DATA TRANSMISSION USING AES ENCRYPTION VIA EMAIL AND SMS

ARIF HANAFI BIN SHARIF KATULLAH

UNIVERSITY MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS

JUDUL: **DATA TRANSMISSION USING AES ENCRYPTION VIA EMAIL AND SMS**

SESI PENGAJIAN: **2013/2014**

Saya **ARIF HANAFI BIN SHARIF KATULLAH**
(HURUF BESAR)

mengaku membenarkan tesis (PSM/~~Sarjana/Doktor Falsafah~~)* ini disimpan di Perpustakaan Universiti Malaysia Pahang dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Universiti Malaysia Pahang.
2. Perpustakaan Perpustakaan Universiti Malaysia Pahang dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (✓)

□ SULIT (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

□ TERHAD (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

□ TIDAK TERHAD

Disahkan oleh

______________________________ ______________________________

(TANDATANGAN PENULIS) (TANDATANGAN PENYELIA)

Alamat Tetap: JB 405 Taman Sri Kolam
Jalan Sultan Sulaiman
Kuala Terengganu

Penyelia:

Dr Rohani Binti Abu Bakar

Tarikh:____________________

Tarikh:____________________

CATATAN:

* Potong yang tidak berkenaan.

** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali sebab dan tempoh tesis ini perlu dikelaskan sebagai SULIT atau TERHAD.

◆ Tesis dimaksudkan sebagai tesis bagi Ijazah Doktor Falsafah dan Sarjana secara penyelidikan, atau disertai bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM).

DATA TRANSMISSION USING AES ENCRYPTION VIA EMAIL AND SMS

ARIF HANAFI BIN SHARIF KATULLAH

A thesis submitted in fulfillment of the requirements
for the award of the degree of the
Bachelor of Computer Science (Software Engineering)

FACULTY OF COMPUTER SYSTEMS & SOFTWARE ENGINEERING
UNIVERSITY MALAYSIA PAHANG

2013

DECLARATION

I hereby declare that this thesis entitled Data Transmission Using AES Encryption via Email and SMS is the result of my own research except for quotations and summaries which have been duly acknowledged. The thesis has not been accepted for any degree and is not concurrently submitted in the candidature of any other degree.

Signature : ....

Name : Arif Hanafi Bin Sharif Katullah

ID Number : CB10031

Date : ....

SUPERVISOR DECLARATION

I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of the degree of Bachelor of Computer Science (Software Engineering)

Signature : …………………………………

Supervisor : Dr Rohani Binti Abu Bakar

Date : ………………………………….

ACKNOWLEDGEMENT

This thesis would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

First and foremost praise be to Almighty Allah for all His blessings for giving me patience and the strength to plod on despite my constitution wanting to give up throughout the duration of this undergraduate research, and to Prophet Muhammad Peace Be Upon Him for his way of life, sayings (Hadith), and perseverance.

I would like to express my sincere gratitude to my supervisor, Dr Rohani Binti Abu Bakar for the continuous support of my undergraduate study and research, for his motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better supervisor and mentor for my undergraduate research.

This thesis would not have been possible without my many colleagues who stand by my side during this undergraduate research offering brilliant ideas and unwavering support.

Last but not the least, there are no words big enough to express my gratitude to my parents and family members for being my inspirations, for believing in me especially during the hard times and for their undivided love throughout my entire life.

ABSTRACT

A computer, and other mobile computing devices, are generally accepted globally as a personal computing platform. Given the amount of sensitive information gathered by these devices, there are serious privacy and security implications for both individual use and enterprise organization. Confidentiality of the data can be effectively restricted by deploying an encryption file technique. All major computers Operating System now integrates some form of encryption. In certain situations, this is inadequate, as users may be forced into disclosing their decryption keys. In this case, the data must be hidden so that it is very certainly transmit sensitive information but unable to intercept easily if not encrypt before sending to the intended person. Encryption and decryption technique while the interior structure's strength of the data file depends on the key/password it used. This thesis priority is to avoid the using of the manual process such as hand-to-hand method transmission while maintaining the concealment of their sensitive or confidential data and explores the usefulness of encryption for computer devices. Users especially choose weak technique. The goal is to ensure the confidential data is not reachable for unauthorized personnel by implementing the use of passwords that are for encryption keys and to make ease for the transmission confidential data via email platform.

ABSTRAK

Sebuah komputer, dan peranti pengkomputeran mudah alih yang lain, diterima secara umum di peringkat global sebagai platform pengkomputeran peribadi. Memandangkan jumlah maklumat sensitif yang dikumpul oleh alat-alat, terdapat sulit dan keselamatan implikasi yang serius untuk kegunaan individu dan organisasi perusahaan. Kerahsiaan data boleh berkesan disekat oleh menggunakan teknik fail penyulitan. Semua komputer utama Sistem Operasi kini mengintegrasikan beberapa bentuk penyulitan. Dalam keadaan tertentu, ini adalah tidak mencukupi, sebagai pengguna boleh dipaksa mendedahkan kunci penyahsulitan mereka. Dalam kes ini, data mesti disembunyikan sedemikian bahawa sangat kepastian menghantar maklumat sensitif tetapi tidak dapat memintas dengan mudah jika tidak menyulitkan sebelum menghantar kepada orang yang dimaksudkan. Penyulitan dan penyahsulitan teknik manakala kekuatan struktur dalaman fail data bergantung pada kekunci / kata laluan yang digunakan. Ini keutamaan tesis adalah untuk mengelakkan menggunakan proses manual seperti tangan-ke-tangan penghantaran kaedah yang sama mengekalkan penyembunyian data sensitif atau sulit mereka dan meneroka kegunaan penyulitan untuk peranti komputer. Pengguna terutamanya memilih teknik penyulitan yang lemah. Matlamatnya adalah untuk memastikan data sulit tidak dapat dihubungi untuk kakitangan yang tidak dibenarkan dengan melaksanakan penggunaan kata laluan yang untuk kunci penyulitan dan membuat mudah bagi data sulit penghantaran melalui platform e-mel.

Table of Content

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDIX

CHAPTER 1

INTRODUCTION

This section briefly describes the entire overview research that includes five parts. Foremost, is the introduction, followed by the problem statement. Then, is the objective and of study.Finally terminology and thesis organization.

1.1 Background

The desire to transmit messages securely is not new for centuries. Community kept communication secret. Nowadays, with the latest technology, particularly the Internet vast amount data transmit sensitive data such as confidential trade secrets, military strategies and Government communication (Eskicioglu, 2001).

Encryption is the main key structure to prevent malicious attacks and is an essential role of Information Security. However, providing the alternative in information privacy, but it acts as authenticate coming from the sender. As a result, the rationale of adopting the correct encryption techniques is vital the reliability of data may be leaked once weakness in the encryption key was detected by preventing tampering, falsification and counterfeiting. Transmitting messages is not new for centuries community continues look way or method to transmitted data with efficient and optimizes time. Encryption and decryption technique security heavily relied on the interior structure of the strength depend on the key it uses (Xin et al, 2011). Cryptography or secret codes have been known some 4000 years.

Now, first being used by the ancient Egyptians, the prospect of writing something that only selected persons can decipher has proven its usefulness through many years. Cryptography has played decisive roles in both World War I and II (Maartmann-Moe, 2007). This deployment is not new since ancient times, Julius Caesar (100 B.C.E. – 44 B.C.E) employed by sending secret messages. This method is shift performed modular 26. The plaintext A become D, B became E and Z substiture to C (Eskicioglu, 2001). Figure 1.1 describes how plaintext encrypted becomes a cipher text before been transmitted to insecure channel. After receiving the cipher text it needs to be decrypted before it able to get the plaintext.

Many times when sensitive data is exchanged electronically the privacy of the data is a requirement. The use of encryption restricts unintended receivers from viewing the confidential data, which are deemed confidential and potentially dangerous if made known to irresponsible persons. Today, encryption is the procedure of transforming plaintext, data that can be read by anyone, to cipher text, data that can only be read by someone with a secret decryption key. A message before being changed in any way is called plaintext. Plaintext messages are converted to cipher text via some encryption method. A particular such method is called a **cryptosystem**. Cryptosystem would be a suitable method in implement to the examination paper (confidential data) to avoid any sensitive data fall into someone else that not involve.

A cryptosystem is designed so that decryption can be accomplished only under certain conditions, which generally means only by persons in possession of both a decryption engine (these days, generally a computer program) and a particular piece of information, called the decryption key, which is supplied to the decryption engine in the process of decryption. Plaintext is converted into cipher text by means of an encryption engine (again, generally a computer program) whose operation is fixed and determinate (the encryption method) but which functions in practice in a way dependent on a piece of information (the encryption key) which has a major effect on the output of the encryption process. A cryptosystem could be designed which made use of several different methods of encryption, the particular method chosen for a particular encryption process being key-dependent.

The combination of encryption methods results again in an encryption method, which is just as deterministic as a simpler cryptosystem, although probably harder for a cryptanalyst to crack. A good cryptosystem should in fact vary the details of its encryption method in a key-dependent way, though high security does not require the combination of distinct encryption algorithms. The result of using the decryption method and the decryption key to decrypt cipher text produced by using the encryption method and the encryption key should always be the same as the original plaintext (except perhaps for some insignificant differences).In this process the encryption key and the decryption key may or may not be the same.

The Advanced Encryption Standard, in the following referenced as **AES**, is the winner of the contest, held in 1997 by the US Government, after the Data Encryption Standard was found too weak because of its small key size and the technological advancements in processor power. Fifteen candidates were accepted in 1998 and based on public comments the pool was reduced to five finalists in 1999. In October 2000, one of these five algorithms was selected as the forthcoming standard: a slightly modified version of the Rijndael. The Rijndael, whose name is based on the names of its two Belgian inventors,Joan Daemen and Vincent Rijmen, is a Block cipher, which means that it works on fixed-length group of bits, which are called blocks. It takes an input block of a certain size, usually 128, and produces a corresponding output block of the same size.

The transformation requires a second input, which is the secret key. It is important to know that the secret key can be of any size (depending on the cipher used) and that AES uses three different key sizes: 128, 192 and 256 bits. While AES supports only block size of 128 bits and key sizes of 128, 192 and 256 bits, the original Rijndael supports key and block sizes in any multiple of 32, with a minimum of 128 and a maximum of 256 bits.

1.2 Problem Statements

The confidential data that are involved in assembling or transferring the confidential data still using the traditional method (manual process hands-to-hands). The confidential data that process manually leads to the exposing of human errors, this problem can produce the security issues. What if the papers that contain confidential information were lost and possessed by irresponsible person it would jeopardize the integrity of the data itself. The manual process should develop a good security and only the authorized personnel can access the confidential data when the process of preparing examination takes place. The security of the manual process is still in uncertainty in how the organization accomplishes the examination paper security in each process. When there are changes in the examination question the manual process of transferring confidential data also creates a difficulty to lecturers to get seat with the panel for the vetting process and evaluation session when they are in a different time and place.

1.3 Objectives

In this section the objectives of the project will be clear out. Thus, the objectives of this project are as follows:-

- To develop a prototype that encrypts the confidential files.

