementation of three keystream generators that have been built by the proposed package.

A. Build Geffe Generator

Fig.12 shows the components of geffe and connection lines among them as executed in the package. The designed geffe generator consists of three LFSR, two AND gate, one Not gate, one XOR function. All LFSRs are initialized with value "fff".

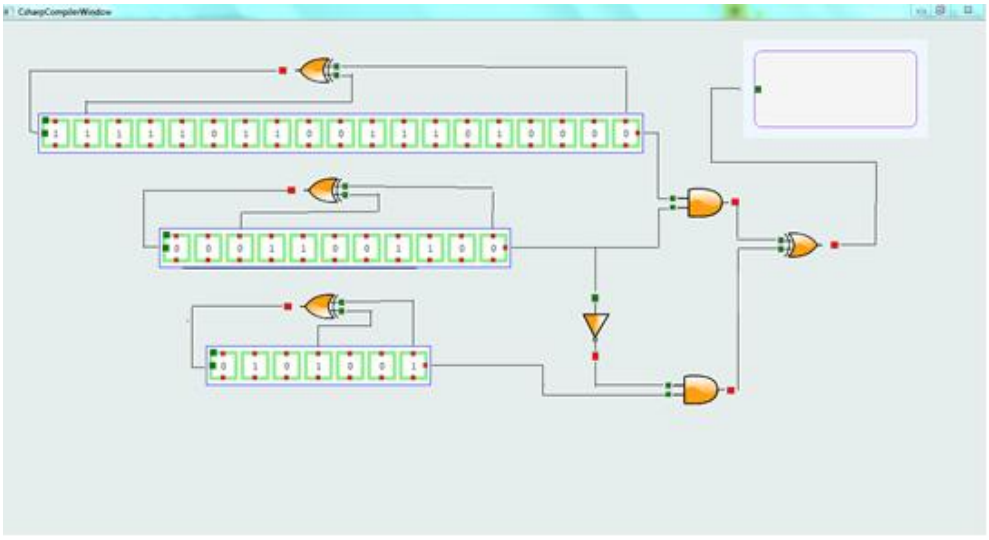

Fig.12 Designed geffe algorithm

B. Design New Keystream Generator

Some information about the new designed generator:

1- Number of used LFSRs are 6; LFSRs have lengths: (11, 4, 7, 19, 5, 13).
2- Other components used are: Police, JK flip flop, Table, XOR gate.
3- Length of Table component is 256, and initialize with value " abcdefghijklmnopqrstuvwxyzabcde".

Fig.13 show the new designed algorithm.

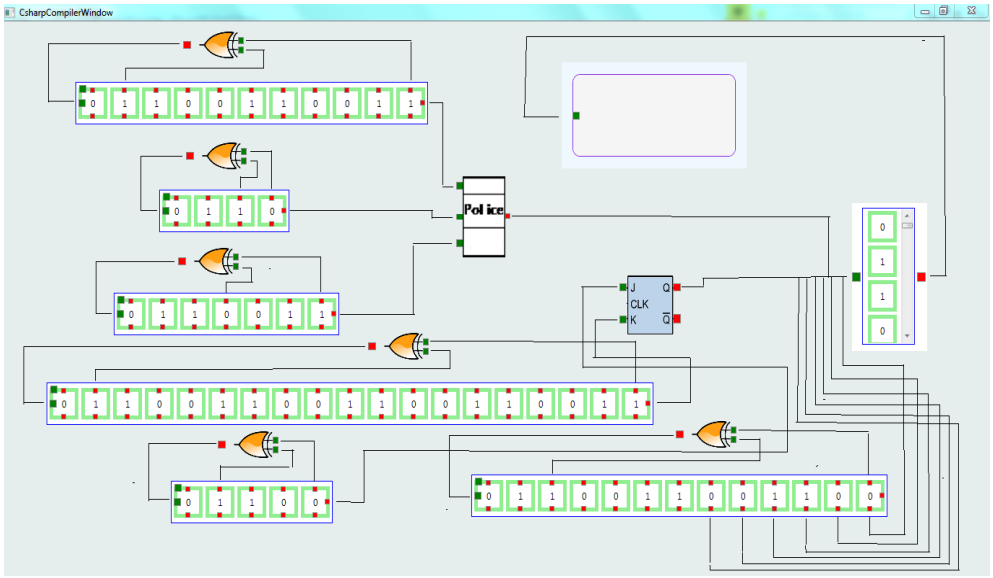

Fig.13 New keystream generator

C. Implement RC4 Stream Cipher Algorithm

Fig.14 shows build stream cipher algorithm by the proposed package, the algorithm is RC4 and has been built by enter C# code of RC4.

