Islamic University-Gaza
Computer Engineering Department
Faculty of Engineering
Deanery of Higher Studies

An Efficient Approach For Data Encryption Using Two Keys

Belal Y Abusalim

120100650

Supervisor

Prof. Mohammad A. Mikki

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering

1436H(2015)

ACKNOWLEDGMENT

نموذج رقم (1)

إقـــــرار

أنا الموقع أدناه مقدم الرسالة التي تحمل العنوان: تصميم طريقة فعالة جديدة لتشفير البيانات باستخدام مفتاحين

An Efficient Approach For Data Encryption Using Two Keys

أقر بأن ما اشتملت عليه هذه الرسالة إنما هو نتاج جهدي الخاص، باستثناء ما تمت الإشارة إليه حيثما ورد، وإن هذه الرسالة ككل أو أي جزء منها لم يقدم من قبل لنيل درجة أو لقب علمي أو بحثي لدى أي مؤسسة تعليمية أو بحثية أخرى.

DECLARATION

The work provided in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted elsewhere for any other degree or qualification

اسم الطالب: بلال يحيى أبو سليم Student's name: Belal Abu Sulim

التوقيع: Signature:

التاريخ: 2015-10-13 Date: 13-10-2015

بسم الله الرحمن الرحيم

الجامعة الإسلامية – غزة
The Islamic University - Gaza

مكتب نائب الرئيس للبحث العلمي والدراسات العليا هاتف داخلي 1150

الرقم Ref ج س غ/35/

التاريخ Date 2014/07/07

نتيجة الحكم على أطروحة ماجستير

بناءً على موافقة شئون البحث العلمي والدراسات العليا بالجامعة الإسلامية بغزة على تشكيل لجنة الحكم على أطروحة الباحث/ **بلال يحيى عبدالكريم أبو سليم** لنيل درجة الماجستير في كلية *الهندسة* قسم **هندسة الحاسوب** وموضوعها:

تصميم طريقة فعالة جديدة لتشفير البيانات باستخدام مفتاحين

An Efficient Approach For Data Encryption Using Two Keys

وبعد المناقشة التي تمت اليوم الثلاثاء 20 رمضان 1436هـ، الموافق 2014/07/07م الساعة الحادية عشرة صباحاً، اجتمعت لجنة الحكم على الأطروحة والمكونة من:

أ.د. محمد أمين مكي	مشرفاً و رئيساً	.....
د. أيمن أحمد أبو سمرة	مناقشاً داخلياً	.....
أ.د. سامي سليم أبو ناصر	مناقشاً خارجياً	.....

وبعد المداولة أوصت اللجنة بمنح الباحث درجة الماجستير في كلية *الهندسة* / قسم وموضوعها: **هندسة الحاسوب**.

واللجنة إذ تمنحه هذه الدرجة فإنها توصيه بتقوى الله ولزوم طاعته وأن يسخر علمه في خدمة دينه ووطنه.

والله ولي التوفيق،،،

مساعد نائب الرئيس للبحث العلمي وللدراسات العليا

أ.د. فؤاد علي العاجز

P.O. Box 108, Rimal, Gaza, Palestine fax: +970 (8) 286 0800 فاكس Tel: +970 (8) 286 0700 ص.ب. 108 الرمال، غزة، فلسطين هاتف
public@iugaza.edu.ps www.iugaza.edu.ps

ACKNOWLEDGMENT

All praise and glory are due to Allah the almighty who provided me with the much needed strength and stamina to successfully accomplish this work. The completion of this thesis also owes a great deal to the help and unrelenting support of the people around me. I am especially indebted to my advisor, Prof. Mohammed Mikki, for his guidance, support and patience with me throughout the course of my research. He taught me the essence and principles of research and guided me through until the completion of this thesis. I could no have asked for a better advisor.

I owe my largest debt to my family, and I wish to express my heartfelt gratitude to all of them for their encouragement, constant prayers, and continued support. My dear brothers and sisters who have given me all their love and support over the years; I thank them for their unwavering commitment through good times and hard times.

I would like to thank my best friends Kanaan Albhasy and Nader Abu Attaya for their kind helps to me.

My profound gratitude goes to my wife for her support, endurance and patience. I dedicate this thesis to her.

Finally, I thank all of those whose names are not mentioned here but have helped me in any way to accomplish this work.

List Of Contents

LIST OF ACRONYMS

AES	Advanced Encryption Standard
NIST	National Institute of Standards and Technology
DES	Data Encryption Standard
BC	Before Christ
US	United States
FBI	Bureau of Investigation
IBM	International Business Machines
NSA	National Security Agency
IDEA	International Data Encryption Algorithm
PGP	Pretty Good Privacy
RSA	Ron Rivest, Adi Shamir, and Leonard Adleman
RC5	Symmetric-key block Cipher
LUT	Look Up Table
FPGA	Field-Programmable Gate Array
FPD	Field Programmable Devices
VLSI	Very Large Scale Integration
CPU	Central Processing Unit
PKI	Public Key Infrastructure
PC	Personal Computer
AEAD	Authenticated Encryption with Associated Data
AD	Associated Data
MIC	Message Integrity Code
MAC	Message Authentication Code
KDF	Key Derivation Function

LIST OF FIGURES

الملخص

شهد أمن الانترنت في الثلاثة عقود الماضية نمو كبير ومثير. تحت هذا السقف من الإثارة يكمن الكثير من التحديات الفنية والتجارية ولكن دون حل هذه التحديات, الانظمة الامنة لن تصل إلى الكفاءة المتوقع لها. يمكن ان يتحقق امن الانظمة بواسطة التشفير حيت ان التشفير يستخدم مفتاح لتشفير المعلومات وفكها. بدون وجود مفتاح التشفير المعلومات المشفرة لا يمكن تحويلها إلى نصها الأصلي.

التطبيق الانتقائي للضمانات التكنولوجيا والإجرائية المتصلة بها هي مسؤولية هامة من كل مؤسسة من حيث توفير الأمن الكافي لأنظمة بياناتها الإلكترونية. حماية البيانات أثناء نقلها أو أثناء التخزين قد يكون ضروريا للحفاظ على سرية وسلامة المعلومات الممثلة بالبيانات. تم إدخال معيار التشفير المتقدم (AES) الذي يستخدم 128 بت حجم الكتلة وكذلك 128 بت حجم مفتاح.

التشفير هو طريقة التي عن طريقها يتم تحويل– البيانات– الأرقام وغير ذلك إلى شكل مشفر ولا يمكن فك تشفيرها وقراءتها إلا اذا كان المستخدم يمتلك مفتاح التشفير المناسب. وهي تمثل واحدة من العديد من التقنيات التي يمكن تطبيقها لحماية البيانات البحثية من الوصول الغير مصرح به.

معيار التشفير المتقدم (AES) خيار شائع جدا في الاتصالات الأرضية يقل ظهوره كخيار مفضل في صناعة الطيران بما في ذلك الأقمار الصناعية. AES يعمل على تشفير البيانات ككتلة واحدة حيث انه يشفر كتلة بطول معين مرة واحدة.

يستخدم التشفير المتناظر على نطاق واسع لتزويد أمن البيانات وعمل العديد من اولويات التشفير.

هذه الاطروحة تقترح نهج جديد فعال لتشفير المعلومات بشكل متناظر ولكن باستخدام مفتاحين، مفتاح مقدم من النظام ومفتاح اخر مقدم من العميل لدى هذا النظام ، سأقوم بدمج كلا المفتاحين للحصول على مفتاح رئيسي اقوم باستخدامه في تشفير المعلومات، سأستخدم معيار التشفير المتقدم في تصميم هذا النهج الجديد ، معيار التشفير المتقدم هو خوارزميه من خوارزميات التشفير المتناظر التي تستخدم مفتاح واحد في التشفير ولذلك نظام التشفير المقترح نظام تشفير متناظر .

باستخدام النهج الجديد في الأطروحة ،الكود الناتج سيكون له العديد من الخصائص المفيدة اولا سيكون سريع من حيث التنفيذ ثانيا سهل التصميم في الأجهزة وثالثا سيعطينا مدى امن عالي جدا .تماما مثل معيار تشفير البيانات(DES) عندما يتغير بت واحد في المعلومات فان العديد من البت في كود التشفير ستتغير .يمكن استخدام هذا النهج الجديد مع امكانية ان يكون طول كتلة البيانات مختلف ،النهج الجديد يمكن استخدامه من قبل المؤسسات والحكومات والبنوك بالإضافة الى المستخدمين العاديين

ABSTRACT

Internet security has witnessed an explosive and exciting growth in the past three decades . Under the surface of excitement lies a mine of technical and commercial challenges. Without solving these challenges, secure systems will not reach the expected potential. Security can be achieved via encryption. Encryption uses b“keys” to encrypt and decrypt the information. Without having the cryptographic key, the ciphered information will never be converted into its original text.

The selective application of technological and related procedural safeguards is an important responsibility of every organization in providing adequate security to its electronic data systems. Protection of data during transmission or while in storage may be necessary to maintain the confidentiality and integrity of the information represented by the data. The Advanced Encryption Standard (AES) that uses 128 bit block size as well as 128 bits key size was introduced by NIST.

Encryption is a method by which data – digital or otherwise – is converted into an encoded form that can only be decoded and read if the user possesses an appropriate encryption key. It represents one of several techniques that may be applied to protect research data from unauthorized access.

The Advanced Encryption Standard (AES), which is a very popular choice in terrestrial communications, is slowly emerging as the preferred option in the aerospace industry including satellites. AES is a block cipher, which encrypts one block of fixed length data at a time.

Symmetric encryption is widely used to enforce data security and provide other cryptographic primitives.

This thesis proposed a new efficient approach for data encryption algorithm using two keys. One supplied and generated by the system that's specific to an individual user and one from the user (user key).I hashed the user key with the system key to get a master key which will be used at the end to encrypt data. I used AES as the basis of our approach. AES is symmetric key that uses only one key. So my proposed cryptographic system is symmetric.

By two keys method, the new algorithm had several useful properties. First, it is fast, Second, can be easily implemented in hardware. Third, it is highly secure. Similar to DES, when one bit in the data is charged the encryption code will have many bits

charged. The length of data block can vary. The encryption mode can be changed easily. The proposed approach may be used by large organizations, governments, banks in addition to individuals.

Chapter 1

Introduction

1.1 Background

As the importance and the value of exchanged data over the internet or other media types are increasing, the search for the best solution to offer the necessary protection against the data thieves' attacks along with providing the services under timely manner is one of the most active subjects in the security related communities. The increase demand on exchanging data or other media types over the internet stimulate engineers to search for the best solution to offer the necessary protection against the data thieves' attacks along with providing the services under timely manner are some of the most active subjects in the security related communities [1].

In this digital world, digital data (data flows) play a core role in the communications between computers and networks. Among these digital formats, the binary format is the foundation of others. And thus digital data can always be considered as some type of binary data directly or indirectly. Based on the use of the digital data, we can roughly classify them to three types: public data such as online newspapers and government service information, which are open to everyone; private data such as the personal online albums and the personal blogs, which are shared within a small group of people; and secret data such as military databases, medical records, and classified online documents, which are accessible only by authorized users. Therefore, the demand of data security is very high and increasing rapidly. Encryption is one of ways for enhancing data security, where the term 'encryption' refers to the process of converting ordinary information into unintelligible [2].

Encryption is a method by which data – digital or otherwise – is converted into an encoded form that can only be decoded and read if the user possesses an appropriate encryption key. It represents one of several techniques that may be applied to protect research data from unauthorized access.

. Similar to DES, when one bit in the data is changed the encryption code will have many bits changed. The length of data block can vary. The encryption mode can be changed easily. The proposed approach may be used by large organizations,

governments, banks in addition to individuals. I used AES (advanced Encryption Standard) as the basis of this new approach.

1.2 Brief History of Cryptography

Cryptography, the science of encrypting and decrypting information, dates as far back as 1900 BC when a scribe in Egypt first used a derivation of the standard hieroglyphics of the day to communicate.[3] There are many notable personalities who participated in the evolution of Cryptography. For example, "Julius Caesar (100-44 BC) used a simple substitution with the normal alphabet (just shifting the letters by 3 positions(in government communications", [3] and later, Sir Francis Bacon in 1623, who described a cipher is known today as a 5-bit binary encoding. He advanced it as a steganography device by using variation in type face to carry each bit of the encoding". For all the historical personalities involved in the evolution of cryptography, it is William Frederick Friedman، founder of Riverbank Laboratories, cryptanalyst for the US government, and lead code-breaker of Japan's World War II Purple Machine, who is "honored as the father of US cryptanalysis". In 1918 Friedman authored The Index of Coincidence and their Applications in cryptography, which is still considered by many in this field as the premiere work on cryptography written this century, During the late 1920s and into the early 1930s, the US Federal Bureau of Investigation (FBI) established an office designed to deal with the increasing use of cryptography by criminals. At that time the criminal threat involved the importation of liquor. According to a report written in the mid-1930s by Mrs. Elizabeth Friedman, a cryptanalyst employed by the US government like her husband, William F. Friedman, the cryptography employed by bootleggers. Although cryptography was employed during World War I, two of the more notable machines were employed during World War II: the Germans' Enigma machine, developed by Arthur Scherbius, and the Japanese Purple Machine, developed using techniques first discovered by Herbert O. Yardley.

In the 1970s, Dr. Horst Feistel established the precursor to today's Data Encryption Standard (DES) with his 'family' of ciphers, the 'Feistel ciphers', while working at IBM's Watson Research Laboratory. In 1976 The National Security Agency (NSA) worked with the Feistel ciphers to establish FIPS PUB-46, known today as DES.