- To employ an AES encryption technique in the process of transferring confidential files.

- To test the propose prototype in order to evaluate the provided functions able to execute smoothly.

1.4 Scope Of Study

- The main scope of the prototype is to encrypt the confidential data file (personal data, credit card data, examination paper and many more) .
- This system is developed by using VB.net language and using GSM Modem in the application that run on Windows.
- The file extension that can be encrypted/decrypted (.doc, .txt , .pdf, .png, jpeg, jpg)
- The SMS and E-mail can be sent directly when using the system

1.5 Terminology

Web-Based

- A web application is an application that is accessed over a network such as the Internet or an intranet

Encryption

- The activity of converting data or information into code.

Cryptanalysis

- Science and sometimes art of breaking cryptosystems.

Cryptosystem

- System for encoding and decoding secret messages

1.6 Thesis Organization

This thesis consists of six (6) chapters.

Chapter 1 will discuss on introduction to the system. The discussion consists of system overview. Problem statement discuss on the problem that faced by the current system. On objectives, the reasons of the development of project are listed. Scope of the project is discussed on project and user limitation.

Chapter 2 is literature review which will discuss on current system and the technique or software that is used in the current system.

Chapter 3 will discuss on system methodology. It will be discuss on the method that is used to develop the system and project planning. In this chapter also will discuss the needs of the project such as the software and the device that are needed to develop the system.

Chapter 4 will discuss on project implementation. This chapter will discuss on design of project development.

Chapter 5 will discuss on the discussion and result that receive from the data and data analysis, project constrains and, fix and suggestion of the system. Project analysis will discuss on project objective which continuously with project problem.

Chapter 6 will discuss on conclusion of the project. This is including the conclusion of the data that are received and conclusion of the methodology and used research implementation

CHAPTER 2

LITERATURE REVIEW

2.0 Introduction

The purpose of reviewing previous work is to guide through the kind of work that others have done related to the project field. A literature review is a body of text that aims to review the critical points of current knowledge including substantive findings as well as theoretical and methodological offerings to a particular topic. This chapter will take brief explanations on study of previous or existing system that related to the proposed system, based on the development process, tools and platform used.

Many technologies exist that can be adapted in order to integrate a stand-alone system. By making a research and analyze the system strength and weaknesses, it will be easier to me developed system for AES data encryption system. An encryption process uses an algorithm and a key to transform plain text which is original data into cipher text. The inverse of the encryption process is decryption. Only people who have the secret key or password can decrypt the message into plain text encrypted messages can sometimes be broken by cryptanalysis, which is also called code breaking although modern cryptography techniques are virtually unbreakable electronic security becomes increasingly important as nowadays the internet and other forms of electronic communication become more prevalent and not secure. It is the technique of the principle means to protect information security. Besides ensuring the information is confidential, it also provides digital signature, authentication, secret sub-storage, system security and any other functions

2.1 Overview of Data File Transmission Using AES Encryption via Email and SMS

Currently the transferring confidential data for FSKKP faculty still using hands-to-hands method. This might lead to the exposing the human errors if the staff that handle the confidential data lost or misplaced it. The system that will develop will be able to encrypt the data using the user own password or secret key. The users that key-in the password should take great care when the users select the password because the prototype has no way to retrieve a lost password. The user can send the password to the person they want straightly to receiver mobile by insert the receiver mobile phone number. The user also can send the data that is fully encrypt via email platform that is also provide in the system. This encryption system used 256-bit Advanced Encryption Standard (AES) method. When the receiver received the encrypted files the receiver can open the files by using the password that was sent via mobile. This will provide a good security level for the confidential files. The prototype will protect the confidential and the integrity of the files.

2.2 Comparison between cryptographic techniques

DES: (Data Encryption Standard), was the first encryption standard to be recommended by NIST (National Institute of Standards and Technology).DES is (64 bits key size with 64 bits block size). Since that time, many attacks and methods recorded the weaknesses of DES, which made it an insecure block cipher. 3DES is an enhancement of DES it is 64 bit block size with 192 bits key size. In this standard the encryption method is similar to the one in the original DES but applied 3 times to increase the encryption level and the average safe time. It is a known fact that 3DES is slower than other block cipher methods (Coppersmith, D. 1994).

RC2 is a 64-bits block cipher with a variable key size that range from 8 to 128 bits. RC2 is vulnerable to a related-key attack using 234 chosen plaintexts (Coppersmith, D. 1994).

Blowfish is block cipher 64-bit block - can be used as a replacement for the DES algorithm. It takes a variable-length key, ranging from 32 bits to 448 bits; default 128 bits. Blowfish is unpatented, license-free, and is available free for all uses. Blowfish has variants of 14 rounds or less. Blowfish is successor to Twofish.

AES is a block cipher .It has variable key length of 128, 192, or 256 bits; default 256. It encrypts data blocks of 128 bits in 10, 12 and 14 round depending on the key size. AES encryption is fast and flexible; it can be implemented on various platforms especially in small devices. Also, AES has been carefully tested for many security applications. The cipher key used in the algorithm is of 128 bits. Therefore, to break the cipher key an attacker has to check 2128 possibilities which are practically almost impossible. Therefore, the brute-force attack fails on this algorithm. The flow of the algorithm makes sure that there is no fixed pattern in any of the steps of the algorithm. The components of the proposed algorithm have brought about strong diffusion and confusion. Therefore, statistical and pattern analysis of the ciphertext fails. The most important security advantage is that no differential or linear attacks can break this algorithm (Rayarikar et al, 2012).

RC6 is block cipher derived from RC5. It was designed to meet the requirements of the Advanced Encryption Standard competition. RC6 proper has a block size of 128 bits and supports key sizes of 128, 192 and 256 bits. Some references consider RC6 as Advanced Encryption Standard (Khate, 2009).

Table 2.1 : Comparison of encryption techniques

Factors	AES	3DES	DES
Key Length	128,192 or 256 bits	(k1,k2 and k3) 168 bits (k1 and k2 is same 112 bits	56 bits
Cipher Type	Symmetric	Symmetric	Symmetric
Block Size	128,192 or 256 bits	64 bits	64 bits
Developed	2000	1978	1977
Cryptanalysis Resistance	Strong to any attack	Vulnerable in some attacks	Vulnerable
Security	Considered Secured	Intermediate	Proven inadequate
Possible Keys	2^128, 2^192 and 2^256	2^112 and 2^168	2^56
Time Require to check keys at 50 billion keys per second	For 128 bits = 5x10^21 years	For 112 bit key = 800 days	For 56 bit keys = 400 days

2.3 Why AES?

The latest trends in e-mail system that utilize handling of two categories cryptographic (RSA by means of asymmetric keys and AES through symmetric key). These methods build a strong encryption base capable of enduring numerous types of hit, uncovering and reverse engineering. The computer which using AES 128 bit was designed to negate the major defect existing in other encryption communication (Hopper et al 2009 and Yang 2006). The AES may remain the undisputed encryption process that able to withstand the entire process of the "weakest link in the encryption" both leave behind several stages of protection (Mare et al, 2011).

2.4 Types of cryptography

Willingly available are numerous conduct of sort cryptographic techniques. At this point value amount of keys that use for encryption and decryption. The two types of techniques are:

a) **Secret key cryptography** - occasionally known as symmetric cryptography. It long-established form of cryptography, single key used to encrypt and decrypt a message. In some situation not only handle encryption, but also conformity with authentication. These techniques were used by message authentication codes (Kaufman et al, 2002).

b) **Public Key Encryption** - Most noteworthy recent growth cryptography the last few centuries was mentioned by Stanford University lecturer Martin Hellman along with Graduate apprentice Whitfield Diffie in 1976. Their research about possible that both users can exchange different key via secure communiqué over a secure communications outlet as private key not make known other party (Kaufman et al,2002).

2.5 Existing of AES data-encryption system

This part will describe briefly any existing system that using the same technique of encryption that is available in the website. The users need to purchase the product to use any of the existing system in the online website.

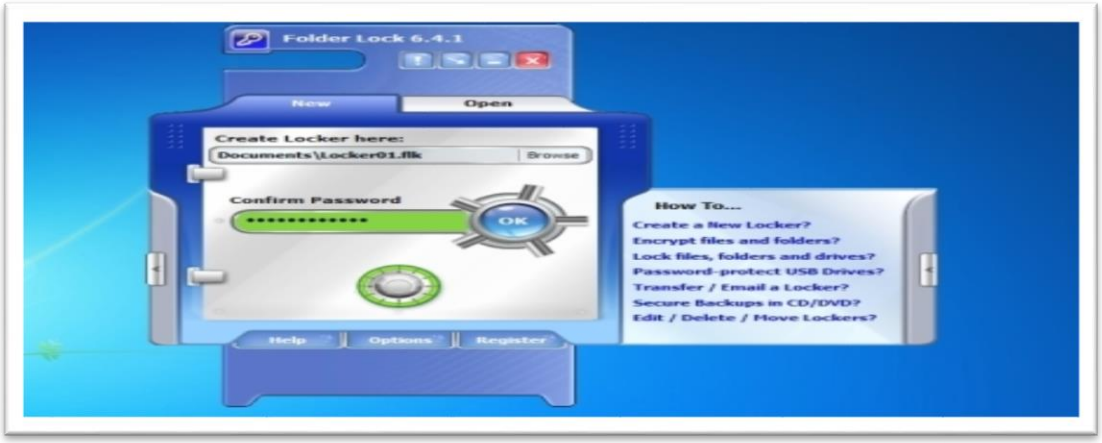
2.5.1 Folder Lock 6.4.1

Folder Lock's security is robust with **256 bit AES encryption**. Decryption was transparent and showed no signs of lag even when playing a 1.7GB video file. The original files are shredded after encryption, leaving no remains on the user's PC, and file integrity was maintained after decryption. To insure a quality password is chosen, a password meter is displayed as a red or green ring around the safe's combination dial, providing immediate feedback regarding the password's strength. Should the user decide not to create a password, the password generator utility will securely accomplish the task.

The virtual keyboard method for inputting passwords prevents key-loggers from capturing key presses from the keyboard; an advanced security precaution utilized by many online banks. Another enhancement is that the "lockers," which contain all your files, cannot be deleted unless the password is known. This prevents others from destroying your data, whether intentional or not. Some right click context menu options lets the user choose whether to "lock" or "encrypt" the files or folders. Both methods ensure the data is well protected. The history cleaning option can remove document history and clear clipboard data, which is an added bonus for this type of software. Stealth mode can completely hide the program from Windows and can be engaged with a user definable hot key.

Folder Lock is an easy-to-use encryption software package. After a short setup of creating a new locker, accessing files and folders is a breeze and there is no noticeable slowdown when accessing your data. The many security features of this software set it apart from all the other encryption programs. One gem is the portability feature of Folder Lock, which allows you to take all your files on-the-go, for example, transferring your locker to a USB drive. After creating a locker, you merely select the portability button, which creates an auto run executable file and moves both to the USB drive. This creates a mini version of Folder Lock, and can be used on any Windows computer without the original software.

This usage limit doesn't give a user enough experimentation time to try all the features. In addition, when right clicking on files or folders, sometimes the folder lock option was available, other times it was not. Lastly, the software was often slow when opening and closing, but was still 100% functional.