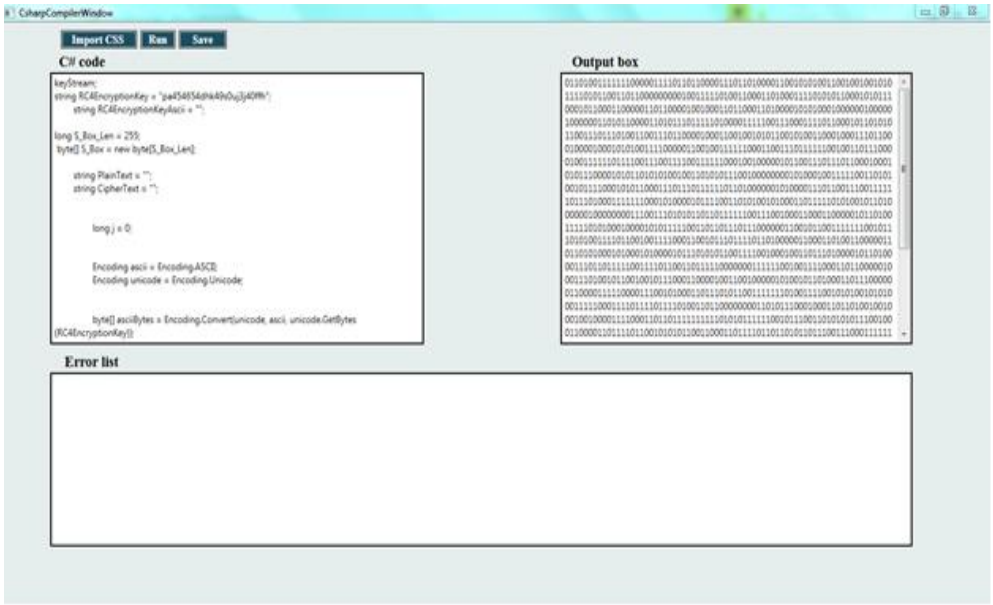

Fig.14 The implementation of RC4

After save the key produced by the code of RC4 in text file, the key should save as binary sequence of '1' and '0'. The designed RC4 algorithm produced 2040 bits.

VIII. CONCLUSION

1- The package provides easier and convenient tool to build unlimited number of stream cipher algorithm, either visually by use existing stream cipher components or by enter C# code.
2- The package provides the ability to the designer to implement any stream cipher algorithms by enter the C# code of stream cipher algorithm.
3- If the used LFSRs are more than one in the algorithm that designed visually in the package, all the LFSRs are shifting in parallel.
4- The encryption and decryption processes in the package are executed in parallel.

REFERENCES

[1] Rainer A. Rueppel, *Analysis and Design of Stream Ciphers*, Berlin, Springer, 1986.
[2] Darrel Hankerson, Alfred Menezes and Scott Vanstone, *Guide to Elliptic Curve Cryptography*, New York, Springer, 2004.
[3] Mohammad Ubaidullah Bokhari, Shadab Alam, Syed Hamid Hasan, "*A Detailed Analysis of Grain family of Stream Ciphers*", *I.J. Computer Network and Information Security*, 2014
[4] Richard R. Brooks, *Introduction to Computer and Network Security, Boca* Raton, CRC Press, 2014.
[5] Christof Paar and Jan Pelzl, *Understanding Cryptography a Textbook for Students and Practitioners*, Berlin, Springer, 2010.
[6] M.J.B. Robshaw, *Stream Ciphers RSA Laboratories Technical Report TR-701*, Redwood City, CA, RSA Laboratories, 1995.
[7] Andreas Klein, *Stream Cipher*, London, Springer, 2013.

