Republic of Iraq

Ministry of Higher Education

&

Scientific Research

University of Technology

Department of Computer Science

Network Management Branch

Design and Implement Secure Communication Channel

-(Peer to Networks)

"الاجتهاد عنوان التميز"

بذلك ثابرنا لنقدم إلى قسم علوم الحاسوب وادارة الشبكات

(الجامعة التكنولوجية)

مشروع تخرجنا كجزء أساس لنيل شهادة البكالوريوس

فـــي

(علـــوم الحـــاســـوب)

الإشراف بفخر

(الدكتورة الفاضلة سكينة حسن هاشم)

إعداد الطالب

(إبراهيم محمد مدحت)

و

الطالبة

(رؤى علي عبد الوهاب)

بسم الله الرّحمن الرحيم

(يرفعِ اللهُ الَّذين آمنوا منكم
والَّذينَ أوتوا العِلمَ درجاتٍ)

صدق الله العظيم

الإنــسانيون ..

رسل وأنبياء .. علماء وأفاضل

لبناة مجد الوطن ..

الأم والأب.. الأستاذ والمعلم وأخوة الأرض والعقيدة..

لجميعكم أهُـدي هذا المسعى..

تقبلوهُ مني بتواضع..

"هو خيرُ ما أبدعتُ واجتهدتُ لأجلهِ لأكونَ كما تمنيتم لي"

أن وجدتموني أهلا للعلم .. فلا تردوا الهدية

الشكر والثناء ..

لست من يمنح التقدير لمعلمه لأنه أبلغُ علما وسيدوم، غير إن إبداء الامتنان بصدر رحب ورأس شامخة هو ما املك لأني سأظل الطالب العبد لأستاذي الذي علمني بدل الحرف حروفا..

وكيف الحال لو كان الأستاذ نفر لا يسمى بعدد؟! بل هم ثلة خيرة من الأفاضل والفاضلات من الأستاذة والمعيدين في قسم علوم الحاسوب وادارة الشبكات ..

جميعكم محط فخر وعطاء.. سيبقى القلب قبل العقل واللسان يفوح بما غمرتموني من علوم وإنسانية وعون ألهمني حب ما أقدم للمجتمع يوم تخرجي.

لكن للثناء أيضا استثناء مع فاضلة منحتني الأمل

(الدكتورة سكينة حسن هاشم) التي كان لي فخر إشرافها على انجاز هذا المشروع..

دامت روحها وأرواحكم محبة لبناء أجيال تلو أخرى تخدم العلم والوطن.

Certificate

I certify that the preparation of this project

"Design and Implement Secure Communication Channel – (Peer to Networks) "

Where prepared by:

" Ibrahim M. Medhat " & " Ruaa Ali "

was made under my supervision in the Computer Sciences Department at the University of Technology as partial fulfillment of the requirements for degree of the BSC in Computer Science.

Signature:

Name:

Date:

Certificate

We certify, as an examining committee, that we have read this project entitled ***"Design and Implement Secure Communication Channel – (Peer to Networks) "***

Examined the students (" Ibrahim M. Medhat " & " Ruaa Ali ") in its content and found it meets standard of final year project for ***the degree of Bachelor of Science of Computer Science***

Signature:

Name:

Date:

Signature:

Name:

Date:

Signature:

Name:

Date:

TABLE OF CONTENTS

Chapter One

General Overview

1.1 Introduction

Network Security is an organization's strategy and provisions for ensuring the security of its assets and of all network traffic. Network security is manifested in an implementation of security policy, hardware, and software. For the purposes of this discussion, the following approach is adopted in an effort to view network security in its entirety: [1]

1. Policy
2. Enforcement
3. Auditing

Policy

The IT Security Policy is the principle document for network security. Its goal is to outline the rules for ensuring the security of organizational assets. Employees today utilize several tools and applications to conduct business productively. Policy that is driven from the organization's culture supports these routines and focuses on the safe enablement of these tools to its employees. The enforcement and auditing procedures for any regulatory compliance an organization is required to meet must be mapped out in the policy as well. [3]

Enforcement

Most definitions of network security are narrowed to the enforcement mechanism. Enforcement concerns analyzing all network traffic flows and should aim to preserve the confidentiality, integrity, and availability of all systems and information on the network. These three principles compose the CIA triad: [3]

- Confidentiality - involves the protection of assets from unauthorized entities
- Integrity - ensuring the modification of assets is handled in a specified and authorized manner
- Availability - a state of the system in which authorized users have continuous access to said assets.

Strong enforcement strives to provide CIA to network traffic flows. This begins with a classification of traffic flows by application, user, and content. As the vehicle for content, all applications must first be identified by the firewall regardless of port, protocol, evasive tactic, or SSL. Proper application identification allows for full visibility of the content it carries. Policy management can be simplified by identifying applications and mapping their use to a user identity while inspecting the content at all times for the preservation of CIA. [3]

The concept of defense in depth is observed as a best practice in network security, prescribing for the network to be secured in layers. These layers apply an assortment of security controls to sift out threats trying to enter the network:

- Access control
- Identification
- Authentication
- Malware detection
- Encryption
- File type filtering
- URL filtering
- Content filtering

These layers are built through the deployment of firewalls, intrusion prevention systems (IPS), and antivirus components. Among the components for enforcement, the firewall (an access control mechanism) is the foundation of network security.

Providing CIA of network traffic flows was difficult to accomplish with previous technologies. Traditional firewalls were plagued by controls that relied on port/protocol to identify applications—which have since developed evasive characteristics to bypass the controls—and the assumption that IP address equates to a users identity.

The next generation firewall retains an access control mission, but reengineers the technology; it observes all traffic across all ports, can classify applications and their content, and identifies employees as users. This enables access controls nuanced enough to enforce the IT security

policy as it applies to each employee of the organization, with no compromise to security. [3]

Additional services for layering network security to implement a defense in depth strategy 8have been incorporated to the traditional model as add-on components. Intrusion prevention systems (IPS) and antivirus, for example, are effective tools for scanning content and preventing malware attacks. However, organizations must be cautious of the complexity and cost that additional components may add to its network security, and more importantly, not depend on these additional components to do the core job of the firewall.[3]

Auditing

The auditing process of network security requires checking back on enforcement measures to determine how well they have aligned with the security policy. Auditing encourages continuous improvement by requiring organizations to reflect on the implementation of their policy on a consistent basis. This gives organizations the opportunity to adjust their policy and enforcement strategy in areas of evolving need. [4]

1.2 Aim of Project

This project aim to advance the network security; by Integrate the message using message digest and then encrypt the overall message (plaintext and digest) with stream cipher. Receiver get the integrated/encrypted message will decrypt it and then separate the digest from the original message. Finally apply the message digest on the original plaintext message and compare the resulted digest with the separated one. If they equal the message is true else message is modified.