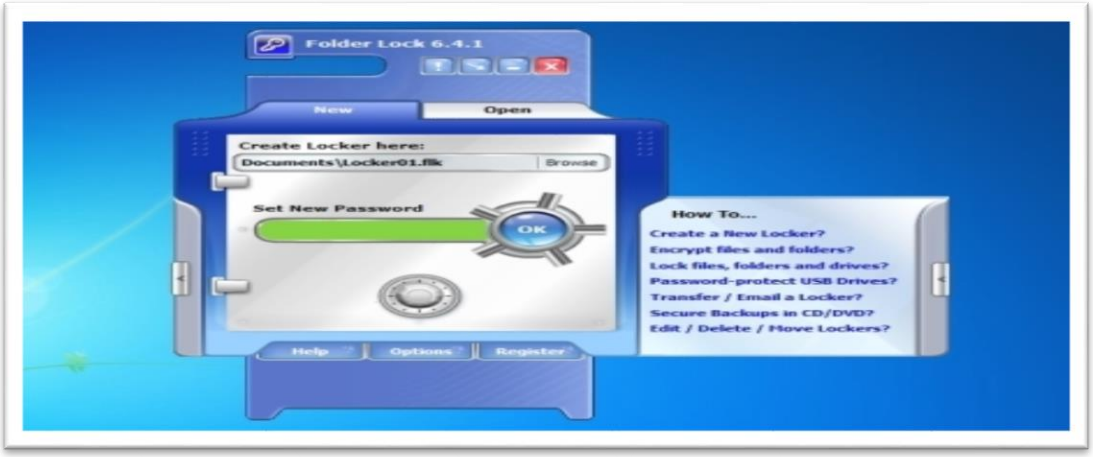


Create Locker here: Documents\Locker01.flk

Folder Lock 6.4.1
New
Open
Browse
Confirm Password
OK
How To...
Create a New Locker?
Encrypt files and folders?
Lock files, folders and drives?
Password-protect USB Drives?
Transfer / Email a Locker?
Secure Backups in CD/DVD?
Edit / Delete / Move Lockers?
Help
Options
Register

Figure 2.1 : Interface of Folder Lock 1

As shown in the figure above the auto protection option can be toggled on or off. If turned on, it can be set from 5 to 360 minutes of idle time to protect your locker.

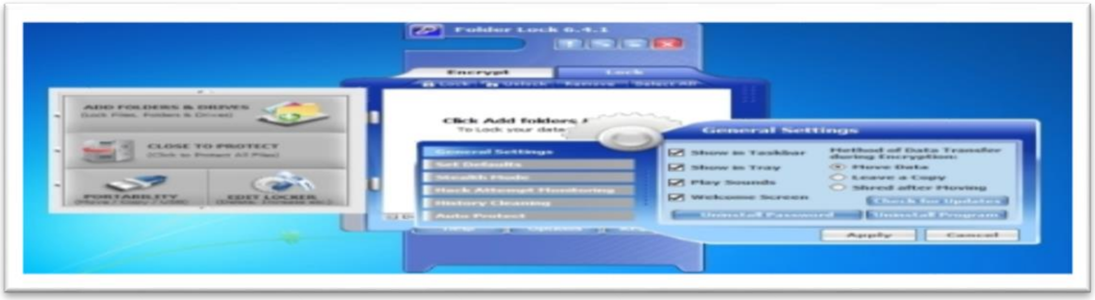


Create Locker here: Documents\Locker01.flk

Folder Lock 6.4.1
New
Open
Browse
Set New Password
OK
How To...
Create a New Locker?
Encrypt files and folders?
Lock files, folders and drives?
Password-protect USB Drives?
Transfer / Email a Locker?
Secure Backups in CD/DVD?
Edit / Delete / Move Lockers?
Help
Options
Register

Figure 2.2: Interface of Folder Lock

As shown in the figure above the hack attempt monitoring feature checks for 5 consecutive fake password attempts. It can be set to either log off the PC or shut it down.



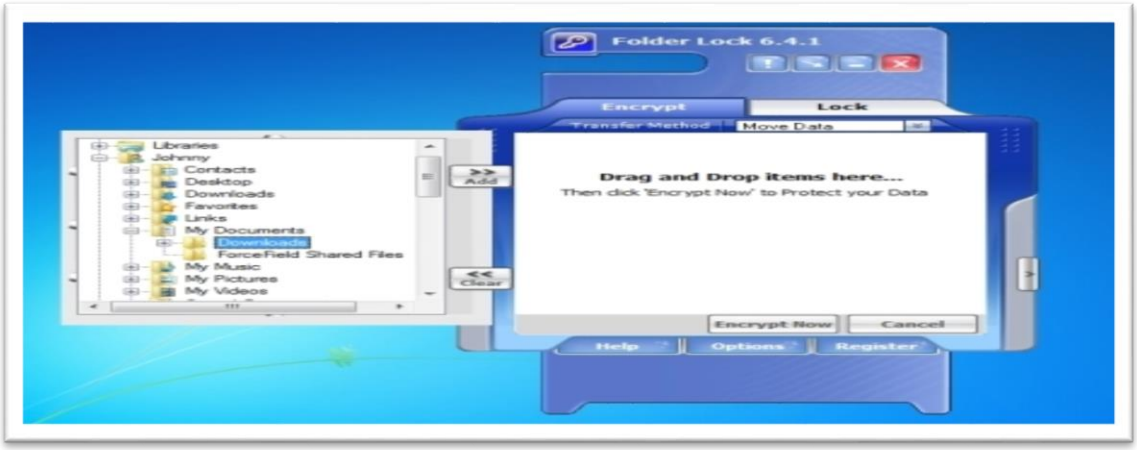
General Settings

☑ Show in Taskbar
☑ Show in Tray
☑ Play Sounds
☑ Welcome Screen

Method of Data Transfer during Encryption:
● Move Data
○ Leave a Copy
○ Shred after Moving

Figure 2.3: Interface of Folder Lock

The figure above shows the history cleaning window will remove unwanted traces from your computer. It's limited to clearing recent document history, files and folder history and clipboard data. Further cleaning requires purchase of a separate product.



Transfer Method: Move Data

Drag and Drop items here...
Then click 'Encrypt Now' to Protect your Data

Figure 2.4 : Interface of Folder Lock

The figure above shows select the files you want to encrypt from the mini explorer window. Drag to the vault, and once you see them there, click "encrypt". This move and scrambles the original files, storing them in the encrypt tab, while the originals are permanently erased.

2.5.2 Advanced Encryption Package Pro 5.3.6

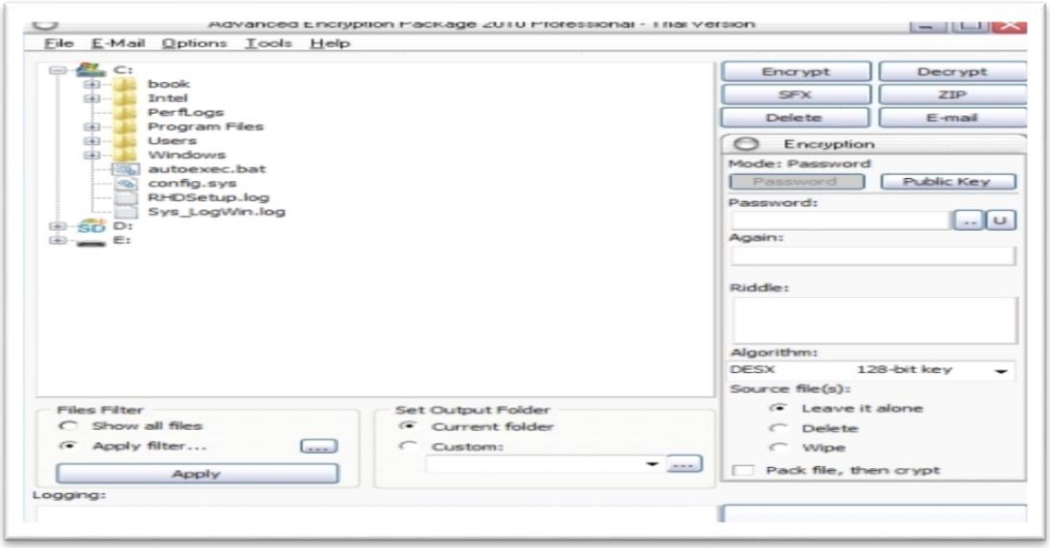
Advanced Encryption Package 2010 Professional sports useful tools in an unthreatening, colorful environment, integrating state-of-the-art encryption. This program is headed in the right direction and it presents the user with many advanced tools, including a simplified, clean interface. This package can meet your security needs; however caution should be exercised when encrypting critical information.

Advanced Encryption Package 2010 Professional really shines with all of its included security components. It has some of the strongest algorithms available of any encryption software, up to 2048-bit. Data destruction is permanent, insured by a variety of 18 shredding algorithms, which is augmented by a manual wipe option. Files are returned to their original state after decryption.

This program also features a PKI Key manager, allowing the creation of both public and private keys instead of passwords. If a manual password is chosen, there is a quality meter to insure its strength. In addition, when setting options for a password, checkboxes can be ticked to disallow for weak or dictionary passwords. There is also a utility to generate passwords up to 15 characters in length. Clicking the two dots next to the password box opens up the virtual keyboard, which can be used if there is the worry of key loggers on your system. In this same window there is a button which generates ultra-long random passwords tailored to the algorithm chosen. From the tools menu, selecting “Clear Computer History” brings up the Privacy Master. Use this module to erase all history from the Windows operating system and internet browsers (Microsoft Internet Explorer and Mozilla Firefox).

Using the password generator utility (which is very slow) allows the user to create a password up to 15 characters in length. The only way to use this password is by either writing it down, or copying it in Windows. This password will then reside in memory until the next copy command, which is not a very secure method considering this software is supposed to protect your privacy. In addition, there is no password strength meter in the text encryption tool or when creating a master password to secure the USB keys. This meter should be available whenever a password is being created throughout the use of the software.

As for usability issues, when creating a new password and saving it to the USB drive as a key, you will discover that clicking "ok" does nothing. After repeated attempts and experimentation, it was found that text must be entered into the description box, which labels your key, which is a necessity. A popup box should be incorporated demanding the user to input a text label before continuing.



Mode: Password

Algorithm: DESX 128-bit key

Source file(s):
● Leave it alone
○ Delete
○ Wipe
□ Pack file, then crypt

Files Filter: ○ Show all files ● Apply filter...