Today, triple-DES is the security standard used by U.S. financial institutions. Also in 1976, two contemporaries of Feistel, Whitfield Diffie and Martin Hellman first introduced the idea of public key cryptography in a publication entitled "New Directions in Cryptography". Public key cryptography is what PGP, today's industry standard, uses in its software. In the September, 1977 issue of The Scientific American, Ronald L. Rivest, Adi Shamir and Leonard M. Adleman introduced to the world their RSA cipher, applicable to public key cryptography and digital signatures. The authors offered to send their full report to anyone who sent them self-addressed stamped envelopes, and the ensuing international response was so overwhelming the NSA balked at the idea of such widespread distribution of cryptography source code.

In the mid-1980s ROT13 was employed by USENET groups to prevent the viewing of "objectionable material [by] innocent eyes", and soon thereafter, a 1990 discovery by Xuejia Lai and James Massey proposed a new, stronger, 128-bit key cipher designed to replace the aging DES standard named International Data Encryption Algorithm (IDEA). This algorithm was designed to work more efficiently with "general purpose" computers used by everyday households and businesses. Concerned by the proliferation of cryptography, the FBI renewed its effort to gain access to plaintext messages of US citizens. In response, Phil Zimmerman released his first version of Pretty Good Privacy (PGP) in 1991 as a freeware product, which uses the IDEA algorithm. PGP, a free program providing military-grade algorithm to the internet community, has evolved into a cryptographic standard because of such widespread use. The initial versions of PGP were geared towards the more computer literate individual, but to the individual nonetheless. Phil Zimmerman could be compared to Henry Ford in his efforts to provide PGP to every home by making it free, and therefore, affordable. Today, PGP's updated version is offered free to the public. In 1994, Professor Ron Rivest, co-developer of RSA cryptography, published a new algorithm, RC5, on the Internet. It had been claimed that RC5 is stronger than DES. [3]

The interview about cryptography is usually referred to as "the study of secret", while nowadays is most attached to the definition of encryption. Encryption is the process of converting plain text "unhidded" to a cryptic text "hidded" to secure it

against data thieves. This process has another part where cryptic text needs to be decrypted on the other end to be understood.

The most commonly used techniques for producing confidentiality in data transmission is symmetric encryption [4], [5]. Symmetric encryption scheme, also is referred to as single key encryption method, has five main modules [5]. The term plaintext is used to denote the original incoming data. The key is an essential part of the encryption process and it provides the secure data traffic among the sender and the recipient. Encryption algorithm performs various mathematical and logical functions on the plaintext by using the key. Cipher data is the encrypted message produced by encryption algorithm by using the key and the plain text.

Decryption algorithm is employed on cipher data and performs reverse action to generate the plain text. The symmetric key encryption algorithm shares the same key between sender and receiver and a strong algorithm for encryption and decryption processes is essential to provide an efficient security mechanism [5]. The block cipher algorithms are the commonly used symmetric encryption techniques which analyze the input data as fixed size blocks and produce the scrambled data as equal size as the original data [5]. One of the most widely used cipher block is DES [4].

Due to heavily increased volume of sensitive information, recently the implementations of cryptographic algorithms have been increased as countermeasures for security threats. The diversity of the security applications has also been raised dramatically. This trend has introduced additional challenges not only in terms of highly secure algorithms but also in terms of fast solution approaches for high performance applications. Cryptographic designers have explored not only realizations on software platforms but at the same time on hardware platforms [6], [4], [7], [8], [9].

There are six main goals behind using cryptography. Every security system must provide a bundle of security functions that can assure the secrecy of the system. These functions are usually referred to as the security system. They are [10]:

- CONFIDENTIALLY- Information in computer transmitted information is accessible only for reading by authorized parties.

- AUTHENTICATION- Origin of message is correctly identified with an assurance that identity is not false.
- INTERGRITY- Only authorized parties are able to modify transmitted or stored information.
- NON REPUDIATION- Requires that neither the sender, nor the receiver of message be able to deny the transmission.
- ACCESS CONTROL- Requires access may be controlled by or for the target system.
- AVAILIBILITY- Computer system assets are available to authorized parties when needed.[10]

1.3 Types of Cryptography

Many encryption algorithms are widely available and used in information security. They can be categorized into symmetric (private) and asymmetric (public) keys encryption. In symmetric keys encryption or secret key encryption, only one key is used to encrypt and decrypt data. The key should be distributed before transmission between entities. Keys play an important role. If weak key is used in algorithm then everyone may decrypt the data. Strength of symmetric key encryption depends on the size of key used. For the same algorithm, encryption using longer key is harder to break than the one done using smaller key. There are many examples of strong and weak keys of cryptography algorithms like RC2, DES, 3DES, RC6, Blowfish, and AES. RC2 uses one 64-bit key .DES uses one 64- bits key. Triple DES (3DES) uses three 64-bits keys while AES uses various (128,192,256) bits keys. Blowfish uses various (32-448); default 128bits while RC6 is used various (128,192,256) bits keys [11-12]. But main problem with this is secure transmission of key over the malicious network. Asymmetric key encryption or public key encryption is used to solve the problem of key distribution. In asymmetric keys, two keys are used; private and public keys. Public key is used for encryption and private key is used for decryption (E.g. RSA and Digital Signatures). Users use two keys: public key, which is known to the public and private key which is known only to the user. There is no need for distributing them prior to transmission. However, public

key encryption is based on mathematical functions, computationally intensive and is not very efficient for small mobile devices [11].

Chapter 2

RELATED WORKS

Between the two types of cryptography techniques, the symmetric key encryption is fast and most commonly used compared to the asymmetric key encryption. In case of symmetric key encryption only one key is used on both sides of encryption and decryption. Few commonly used symmetric key algorithms are DES, RC2, RC4 etc. [13] According to the new encryption algorithms mentioned in [13], the ASCII value of the letters are considered and it is divided by the secret key and the cipher text is obtained from the quotient and remainder of the division. The 4-bit secret key needs to be greater than 1000.Even the public key cryptography [14] doesn't work out well for small amount of data. This paper proposed a fully-parallel, bit-sliced unified architecture designed to perform modular multiplication/exponentiation and GF(2M) multiplication as the core operations of RSA and EC cryptography. The architecture uses a radix-2 Montgomery technique for modular arithmetic, and a radix-4 MSD-first approach for GF(2M) multiplication. To the best of our knowledge, it is the first unified proposal based on such a hybrid approach. The architecture structure is bit-sliced and is highly regular, modular, and scalable, as virtually any data path length can be obtained at a linear cost in terms of hardware resources and no costs in terms of critical path. Our proposal outperforms all similar unified architectures found in the technical literature in terms of clock count and critical path. The architecture has been implemented on a field-programmable gate array (FPGA) device. A highly compact and efficient design was obtained taking advantage of the architectural characteristics[14]. The authentication system has been well explained in [15]. In this model, the password is not stored in a single authenticating server but rather shared among a set of n servers so that an adversary can learn the password only by breaking into t+ 1 of them. The protocols require n > 3t servers to work. The goal is to protect the password against hackers attacks that can break into the authenticating server and steal password information. All known centralized password authentication schemes are susceptible to such an attack[15].

Symmetric encryption is a form of cryptosystem in which both encryption and decryption are performed by using the same key [16]. Symmetric-key encryption can use either stream ciphers or block ciphers. Some of the basic techniques that can be

considered as building blocks to construct symmetric encryption algorithms include Feistel Network, Key Schedule،Product cipher, S-Box, P-Box [16]. Various popular symmetric encryption algorithms like Advanced Encryption Standard (AES)/RijnDael and Data Encryption Standard (DES)were developed through a combination of above mentioned techniques. On the other hand, Double DES [16] and Triple DES with two keys [17] and Triple DES with three keys [18] are some well-known examples where an existing symmetric algorithm, i.e. DES was used to create new symmetric encryption algorithms. I thus find that symmetric encryption algorithms can be developed either by a combination of raw encryption techniques or by using existing symmetric encryption algorithms as building blocks for creating new schemes.

Recent AES implementations have focused on speed gains obtained by manipulations in the Sub Bytes and Mix Columns, two of the more time-consuming function in the algorithm. One such software technique, called the T-Box algorithm, merges Sub Bytes and Mix Columns in encryption and Inverse Sub Bytes and Inverse Mix Columns in decryption [19]. Another widespread technique used was a BDD architecture and two-level logic to simplify the S-Box. The use of such techniques increases throughput to above 10-Gbps. However, such implementations are fabricated using 0.13 μm technology with clock rates approaching 900 MHz. Even in 0.18 μm CMOS, only 1.6 Gbps is achieved [20]. It is clear that out of all the functions, manipulating SubBytes is the key to increasing performance. Although, custom implementations of modifications to the SubBytes and Mix Columns functions will often result in increased sensitivity to noise and operating temperature. As well as extremely large fan-outs, adding higher propagation delays.

There has been little to no research done regarding lowering power requirements and cost by deemphasizing processing speed. Relatively high throughput (2.381Gbps) for 128-bit key mode was achieved in one FPGA implementation with a cost of only 58.5K gates [21]. This was done by introducing a 4-stage pipeline for the main functions and performing a basis transformation on SubBytes. The alternative S-Box design would be to use a Look Up Table (LUT).

There are several hardware implementations for AES that can be found in literature. These implementations were done for the original standard AES (not more than 256

bits key size), such as the work presented in [8-22] [9-23] [10-24]. Before choosing Rijindael to be the AES in November of 2001, many related implementations were proposed to how much the structure of the proposed candidates for the AES competition is suitable for hardware implementation [25][26] [27] [28] [29]. They provide a multiple architecture options for AES finalist candidates with an implementation analysis for each architecture based on both area and speed optimization. Also, the authors in [26] and [22] provided a detailed comparison of the FPGA hardware performance of the AES candidates. A Field Programmable Devices (FPD) implementation for Rijndael was presented in [4-30] with a comparison with Xilinx FPGA implementation. The authors claim a performance gain (speed) using FPD over other FPGA implementation. An optimized S-Box design was presented in [31] for an efficient compact hardware design. Another compact AES design was presented in [26]. The compact designs target the small embedded systems market with low memory constraints. On the other hand, a VLSI Implementation with 1.82-Gbits/sec proposed and evaluated in [24]. In [32], application based speed/area tradeoffs design was presented for the AES. A high speed and low power sub- pipelined structure proposed in for compact AES design[33]. An FPGA parallel implementation of the AES proposed in [32]. On the other hand, in [33] and [34] a 128, 192, and 256-bit key size ASIC implementation of AES with high-throughput cost-effective design were proposed.

Chapter 3

SYMMETRIC ENCRYPTION

3.1 General Overview

In symmetric-key encryption, each computer has a secret key (code) that it can use to encrypt a packet of information before it is sent over the network to another computer. Symmetric-key requires that you know which computers will be talking to each other so you can install the key on each one. Symmetric-key encryption is essentially the same as a secret code that each of the two computers must know in order to decode the information. The code provides the key to decoding the message.

Think of it like this: You create a coded message to send to a friend in which each letter is substituted with the letter that is two down from it in the alphabet. So "A" becomes "C," and "B" becomes "D". You have already told a trusted friend that the code is "Shift by 2". Your friend gets the message and decodes it. Anyone else who sees the message will see only nonsense.

The same goes for computers, but, of course, the keys are usually much longer. The first major symmetric algorithm developed for computers in the United States was the Data Encryption Standard (DES), approved for use in the 1970s. The DES uses a 56-bit key.

Because computers have become increasingly faster since the '70s, security experts no longer consider DES secure -- although a 56-bit key offers more than 70 quadrillion possible combinations (70,000,000,000,000,000), an attack of brute force (simply trying every possible combination in order to find the right key) could easily decipher encrypted data in a short while. DES has since been replaced by the Advanced Encryption Standard (AES), which uses 128-, 192- or 256-bit keys. Most people believe that AES will be a sufficient encryption standard for a long time coming: A 128-bit key, for instance, can have more than 300,000,000,000,000,000,000,000,000,000,000,000 key combinations [35].

This chapter discusses some of the basic considerations of symmetric encryption from the angle of the ultimate security that is provided .

Example of Symmetric encryption include: Advanced Encryption Standard (AES).

3.2 How Does Symmetric Encryption Work?

Symmetric encryption involves the use of a key that I called the Symmetric Key As in Figure 3.1.

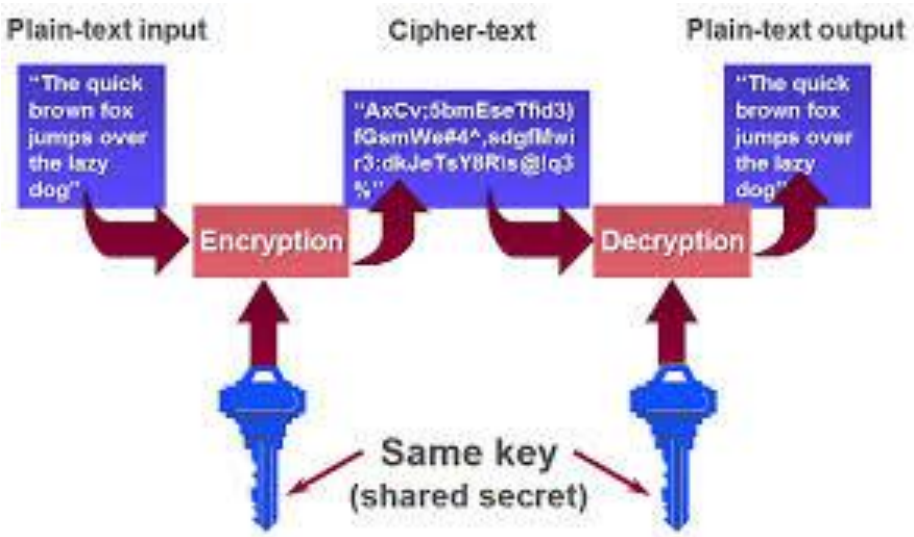

Figure 3.1 Symmetric Key Encryption

In today's computer-based systems this Symmetric Key is a series of numbers and letters. Example: f8kW2B60mVa2Kjue, this Symmetric Key will be used to encrypt a message. This very same Symmetric Key must be used to decrypt the message. There is only one key in Symmetric Encryption.