1.3 Project Outlines

1. Chapter two: “message digest and stream cipher”
2. Chapter Three : “ Design and implement proposal”
3. Chapter Four : “ Conclusions and Recommendations”

Chapter Two

Message Digest and Stream Cipher

2.1 Introduction

Message digest functions also called *hash functions* , are used to produce digital summaries of information called message digests. Message digests (also called *hashes*) are commonly 128 bits to 160 bits in length and provide a digital identifier for each digital file or document. Message digest functions are mathematical functions that process information to produce a different message digest for each unique document. Identical documents have the same message digest; but if even one of the bits for the document changes, the message digest changes. [1]

Stream cipher is a symmetric key cipher where plaintext digits are combined with a pseudorandom cipher digit stream (keystream). In a stream cipher each plaintext digit is encrypted one at a time with the corresponding digit of the keystream , to give a digit of the ciphertext stream. [2]

2.2 message-digest algorithm (MDA)

A message-digest algorithm is also called a hash function or a cryptographic hash function. It accepts a message as input and generates a fixed-length output, which is generally less than the length of the input message. The output is called a hash value, a fingerprint or a message digest. Figure (2.1) shows the basic message digest process. [2]

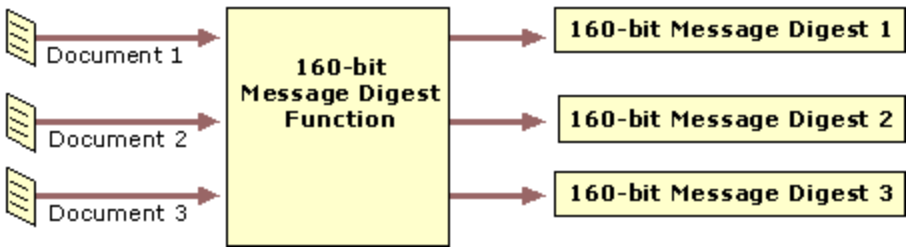


Figure (2.1) Example of the Message Digest Process

2.2.1 Properties of a message-digest algorithm

When people plan to design a message-digest algorithm, they try to make the algorithm satisfy the following properties:

- . It should be one-way. Given the message digest, it is hard to get the original message.
- . Given both input and output, it is difficult to find another input message which generates same output.
- . It should be collision-resistant. It is computationally infeasible to find two messages, which generates same message digest. This property is not same as the second property. It is easier to make attack on this property than on the second property.
- . The message digest should satisfy pseudo-randomness.

When all of the above properties are satisfied, we call the algorithm a collision-resistant message-digest algorithm. It is unknown whether collision-resistant message-digest algorithm can exist at all. [2]

2.2.2 Possible applications

Message-digest algorithms are mainly used in implementing digital signature. In this case, all of the above properties are required. However, the requirement is quite different when different applications use these algorithms. An application may rely upon some or all of the properties of the MDA. For example, some applications use the one-way property of a MDA. Because of its property of pseudo-randomness, MDA is also used to be part of the mechanism for random number generation. [2]

2.2.3 Why is MD5 fast?

There are three kinds of operations in MD5:

- . Bitwise Boolean Operation
- . Modular Addition
- . Cyclic Shift Operation

All these three operations are very fast on a 32-bit machine. So MD5 is quite fast. [2]

2.2.4 The mechanism of MD5

MD5, as well as MD2 and MD4, follows a design principle proposed by Merkle and Damagard. Its basic idea is to do hash in a block-wise mode. In a word, MD5 consists of two phases: padding phase and compression phase. In the padding phase, some extra bits (1 to 512bits) are appended to the input message. The result bits is congruent to 448 mod 512. Then the length of the initial message is transformed to a 64-bit binary-string(if the length is greater than 2^{64}, the lower 64-bit is used) and this 64 bits is added to the tail of the

message too. So the padding phase ends with a bit stream that consists of one or more 512-bit blocks. In the compression phase, a compression function is used on each 512-bit block and generates a 128-bit output. The output is always involved in the calculation of next round.

For convenience, we describe the algorithm through the following five steps:

(a) Add padding bits behind the input message, This step is to elongate the initial message and make its length be congruent to 448 mod 512. First, a single bit "1" is appended to the message. Then, a series of "0" bits are appended so that Length(the padded message) ≡ 448 mod 512

For example, suppose the initial message has 1000 bits. Then this step will add 1 bit "1" and 471 bits "0". As another example, consider a message with just 448 bits. Since the algorithm doesn't check whether the initial length is congruent to 448 mod 512, one bit "1" and 511 bits "0" will be appended to the message. Therefore, the padding bits' length is at least one and at most 512.

(b) Add a 64-bit binary-string which is the representation of the message's length, Here; please pay attention to the meaning of the 64-bit binary-string. You shouldn't regard it as first 64 bits of the initial message. It is indeed the binary representation of the length of the initial message. For example, suppose the message is 1000bits long. Its 64-bit binary representation would be 0x00000000000003E8. If the message is very long, greater than 2^{64}, only the lower 64 bits of its binary representation are used.

(c) Initialize four 32-bit values; these four 32-bit variables would be used to compute the message digest. We denote them by A, B, C, D. Their initial values are:

A = 0x67452301

B = 0xEFCDAB89

C = 0x98BADCFE

D = 0x10325376

(d) Compress every 512-bit block

(e) Generate the 128-bit output

There are four rounds and sixteen steps in each round. [2]

2.3 Stream Cipher

An alternative name is a state cipher, as the encryption of each digit is dependent on the current state. In practice, a digit is typically a bit and the combining operation anexclusive-or (xor). The pseudorandom keystream is typically generated serially from a random seed value using digital shift registers. The seed value serves as the cryptographic key for decrypting the ciphertext stream. Stream ciphers represent a different approach to symmetric encryption from block ciphers. Block ciphers operate on large blocks of digits with a fixed, unvarying transformation. [1]

This distinction is not always clear-cut: in some modes of operation, a block cipher primitive is used in such a way that it acts effectively as a stream cipher. Stream ciphers typically execute at a higher speed than block ciphers and have lower hardware complexity. However, stream ciphers can be susceptible to serious security problems if used incorrectly (see stream cipher attacks); in particular, the same starting state (seed) must never be used twice. Whenever your browser establishes a "secure" connection to a web site, it encrypts the data. The encryption often takes place byte-by-byte, since the software can't always predict how much data will be sent. This encryption style requires a *stream cipher*. [1]

Stream ciphers use a deceptively simple mechanism: you combine the plaintext data, bit by bit, with "key" bits, using the*exclusive or* operation. This is often abbreviated *xor*, and denoted by ⊕ – a circle with a cross. Figure (2.2) present stream cipher. [4]