Set Output Folder: ● Current folder ○ Custom:

Figure 2.5: Interface of Advanced Encryption Package Pro 5.3.6

The figure above shows this is the main screen that displays the explorer type interface which is very similar in usage to Windows Explorer

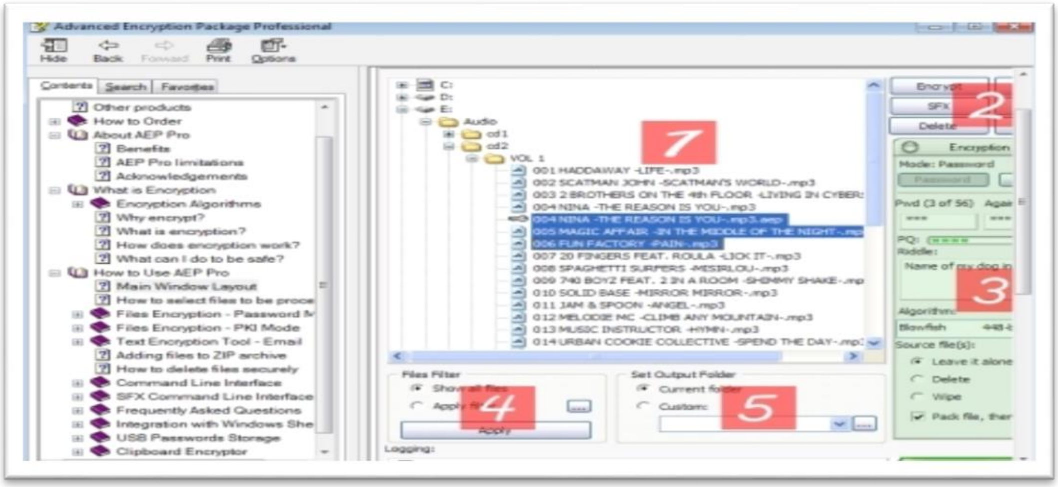
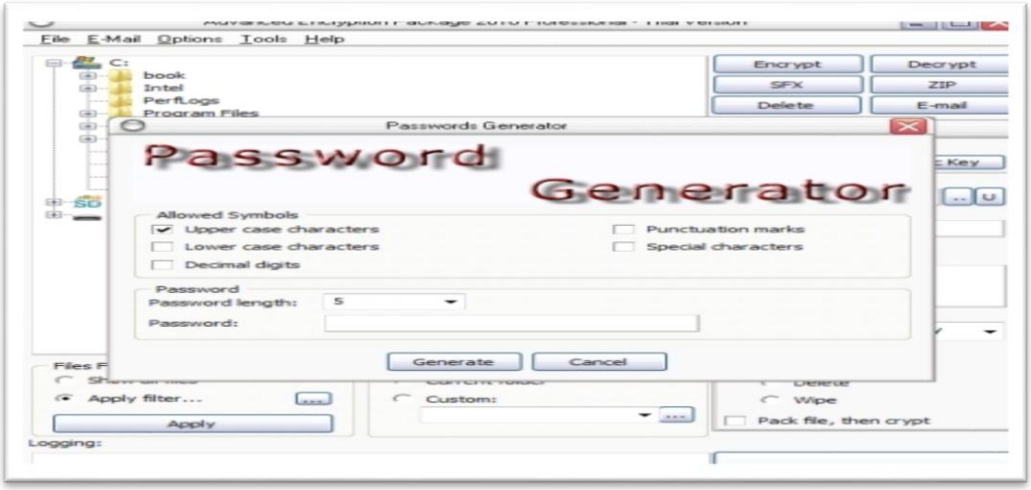

Advanced Encryption Package Professional
Hide
Back
Forward
Print
Options
Contents
Search
Favorites
Other products
How to Order
About AEP Pro
Benefits
AEP Pro limitations
Acknowledgements
What is Encryption
Encryption Algorithms
Why encrypt?
What is encryption?
How does encryption work?
What can I do to be safe?
How to Use AEP Pro
Main Window Layout
How to select files to be proce
Files Encryption - Password M
Files Encryption - PKI Mode
Text Encryption Tool - Email
Adding files to ZIP archive
How to delete files securely
Command Line Interface
SFX Command Line Interface
Frequently Asked Questions
Integration with Windows She
USB Passwords Storage
Clipboard Encryptor
C:
D:
E:
Audio
cd1
cd2
VOL 1
001 HADDAWAY -LIFE-.mp3
002 SCATMAN JOHN -SCATMAN'S WORLD-.mp3
003 2 BROTHERS ON THE 4th FLOOR -LIVING IN CYBER:
004 NINA -THE REASON IS YOU-.mp3
004 NINA -THE REASON IS YOU-.mp3.aep
005 MAGIC AFFAIR -IN THE MIDDLE OF THE NIGHT-.mp
006 FUN FACTORY -PAIN-.mp3
007 20 FINGERS FEAT. ROULA -LICK IT-.mp3
008 SPAGHETTI SURFERS -MESIRLOU-.mp3
009 740 BOYZ FEAT. 2 IN A ROOM -SHIMMY SHAKE-.mp
010 SOLID BASE -MIRROR MIRROR-.mp3
011 JAM & SPOON -ANGEL-.mp3
012 MELODIE MC -CLIMB ANY MOUNTAIN-.mp3
013 MUSIC INSTRUCTOR -HYMN-.mp3
014 URBAN COOKIE COLLECTIVE -SPEND THE DAY-.mp3
Encrypt
SFX
Delete
2
7
Encryption
Mode: Password
Password
Pwd (3 of 56)
Again
PQ:
Riddle:
Name of my dog in
3
Algorithm:
Blowfish 448-
Source file(s):
Leave it alone
Delete
Wipe
Pack file, then
Files Filter
Show all files
Apply filter
Apply
4
Set Output Folder
Current folder
Custom:
5
Logging:

Figure 2.6: Interface of Advanced Encryption Package Pro 5.3.6

The figure above shows this software has a built in Windows help system. Each feature of the program is briefly explained in the "Main Window Layout" section shown here.



Allowed Symbols
☑ Upper case characters
□ Lower case characters
□ Decimal digits
□ Punctuation marks
□ Special characters

Password
Password length: 5
Password:

○ Show all files
● Apply filter...
□ Pack file, then crypt

Figure 2.7: Interface of Advanced Encryption Package Pro 5.3.6

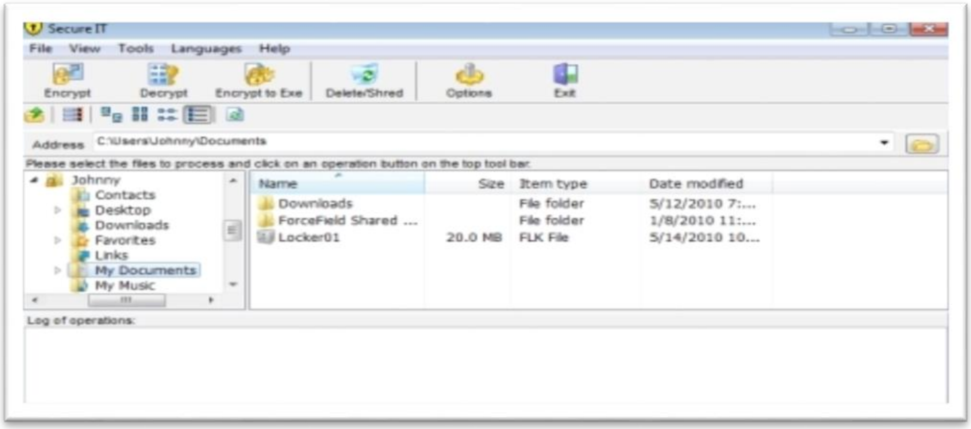
The figure above shows the password generator is a built in tool that will generate strong password up to 15 characters in length, using user selectable symbols.

2.5.3 SecureIT 4.2.0

SecureIT employs two top notch encryption algorithms, **256-bit AES** or 448-bit Blowfish, which are more than sufficient for any user. Files are securely encrypted and original files are forever erased. Integrity is maintained when decrypting and no data loss is evident. There is also a manual wipe feature with the option to delete the file to the recycle bin or to permanently eradicate it.

SecureIT is a dependable, bug-free encryption software utility and accomplishes all of its goals, but is better suited for the encryption of single files. When encrypting a folder, a container type system with a virtual drive is quicker. With this software, to access a file from a group of files, you have to decrypt the entire .sit file. This is nearly identical to the way a zipped file works. Only when the .sit file is decrypted and deflated can you access the data you need. This can take a long time depending on how many files you have. Self-extraction works well, with options to overwrite existing files or open the target folder when done, with a password hint available if one was given when the files were encrypted.

SecureIT's explorer type interface is almost identical to that of Windows, however there was no option to right click and create a new folder. This usability issue can be irritating, as it requires the user to open Windows explorer to create a folder. Upon refreshing the SecureIT window, the new folder will then appear. All in all, SecureIT is a stable, capable and reliable program with strong encryption and many shredding options that should prove suitable for any user.



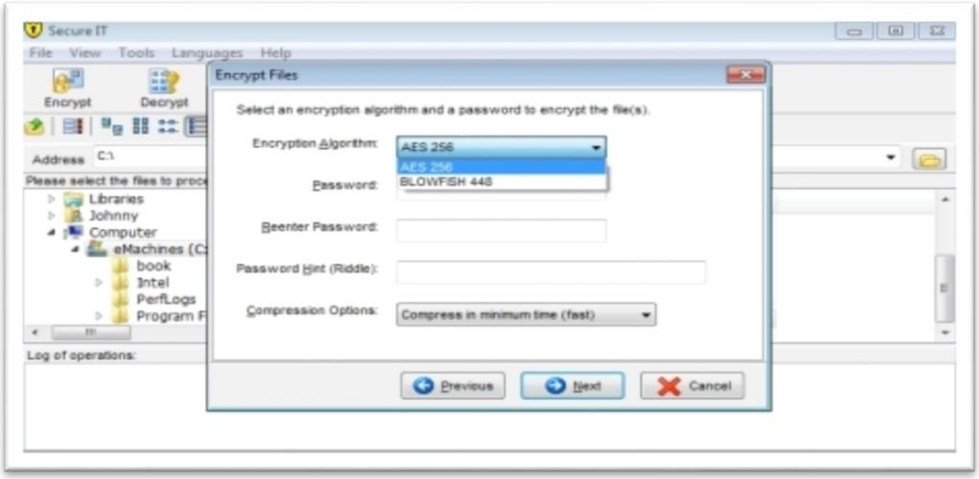
Address C:\Users\Johnny\Documents

Please select the files to process and click on an operation button on the top tool bar.

Name	Size	Item type	Date modified
Downloads		File folder	5/12/2010 7:...
ForceField Shared ...		File folder	1/8/2010 11:...
Locker01	20.0 MB	FLK File	5/14/2010 10...

Figure 2.8 : Interface of SecureIT 4.2.0

The figure above shows this is the main screen of the program. The explorer type window provides four action buttons to choose from after selecting a file or folder.



Select an encryption algorithm and a password to encrypt the file(s).