3.3 Example of Symmetric Encryption

Ali wants to send an encrypted message to Omar, say over the Internet. Therefore, Ali has to find a way to safely get the Symmetric Key to Omar .In as much as Ali encrypts his message before sending it over the Internet, it is clear that he does not consider the Internet to be a secure way to send messages. He fears, rightly so, that his message could somehow get into the hands of some person other than the intended recipient. Since Ali's premise is that the Internet is not secure, he may use traditional postal mail to send the Symmetric Key to Omar. he could also hand the Symmetric Key to Omar, saying, "Any time in the future I send you a secure message, I will use this Symmetric Key. With this Symmetric Key you can decrypt

the message I send to you." Interestingly, this one Symmetric Key can be used for either of these two parties to send encrypted messages to each other .

3.4 What Is Good About Symmetric Encryption?

Symmetric Encryption has almost always been the approach-of-choice for governments. Since the financial resources of a government to evolve cryptography (or any initiative for that matter) are considerable, thus far the "big money" that has "big secrets to protect" is going with Symmetric Encryption .

The popular Symmetric Encryption approaches have enjoyed speed advantages over Asymmetric approaches. In many applications, this speed of encrypting and decrypting is quite important.[36]

3.5What Is Undesirable About Symmetric Encryption?

The principle disadvantage of Symmetric Encryption involves a logistics problem of conveying the Symmetric Key. Recall that in Symmetric Encryption, there is only one key, the Symmetric Key. An example will illustrate it.

Example 1

Perhaps the most common occurrence of a situation in which encryption is needed and Symmetric Encryption does not currently serve well is that of a Web-based secure form .In this situation, the User (a person) operates a PC, using a Web browser such as Internet Explorer or Navigator, to connect to an Internet Web server that will take an order for some merchandise. A "secure form" is the computer-based form that the User fills in by typing into his/her browser. The User wishes to enter his/her credit card number into this secure form and have the computer/browser send this information, over the Internet, to the Web server taking the order. The User wants this transmission of the credit card number to be encrypted so that if the transmission is intercepted, it will appear as meaningless gibberish to the person who intercepted it .

If Symmetric Encryption were to be used here, then somehow, ahead of time, this User must have communicated with the company that runs the web server and conveyed to them the Symmetric Key needed for Symmetric Encryption. This is the only way the credit card information could be decrypted by the company that is taking the order. This can be quite inconvenient to both the User and the merchant and would tend to stifle internet commerce (e-commerce). I discussed the solution

for this problem in the methodology part ,I tried to give a solution be giving an authority for the user over the System.

3.6 Why Symmetric May Be Vastly Better?

The CPU time needed for the RSA encryption is more than that required for AES. AES was carefully tuned to allow very low power chips, embedded in smart cards, to encrypt and decrypt. AES was also tuned to be efficient with required memory. If AES at 256 bits key length takes fewer CPU cycles and less memory to encrypt/decrypt than RSA at 256 bits, consider comparing AES at 256 bits to RSA at 15360 bit.

Various online and print sources indicate that one of the most popular Asymmetric encryption systems has key recovery built in. Who has the access to this key recovery? Are you concerned about that entity having instant access to your messages? Also, assuming that key recovery is indeed buried in that product, are you concerned that a math graduate student hacks on one of his/her own encrypted messages and figures out the key recovery process and publishes the results online?

I mentioned in this research that sending a Public Key over the Internet did not pose problems in certain cases since it did not allow an intercepting party to decrypt messages. It does allow an intercepting party to create a message and send it as if from the intended party .

Example :

Ali wants to send an encrypted message to Omar, say over the Internet. In order to do this, Ali has to request that Omar create a Public Key - Private Key pair, and send the Public Key to him. Omar does so, but when he sends the Public Key to Ali, it is intercepted by Hassan.

Now Hassan can send an encrypted message to Omar, falsify the email return address to make it look like it is from Ali. When Omar receives the email and when he decrypts the message, he thinks it is from Ali .

This problem is addressed by the use of Authentication within the framework of a Public Key Infrastructure (PKI). PKI is beyond the scope of this theses . Suffice it to say that PKI has its own set of problems, also beyond the scope of this theses.

The point here is that Symmetric Encryption does not have this inherent Authentication weakness .

It is clear that sending a Public Key over the Internet did not pose problems in certain cases since it did not allow an intercepting party to decrypt messages, but that it does allow an intercepting party to create a message and send it as if from the intended party .

Ali wants to send an encrypted message to Omar, say over the Internet. In order to do this, Ali has to request that Omar create a Public Key - Private Key pair, and send the Public Key to him. Omar does so, and Ali sends him a message .

When Omar receives his message he is very unhappy about it. It doesn't matter why, he is just unhappy about it. When he confronts Ali with his unhappiness, he denies having sent it and suggests that someone intercepted the transmission of the Public Key over the Internet and sent Omar a message as if it had been from him.[36][37][38]

3.7 AES (Advanced Encryption Standard) as an example

The Advanced Encryption Standard (AES) is an encryption algorithm for securing sensitive but unclassified material by U.S. government agencies and, as a likely consequence, may eventually become the de facto encryption standard for commercial transactions in the private sector. (Encryption for the US military and other classified communications is handled by separate, secret algorithms.

The Advanced Encryption Standard (AES), also known as Rijndael, is the latest encryption standard approved by that National Institute of Standards and Technology (NIST). It was approved as a standard in 2001 following a five-year standardization process that involved a number of competing encryption algorithms .

AES was published by NIST as FIPS PUB 197 in November 2001.[39]

The AES encryption algorithm is a "block cipher" originally created by two Belgians named Joan Daemen and Vincent Rijmen. Since its adoption as a standard, AES has become one of the world's most popular encryption algorithms that uses symmetric keys for encryption and decryption .

AES Crypt is a file encryption software available on several operating systems that uses the industry standard Advanced Encryption Standard (AES) to easily and securely encrypt files .

You do not need to be an expert to use AES Crypt, nor do you need to understand cryptography. When using Windows, the only thing you need to do is right-click on a file, select AES Encrypt or AES Decrypt, enter a password, and AES Crypt will do the rest. On a Mac, you can drag a file to the AES Crypt program and provide the required password. On the command line, one can execute the "aescrypt" command with name of the file and password to use to encrypt or decrypt. For Java and C# developers, there is also a Java and C# library available that can read and write AES-encrypted files from within your application .[40]

Using a powerful 256-bit encryption algorithm, AES Crypt can safely secure your most sensitive files. Once a file is encrypted, you do not have to worry about a person reading your sensitive information, as an encrypted file is completely useless without the password. It simply cannot be read .

AES Crypt is the perfect tool for anyone who carries sensitive information with them while traveling, uploads sensitive files to servers on the Internet, or wishes to protect sensitive information from being stolen from the home or office. AES Crypt is also the perfect solution for those who wish to backup information and store that data at a bank, in a cloud-based storage service, and any place where sensitive files might be accessible by someone else .

Best of all, AES Crypt is completely free open source software. Since it is open source, several people have contributed to the software and have reviewed the software source code to ensure that it works properly to secure information. Most important to most users, though, is the fact that the software is available at no cost. You are free to use this software in your business, at home, or in your own open source development projects.[41]

3.7.1 Description of the cipher

AES is based on a design principle known as a substitution-permutation network, combination of both substitution and permutation, and is fast in both software and hardware.[42] Unlike its predecessor DES, AES does not use a Feistel network. AES is a variant of Rijndael which has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits. By contrast, the Rijndael specification per se is specified with block and key sizes that may be any multiple of 32 bits, both with a minimum of 128 and a maximum of 256 bits.

AES operates on a 4×4 column-major order matrix of bytes, termed the state, although some versions of Rijndael have a larger block size and have additional columns in the state. Most AES calculations are done in a special finite field.

The key size used for an AES cipher specifies the number of repetitions of transformation rounds that convert the input, called the plaintext, into the final output, called the cipher text. The number of cycles of repetition are as follows:

10 cycles of repetition for 128-bit keys.

12 cycles of repetition for 192-bit keys.

14 cycles of repetition for 256-bit keys.

Each round consists of several processing steps, each containing four similar but different stages, including one that depends on the encryption key itself. A set of reverse rounds are applied to transform cipher text back into the original plaintext using the same encryption key.

3.7.2 High-level description of the algorithm.

3.7.2.1 Key Expansions

Round keys are derived from the cipher key using Rijndael's key schedule. AES requires a separate 128-bit round key block for each round plus one more.

3.7.2.2 Initial Round

Add Round Key—each byte of the state is combined with a block of the round key using bitwise Xor.

3.7.2.3 Rounds

A-SubBytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.

B-Shift Rows—a transposition step where the last three rows of the state are shifted cyclically a certain number of steps.

C-Mix Columns—a mixing operation which operates on the columns of the state, combining the four bytes in each column.

D-Add Round Key

3.7.2.4 Final Round (no Mix Columns)

A-SubBytes

B-Shift Rows

C-Add Round Key.

A- The SubBytes step

In the SubBytes as in figure 3.2 , each byte Ai,j in the state matrix is replaced with a Sub Byte S(Ai,j) using an 8-bit substitution box, the Rijndael S-box. This operation provides the non-linearity in the cipher. The S-box used is derived from the multiplicative inverse over GF(2 to the power 8), known to have good non-linearity properties. To avoid attacks based on simple algebraic properties, the S-box is constructed by combining the inverse function with an invertible affine transformation. The S-box is also chosen to avoid any fixed points (and so is a derangement), i.e., S(Ai,j) ≠ Ai,j , and also any opposite fixed points, i.e., S(Ai,j) \o plus Ai,j ≠0xFF . While performing the decryption, Inverse SubBytes step is used, which requires first taking the affine transformation and then finding the multiplicative inverse (just reversing the steps used in SubBytes step).

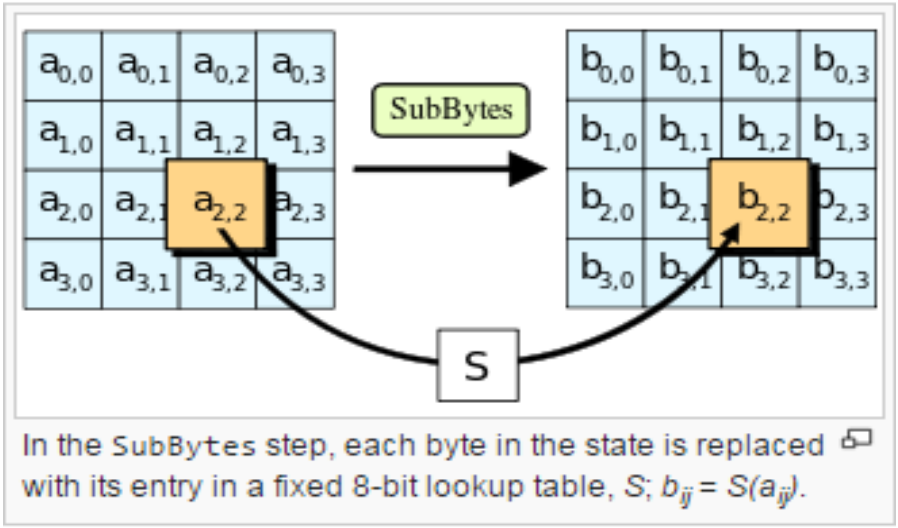

Figure 3.2 Sub Bytes Step[54]

B- The ShiftRows step

The Shift Rows step (as in figure3.3) operates on the rows of the state; it cyclically shifts the bytes in each row by a certain offset. For AES, the first row is left unchanged. Each byte of the second row is shifted one to the left. Similarly, the third and fourth rows are shifted by offsets of two and three respectively. For blocks of sizes 128 bits and 192 bits, the shifting pattern is the same. Row n is shifted left circular by n-1 bytes. In this way, each column of the output state of the Shift Rows

step is composed of bytes from each column of the input state. (Rijndael variants with a larger block size have slightly different offsets). For a 256-bit block, the first row is unchanged and the shifting for the second, third and fourth row is 1 byte, 3 bytes and 4 bytes respectively—this change only applies for the Rijndael cipher when used with a 256-bit block, as AES does not use 256-bit blocks. The importance of this step is to avoid the columns being linearly independent, in which case, AES degenerates into four independent block ciphers.

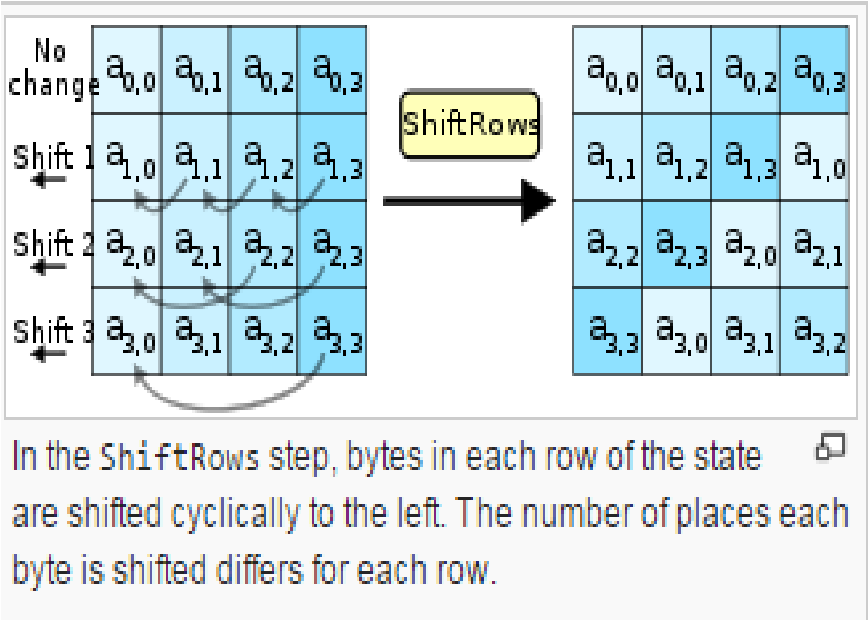

Figure 3.3 Shift Rows Step[54]

C- The MixColumns step

In the Mix Columns step (as in figure 3.4), the four bytes of each column of the state are combined using an invertible linear transformation. The MixColumns function takes four bytes as input and outputs four bytes, where each input byte affects all four output bytes. Together with ShiftRows, MixColumns provides diffusion in the cipher.

