University of Thi-Qar Journal Vol.10 No.2 June 2015
Web Site: https://jutq.utq.edu.iq/index.php/main *Email: journal@jutq.utq.edu.iq*

Influence of Multiple Encryptions on the Security and Accuracy of Sent and Received Data

https://doi.org/10.32792/utq/utj/vol10/2/5

Jehan K. Shareef
Information systems (IS)
jihansh@yahoo.com
Media Faculty/ Thi- Qar University

Abstract

Encryption of data using multiple encryptions has been suggested in a range of contexts and can be used to protect the text versus cryptanalysis. Most work on this topic has intensive on the security of multiple encryptions against plaintext – ciphertext attacks, and has displayed creations secure by merging RC5 and CFB algorithm.

This research explains in detail the application which it develops to encrypt and decrypt information depending on RC5 algorithm with the cipher feedback (CFB) mode. RC5 is a block cipher which means the same key will be used for both encryption and decryption. The most commonly chosen is the RC5-32/12/16. Cipher text feedback (CFB) is a mode of operation for a block cipher. This application has both server and client's side which connect to each other using some ports to exchange encrypted and decrypted data by the RC5 algorithm with CFB mode. Clients connect to server to transmit data in a secure channel. The main aim is to simplify and describe the application which it developed using the RC5 and CFB mode and shows the secure usability of those algorithms.

Keywords: Multiple encryption, RC5, CFB, and Data security.

1. Introduction

1.1 Block Cipher

In cryptography, a mode of operation is an algorithm that uses a block cipher to provide an information service such as confidentiality or authenticity. A block cipher by itself is only suitable for the secure cryptographic transformation (encryption or decryption) of one fixed- length group of bits called a block. A mode of operation describes how to repeatedly apply a cipher's single-block operation to securely transform amount of data larger than a block (Menezes, Oorschot and Vanstone 1996) (Huang, Chiu and Shen 2013) .

1.2 RC5 Algorithm

RC5 is a fast block cipher planned by R. Rivest in 1994, RC5 is designated as RC5-w/r/b, the letters parameters w, r and b denote the word size (w) in bits (the possible value of w is 16, 32, and 64); r refers to a number of rounds in the range from 0 to 255; and b = 10-byte (80-bit) secret key variable (Rivest , The RC5 Encryption Algorithm. 1994) (Baldwin and Rivest 1996) (Rivest, Block Encryption Algorithm with Data-Dependent Rotations 1998). RC5 is a block cipher which means the same key will be used for both encryption and decryption. The most commonly chosen is the RC5-32/12/16. The three routines in RC5 are as the following:

1. Words addition module denoted by 2^w
2. Bit-wise X-OR
3. Data-dependent left rotation of x by y denoted by x<<y

Although RC5 is particularly simple and easy to execute, it delivers a good level of confidence in security, as long as adequate key length and sufficient rounds are employed.

Assume that the input block is given in two w-bit registers A and B. Also assume that key expansion has already been performed. So that the array S[0… t-1] has been computed. The following steps clear the structure work of RC5 algorithm:

- **The first step is to copy the secret key:**

K [0...b-1] **to an array** L[0..c-1] of c=[Image] words, where u=w/8 the number of bytes /word.

- **The second step of the key expansion:** is defending array *S* to a specific fixed key, pseudo-random bit pattern, using arithmetic progression modulo [IMAGE] which is determined by the magic constants [IMAGE] AND [IMAGE]. As example:

s[0] = P_{32};
for (int i = 1; i <= t - 1; i++)
{
s[i] = s[i - 1] + Q_{32};
}

- **The Third step:**

In the algorithm is to mix the user's secret key in three passed over the both S and L array which mean because of the potential different size of S and L. encryption and decryption implemented easily after the key expansion the array will be processed three times and the other may be handled more time.

The key expansion routine expands the user's secret key *K* to fill the expanded key array *S*, so that *S* resembles an array of *t=2(r+1)* random binary words determined by *K*. After the key expansion complete, the encryption implement using the following algorithm code:

A=A+S[0];

B=B+S[1];

For i=1 to r do

*A=((A^B)<<<B)+S[2*i];*

*B=((B^A)<<<A)+s[2*i+1];*

Also, decryption algorithm implements using the following algorithm code:

For i=r downto 1 do

*B=((B-S[2*i+1])>>>A)^A;*

*A=((A-S[2*i])>>>B)^B;*

B=B-S[1];

A=A-S[0];

The output is in the registers A and B. Figure (1) describes the encryption and decryption steps work of RC5 algorithm (Stallings 2002).