Encryption Algorithm: AES 256

AES 256
BLOWFISH 448

Password:

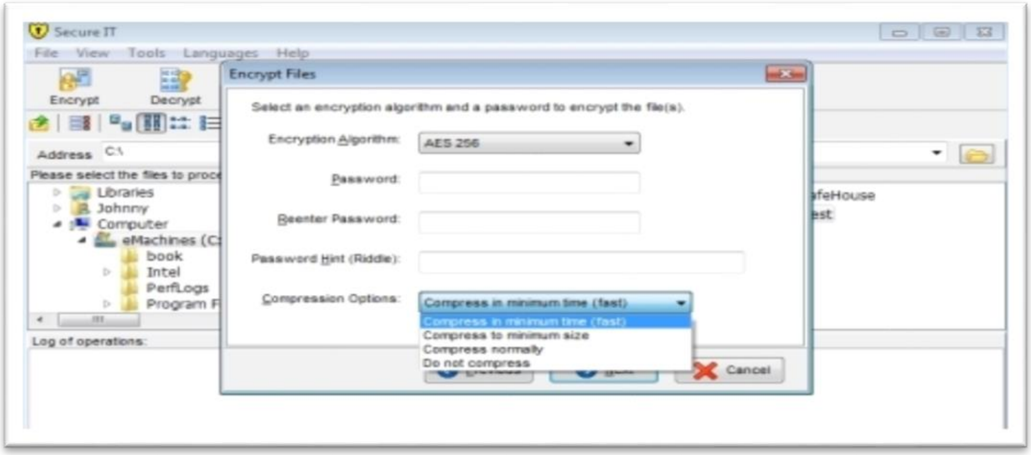
Reenter Password:

Password Hint (Riddle):

Compression Options: Compress in minimum time (fast)

Figure 2.9 : Interface of SecureIT 4.2.0

The figure above shows encrypting a file is simple in this program. Choose an encryption algorithm, a password, a password hint and whether or not you want compression.



Select an encryption algorithm and a password to encrypt the file(s).

Encryption Algorithm: AES 256

Password:

Reenter Password:

Password Hint (Riddle):

Compression Options: Compress in minimum time (fast)

- Compress in minimum time (fast)
- Compress to minimum size
- Compress normally
- Do not compress

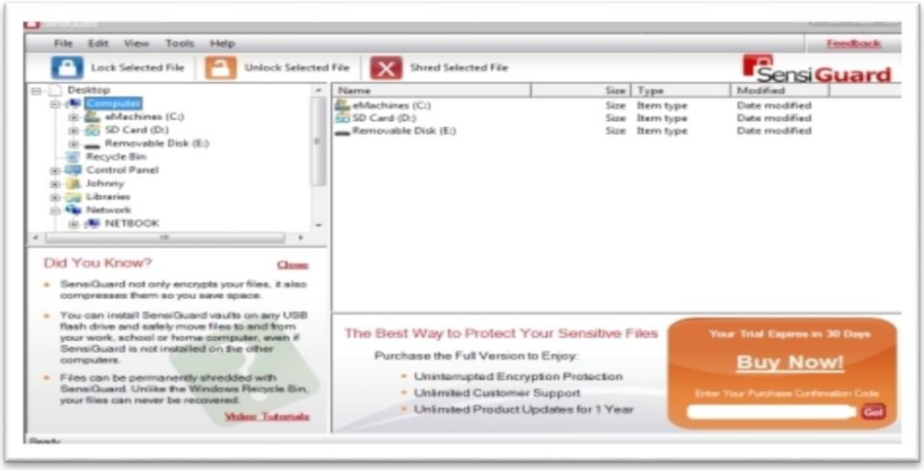
Figure 2.10: Interface of SecureIT 4.2.0

The figure above shows the "Encrypt to Exe" function provides the same options as regular encrypting. The only difference is that the created archive can be emailed or shared with others as long as they know the password.

2.5.4 SensiGuard 3.1

This program utilizes **256-bit AES encryption**; however, this is not evident anywhere in the encryption software, but the SensiGuards' website states that it is so. Data reliability is evident after decrypting and the original files are securely shredded. There is also a manual wipe option to erase obsolete data. SensiGuard kept their software simple with only one option, which is the ability to change the users' password. Mandatory features were excluded. For example, there are no buttons to go up one level in the directory window; you must click on the parent folder in the left hand pane to go back. This is not efficient and quickly becomes irritating. Also, double clicking on a folder will open that folder, but then when double clicking on a file to view it, the hourglass symbol appears and nothing happens.

To open the selected file, you must right click and select “open.” Only during the installation of the software are personal security questions asked with fields to enter answers; this cannot be changed later. This safeguard is provided in case the user forgets his password when decrypting. When clicking on “I forgot my password” during file decryption, a dialog box flashed and then disappeared. This function seems to work at random.SensiGuard's encryption software works reliably and is very basic, yet slow when manipulating large files. The ability to create USB “vaults” is a plus. SensiGuard is a good option for new computer users as long as they are prepared to deal with the minor inconveniences and bugs in the software



Name	Size	Type	Modified
eMachines (C:)	Size	Item type	Date modified
SD Card (D:)	Size	Item type	Date modified
Removable Disk (E:)	Size	Item type	Date modified

Did You Know?

- SensiGuard not only encrypts your files, it also compresses them so you save space.
- You can install SensiGuard vaults on any USB flash drive and safely move files to and from your work, school or home computer, even if SensiGuard is not installed on the other computers.
- Files can be permanently shredded with SensiGuard. Unlike the Windows Recycle Bin, your files can never be recovered.

The Best Way to Protect Your Sensitive Files

Purchase the Full Version to Enjoy:

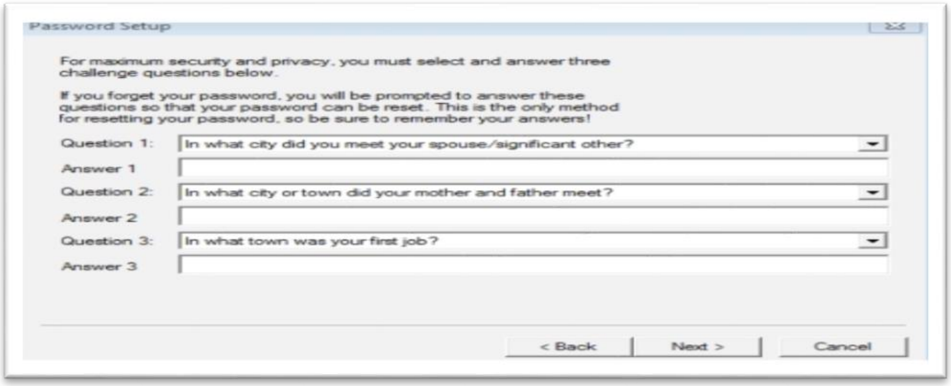
- Uninterrupted Encryption Protection
- Unlimited Customer Support
- Unlimited Product Updates for 1 Year

Your Trial Expires in 30 Days

Enter Your Purchase Confirmation Code

Figure 2.11 : Interface of SensiGuard 3.1

The figure above shows the main screen of the basic explorer-type interface. The three main features of the software are "Lock," "Unlock" and "Shred," which are indicated by large colorful buttons.


Password Setup

For maximum security and privacy, you must select and answer three challenge questions below.

If you forget your password, you will be prompted to answer these questions so that your password can be reset. This is the only method for resetting your password, so be sure to remember your answers!

Question 1: In what city did you meet your spouse/significant other?

Answer 1

Question 2: In what city or town did your mother and father meet?

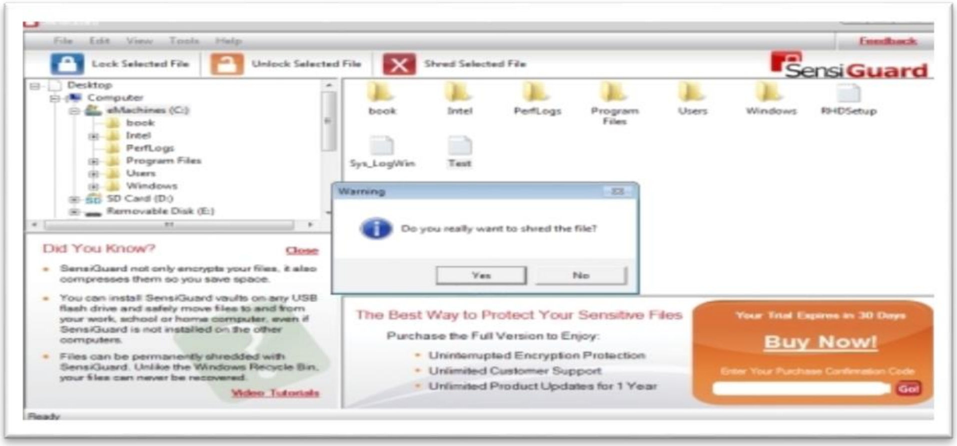
Answer 2

Question 3: In what town was your first job?

Answer 3

Figure 2.12 : Interface of SensiGuard 3.1

Figure above shows this setup screen displays a variety of pointed questions. Answer the questions based on your personal history. Should you ever forget your password, merely answer the questions correctly for access.



Do you really want to shred the file?

Did You Know?

- SensiGuard not only encrypts your files, it also compresses them so you save space.
- You can install SensiGuard vaults on any USB flash drive and safely move files to and from your work, school or home computer, even if SensiGuard is not installed on the other computers.
- Files can be permanently shredded with SensiGuard. Unlike the Windows Recycle Bin, your files can never be recovered.

The Best Way to Protect Your Sensitive Files

Purchase the Full Version to Enjoy:

- Uninterrupted Encryption Protection
- Unlimited Customer Support
- Unlimited Product Updates for 1 Year

Figure 2.13 : Interface of SensiGuard 3.1

Figure above shows the warning box that appears when you choose to shred files or folders. This is a nice preventative measure to avoid accidentally erasing needed data.

2.6 Comparison between existing system and the prototype.

Table 2.2: Comparison between existing system and the prototype

Description	**Folder Lock**	**Advanced Encryption Package Pro 5.3.6**	**SecureIT 4.2.0**	**SensiGuard 3.1**	**Fskkp Data Transmission Using AES Encryption via SMS and Email**
Purpose	Protect your sensitive files and folders with unique encryption software.	Encrypt data using 1 of 17 different algorithms.	smartly designed encryption software.	employs 256-bit encryption to "lock" your files.	To protect the confidential and integrity of the file by using AES algorithm encryption.
System Requirement	Windows XP / Vista / 7 (all editions) - At least 512 MB of available RAM (1 GB recommended)	Microsoft Windows 2000 / XP / Vista (32- and 64-bit) / 7 (32- and 64-bit) -at least 10 MB of free hard drive space	Microsoft Windows 2000 / XP / Vista / 7 -At least 6.5 Mb free on hard drive space	WinXP, WinVista, WinVista x64, Win7 -At least 10 MB free hard drive space	Windows XP / Vista / 7 (all editions) - At least 512 MB of available RAM (1 GB recommended) -Internet Access -GSM Modem
Advantages	Folder Lock has lots of useful	The software offers a complete	SecureIT includes all the necessary	The software uses strong encryption in	The prototype can send the Password or

	features and quickly encrypts all the user top secret documents.	security package with a variety of military grade encryption algorithms, text tools, and computer cleaning options.	security elements while keeping the software intuitive and manageable.	a clean interface that is very easy to use.	any note using SMS form. User also can send encrypted or decrypted file using email service that is provided by the prototype.
Limitation	The software was sometimes slow to open and close, and right click options weren't always available.	. Advanced Encryption Package Pro has minor usability issues and was unstable on occasion.	The application interface doesn't mimic Windows explorer, and accessing files from an archive can be slow.	SensiGuard's interface needs some minor work and the software has a few bugs.	The software cannot encrypt folder. Its need to be compressed first by third party software and the email and sms time execute depend on the connection of the Internet speed.
Availability of user	Large scale of user.	Large scale of user.	Large scale of user	Large scale of user	Large scale of user

2.7 Technique

On this part, we will review about the techniques that we used to develop the system. Which are a stand-alone windows application, GSM modem, Advanced Encryption Standard (AES) Method and programming language.