During this operation, each column is multiplied by a fixed matrix:

Matrix multiplication is composed of multiplication and addition of the entries, and here the multiplication operation can be defined as this: multiplication by 1 means no change, multiplication by 2 means shifting to the left, and multiplication by 3 means shifting to the left and then performing XOR with the initial un shifted value. After shifting, a conditional XOR with 0x1B should be performed if the shifted value is

larger than 0xFF. (These are special cases of the usual multiplication in GF(28).) Addition is simply XOR.

In more general sense, each column is treated as a polynomial over GF(28) and is then multiplied modulo x4+1 with a fixed polynomial c(x) = 0x03 · x3 + x2 + x + 0x02. The coefficients are displayed in their hexadecimal equivalent of the binary representation of bit polynomials from GF(2)[x]. The MixColumns step can also be viewed as a multiplication by the shown particular MDS matrix in the finite field GF(28). This process is described further in the article Rijndael mix columns.

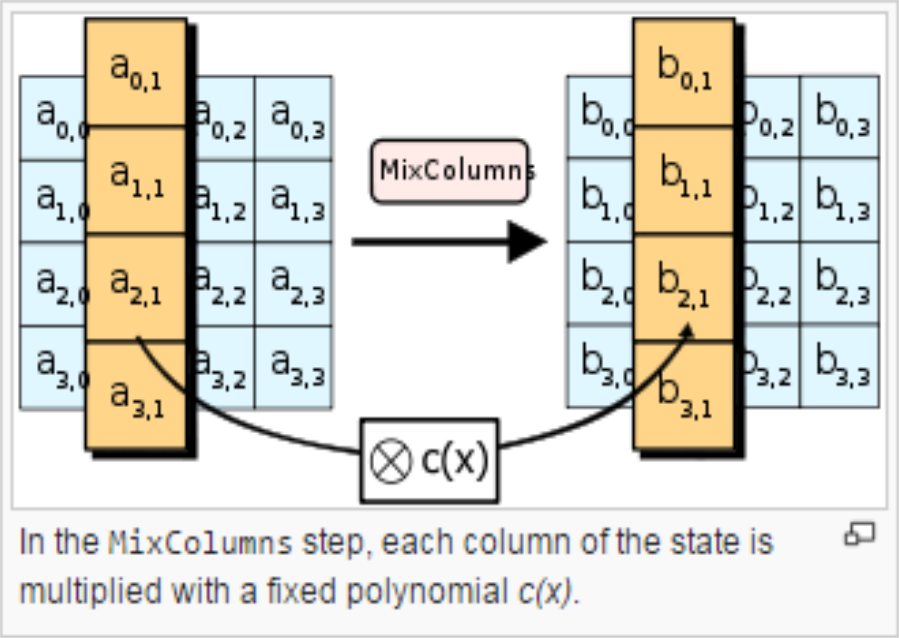

Figure 3.4 Mix Columns Step[54]

D- The AddRoundKey step

In the Add Round Key step (as in figure 3.5) , the subkey is combined with the state. For each round, a subkey is derived from the main key using Rijndael's key schedule; each subkey is the same size as the state. The subkey is added by combining each byte of the state with the corresponding byte of the subkey using bitwise XOR.[43]

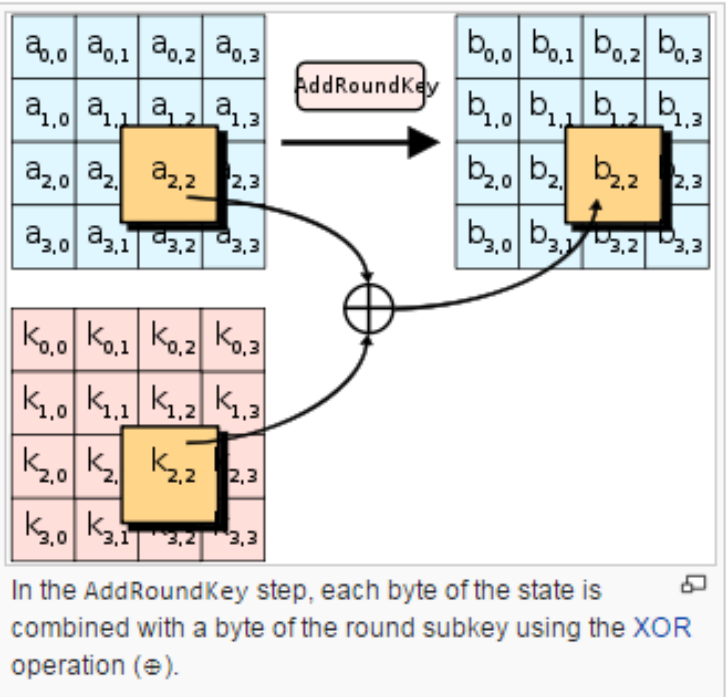

Figure 3.5 Add Round Key Step[54]

3.8 Optimization of the cipher

On systems with 32-bit or larger words, it is possible to speed up execution of this cipher by combining the SubBytes and ShiftRows steps with the MixColumns step by transforming them into a sequence of table lookups. This requires four 256-entry 32-bit tables, and utilizes a total of four kilobytes (4096 bytes) of memory — one kilobyte for each table. A round can then be done with 16 table lookups and 12 32-bit exclusive-or operations, followed by four 32-bit exclusive-or operations in the AddRoundKey step.[43]

If the resulting four-kilobyte table size is too large for a given target platform, the table lookup operation can be performed with a single 256-entry 32-bit (i.e. 1 kilobyte) table by the use of circular rotates.

Using a byte-oriented approach, it is possible to combine the SubBytes, ShiftRows, and MixColumns steps into a single round operation.[44]

3.9 Using AES Crypt for Java

This Java AES Crypt package contains the Java class es.vocali.util.AESCrypt, which provides file encryption and decryption using aescrypt file format .

Versions 1 & 2 are supported both for reading and writing. Version 0 is not supported currently .

3.9.1 Requirements for using AES Crypt for Java

The Java AES Crypt package only works in Java 6, but can be easily adapted to Java 5 by replacing the call to NetworkInterface.getHardwareAddress() with something else .

In order to use 256 bit AES keys, you must download and install "Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files" from Oracle .

3.9.2 How to Use AES Crypt for Java?

Embedded in Java Project

See JavaDoc and typical use in AESCrypt.main() method. There are no dependencies other than Java 6 API .

From the Command Line

java -cpbin es.vocali.util.AESCrypted password fromPath toPath

Operation mode is selected with (e)ncrypt or (d)ecrypt .

Using AES Crypt for CIL/.NET

This .NET AES Crypt package contains the C# class SharpAESCrypt.SharpAESCrypt, which provides file encryption and decryption using aescrypt file format .

Version 2 of the AES File Format is supported for reading and writing. Versions 0 and 1 are not verified, but there is code to read and write the formats .

Requirements for Using AES Crypt for CIL/.NET

The SharpAESCrypt package works with .NET 2.0+, and is tested with :

- Windows, Microsoft.Net, 32bit and 64bit
- Linux, various distributions, Mono 2.6+, 32bit (Mono limitation)
- OSX 10.6+, Mono 2.6+, 32bit (Mono limitation)

Besides a CLI runtime, no further dependencies are required.

Chapter four

Message Authentication Code (MAC)

4.1 Background

In most people's minds, privacy is the goal most strongly associated to cryptography. But message authentication is arguably even more important. Indeed you may or may not care if some particular message you send out stays private, but you almost certainly do want to be sure of the originate or of each message that you act on. Message authentication is what buys you that guarantee.Message authentication allows one party—the sender—to send a message to another party—the receiver—in such a way that if the message is modified en route, then the receiver will almost

certainly detect this. Message authentication is also called data-origin authentication. Message authentication is said to protect the integrity of a message, ensuring that each message that it is received and deemed acceptable is arriving in the same condition that it was sent out—with no bits inserted, missing, or modified.

Here we'll be looking at the shared-key setting for message authentication (remember that message authentication in the public-key setting is the problem addressed by digital signatures). In this case the sender and the receiver share a secret key, K, which they'll use to authenticate their transmissions. We'll define the message authentication goal and we'll describe some different ways to achieve it. As usual, we'll be careful to pin down the problem we're working to solve **.**Authenticated encryption [45] is a form of encryption that, in addition to providing confidentiality for the plaintext that is encrypted, provides a way to check its integrity and authenticity Authenticated Encryption with Associated Data, or AEAD [46], adds the ability to check the integrity and authenticity of some Associated Data (AD), also called "additional authenticated data،"that is not encrypted.

4.2 What is the Difference Between Encryption and Authentication?

Not considering how to actually do encryption or authentication, it is fairly simple for a native Latin speaker to distinguish between the two. We authenticate something to prove to the receiver of the "something" that it actually came from us. We encrypt a message so nobody, including us, can read it. Why do we authenticate or encrypt? We authenticate so that the receiver is assured that what she received came from us

and not from an imposter. This “thing” can be an item – a coin or painting for instance, or a piece of information, an email attachment or a speed command to a uranium centrifuge. We encrypt information so that only the intended receiver(s) can understand it.

So that was simple. But why do computer gurus go through great efforts to provide means of information authentication? Wouldn’t encrypting information be enough? Couldn’t the sender just include its name and address in the information and then encrypt? Well, no. The problem is that although a “man in the middle” will not understand the information, he will still be able to change it. For instance, in computer communication protocols a destination address (port) might be at a fixed position in a message. An adversary could copy such a message when it is on its way through some wire, change this value randomly, and monitor its own port/s until one of these messages – though still garbled – arrives. Once the adversary has received one message, he can now inject the encrypted port value for his own port for every message. One message would not be enough for a hacker to perform decryption, but many makes this possible. Not only would an adversary then be able to decipher messages that were not meant for her, but she can now also “break the code”, meaning deduce the encryption key. And with that key in hand, she can now send messages that are not authentic.

Therefore, a secure communication consists of authenticating the message and encrypting it.

4.3 Why do you need message authentication in addition to encryption?

Encryption does not automatically protect the data against modification.

For example, let's say we have a stream cipher that is simply a RNG (Random Number Generator), where the key is the seed. Encryption works by generating random numbers in sequence (the key stream) and exclusing- or'ing them with the plaintext. If an attacker knows some plaintext and cipher text bytes at a particular point, he can Xor them together to recover the key stream for those bytes. From there, he can simply pick some new plaintext bytes and Xor them with the key stream.

Often the attacker need not know the plaintext to achieve something. Let's take an example where an attacker simply needs to corrupt one particular field in a packet's

internal data. He does not know what its value is, but he doesn't need to. Simply by replacing those bytes of cipher text with random numbers, he has changed the plaintext.

This is particularly interesting in block ciphers where padding is used, as it opens us up to padding oracle attacks. These attacks involve tweaking cipher text in a way that alters the padding string, and observing the result. Other attacks such as BEAST and the Lucky Thirteen Attack involve modification of cipher text in a similar way. These tend to rely on the fact that some implementations blindly decrypt data before performing any kind of integrity checks.

Additionally, it may be possible to re-send an encrypted packet, which might cause some behavior on the client or server. An example of this might be a command to toggle the enabled state of the firewall. This is called a replay attack, and encryption on its own will not protect against it. In fact, integrity checks often don't fix this problem either.

4.4 What is Message authentication code (MAC) means ?

In cryptography, a message authentication code (MAC) is a short function is only one of the possible ways to generate MAC(s), accepts as input a secret key and an arbitrary-length message to be authenticated, and outputs a MAC (sometimes known as a tag). The MAC value protects both a message's data integrity as well as its authenticity, by allowing verifiers (who also possess the secret key) to detect any changes to the message content.[47]

4.5 How Many Kinds of Message Authentication There Are ?

It is often crucial for an agent who receives a message to be sure who sent it. If a hacker can call into his bank’s central computer and produce deposit transactions that appears to be coming from a branch office, easy wealth is just around the corner. If an unprivileged user can interact over the network with his company’s mainframe in such a way that the machine thinks that the packets it is receiving are coming from the system administrator, then all the machine’s access-control mechanisms are for naught. In such cases the risk is that an adversary A, the forger, will create messages that look like they come from some other party‹S, the (legitimate) sender. The attacker will send a message M to R , the receiver(or verifier) under S’s identity. The

receiver R will be tricked into believing that M originates with S. Because of this wrong belief،R may inappropriately act on M.

The rightful sender S could be one of many different kinds of entities, like a person a corporation, a network address, or a particular process running on a particular machine. As the receiver R،you might know that it is S that supposedly sent you the message M for a variety of reasons. For example, the message M might be tagged by an identifier which somehow names S ,Or it might be that the manner in which M arrives is a route dedicated to servicing traffic from S.

Here we're going to be looking at the case when S and R already share some secret key،K How S and R came to get this shared secret key is a separate question,.There are several high-level approaches for authenticating transmissions.[48]

In figure 4.1 we are authenticating messages with what is, syntactically, just an encryption scheme. The sender transmits a transformed version C of M and the receiver is able to recover M′= M or else obtain indication of failure. Adversary A Controls the communication channel and may even influence messages sent by the sender.