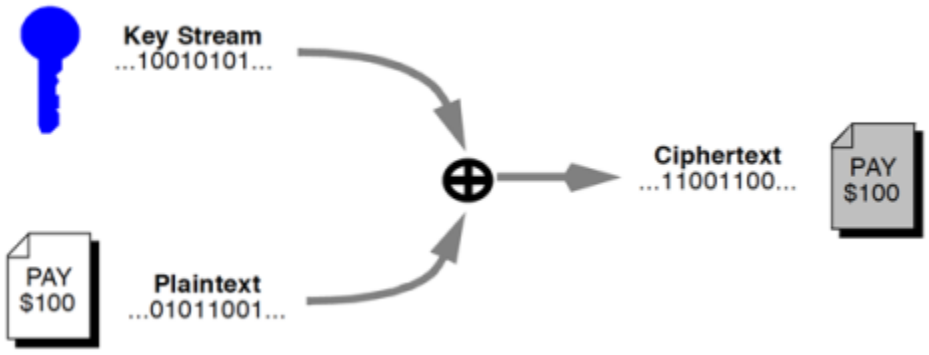


Figure (2.2) Example of the Stream Cipher Process

A "pure" stream cipher consists of three parts:

- a shared secret,
- a process for generating a random-looking bit stream, and

- the xor operation.

Originally, all web sites used Rivest Cipher #4 (RC4) to encrypt their secure connections. RC4 can use 128 bits of shared, secret data to generate a random-looking bit stream. This bit stream is then combined, bit by bit, with the message being sent. [Note that RC4 is no longer safe to use - we use it here purely as an example] [4]

When Alice sends a message to Bob, encryption happens as follows. Ahead of time, Alice and Bob share their secret. When Alice has a message to send to Bob, she uses the shared secret and the RC4 cipher to encrypt it. Upon receipt of the encrypted message, Bob uses the shared secret and the RC4 cipher to decrypt it. This is much more convenient that a one-time pad, which requires a separate shared secret equal to the size of every message sent. [4]

The process for generating the bit stream is the heart of the technique, and usually referred to as the *cryptographic algorithm*. Even if an eavesdropper (call him Peeping Tom) happens to see part of this bit stream, he should not be able to predict other parts of the bit stream. Ideally, Tom would need a copy of the shared secret in order to recover the message. There should be no way to recover the message that's easier than trying to guess the 128-bit secret through trial and error. [2]

This make things much simpler for Alice and Bob. Before sending a message, they share a 128-bit secret. When Alice sends her message, she starts up the RC4 algorithm, feeds it her key, and encrypts her message, bit by bit, using xor. Upon receipt, Bob runs RC4, enters his copy of the shared

secret, and gets back the same bit stream. He decrypts the message by applying xor to the message and the bit stream. [4]

The security of a stream cipher depends on the quality of the algorithm, but it also depends on proper use. In particular, neither Alice nor Bob should intentionally use that shared secret ever again to send a message. In fact, if Bob replies to Alice's message, he must use a different shared secret. If he uses the same shared secret, he will encrypt his message with the same bit stream that Alice used. Then Peeping Tom can retrieve both messages scrambled together, as shown here. [3]

Chapter Three

Proposed Integrated Stream Cipher System

3.1 Intrduction

In this chapter will explain the design and implementation of the proposal; which consist of the following consequence steps;

1. Enter palin text.
2. Convert the plain text into binary.
3. Process it by simple integirty algorithms; which xoring the charaters of the plain text.
4. Add the obtained charater to the end of the plaintext.
5. Enter the key.
6. Convert the key to binary.
7. Apply stream cipher with plaintext + integrity charatect and key.

And for decyption will reverse the steps above, the following section will present the desgin and the implementation of the proposal.

3.2. Desgin of the Proposal

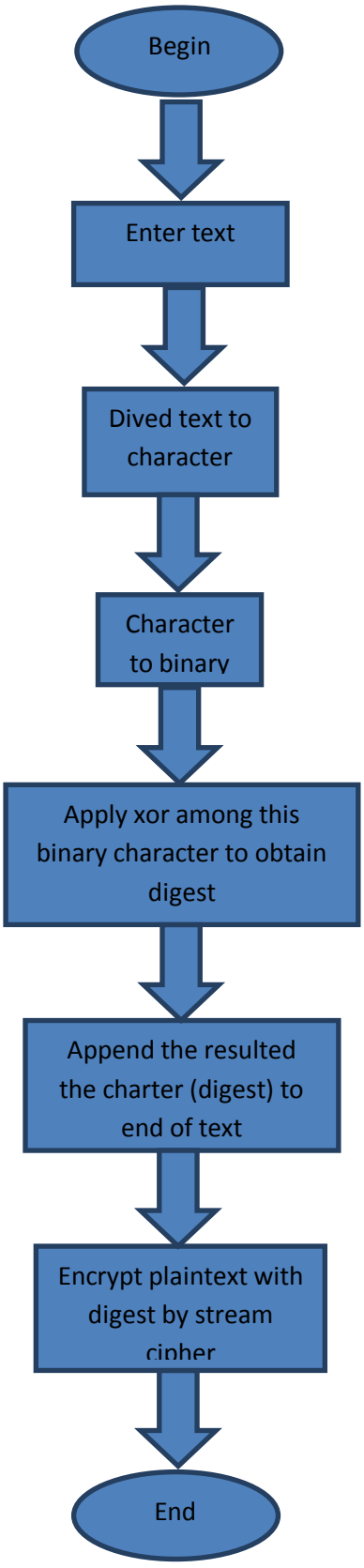
The algorithm of sender will be explaied in algorithm (3.1) and algorithm of reciever will be explaied in algorithm (3.2).