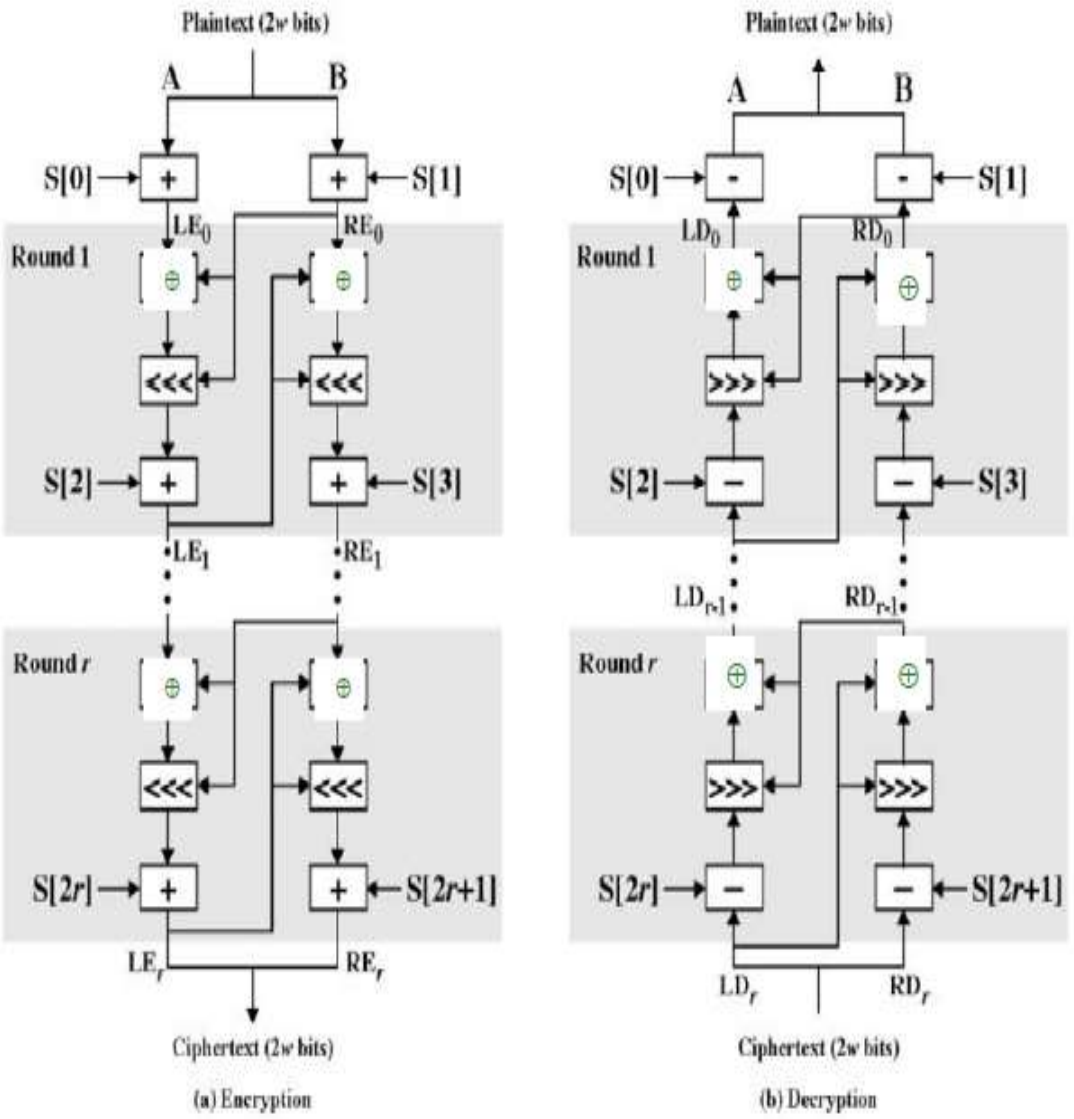


Figure (1): Encryption and Decryption schemes of RC5

1.3 Cipher Feedback Mode

Cipher text feedback (CFB) is a mode of operation for a block cipher. In contrast to the cipher block chaining (CBC) mode, which encrypts a set number of bits of plaintext at a time, it is at times desirable to encrypt and transfer some plaintext values instantly one at a time, for which cipher text feedback is a method. Like cipher block chaining, cipher text feedback also makes use of an initialization vector (IV).