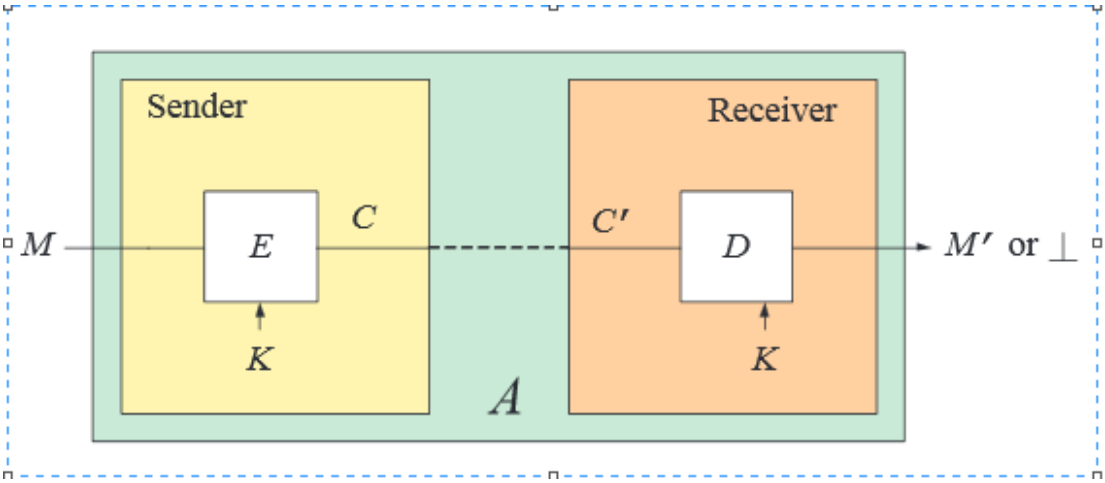

Figure 4.1 An authenticated-encryption

The most general approach works like this. To authenticate a message Musing the key K،the sender will apply some encryption algorithm E to K, giving rise to a cipher text C ,When we speak of encrypting M in this context, we are using the word in the broadest possible sense, as any sort of keyed transformation on the message that obeys are earlier definition for the syntax of an encryption scheme; in particular, we are not suggesting that C conceals M.[49][50][51]

The sender S will transmit C to the receiver R, Maybe the receiver will receive C ،or maybe it will not. The problem is that an adversary A may control the channel on which message s are being sent. Let C′be the message that the receiver actually gets. The receiver R on receipt of C ′will apply some decryption algorithm D to K and C′ .We want that this should yield one of two things: (1) a message M′that is the original message M or (2) an indication ⊥that C′be regarded as inauthentic. Viewed in this way, message authentication is accomplished by an encryption scheme. We are no longer interested in the privacy of the encryption scheme but, functionally, it is still an encryption scheme. See Fig. 4.1. We sometimes use the term authenticated encryption to indicate that we are using an encryption scheme to achieve authenticity.

In figure 4.2 is a special case of the more general frame work from the prior diagram. The authenticated message C is now understood to be the original message M together with a tag T. Separate algorithms generate the tag and verify the pair.

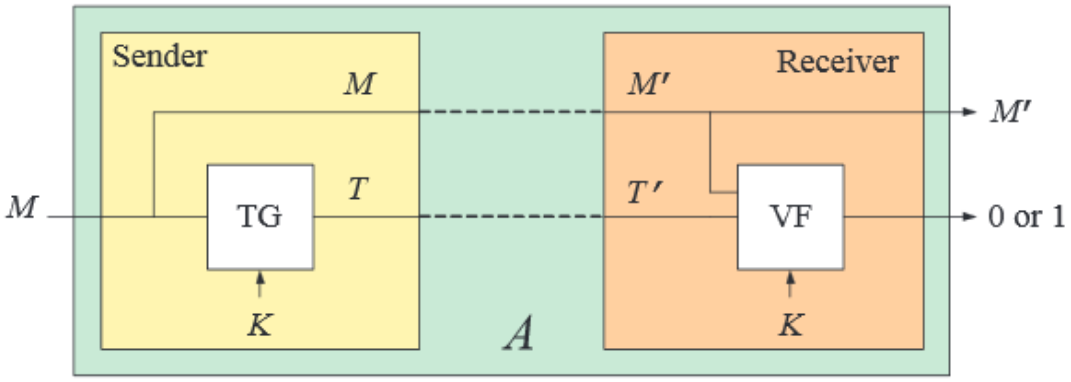

Figure 4.2 A message authentication

Since our authenticity goal is not about privacy, most often the cipher text C that the sender transmits is simply the original message M together with a tag T that is،C=(M, T).When the cipher text is of this form, we call the mechanism a message-authentication scheme.[49][50][51]

In figure 4.3 This is a special case of a message authentication scheme. The authenticated message C is now understood to be the original message M together with a tag T that is computed as a deterministic and stateless function of M and K. The receiver verifies the authenticity of messages using the same MACing algorithm.

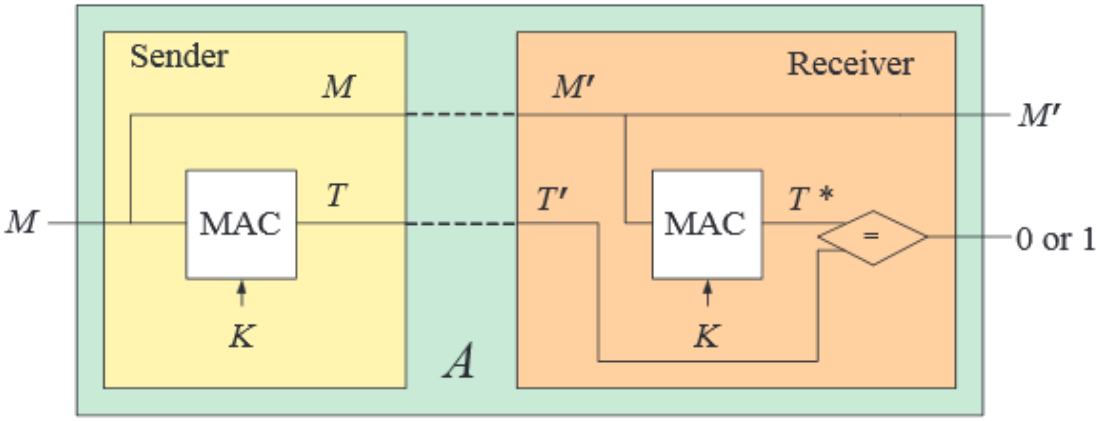

Figure 4.3 Message Authentication Code

the most common possibility of all occurs when the tag-generation algorithm TG is deterministic and stateless. In this case we call the tag-generation algorithm, and the scheme itself، a message authentication code or MAC. When authentication is accomplished using a MAC،we do not need to specify a separate tag-verification algorithm, for tag-verification always works the same way.[49][50][51]

4.6 How does MAC works ?

A message authentication code (MAC) is a small block of data attached to a message that is used by the recipient to verify the integrity of the message. One could think of it as akin to the wax seals that used to be placed on letters and formal correspondence to verify the identity of the sender and confirm that the message had not been opened. Such codes are used when certain types of encrypted or secured data are sent so the sender can check to confirm that the message has not been compromised. Message authentication codes can appear on messages such as electronic funds transfers and emails.

When a message is generated, the MAC is created at the same time. The message is sent to the recipient, and when the recipient opens it, the contents of the message are run through an algorithm to create a new message authentication code. This new code is compared with the code sent along with the original message. If the codes are the same, the message is authenticated. If there is a difference, it indicates that something about the message changed between sender and recipient.[52]

In figure 4.4 I introduced a small example on how does MAC works, In this example, the sender of a message runs it through a MAC algorithm to produce a MAC data tag. The message and the MAC tag are then sent to the receiver. The receiver in turn runs the message portion of the transmission through the same MAC algorithm using the same key, producing a second MAC data tag. The receiver then compares the first MAC tag received in the transmission to the second generated MAC tag. If they are identical, the receiver can safely assume that the integrity of the message was not compromised, and the message was not altered or tampered with during transmission.

However, to allow the receiver to be able to detect replay attacks, the message itself must contain data that assures that this same message can only be sent once (e.g. time stamp, sequence number or use of a one-time MAC). Otherwise an attacker could — without even understanding its content — record this message and play it back at a later time, producing the same result as the original sender. However, to allow the receiver to be able to detect replay attacks, the message itself must contain data that assures that this same message can only be sent once (e.g. time stamp, sequence number or use of a one-time MAC). Otherwise an attacker could — without even understanding its content — record this message and play it back at alater time, producing the same result as the original sender.[53]

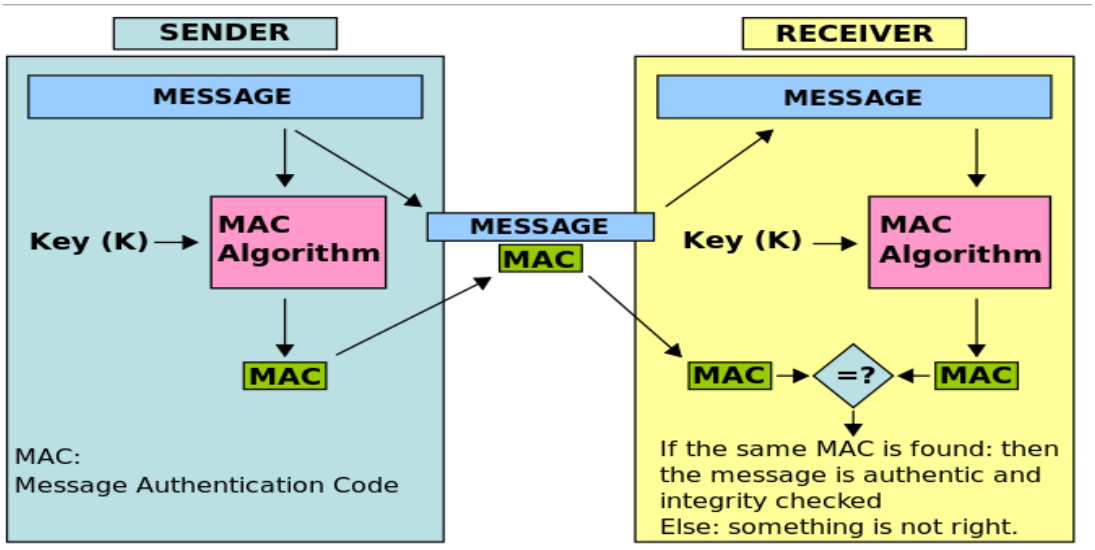

Figure 4.4 How does MAC works[55]

Chapter five

Symmetric Data Encryption Using Two Keys

In data encryption techniques, the DES (with 64bit data block) is still one of utility now, the method with properties of fast operation, and easy achieve hardware, can make digital seal, etc. But its keys management is complex. The RSA algorithm keys management is not complex as DES, but the calculation is heavy. I developed a new method with the advantages of both AES and RSA **.**

5.1Objectives

Thesis objectives are to develop a cryptography system that has two keys. One supplied and generated by the system (System Key) that's specific to an individual user and one from the user (User key). I used AES as the basis cryptographic system, so my proposed cryptographic system is symmetric. Only one master key is used to encrypt data at the end.

5.2 Motivation

The motivation of my proposed approach is to a practical encryption technique with high encryption intensity, simple hardware implementation, versatility encryption way, multi-length of data blocks, multi-usage in data safety and short running delay that could be adopted to enhance the security of information transmission.

5.3 Tools Used

I have used Java programming language to write the source code of the proposed algorithm.

5.4General Overview of The Intended Methodology

I developed a cryptography algorithm that encrypts/decrypts data using a combination of two keys: a client key that would be generated for each client and a single server/application key used by everyone. The idea is to be able to encrypt/decrypt the data only if we are in possession of both keys. So neither the server nor the client could decrypt the data alone. The idea is shown in Figure 5.1 .

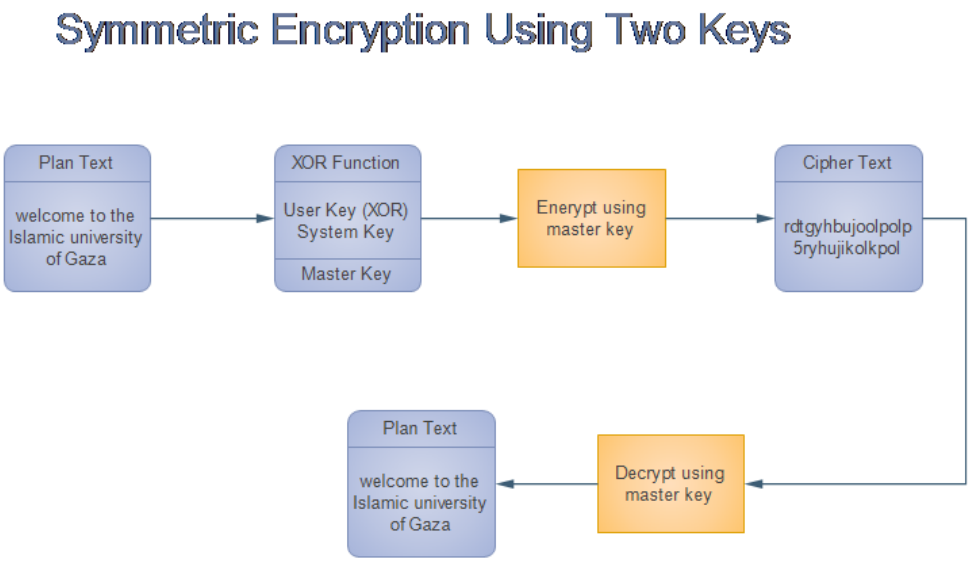

Figure 5.1 Encryption using two keys

It is simple to encrypt the data using the client key and then encrypt the data using the server keys, but if it is possible to just combine the keys in order to create the common decryption key as found of multiple encryption. I looked into Secret Sharing, but from what I got from the crypto++ library, I can only generate n keys from a message (the message would be the merged key in our case). I would rather want to generate a client key, then merge this client key with a fixed server key to generate the final key used for encryption.