2.7.1 Stand Alone Windows Application

Stand-alone software is a software application that does not come rushed within another software application, and does not require another software package to run. Stand-alone software is software installed on your computer. System that capable to operate without other programs, libraries, computers, hardware, network and etc.

2.7.2 GSM Modem

In cellular service there are two main competing network technologies, it's Global System for Mobile Communications (GSM) and Code Division Multiple Access (CDMA). Since its beginning in the '80s, GSM telephone system was developed using cell concept for the network topology. Each cell corresponds to a specific antenna (base station), placed on towers or tall buildings. The GSM standard has been an advantage to both consumers, who may benefit from the ability to roam and switch carriers without replacing phones, and also to network operators.

GSM also has low-cost implementation of the short message service (SMS), also called text messaging, which has since been supported on other mobile phone standards as well. Because of huge coverage of distance, the GSM infrastructure can be an alternative to transmit or receive data from or to a device like sensor, actuator and complex device near or remotely. Compared to analog transmission systems, GSM system provides narrowest bandwidth for a channel, through the use of voice compression algorithm; improving the quality of transmission

2.7.3 Advanced Encryption Standard

AES is a repeated symmetric block cipher, which means that:

- AES works by repeating the same defined steps multiple times.
- AES is a secret key encryption algorithm.
- AES operates on a fixed number of bytes

AES as well as most encryption algorithms is reversible. This means that almost the same steps are performed to complete both encryption and decryption in reverse order. The AES algorithm operates on bytes, which makes it simpler to implement and explain. This key is expanded into individual sub keys, a sub keys for each operation round. This process is called

KEY EXPANSION, which is described at the end of this document. As mentioned before AES is an iterated block cipher. All that means is that the same operations are performed many times on a fixed number of bytes. These operations can easily be broken down to the following functions:

Technique	Function
Addround Key	Each byte of the state is combined with the round key using a bit-wise operation**.**
Byte Sub	A non-linear substitution step where each byte is replaced with another according to a lookup table
Shift Row	A transposition step where each row of the state is shifted cyclically a certain number of steps
Mix Column	A mixing operation which operates on the columns of the state, combining the four bytes in each column.

	An iteration of the above steps is called a round. The amount of rounds of the algorithm depends on the key size.

Table 2.3 : Table of round

Key Size (bytes)	Block Size (bytes)	Round
16	16	10
24	16	12
32	16	14

The table show the only exception being that in the last round the **Mix Column** step is not performed, to make the algorithm reversible during decryption.

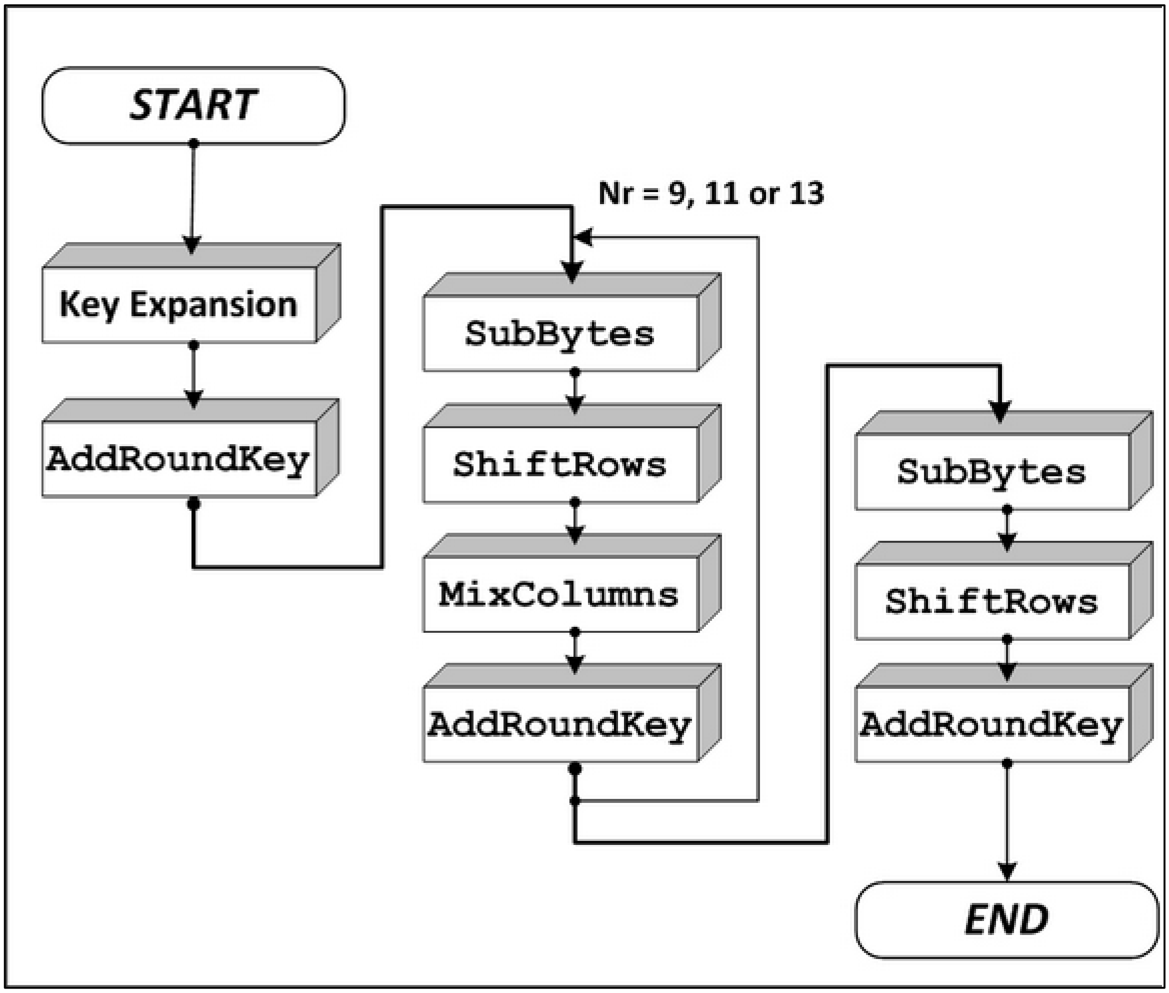


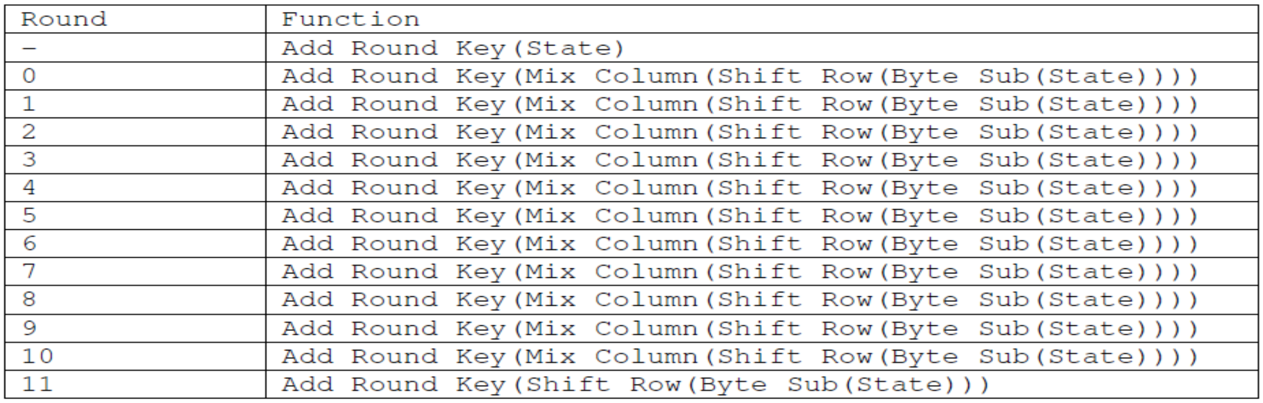
Figure 2.14: AES Forward Cipher Flow Graph

2.7.3.1 Encryption

Table 2.4: AES encryption cipher using 16 byte key.

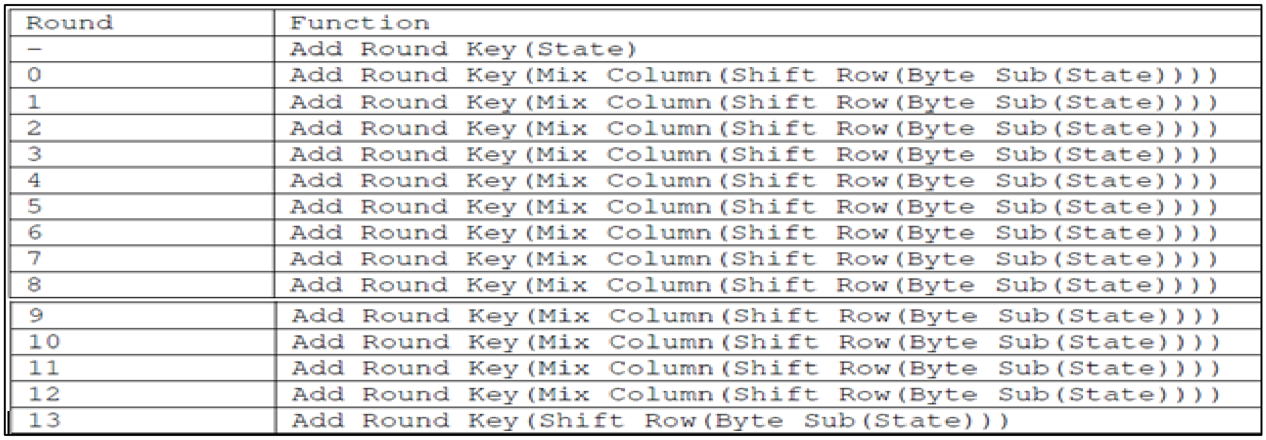
Round	Function
–	Add Round Key(State)
0	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
1	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
2	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
3	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
4	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
5	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
6	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
7	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
8	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
9	Add Round Key(Shift Row(Byte Sub(State)))

Table 2.5: AES encryption cipher using 24 byte key.