Algorithm (3.1) the sender user
Input: plaintext Output: ciphertext
Begin Read (string) Divide string to character Covert charter to ASCII Work xor between the ASCII Covert the result of ASCII to charter Add the charter to end of string Read the key Conformation of the length of key equal the length of the string (with the add character) Covert the key to ASCII Work xor between ASCII (string) And ASCII (key) The result is cipher End

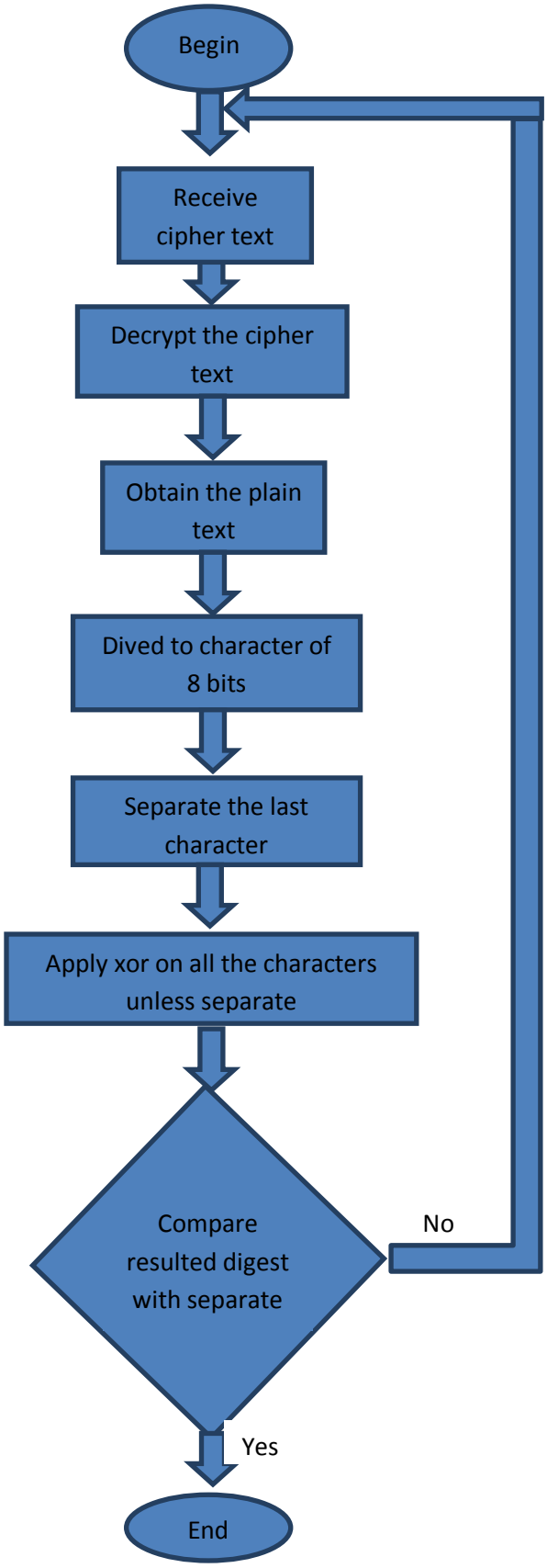
Algorithm (3.2) the reciever user
Input: ciphertext Output : plaintext

Begin
Read (cipher)
Divide cipher to character
Read (key)
Convert key to ASCII
Work xor between ASCII (cipher) and ASCII (key)
Find xor between the result (each character without end character)
Conformation the result of xor equal to end character
If equal
Ok ---- true
Delete the end character
Else
Not equal
Error ----- false
End

Figure (3.1) will explain the sender side and figure (3.2) will explain the reciever side.



Figure(3.1)shows the algorithm of sender

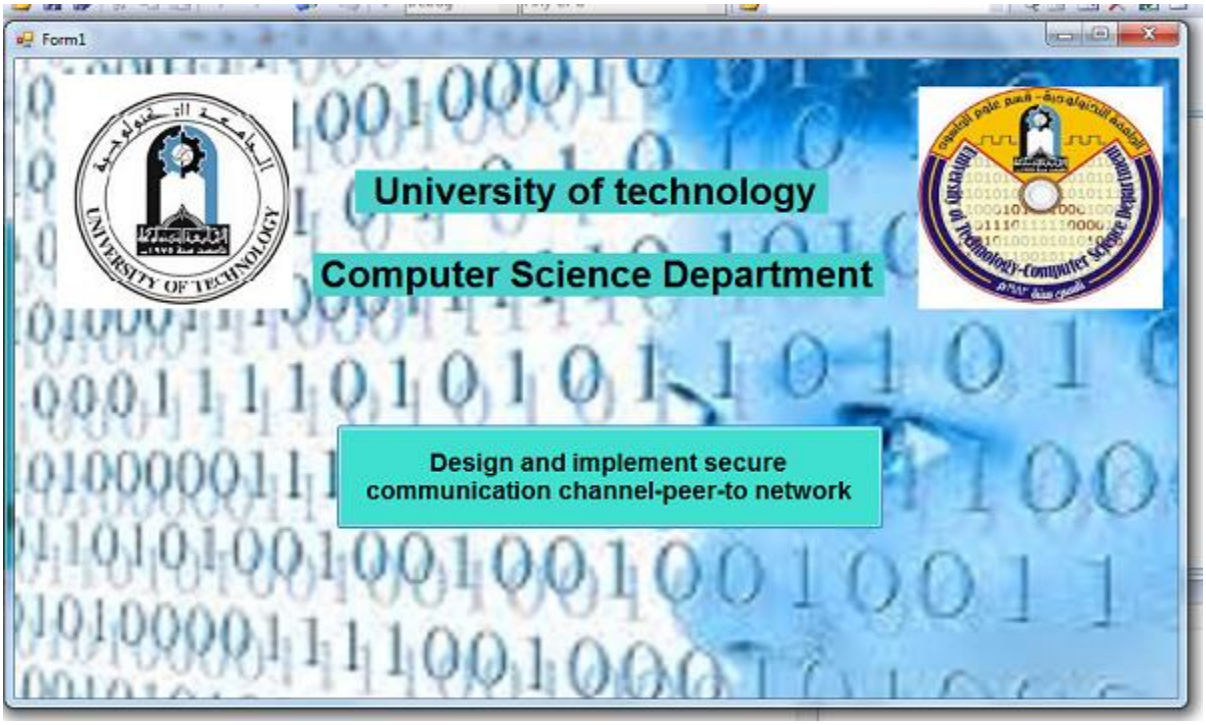


Figure(3.2)shows the algorithm of receiver

3.3 Implmentation of the proposal

In this project we use the c# programming language to implement the coding .