I had two options when I put the idea of the two keys ,the first was concatenated the two keys to make a "master" key and then encrypt the data, the second is to encrypt with one key and the encrypt again with the other key.

Every option has its benefits. Let's see an example [16], We are going to assume:

*User trusts System

*System trusts that User would not pass KeyUser to any entity that User does not trust to be System.

This implies, in particular:

*System is already authenticated to User, and User already has a channel for passing KeyUser to System with preserved authenticity, integrity and confidentiality.

The environment of System is such that there is no risk KeyUser or KeySystem, User will be extracted from System by an attacker. Alternatively, there is no risk it will

happen without detection, in which case an auxiliary protocol exists for replacing the keys.

If the above conditions are met, both methods we have described above are fine for generating an effective content encryption scheme. Just a few points:

*Generating an effective key simply by concatenating two pieces contributed by two separate entities (as in the first method) is an unusual approach. I probably preferred to combine them using a KDF (Key Derivation Function), I used XOR function , instead of simply concatenating them. However, given that System and User both trust each other, the only compelling reason for using a KDF instead of concatenation, would be if, in some sense, it would be worse if both entities are compromised, compared to if just one of them is compromised. A KDF will be better at hiding the other part of the key material, compared to just concatenation before the AES key schedule.

*Encrypting the payload contents twice (as in second method) is also a bit unorthodox, but it can be done in such way that you will get at least the same security, as with just one encryption key. Hence, you could in such case rather just encrypt KeySystem, User with KeyUser using a key wrap encryption scheme, and store the resulting cipher text System side. Using key wrap encryption might provide additional benefits, such as the ability for User to change KeyUser without System having to decrypt and re-encrypt the entire content cipher test (if the key is used for stored data), and the ability for System to give multiple users ability to decrypt the same cipher text, but wrapping the same content encryption key using multiple user keys.

*There are many standardized key derivation functions designed specifically for this purpose. It would generally be better to use one of those rather than rolling your own. For example, HKDF (RFC 5869) should work nicely for your purposes, and it's versatile enough to let you also derive any other key material you might need (e.g. for message authentication) from the same master key(s).

At the end, there will be no obvious problems with either of proposed suggestions. One potential drawback of simply splitting the key into two parts and concatenating them is that the parts must, necessarily, be shorter than the full key. This could be an issue for ciphers with short key lengths, if one of the key parts was compromised, but

it shouldn't be a problem if you use a cipher with a key at least 256 bits long, such as AES-256. In any case, you could avoid this issue entirely by XORing the key parts instead of concatenating them (or, more generally, by using a secret sharing scheme.(We can hash the two keys, and use the result as our symmetric key for encryption. Example: compute the encryption key as K=SHA256(len(k1)||k1||k2), where k1,k2 are the two key parts provided as input. Then, encrypt the message using key K using a standard scheme (e.g., an authenticated encryption scheme like EAX or AES-GCM).

5.5 Practical Part

I implemented a java code to Encrypt and Decrypt data, I used AES algorithm to do so .In figure (5.2), you will see that there is Syspass which is the System encryption key, Userpass which is the user encryption key, Masterpass which is the master key generated by XOR the system encryption key and the user encryption key In the Text I can Enter the text which we want to encrypt or decrypt .I used the Icons Encrypt and Decrypt to do the two operations, I used the Reset button to empty the fields.

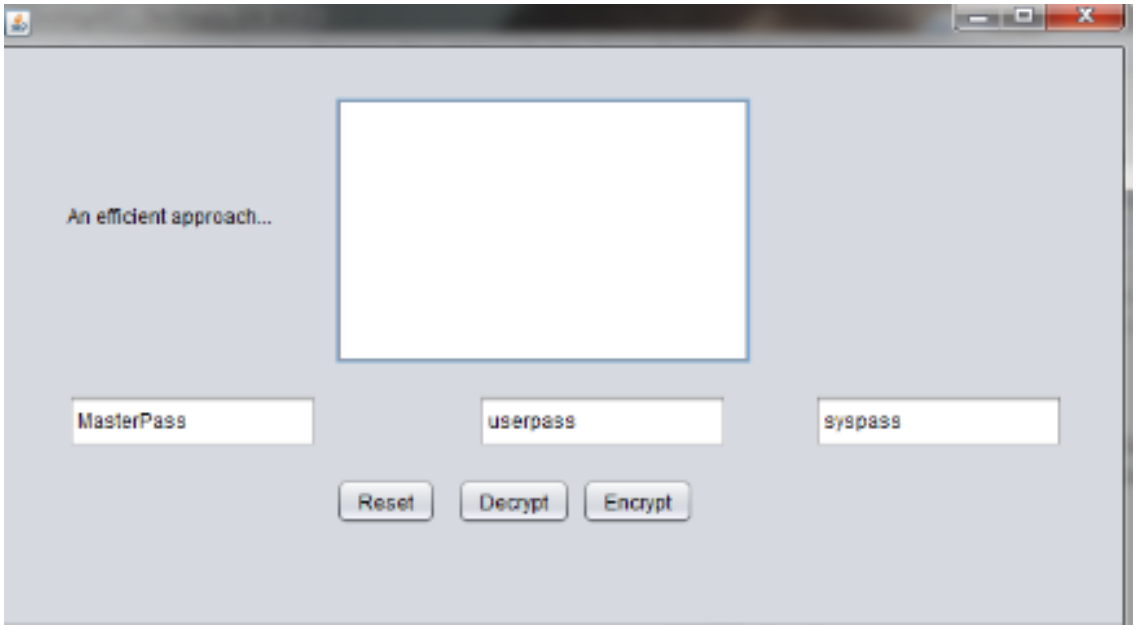

Figure 5.2 Implemented Code view

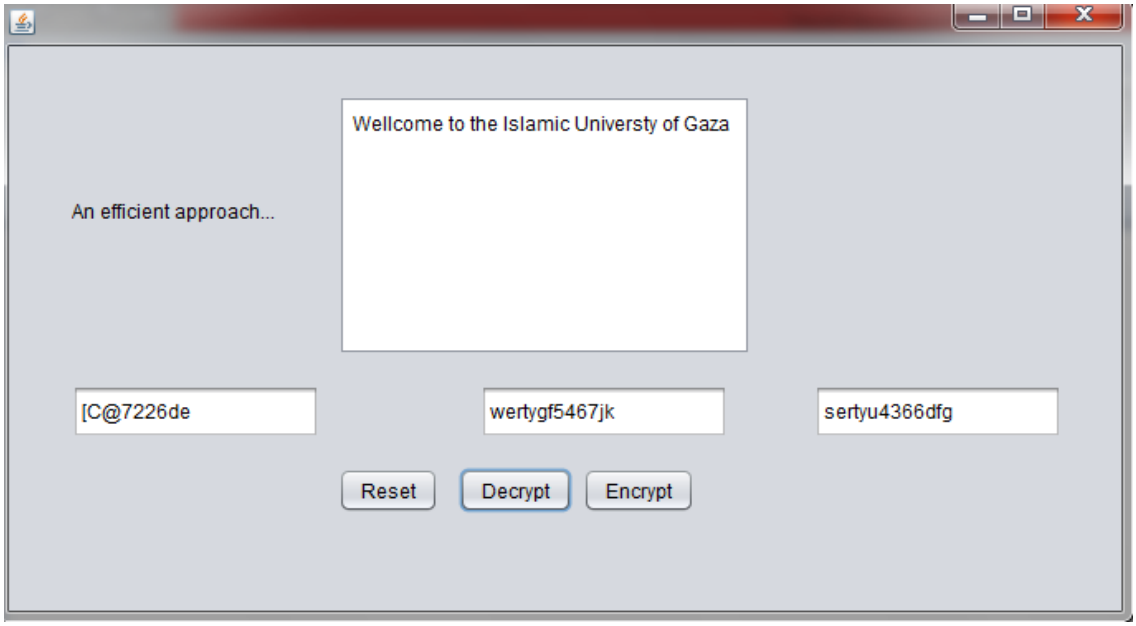

Figure 5.3 Getting the Master key

I used the java code to encrypt the plaintext "Wellcome to the Islamic University of Gaza", I used Syskey = sertyu4366dfg and Userkey= wertygf5467jk, when I run the Encrypt icon the result the master Key was =[C@7226de as in figure 5.3 and using this key the cipher text was=Wy9qvzeTQTTenCIAGsah/bcVgFikaNh00pwyBOVY80rGcPy5dNfUBlYpy9 L3dmP0 as in figure 5.4

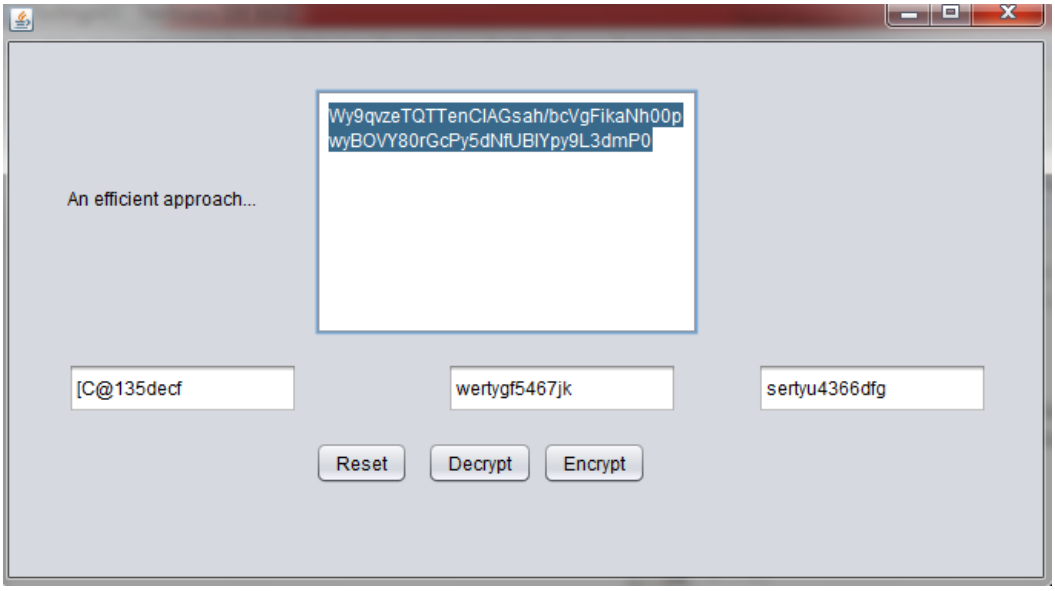

Figure 5.4 The code Result

When I implemented this java code I found that it can be used for any language combination Arabic or English as mentioned in the next examples .