Round	Function
–	Add Round Key(State)
0	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
1	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
2	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
3	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
4	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
5	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
6	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
7	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
8	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
9	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
10	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
11	Add Round Key(Shift Row(Byte Sub(State)))

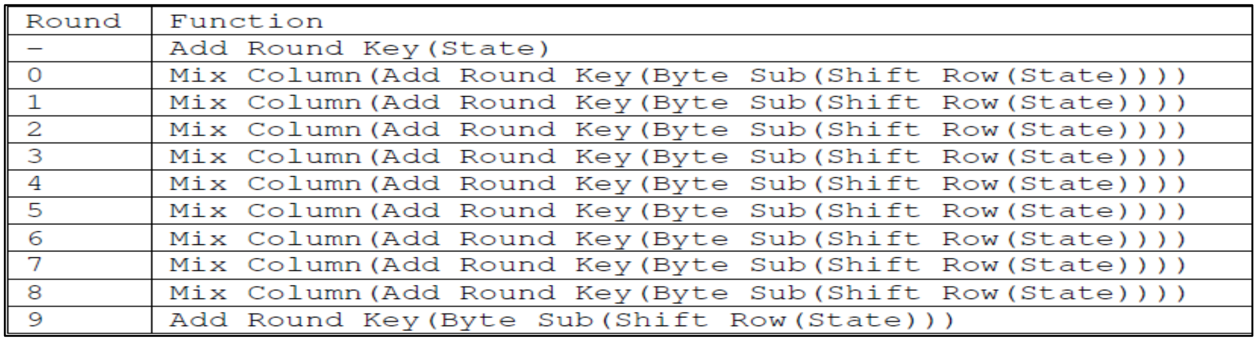
Table 2.6: AES encryption cipher using 32 byte key.



Round	Function
–	Add Round Key(State)
0	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
1	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
2	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
3	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
4	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
5	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
6	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
7	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
8	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
9	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
10	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
11	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
12	Add Round Key(Mix Column(Shift Row(Byte Sub(State))))
13	Add Round Key(Shift Row(Byte Sub(State)))

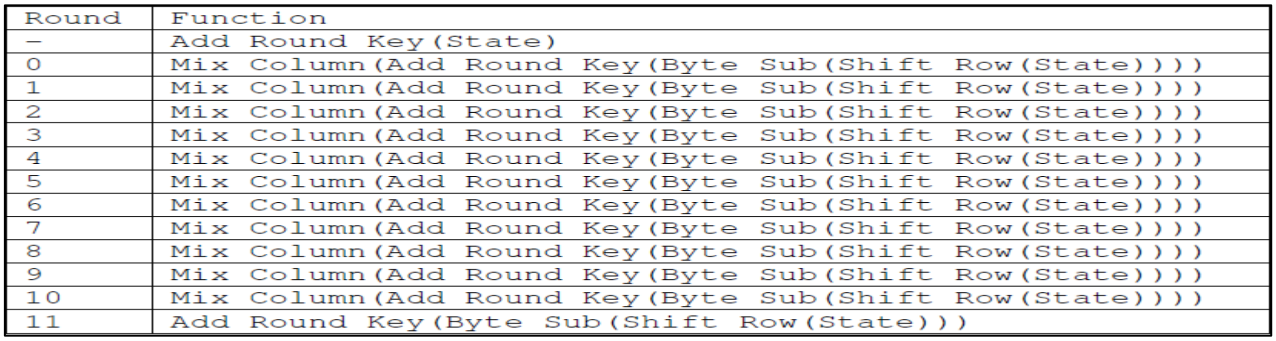
2.7.3.2 Decryption

Table 2.7:AES decryption cipher using 16 byte key.



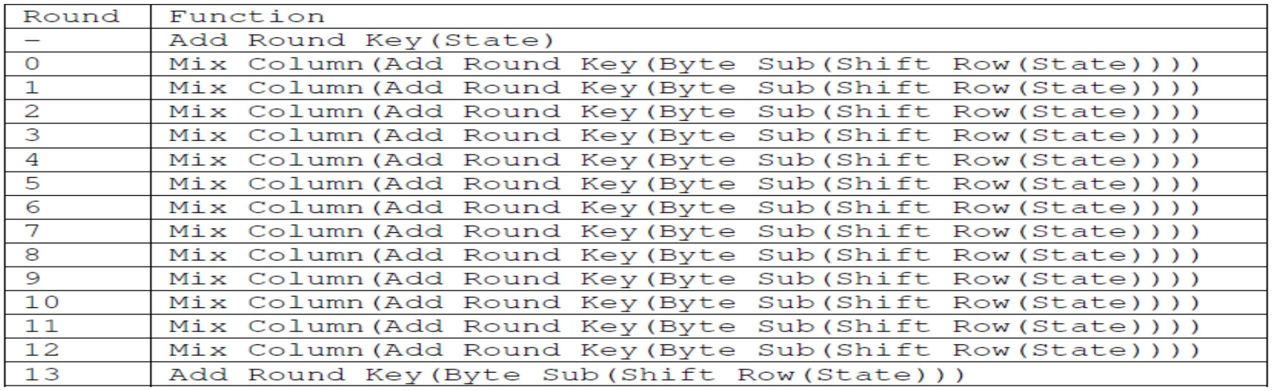
Round	Function
–	Add Round Key(State)
0	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
1	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
2	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
3	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
4	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
5	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
6	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
7	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
8	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
9	Add Round Key(Byte Sub(Shift Row(State)))

Table 2.8:AES decryption cipher using 24 byte key.



Round	Function
–	Add Round Key(State)
0	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
1	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
2	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
3	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
4	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
5	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
6	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
7	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
8	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
9	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
10	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
11	Add Round Key(Byte Sub(Shift Row(State)))

Table 2.9:AES decryption cipher using 32 byte key.



Round	Function
–	Add Round Key(State)
0	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
1	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
2	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
3	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
4	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
5	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
6	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
7	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
8	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
9	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
10	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
11	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
12	Mix Column(Add Round Key(Byte Sub(Shift Row(State))))
13	Add Round Key(Byte Sub(Shift Row(State)))

2.7.3.3 AES Cipher Function

I. AddRound Key

Each of the 16 bytes of the state is XORed against each of the 16 bytes of a portion of the expanded key for the current round. The Expanded Key bytes are never reused. So once the first 16 bytes are XORed against the first 16 bytes of the expanded key then the expanded key bytes 1-16 are never used again. The next time the Add Round Key function is called bytes 17-32 are XORed against the state.

Table 2.10: The first time Add Round Key gets executed

State	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR
Exp Key	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

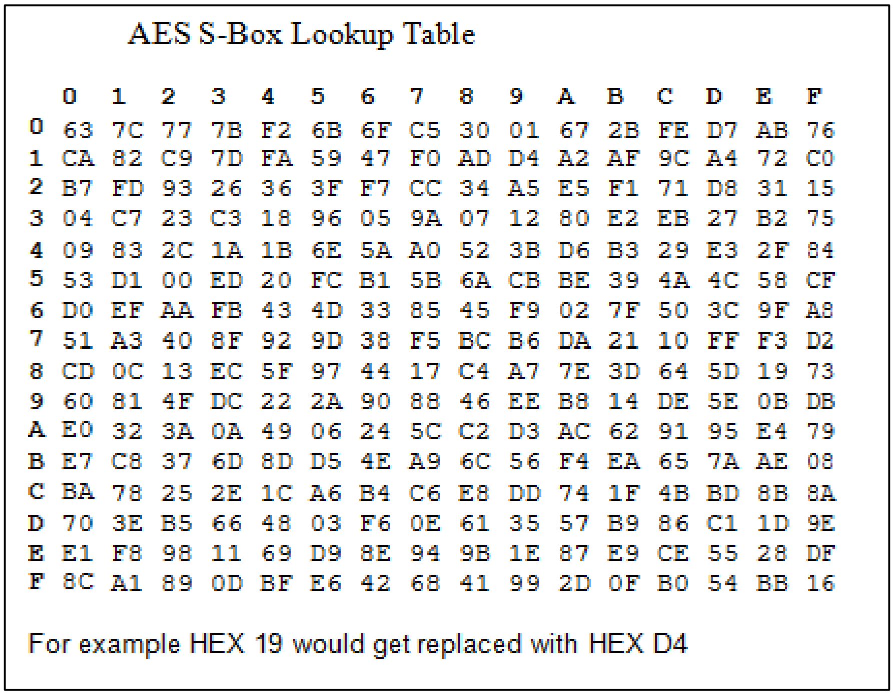
Table 2.11 The second time Add Round Key is executed

State	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR	XOR
Exp Key	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32

And so on for each round of execution.

II. Sub-Byte

- During encryption each value of the state is replaced with the corresponding SBOX value



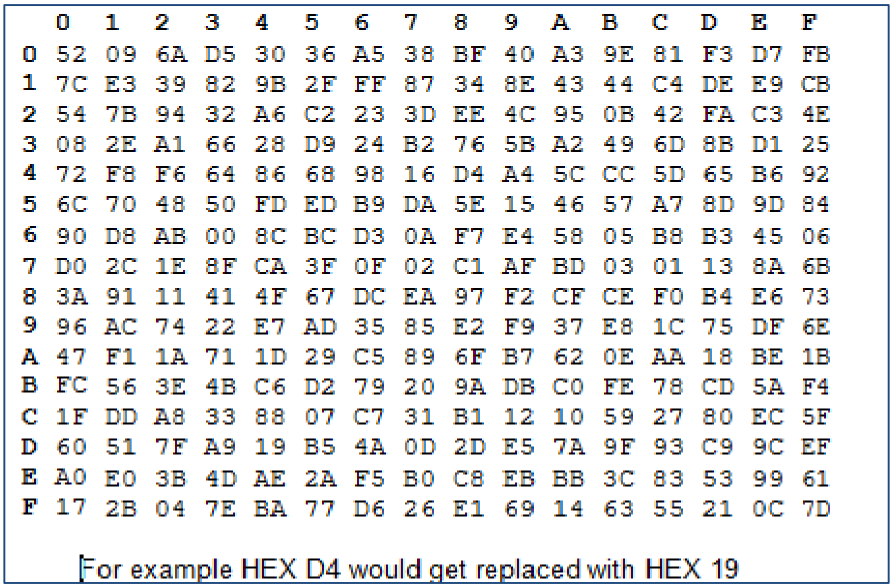
AES S-Box Lookup Table

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

For example HEX 19 would get replaced with HEX D4

Figure 2.15

- During decryption each value in the state is replaced with the corresponding inverse of the SBOX



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

For example HEX D4 would get replaced with HEX 19

Figure 2.16

III. Shift-Row

Arranges the state in a matrix and then performs a circular shift for each row. This is not a bit wise shift. The circular shift just moves each byte one space over. A byte that was in the second position may end up in the third position after the shift. The circular part of it specifies that the byte in the last position shifted one space will end up in the first position in the same row.