The first page is introduction page ,as showen in figure(3.3)

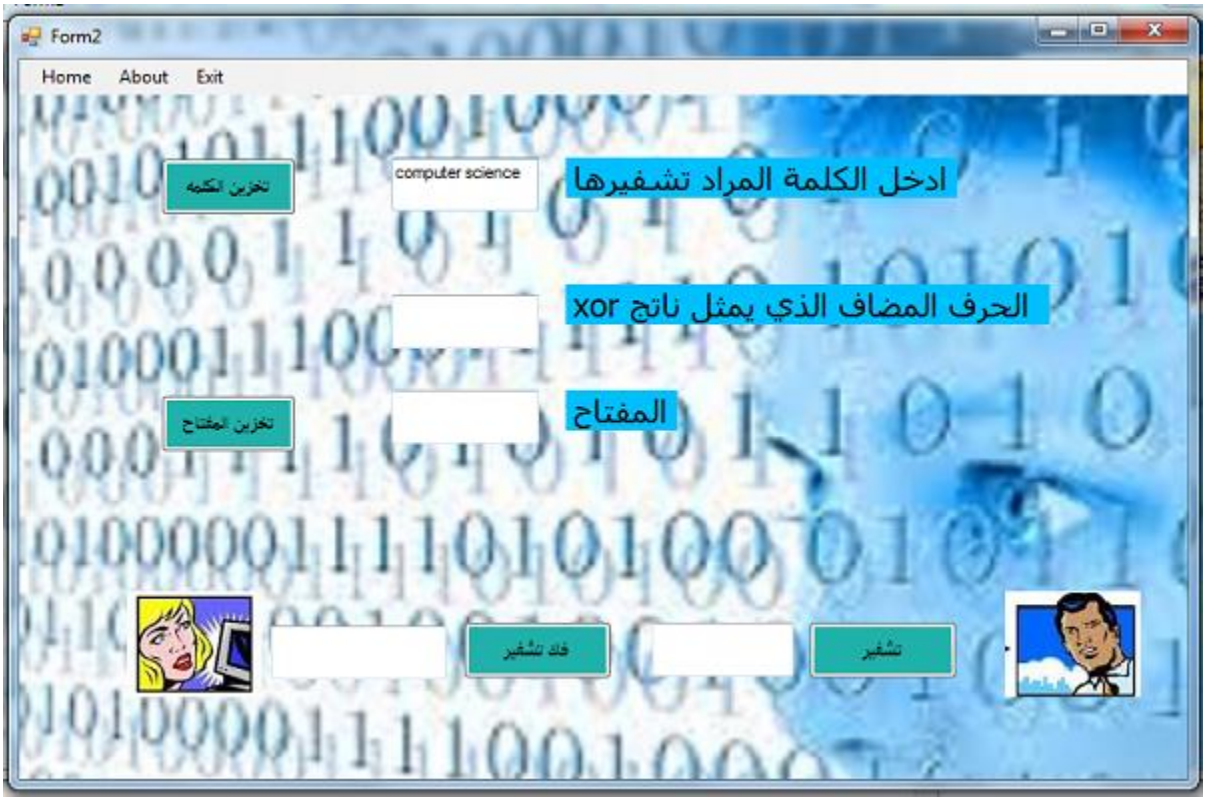


Figure(3.3) shows the introduction of the project.

When we click on the command of project name will appear the implemention page as showen in figure(3.4)

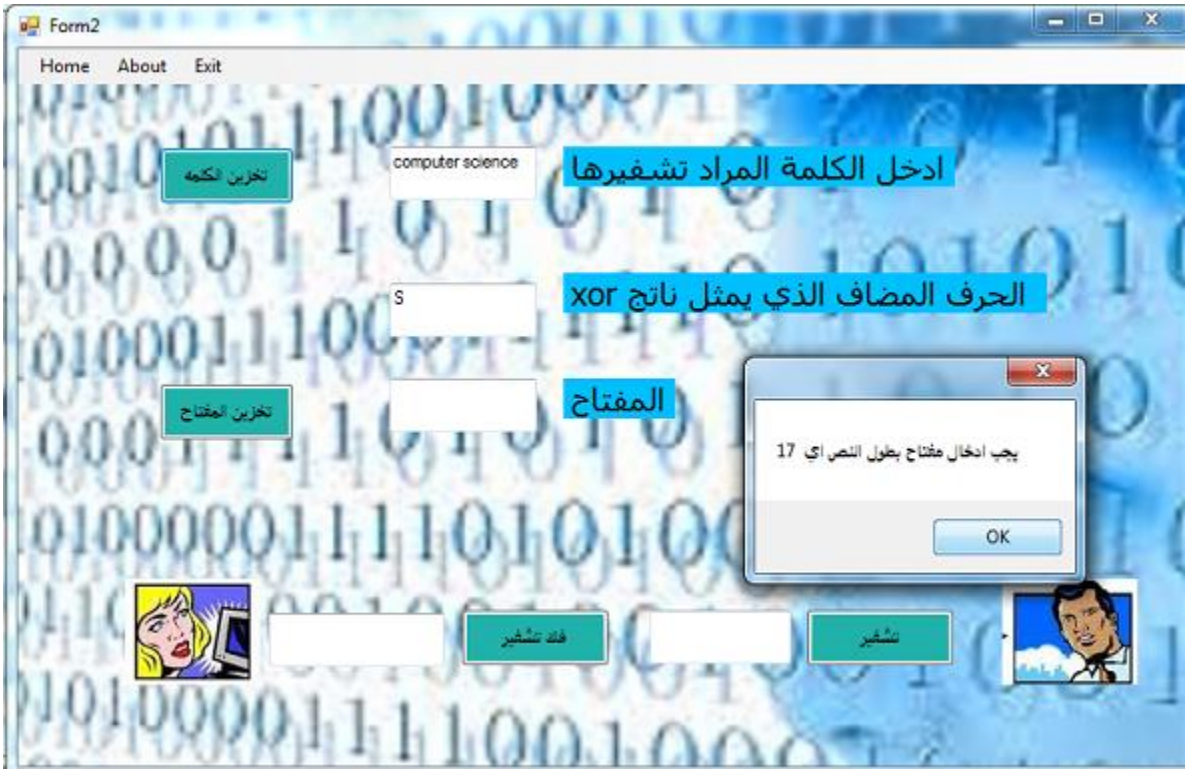
Figure(3.4) showes implemention page

When we enter the text and click on the command of (تخزين الكلمة) as showen in figure(3.5)