Another example is in Figures 5.5,5.6,5.7

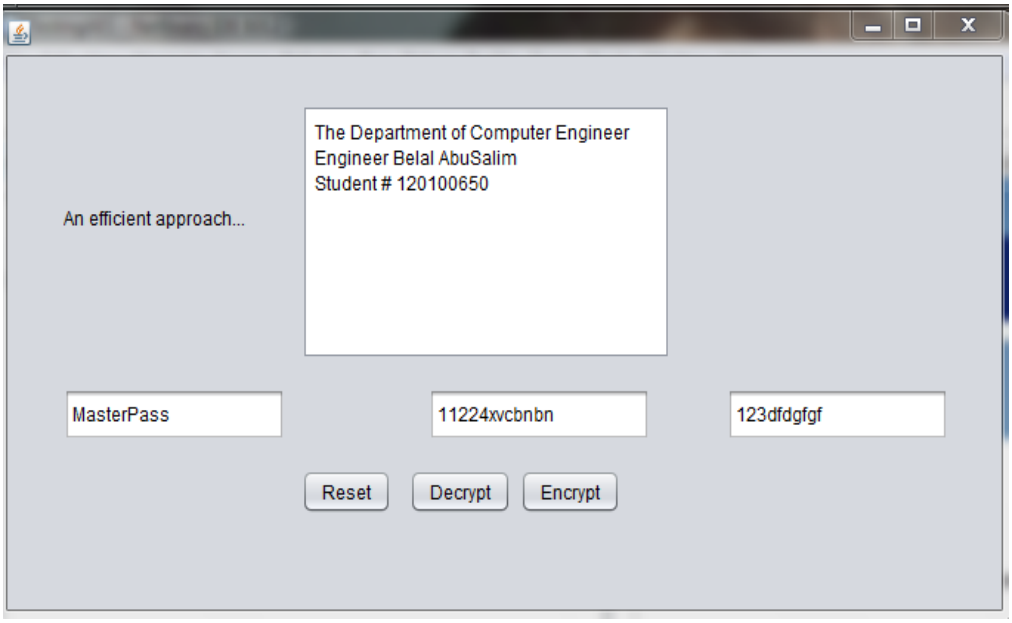

Figure 5.5 Example 2

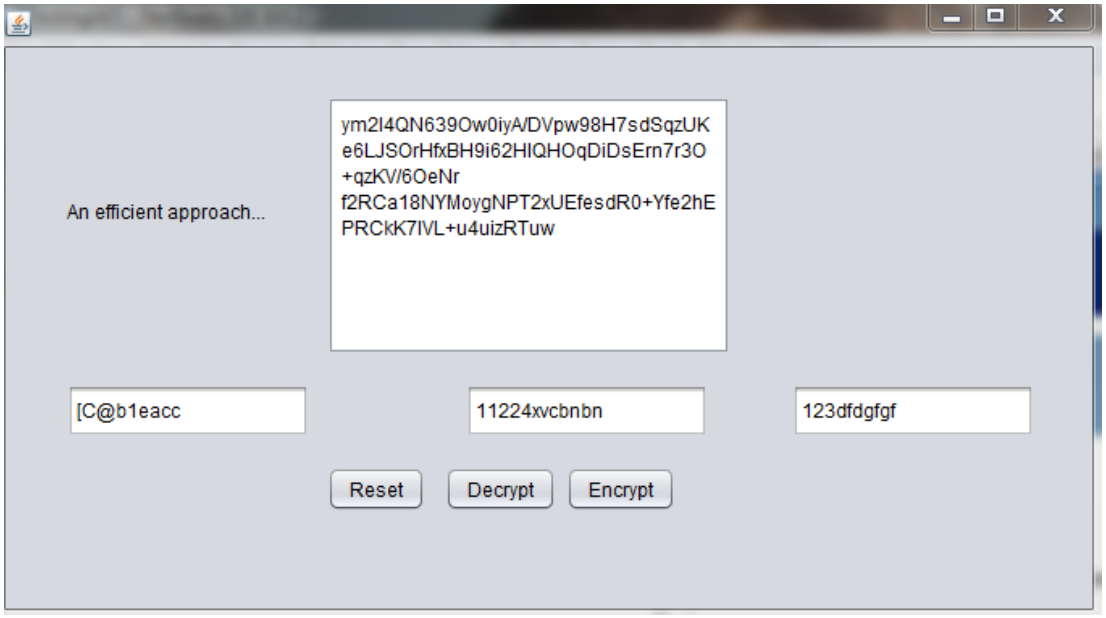

Figure 5.6 Example 2

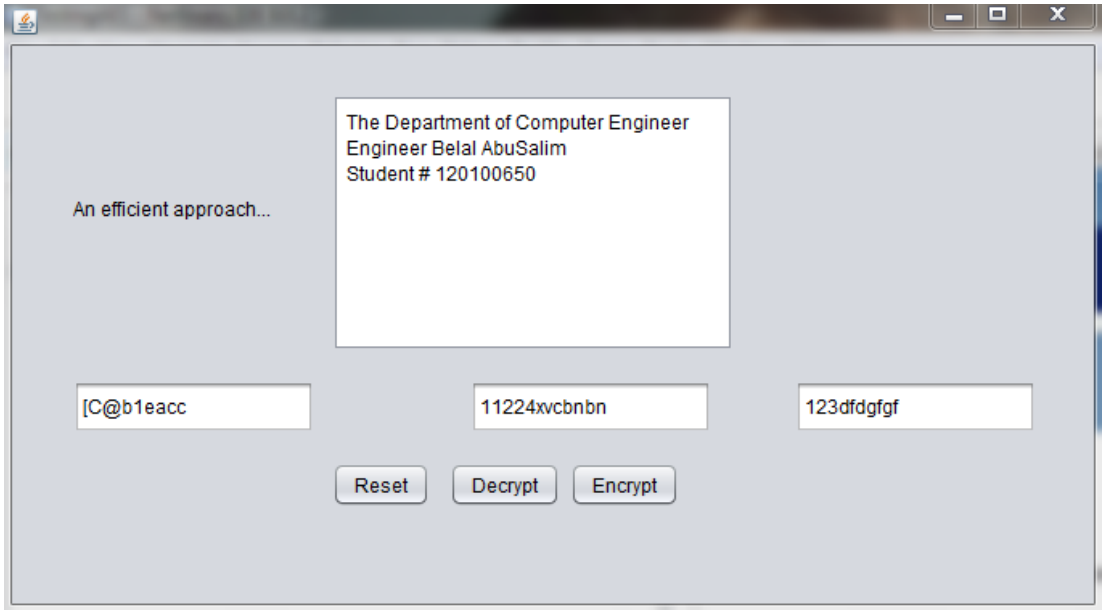

Figure 5.7 Example 2

In figures 5.5,5.6 ,5.7 I followed the same idea in example 1 in figure 5.2,5.3,5.4 I just entered the plain text and entered the system and client (user) keys and pressed the Encrypt icon and java code did the rest .

I mentioned before that it was possible to encrypt Arabic language as well or any languages that the computer (PC) support .In figure 5.8,5.9,5.10 I gave an example to shows this idea

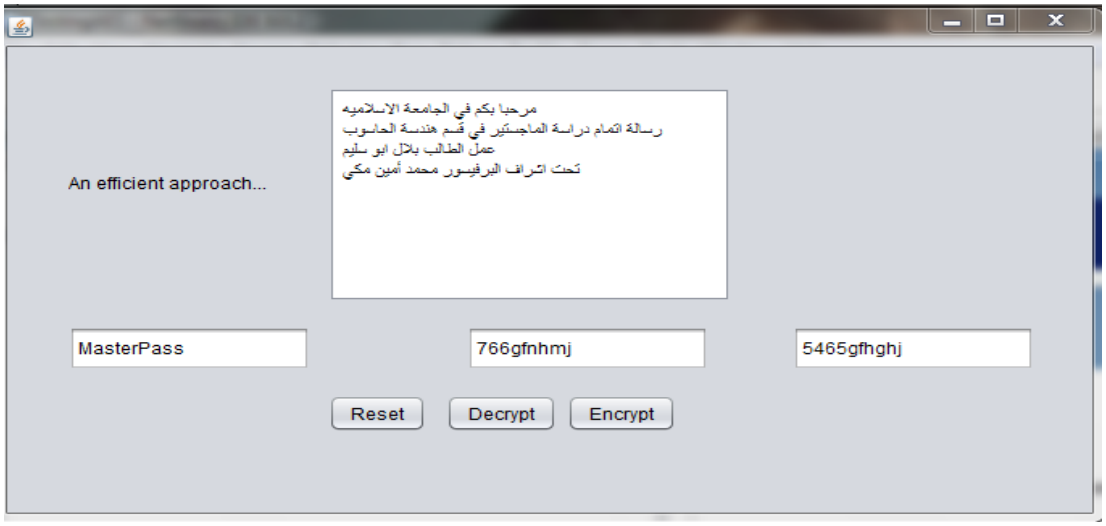

Figure 5.8 Arabic Text Encryption

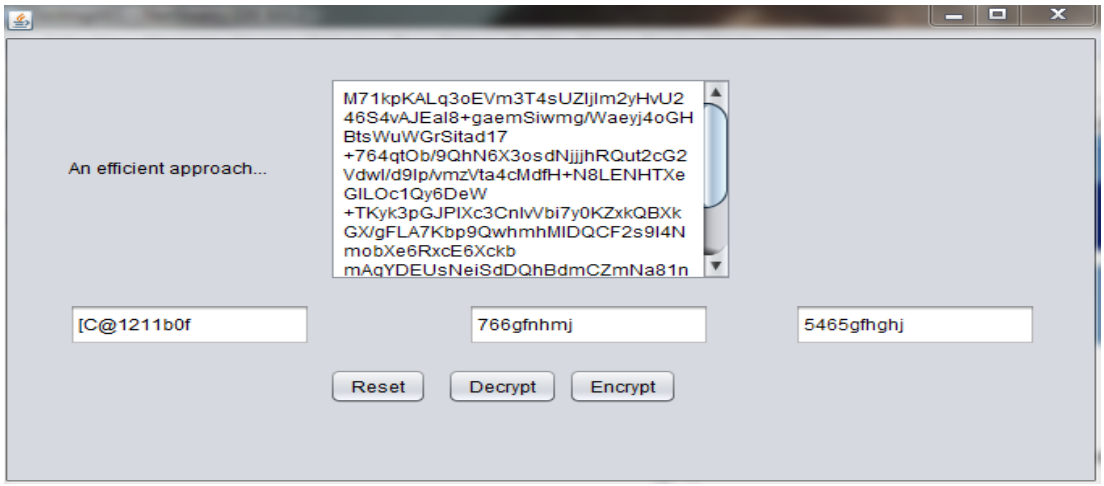

Figure 5.9 Arabic Text Encryption Result

Figure 5.10 Arabic Text Decryption

In the next example as mentioned in figures 5.11,5.12,5.13 I gave a practical view about what could happened When I changed the Hashing code (Xor) , The new Master key in the Xor function will changed to numerical values only .

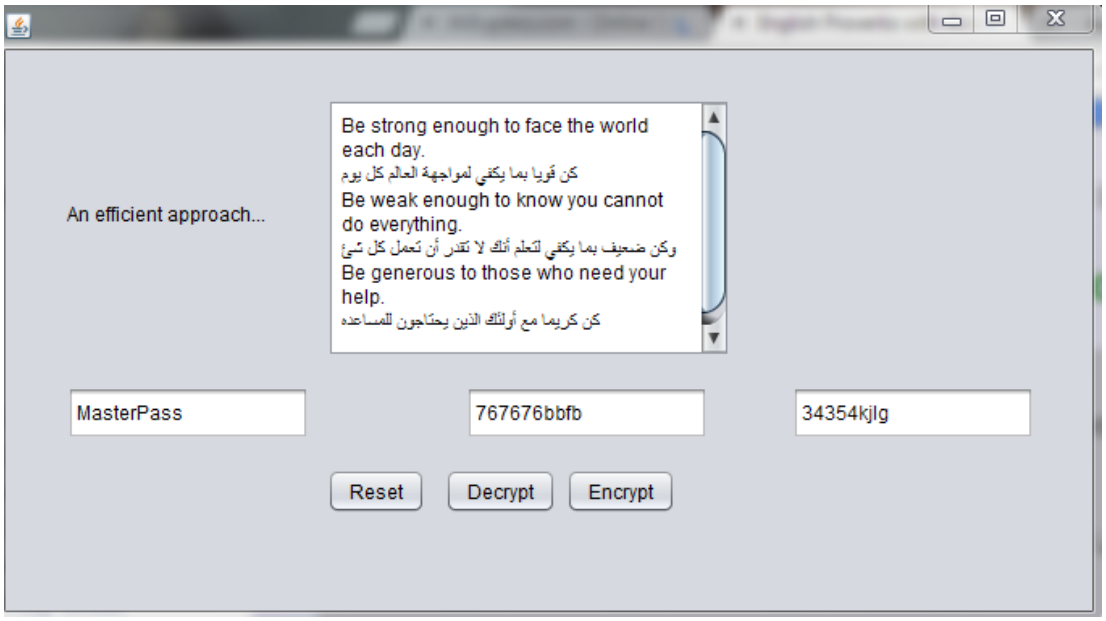

Figure 5.11 Different Hash Function

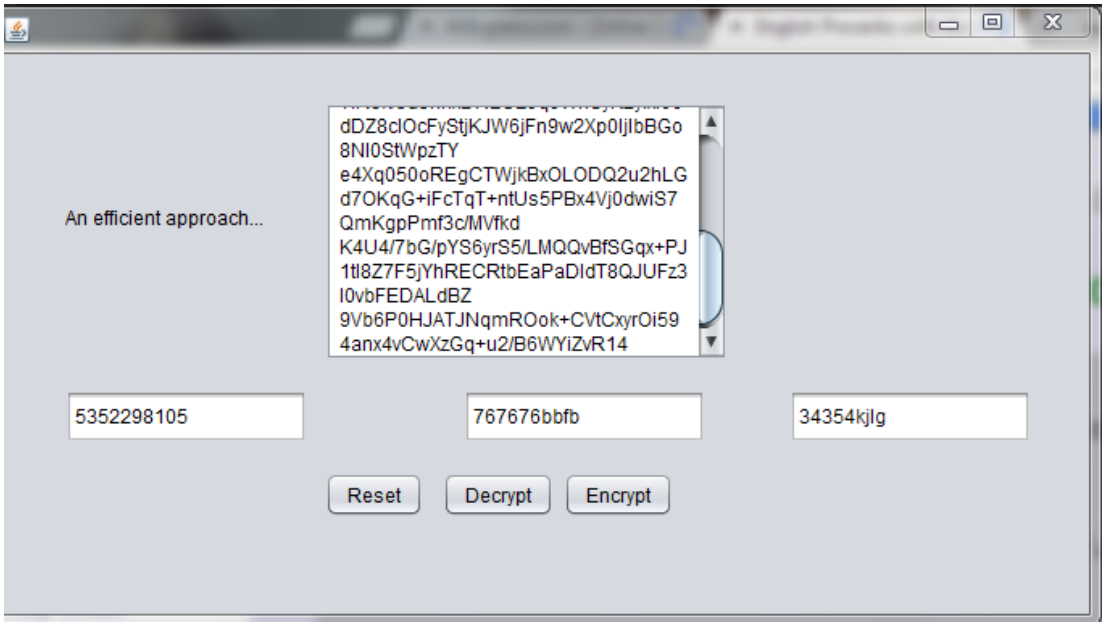

Figure 5.12 Numerical Encryption Key Result

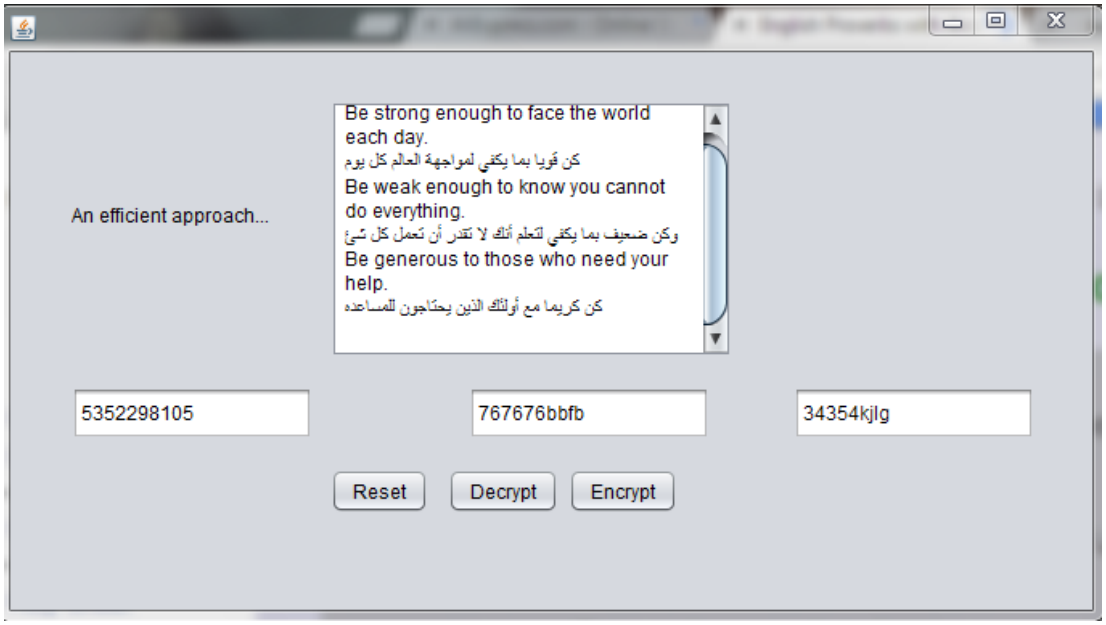

Figure 5.13 Numerical Decryption key Result

5.6 General Discussion

In chapter 3 section 4, I gave an example on how the user may trust untrusted system, when someone want to evaluate my new approach he/she will find that I gave an excellent solution to that problem ,it is simple to say that using my approach neither the user nor the system know each encryption keys ,he/she will use just the master key which is the result of XOR the user key with the System key, there are many hash functions which we can use XOR is just one of them ,So I thing that using my approach the encryption methodology will be more fast and secure .