CFB needs an initialization vector and it is use a block cipher as a component of a random number generator. In CFB mode, the previous cipher text block is encrypted and the output is XORed with the current plaintext block to create the current cipher text block. The XOR operation conceals plaintext patterns. Plaintext cannot be directly worked on unless there is retrieval of blocks from either the beginning or end of the cipher text (Group, NIST Computer Security Division's (CSD) 2013) (Ferguson, Schneier and Kohno 2010). CFB uses same function for encryption and decryption.

In fact, CFB is primarily a mode to derive some characteristics of a stream cipher from a block cipher. In common with CBC mode changing the IV to the same plaintext block results in different output. However the IV need not be secret, some applications would see this desirable. Chaining dependencies are similar to CBC, in that reordering cipher text block sequences alters decryption output, as decryption of one block depends on the decryption of the preceding blocks (Alfred , Paul and Scott 1996) (Kuo-Tsang , Jung-Hui and Sung-Shiou 2013).

The decryption use same process while the received cipher text unit is XORed with the output of the encryption function to produce plaintext unit. Figure (2) shows cipher feedback (CFB) mode encryption (Bond and Bement 2001).

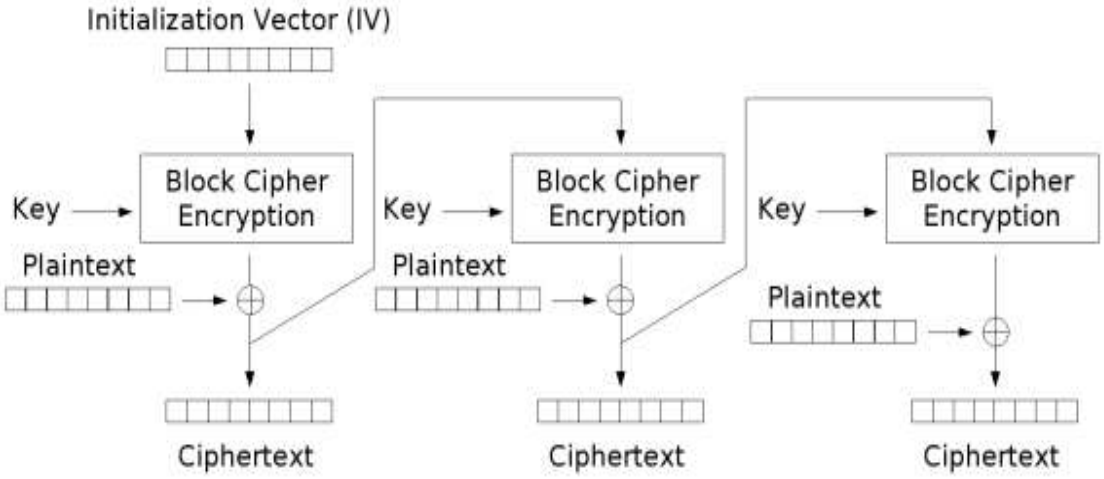


Figure (2): Cipher Feedback (CFB) Mode Encryption

See figure (3) it clears cipher feedback (CFB) mode decryption (Bond and Bement 2001).