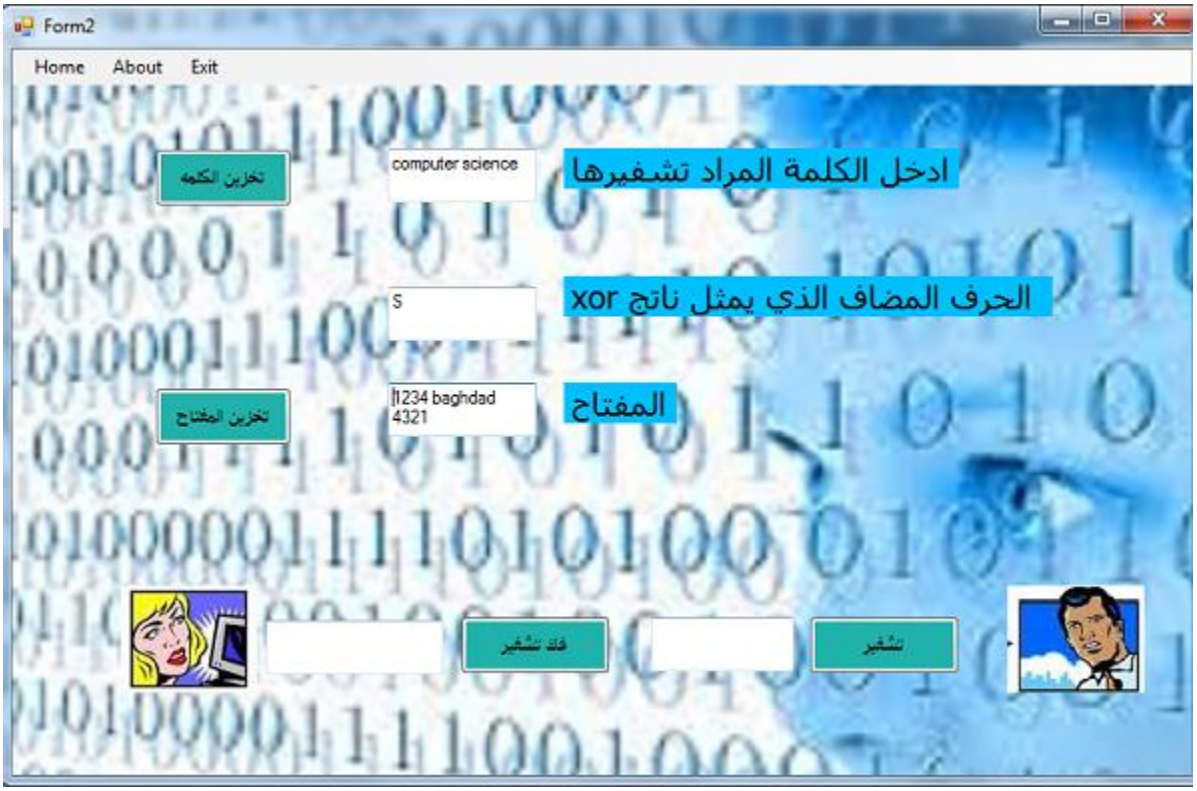
Figure(3.5) shows the text input

And then will appear the figure (3.6) below



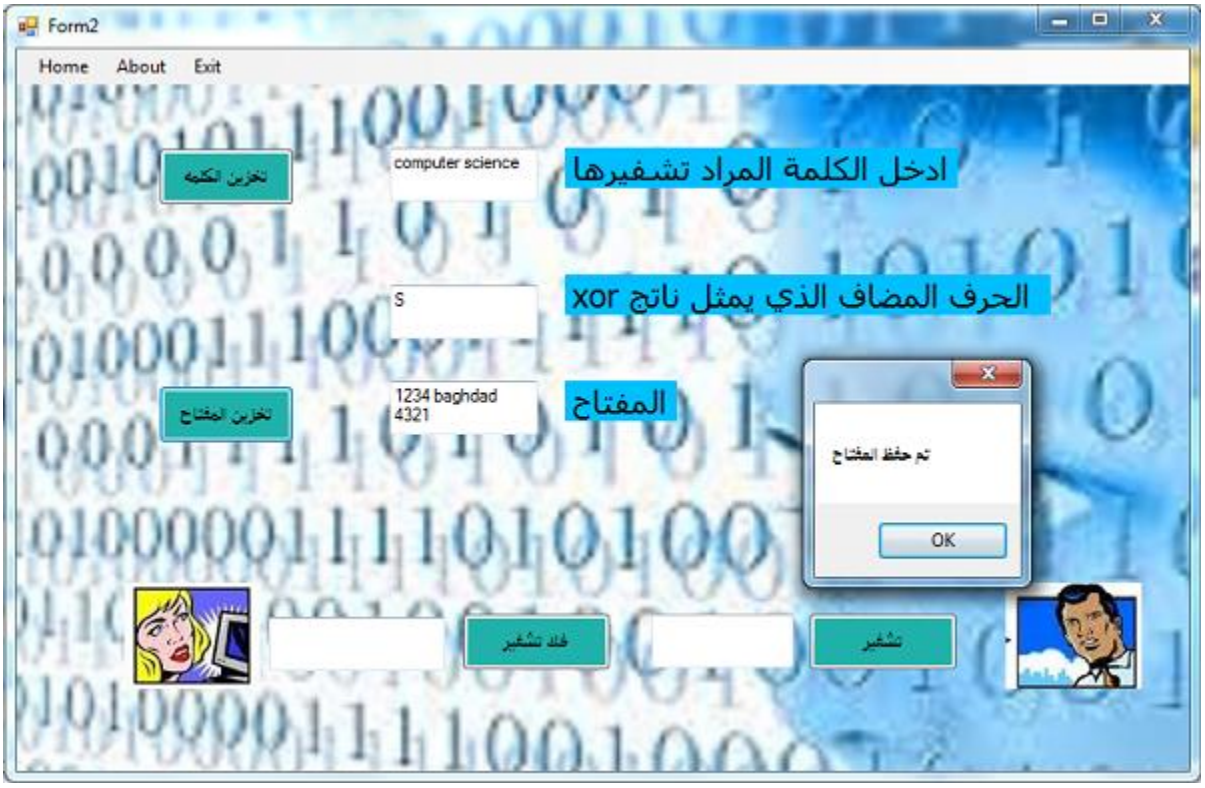
Figure(3.6) shows the text saved and the letter added, which represents the result of the xor

In this figure Enter the key and must be equal to the length of text input as showen in figure(3.7)