Chapter Six

CONCLUSION

6.1 Summary and Conclusion Remarks

In this thesis I developed a new approach both practically and theoretically to encrypt Symmetric Data using two keys ,there is no need to remain that Symmetric Encryption use only one key ,my new approach is to give the user an authority over the system and get a new Master key which is the result of using any hash function, I used XOR function in this theses and Encrypt using this Master key.

I gave a complete study about encryption in General starting from Brief history about encryption and wrote briefly about Types of Cryptography as well.

I gave an overview about Symmetric data encryption and I gave a complete explanation about one of the most popular algorithms used in Symmetric encryption which is AES (Advanced Encryption Standard).

In my thesis study and in the practical section I succeeded to implement Java code that use AES algorithm to encrypt any entered text .The entered Text could be any text whatever is the text language is.

In one of the sections I mentioned about one problem in Symmetric data encryption – Credit Card Example –and I gave the solution using my new approach both practically and theoretically ,note that the Credit Card example could be any sensitive and important data.

6.2 Recommendations and Future Work

The main goal of this thesis is to introduce a new efficient a approach in Symmetric Data Encryption and I have succeeded so far to do so ,All I hope after I passed the discussion concern to this thesis to find a universal company which can adopts the main idea of this work.

References

[1] O P Verma, Ritu Agarwal, DhirajDafouti, Shobha Tyagi "Peformance Analysis Of Data Encryption Algorithms."

[2]Vue Wu, Joseph P. Noonan SosAgaian , "Binary Data Encryption using the Sudoku Block Cipher".

[3] S Hebert," A Brief History of Cryptography" , an article available at http://cybercrimes.net/aindex.html.

[4]Fu Li, Pan Ming, "A simplified FPGA implementation based on an Improved DES algorithm," IEEE Genetic and Evolutionary Computing, 2009. WGEC '09. 3rd International Conference on, pp.227-230.

[5] W. Stallings, L. Brown "Computer Security Principle and Practice," pp.593-600, ISBN: 978-0-13-600424-0, Pearson Education, 2008.

[6] C. Patterson, "High performance DES encryption in Virtex FPGA's using JBits," Field-Programmable Custom Computing Machines, 2000 IEEE Symposium on, pp.113-121.

[7]N. A. Saqib, F. Rodrıguez-Henriquez, and A. Dıaz-Perez, "A compact and efficient fpga implementation of the DES algorithm," 2004.

[8] V. Pasham, S. Trimberger, "High-Speed DES and Triple DES Encryptor/Decryptor," Aug. 2001.

[9] Wong, K., Wark, M., Dawson, E.: A Single-Chip FPGA Implementation of the Data Encryption Standard (des) Algorithm. In: IEEE Globecom Communication Conf., Sydney, Australia (1998) 827–832.

[10] O. P. Verma, Ritu Agarwal, DhirajDafouti, ShobhaTyagi " Peformance Analysis Of Data Encryption Algorithms."

[11] Hardjono, "Security In Wireless LANS And MANS,"Artech House Publishers 2005.

[12] Atulkhate, "Cryptography and Network Security, 2nd Ed," Tata Mcgraw hill, 2009, PP. 87-2004 .

[13] Sarker, M.Z.H.; Parvez, M.S" ،A Cost Effective Symmetric KeyCryptographic Algorithm for Small Amount of Data",9thInternationalMultitopic Conference, IEEE INMIC2005, pages:1-6.

[14] Cilardo, A.; Mazzeo, A.; Mazzocca, N.; Romano, L“ ،.A novel unified architecture for public-key cryptography،”Proceedingsof the conference onDesign, Automation and Test in Europe،2005, Volume 3, Pages: 52 - 57 .

[15] D. Jablon ," Strong Password Only Authenticated KeyExchange"Computer Communication Review،ACMSIGCOMM, Vol 26 No 5, pp 5-26, 1997 .

[16] William Stallings, “Cryptography and Network Security,” 4thed., Pearson Prentice Hall, 2009.

[17] Tuchman, W, “Hellman presents no shortcut solution to DES”،IEEE Spectrum, vol. 16, pp. 40-41, July 1979.

[18] Triple Data Encryption Algorithm Modes of Operation, ANSX9.52-1998.

[19] S. Morioka and A. Satoh"A 10-Gbps Full AES-CryptoDesign with a Twisted BDD S-Box Architecture،”IEEETransactions on VLSI Systems،Vol. 12, No. 7, July 2004, pp.691-686.

[20] V. Fischer and M. Drutarovsky, “Two Methods of RijndaelImplementation in Reconfigurable Hardware،”Proc. CHES،Vol.2001,2162 pp.81-96.

[21] C-P. Su, T-F. Lin, C-T. Huang; and C-W. Wu, “A High-Throughput Low-Cost AES Processor،”CommunicationsMagazine،IEEE, Vol. 41, Issue: 12, December 2003, pp. 86-91.

[22] M. Bean et al. "Hardware Performance Simulation of Round 2 Advanced Encryption Standard Algorithms ",(Nov 2001).

[23] J. Daeman and V Rijmen , " The Rijndael Block Cipher : AES Proposal ," Proc .1st AES Candidate Conf .,1998.

[24]H.Kuo and I Verbauwhede, " Architectural Optimization for a 1.82 Gbits/sec VLSI Implementation of the AES Rijndale Algorithm " Cryptographic Hardware and Embedded Systems (CHES 2001 (Lecture Notes in Computer Science 2162, Springer-Verlag, Heidelberg, Germany, 2001, pp. 53-67.

[25] J.Elbirt, W.Yip,B.Chetwynd,C.Paar,"An FPGA-based performance evaluation of the AES block Cipher Candidate algorithm finalists " IEEE Transaction on VLSI System VOL.9,NO.4,pp 545-557,August 2001.

[26] K.Gaj and P. Chodowise "Comparison of the Hardware Performance of the AES Candidates Using Reconfigurable Hardware" Proc.3rd Advanced Encryption Standard Conference New York ,April 2000 ,pp 40-54.

[27] Elbirt A.J. An FPGA-based performance evaluation of the AES block cipher candidate algorithm finalists , IEEE Transactions onVery Large Scale Integration (VLSI) Systems, Volume: 9 Issue: 4, August 2001.

[28] Dandalis A Comparative Study of Performance of AES Final Candidates Using FPGAs , Cryptographic Hardware and Embedded Systems Workshop (CHES 2000), Worcester, Massachusetts, 2000.

[29] Elbirt A.J., Yip W., Chetwynd B., Paar C. An FPGA Implementation and Performance Evaluation of the AES Block Cipher Candidate Algorithm Finalists , Third Advanced Encryption Standard (AES3) Candidate Conference, New York, 2000.

[30] Viktor Fischer ,Milos Drutarovsky," Two Methods of Rijndael Implementation in Reconfigurable Hardware ",CHES 2001,Proceedings,LNCS Vol ,2162,pp 77-92.

[31] Akashi Satoh, Sumio Morioka ,Kohij Takano and SejiiMunetoh " A compact Rijndael Hardware Architecture with S-box Optimization " ASIAC RYPT 2001,Proccedings ,LNCS Vol.2248, pp 239-254.

[32]Workshop (IDT), 2010 5th International, On page(s): 31 - 36, V,14-15 Dec.2010.

[33]Mourad,O.-c., Lotfy, S.-M., Noureddine, M., Ahmed, B., Camel, T., "AES Embedded Hardware Implementation", Adaptive Hardware and Systems, 2007. AHS 2007. Second NASA/ESA Conference on, On page(s): 103 - 109, V:5-8 Aug. 2007.

[34]Gang Zhou, Michalik, H., Hinsenkamp, L., "Efficient and High- Throughput Implementations of AES-GCM on FPGAs", Field-Programmable Technology, 2007. ICFPT 2007. International Conference on, On page(s): 185 - 192, Volume: Issue: , 12-14 Dec.

[35]Yogesh Kumar1, Rajiv Munjal2, Harsh Sharma" Comparison of Symmetric and Asymmetric Cryptography with Existing Vulnerabilities and Countermeasures".

[36] Sarker, M.Z.H.; Parvez, M.S"A Cost Effective Symmetric KeyCryptographic Algorithm for Small Amount of Date9thInternationalMultitopic Conference, IEEE INMIC2005, pages:1-6.

[37] William Stallings, “Cryptography and Network Security,” 4th ed., Pearson Prentice Hall, 2009.

[38] M. B. ManjunathShashikantChaudhari T. R. RamamohanJayshriNehete, K. Bhagyalakshmi. Mpeg video encryption in real-time usingsecret key cryptography. In Central Research Laboratory BharatElectronics Ltd., Bangalore, 2011.

[39] "Announcing the ADVANCED ENCRYPTION STANDARD (AES)" . Federal Information Processing Standards Publication 197. United States National Institute of Standards and Technology (NIST). November 26, 2001. Retrieved October 2, 2012.

[40] Westlund, Harold B. (2002). "NIST reports measurable success of Advanced Encryption Standard". Journal of Research of the National Institute of Standards and Technology.

[41] Daemen, Joan; Rijmen, Vincent (March 9, 2003). "AES Proposal: Rijndael" . National Institute of Standards and Technology. p. 1. Retrieved 21 February 2013.

[42] Bruce Schneier; John Kelsey; Doug Whiting; David Wagner; Chris Hall; Niels Ferguson; Tadayoshi Kohno et al. (May 2000). "The Twofish Team's Final Comments on AES Selection" .

[43] "Efficient software implementation of AES on 32-bit platforms". Lecture Notes in Computer Science: 2523. 2003.

[44] "byte-oriented-aes - A public domain byte-oriented implementation of AES in C - Google Project Hosting". Code.google.com. Retrieved 2012-12-23.

[45] Bellare, M. and C. Namprempre, "Authenticated encryptionRelations among notions and analysis of the genericcomposition paradigm", Proceedings of ASIACRYPT 2000‹Springer-Verlag, LNCS 1976, pp. 531-545, 2002.

[46] Rogaway, P., "Authenticated encryption with AssociatedData", ACM Conference on Computer and Communication Security (CCS'02), pp. 98-107, ACM Press, 2002‹<http://www.cs.ucdavis.edu/~rogaway/papers/ad.html>.

[47] "IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications". (2007 revision). IEEE-SA. 12 June 2007. doi:10.1109/IEEESTD.2007.373646.

[48] M. Bellare, J. Kilian and P. Rogaway.The security of the cipher block chaining message authentication code.Journal of Computer and System Sciences,Vol. 61, No. 3‹Dec 2000, pp. 362–399.

[49] M. Bellare, R. Canetti and H. KrawczykKeying hash functions for message authentication.Advances in Cryptology – CRYPTO ’96Lecture Notes in Computer Science Vol. 1109, N. Koblitzed., Springer-Verlag, 1996.

[50] J. Black, S. Halevi, H. Krawczyk, T. Krovetz, and P. RogawayUMAC: Fast and secure message authentication.Advances in Cryptology – CRYPTO ’99Lecture Notesin Computer Science Vol. 1666, M. Wiener ed., Springer-Verlag, 1999.

[51] J. Black and P. RogawayCBC MACs for Arbitrary-Length Messages: The Three-KeyConstructions.Advances in Cryptology – CRYPTO ’00Lecture Notes in Computer ScienceVol. 1880, M. Bellare ed., Springer-Verlag, 2000.

[52]N.Zivic,"SecurityAspectsofSoftVerifiedMessagesProtectedbyMessageAuthenticationCodes" ,IEEE 11th .International Conference on Trust, Security and Privacy in Computing and Communications 2012.

[53] "VMAC: Message Authentication Code using Universal Hashing". CFRG Working Group (CFRG Working Group). Retrieved 16 March 2010.

[54] https://en.wikipedia.org/wiki/Advanced_Encryption_Standard.

[55] https://en.wikipedia.org/wiki/Message_authentication_code.