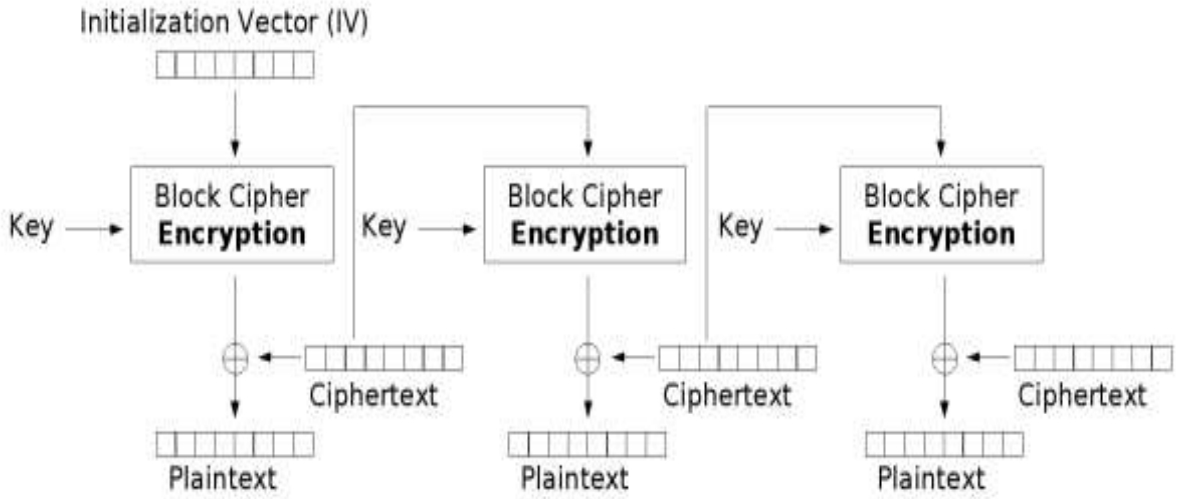


Figure (3): Cipher Feedback (CFB) Mode Decryption

2. The Preparation Procedures of this Work

2.1 The Main Progress

The application in this work has client and server and they are connected using local network via IP. The mechanisms of this work are as the following:

In the beginning, server has the password for encryption and decryption. in client's side of the program it require the user to connect via the connect form by entering his nickname, his IP, and encryption/decryption key, in order a client can lunch the program; after filling all filed with the required data, client will be able to connect. Client is able to communicate with any other clients who are already connected as well to the server. All messages which are flowing through the server are encrypted before being sent to server. Continuing, in server side the server write the message then decrypt the message and send it to other clients, in client side when they receive the encrypted message it decrypt it and write plain text to the screen. The whole progress is: a message encrypted and sent through the server and when clients receive it they decrypt it and see it as plain text.

2.2 Server Side

Server always listen to any request coming from any client if any request to server from client received to send message to other clients the server send the message to the clients which are connected to the server, the server listen to port number 8888 for any request attempt, after that the secret key should be provided to be able to encrypt and decrypt the message arrive. Here how it's done by C# programming language:

```
TcpListener serverSocket = new TcpListener(8888);
serverSocket.Start();
Console.Write("Please enter the key:");
    key = Console.ReadLine();
```

The server automatically will add the client name in client list when connect established and new client list will be created.

clientsList.Add(dataFromClient,clientSocket);

The message received from the client will be sent to all clients who are available in client list

broadcast(dataFromClient + " Joined ", dataFromClient, false);

Client list is a hash table. It stores the client name and create instance of the client socket. Server creates a thread for each client, when client connect to server handle; client class will handle each client as a separate thread, handle class has *dochat* function which will be dealing with all communication between server and the clients

handleClinet client = new handleClinet(key);

client.startClient(clientSocket, dataFromClient, clientsList);

2.3 Client Side

As mentions in server side above, server using the port 8888 so client will be able to connect to server using the port 8888 and client valid IP address. After client connect to server a thread starts to receive message from serve, this done by the following code *clientSocket.Connect(ipValue, 8888);*

Thread ctThread = new Thread(getMessage);

ctThread.Start();

After thread creation, thread will call the function GetMessage to give the ability to send and receive messages via stream sockets, GetStream function will call to return a Network Stream so client can be able to send and receive information, Receive Buffer Size will make the buffer ready to store the incoming data for any read operation. The

read function reads data to the buffer and returns the number of bytes which was successfully read.

serverStream = clientSocket.GetStream(); buffSize = clientSocket.ReceiveBufferSize;

byte[] inStream = new byte[buffSize];

int size = serverStream.Read(inStream, 0, buffSize);

Write method sent the message from clients to the network and then flushes the data from the stream for any other use.

2.4 RC5 class

RC5 takes the *K* parameter and assigns it to key and converts it to byte by calling GetFromStin function,

(B is the length of the key, U is the number of bytes in each word, u=w/ 8, T is total of sub key t=2r+2, and C is the length of the array of copied).