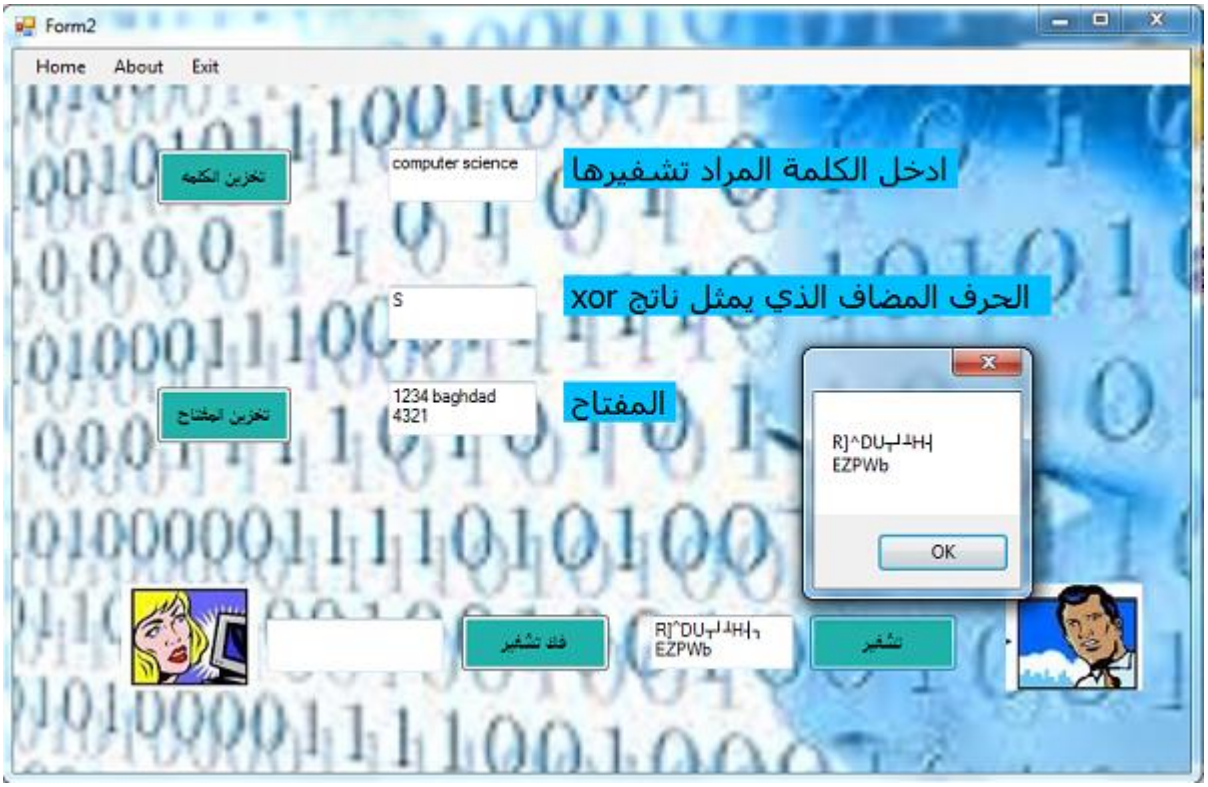
figure(3.7) Enter the key

When we click on the commend of (تخزين المفتاح) will appear the figure (3.8) as showen below



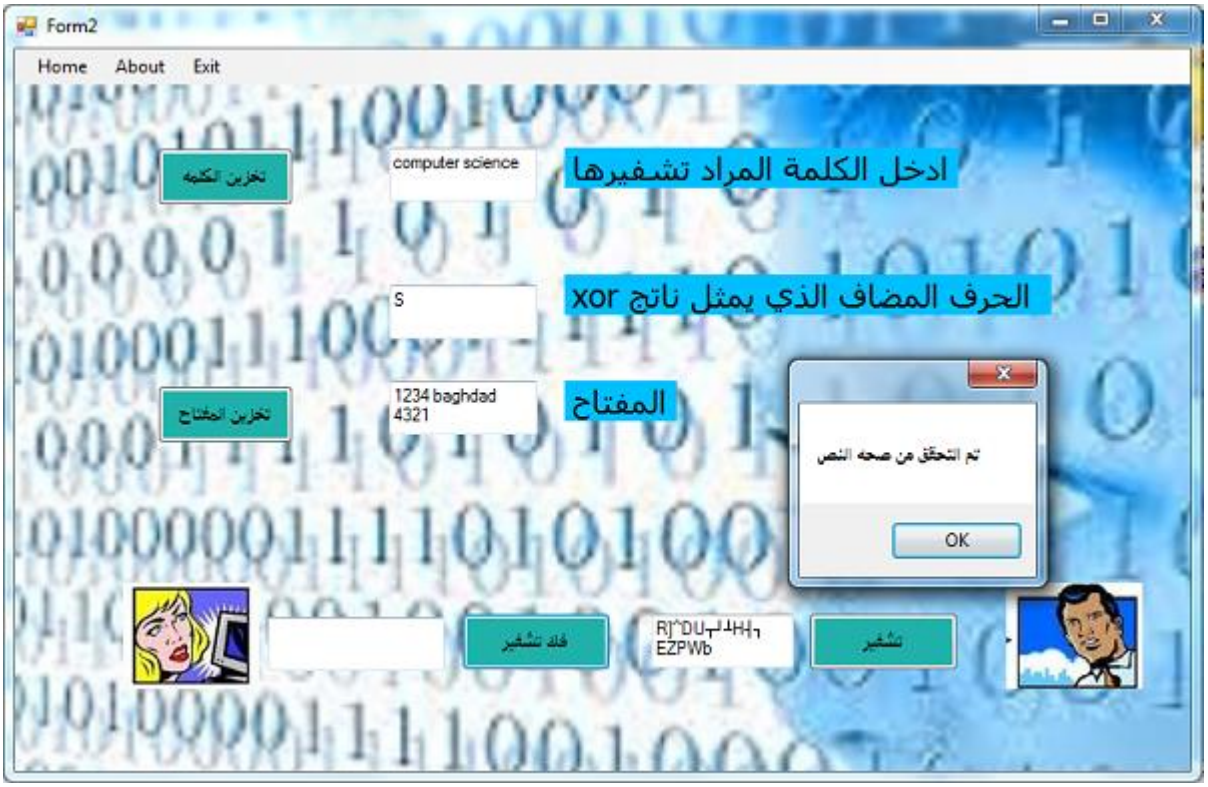
Figure(3.8) store the key

When we click on the command of (تشفير) will appear the implemention as showen in figure(3.9)