The round could be any number between 0-255. In case of this work, default is set to 16. First two word-sized binary constants *Pw* and *Qw* are defined; *w* can be 16, 32 and 64 bit. In this work *w=32* bit. The constant for the P_{32} and Q_{32}, defined respectively b7e15163 and 9e3779b9. The third step creation of *S* array will be based on addition of current element Q_{32} with the previous *S* element.

In the end user's secret key created by combining the 3 stages and *L* array will have different length in order to be able to make the implementation 3 times In the other hand encryption function is defined which has insert as parameter and blocks which are *r1* and *r2*. The Encode function contained 3 parameters *R1,R2* and rounds, so the encryption process done in three steps,

3 R1 XOR R2

4 Left rotate R1 and R2

5 Addition of result with sub key.

Decryption can be derived from encryption routine. It is the reverse of encryption function. The rounds start from last round, key starts from the last key, and rotation is done to the right. More secure communication can provide by cipher feedback mode which is already support it by the RC5 algorithm.

3. Application walkthrough

When server starts, it will ask user about the key to use it for encryption and decryption of the message which it sent from client. Then system will check the key if it is 8 characters or less. Also, the system will ask for more secure key if it is necessary. Figure (4) clear the operation of starting the server.

Figure (4): Starting the Server

This form (figure (5)) will allow client to connect to the server. It is require the password which already entered in the server and valid IP address. In case that client enters wrong key or IP address, system will ignores the entered data. Client also asked to enter a nickname to be able to connect. Without providing nick name, the client will get warning message asking to enter a nickname.

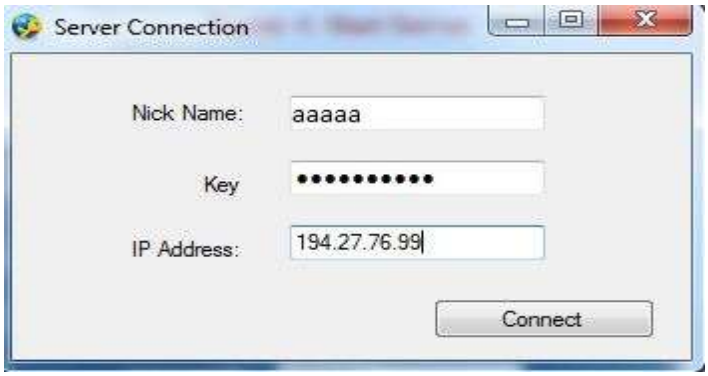
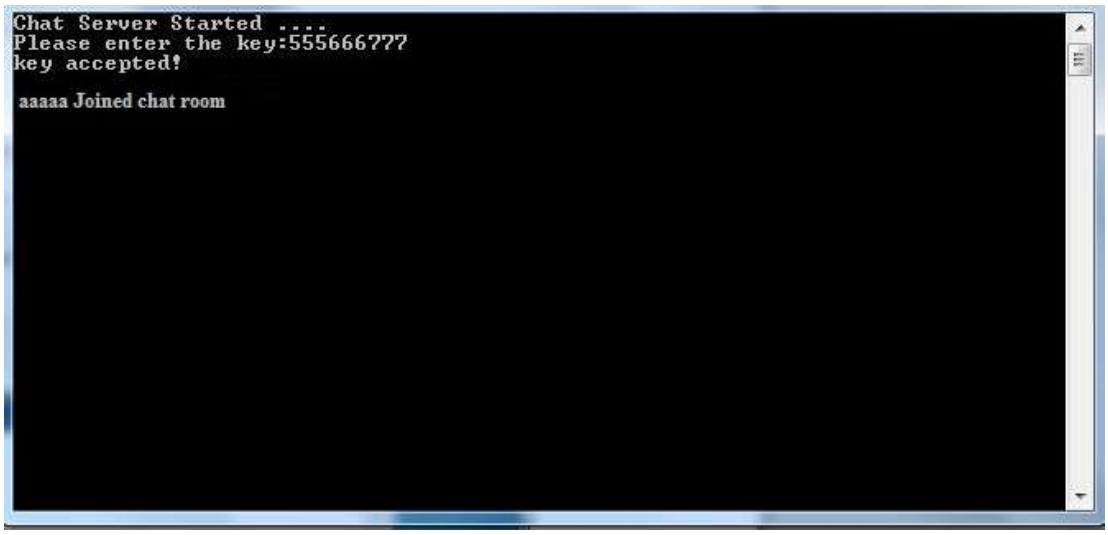


Figure (5): Client Connects to the Server.