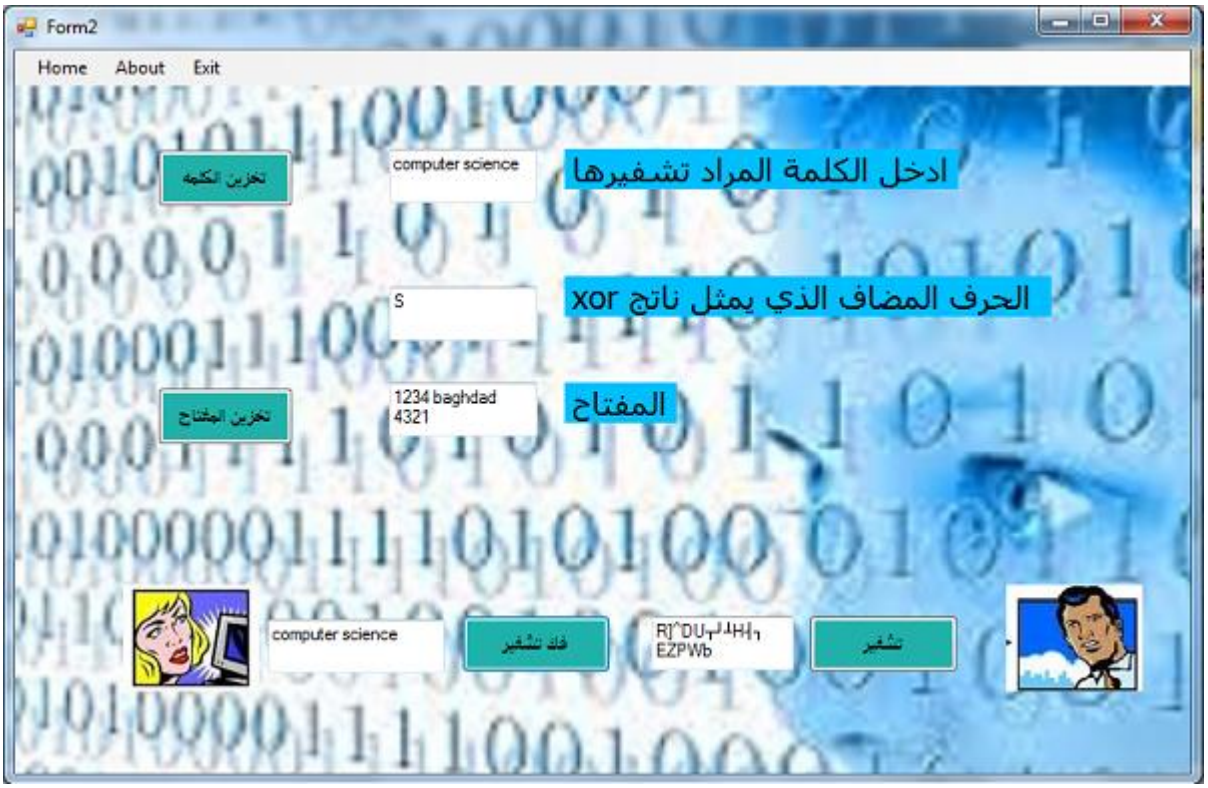
Figure(3.9)shows the encryption message

When we click on the command of (فك تشفير) will appear the implemention as showen in figure(3.10)



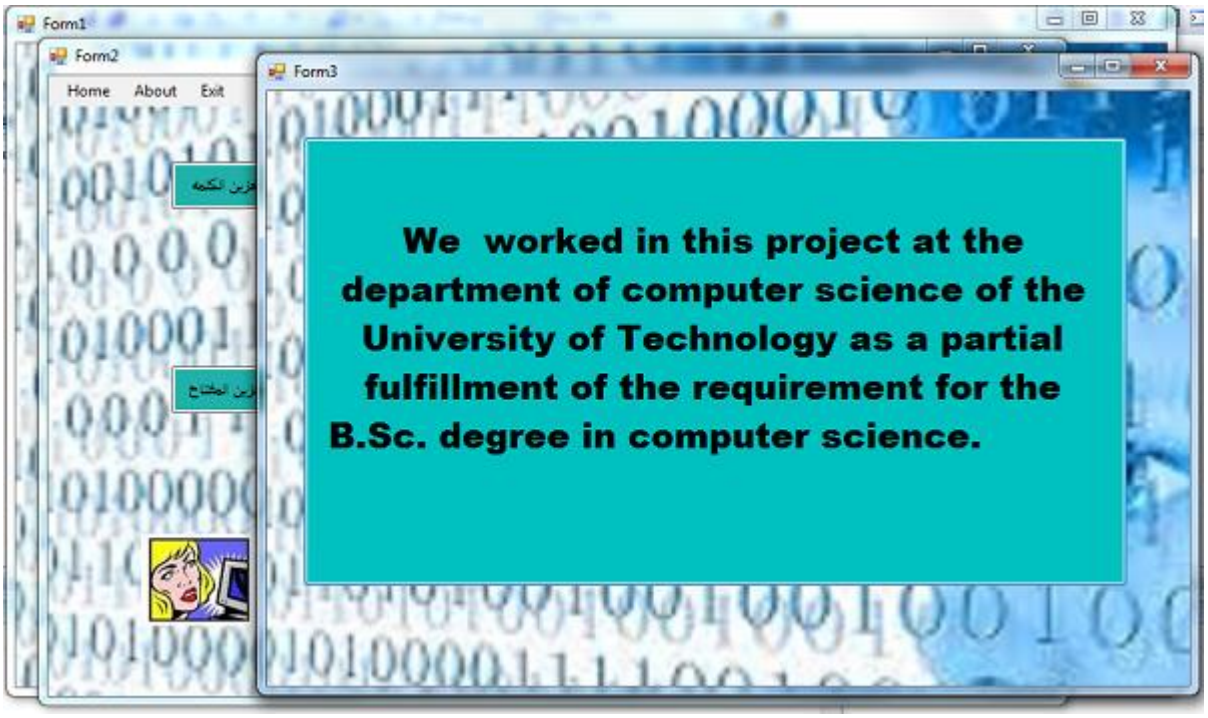
figure(3.10) show massge box (تم التحقق من صحة النص)

figure(3.11) show decryption text



figure(3.11) show the result of decryption

When we click on the (about) will appear the About page as showen in figure(3.12)



figure(3.12) About page

Chapter Four

Conclusions and Recommendations

4.1 Conclusions

Stream ciphers are often used for their speed and simplicity of implementation in hardware, and in applications where plaintext comes in quantities of unknowable length like a secure wireless connection. If a block cipher (not operating in a stream cipher mode) were to be used in this type of application, the designer would need to choose either transmission efficiency or implementation complexity, since block ciphers cannot directly work on blocks shorter than their block size. For example, if a 128-bit block cipher received separate 32-bit bursts of plaintext, three quarters of the data transmitted would be padding. Block ciphers must be used in ciphertext stealing or residual block termination mode to avoid padding, while stream ciphers eliminate this issue by naturally operating on the smallest unit that can be transmitted (usually bytes).

Another advantage of stream ciphers in military cryptography is that the cipher stream can be generated in a separate box that is subject to strict security measures and fed to other devices such as a radio set, which will perform the xor operation as part of their function. The latter device can then be designed and used in less stringent environments.

4.2 Recommendations

We develop a secure key issuing protocol, which adopts KGC and KPAs to issue private keys to peers securely.

Our future work includes developing a scheme to authenticate KPAs, remove malicious ones and finding out alternate ones to join in the system using the BFT protocol.

References

1. Stallings W., and Brown L., "**Computer Security Principles and Practice**", Book, Pearson, 2012.

2. Stallings W., "**Cryptography and Network Security Principles and Practice**", Book, Prentice Hall, Pearson, 2011.

3. Gollmann D., "**Computer Security**", WILEY A John Wiley and Sons, Ltd. Publication, 2011.

4. Eric Cole E., Krutz R. L., Conley J.W., Reisman B., Ruebush M., and Dieter Gollman, "**Network Security Fundamentals**", John Wiley & Sons, Inc., 2008.