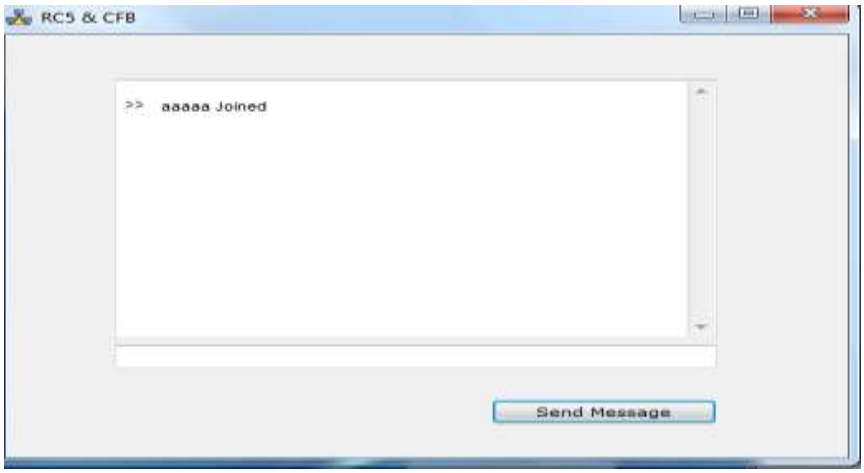
When everything goes right and all information true, the client will be able to connect to server and system will display a message saying that "You are successfully connected ", figure (6).

Figure (6): Successfully Connected

In server side it shows the nickname of client and it say "client name joined chat room". Also, in chat room it shows the same message. Currently, the client can write message to other clients, see figure (7) a and b.



(a)



(b)

Figure (7) a and b: Verification of Correct Password and IP Address

All message in the system are encrypted with mode of cipher feedback then it broadcast through the server and appear to all clients, In other word messages are encrypted by the server using the key, both client and server use the same key so that both will encrypt and decrypt correctly .

4. The Errors

Any empty or wrong information system will show sign near the wrong failed, figure (8).

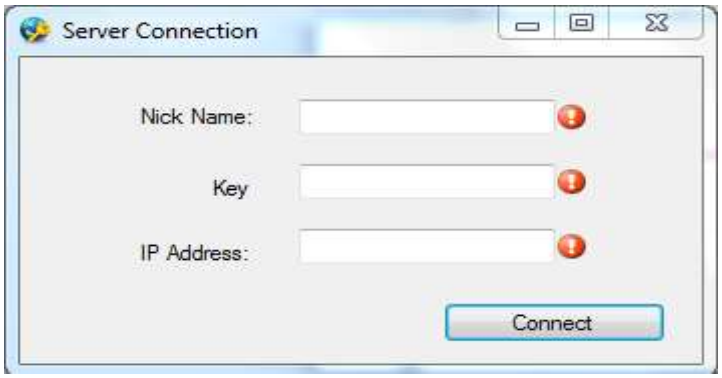


Figure (8): Error Case (1)

In case if the password don't match the server password, system will display a message and client cannot connect to the server in this case, figure (9).

Figure (9): Error case (2)

If clients enter wrong IP adress, system will get the following error message and client cannot connect to the server in this case as well, figure (10).

Figure (10): Error case (3)

5. Conclusions

İn conclusion, the multiple encryption provides strong security. For instance, if some secret encryption keys are cracked or some part of cipher texts are broken, the privacy of original text can still preserved. Such that, secure with multiple encryption will be an important part of electronic commerce in the future and successful level of security is required to earn the interest and trust of clients, merchants and financial administrations for online business over wireless network as well. The result of the application having secure channel of data flow,clients and server connected and changing information using RC5 algorithem with cipher feedback (CFB) mode. The key should be long enough and not something voluble to detect. CFB mode combines a block cipher with a stream cipher, It needs an initialization vector. Also, It use same function for encryption and decryption which makes it possible to choose the faster function and possible to use one-way-functions. Encryption of a plaintext block depends on its predecessors.

Bibliography

Allen, Arnold O. *Computer Performance Analysis with Mathematica.* Academic Press, 1994.

Bond, Phillip J., and Arden L. Bement. *Recommendation for Block Cipher Modes of Operation Methods and Techniques.* WASHINGTON: NIST Special Publication 800-38A, 2001.

Gropp, William, and et. al. *MPICH2 User's Guide.* U.S.A: Mathematics and Computer Science Division DAC Program, 2007.

Huang, Kuo-Tsang, Jung-Hui Chiu, and Sung-Shiou Shen. "A Novel Structure with Dynamic Operation Mode for Symmetric-Key Block Ciphers." *International Journal of Network Security & Its Applications (IJNSA),* no. 5, p1-19, 2013.

Jordan, Harry, and Gita Alaghband. *Fundamentals of Parallel Processing.* 2003.

Lerner , R. G., and G. L. Trigg. *Encyclopaedia of Physics.* 2nd. VHC publishers, 1991.

Wong, Parkson. *MPICH.* http://www.mcs.anl.gov/research/projects/mpich2/, 2012.

Alfred , J. Menezes, C. Van Paul , and A. Vanstone Scott . *Handbook of Applied Cryptography* (CRC Press.), p 228–2331996.

Baldwin, R., and R. Rivest. "The RC5, RC5-CBC,RC5-CBC-Pad, and RC5-CTS Algorithms." *RFC 2040,Network Working Group,* 1996.

Ferguson, N., B. Schneier , and T. Indianapo Kohno. "Design Principles and Practical Applications." *Cryptography Engineering* (Wiley Publishing), p 63-64,2010.

Gropp, william, and et al. *MPI: A Message-Passing Interface Standard.* University of Tennessee, 1996.

Group, NIST Computer Security Division's (CSD) . "Block cipher modes." *Cryptographic Toolkit*, 2013.

Kuo-Tsang , Huang, Chiu Jung-Hui , and Shen Sung-Shiou . "A Novel Structure with Dynamic Operation Mode for Symmetric-Key Block Ciphers." *International Journal of Network Security & Its Applications (IJNSA)*, 2013.

Menezes, Alfred J., Paul C. C.van Oorschot , and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

Rivest , R. L. "The RC5 Encryption Algorithm." *CryptoBytes* 1, p 9-11,1994.

Rivest, R. L. "Block Encryption Algorithm with Data-Dependent Rotations." *U.S. Patents*, no. 5,724,428 and 5,835,600 ,1998.

Rumelhart , D. E., G. E. Hinton, and J. L. McClelland. *A General Framework for Parallel Distributed Processing.* MIT Press: Cambridge, MA, 1986.

Stallings, William . *Cryptography and Network Security: Principles and Practice, 3/E.* Prentice Hall, 2002.

الخلاصة

تشفير البيانات باستخدام طريقة التشفير المتعدد تم اقتراحها على مدى واسع من سياق التشفير والتي من الممكن ان تستخدم لحماية النص ضد محاولات الاشخاص الموجهة لكسر النص المشفر. معظم العمل في هذا البحث تركز على امنية التشفير المتعدد ضد الهجمات على النص المشفر والنص الذي يتم فك شفرته وتم عرض عملية امان مبدعة في هذا المجال باستخدام الدمج بين خوارزمية **RC5** وخوارزمية **CFB**.

هذا البحث يوضح بالتفصيل التطبيقات التي طُورَت من اجل تشفير وكسر شفرة معلومات اعتمادا على الدمج بين خوارزمية RC5 و CFB . طريقة تشفير المعلومات في خوارزمية RC5 هي تشفير مجموعة او كتلة معلومات والتي تستخدم نفس المفتاح في التشفير وفك الشفرة . النموذج الاكثر شيوعا في الاستعمال بالنسبة لخوارزمية RC5 هو RC5-32/12/16 . نظام CFB هو نموذج تشفير مجموعة او كتلة معلومات ايضا. ان هذا العمل عبارة عن ان الخادم والعميل يرتبطون مع بعضهم باستخدام مجموعة (ports) من اجل تبادل البيانات المشفرة والغير مشفرة باستخدام خوارزمية RC5 و CFB . يتم ربط العميل مع الخادم لنقل البيانات عبر قناة امينة. الهدف من هذا العمل هو لتبسيط عملية نقل المعلومات والحفاظ عليها بصورة امينة وتوضيح سهولة استعمال وتبادل البيانات باستخدام الخوارزميتين أعلاه.

الكلمات المفتاحية : التشفير المتعدد، **RC5** ، **CFB** ، وامنية البيانات .