In Detail:

The state is arranged in a 4x4 matrix (square)

The confusing part is that the matrix is formed vertically but shifted horizontally. So the first 4 bytes of the state will form the first bytes in each row

So bytes 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

Each row is then moved over (shifted) 1, 2 or 3 spaces over to the right, depending on the row of the state. First row is never shifted

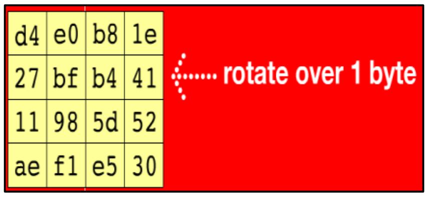
Row1 shift 0

Row2 shift 1

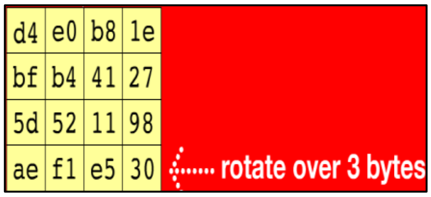
Row3 shift 2

Row4 shift 3

Row 2 shift 1 position Row 4 shift 3 position

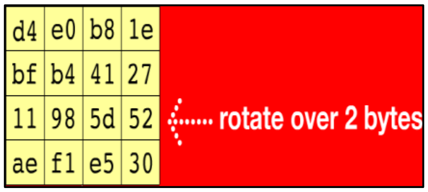


d4	e0	b8	1e
27	bf	b4	41
11	98	5d	52
ae	f1	e5	30



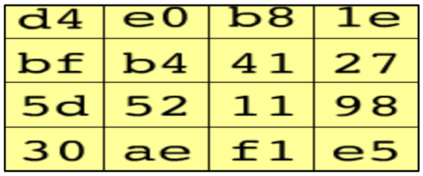
d4	e0	b8	1e
bf	b4	41	27
5d	52	11	98
ae	f1	e5	30

Row 3 shift 2 position



d4	e0	b8	1e
bf	b4	41	27
11	98	5d	52
ae	f1	e5	30

Result after shift row



d4	e0	b8	1e
bf	b4	41	27
5d	52	11	98
30	ae	f1	e5

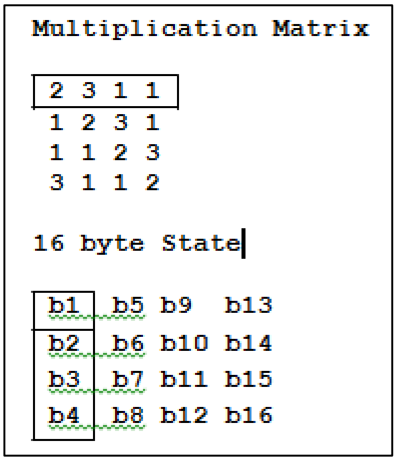
IV. Mix Column

This is perhaps the hardest step to both understand and explain. There are two parts to this step. The first will explain which parts of the state are multiplied against which parts of the matrix. The second will explain how this multiplication is implemented over what's called a Galois Field

- Matrix Multiplication

The state is arranged into a 4 row table (as described in the Shift Row function).

The multiplication is performed one column at a time (4 bytes). Each value in the column is eventually multiplied against every value of the matrix (16 total multiplications). The results of these multiplications are XORed together to produce only 4 result bytes for the next state. Therefore 4 bytes input, 16 multiplications 12 XORs and 4 bytes output. The multiplication is performed one matrix row at a time against each value of a state column



```
Multiplication Matrix

 2  3  1  1
 1  2  3  1
 1  1  2  3
 3  1  1  2

16 byte State

 b1  b5 b9  b13
 b2  b6 b10 b14
 b3  b7 b11 b15
 b4  b8 b12 b16
```

Figure 2.17

The first result byte is calculated by multiplying 4 values of the state column against 4 values of the first row of the matrix. The result of each multiplication is then XORed to produce 1 Byte.

b1 = (b1 * 2) XOR (b2*3) XOR (b3*1) XOR (b4*1)

The second result byte is calculated by multiplying the same 4 values of the state column against 4 values of the second row of the matrix. The result of each multiplication is then XORed to produce 1 Byte.

b2 = (b1 * 1) XOR (b2*2) XOR (b3*3) XOR (b4*1)

The third result byte is calculated by multiplying the same 4 values of the state column against 4 values of the third row of the matrix. The result of each multiplication is then XORed to produce 1 Byte.

b3 = (b1 * 1) XOR (b2*1) XOR (b3*2) XOR (b4*3)

The fourth result byte is calculated by multiplying the same 4 values of the state column against 4 values of the fourth row of the matrix. The result of each multiplication is then XORed to produce 1 Byte

b4 = (b1 * 3) XOR (b2*1) XOR (b3*1) XOR (b4*2)

This procedure is repeated again with the next column of the state, until there are no more state columns.

Putting it all together:

The first column will include state bytes 1-4 and will be multiplied against the matrix in the following manner:

b1 = (b1 * 2) XOR (b2*3) XOR (b3*1) XOR (b4*1)

b2 = (b1 * 1) XOR (b2*2) XOR (b3*3) XOR (b4*1)

b3 = (b1 * 1) XOR (b2*1) XOR (b3*2) XOR (b4*3)

b4 = (b1 * 3) XOR (b2*1) XOR (b3*1) XOR (b4*2)

(b1= specifies the first byte of the state)

The second column will be multiplied against the second row of the matrix in the following manner.

b5 = (b5 * 2) XOR (b6*3) XOR (b7*1) XOR (b8*1)

b6 = (b5 * 1) XOR (b6*2) XOR (b7*3) XOR (b8*1)

b7 = (b5 * 1) XOR (b6*1) XOR (b7*2) XOR (b8*3)

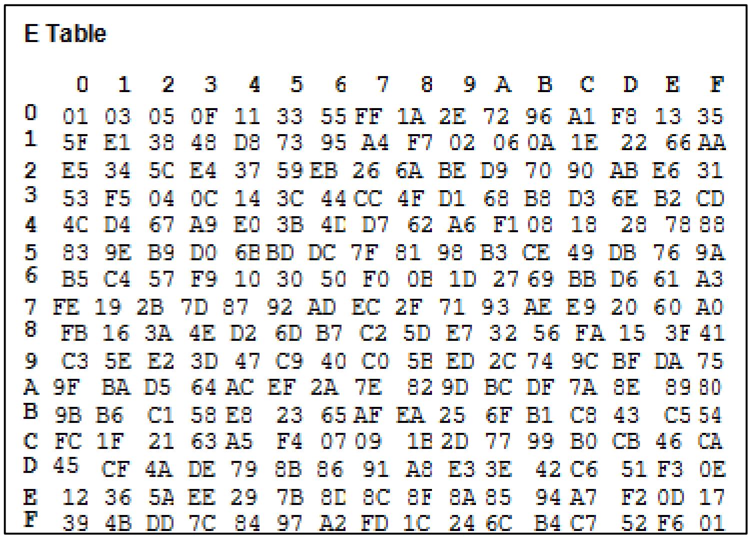
b8 = (b5 * 3) XOR (b6*1) XOR (b7*1) XOR (b8*2)

And so on until all columns of the state are exhausted.

V. Galois Multiplication

The multiplication mentioned above is performed over a Galois Field. The mathematics behind this is beyond the scope of this paper. This section will instead concentrate on the implementation of the multiplication which can be done quite easily with the use of the following two tables in (HEX).

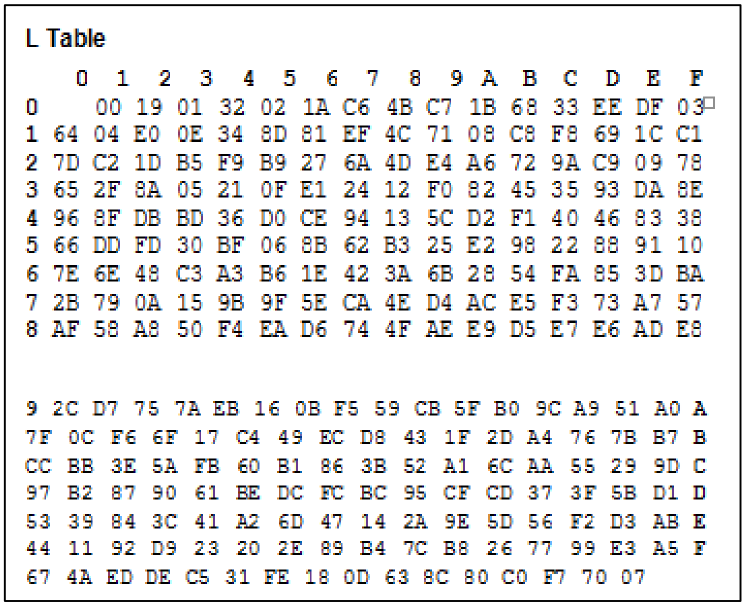
Table 2.12 : E Table



E Table

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	01	03	05	0F	11	33	55	FF	1A	2E	72	96	A1	F8	13	35
1	5F	E1	38	48	D8	73	95	A4	F7	02	06	0A	1E	22	66	AA
2	E5	34	5C	E4	37	59	EB	26	6A	BE	D9	70	90	AB	E6	31
3	53	F5	04	0C	14	3C	44	CC	4F	D1	68	B8	D3	6E	B2	CD
4	4C	D4	67	A9	E0	3B	4D	D7	62	A6	F1	08	18	28	78	88
5	83	9E	B9	D0	6B	BD	DC	7F	81	98	B3	CE	49	DB	76	9A
6	B5	C4	57	F9	10	30	50	F0	0B	1D	27	69	BB	D6	61	A3
7	FE	19	2B	7D	87	92	AD	EC	2F	71	93	AE	E9	20	60	A0
8	FB	16	3A	4E	D2	6D	B7	C2	5D	E7	32	56	FA	15	3F	41
9	C3	5E	E2	3D	47	C9	40	C0	5B	ED	2C	74	9C	BF	DA	75
A	9F	BA	D5	64	AC	EF	2A	7E	82	9D	BC	DF	7A	8E	89	80
B	9B	B6	C1	58	E8	23	65	AF	EA	25	6F	B1	C8	43	C5	54
C	FC	1F	21	63	A5	F4	07	09	1B	2D	77	99	B0	CB	46	CA
D	45	CF	4A	DE	79	8B	86	91	A8	E3	3E	42	C6	51	F3	0E
E	12	36	5A	EE	29	7B	8D	8C	8F	8A	85	94	A7	F2	0D	17
F	39	4B	DD	7C	84	97	A2	FD	1C	24	6C	B4	C7	52	F6	01

Table 2.13 : L table



L Table

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		00	19	01	32	02	1A	C6	4B	C7	1B	68	33	EE	DF	03□
1	64	04	E0	0E	34	8D	81	EF	4C	71	08	C8	F8	69	1C	C1
2	7D	C2	1D	B5	F9	B9	27	6A	4D	E4	A6	72	9A	C9	09	78
3	65	2F	8A	05	21	0F	E1	24	12	F0	82	45	35	93	DA	8E
4	96	8F	DB	BD	36	D0	CE	94	13	5C	D2	F1	40	46	83	38
5	66	DD	FD	30	BF	06	8B	62	B3	25	E2	98	22	88	91	10
6	7E	6E	48	C3	A3	B6	1E	42	3A	6B	28	54	FA	85	3D	BA
7	2B	79	0A	15	9B	9F	5E	CA	4E	D4	AC	E5	F3	73	A7	57
8	AF	58	A8	50	F4	EA	D6	74	4F	AE	E9	D5	E7	E6	AD	E8
9	2C	D7	75	7A	EB	16	0B	F5	59	CB	5F	B0	9C	A9	51	A0
A	7F	0C	F6	6F	17	C4	49	EC	D8	43	1F	2D	A4	76	7B	B7
B	CC	BB	3E	5A	FB	60	B1	86	3B	52	A1	6C	AA	55	29	9D
C	97	B2	87	90	61	BE	DC	FC	BC	95	CF	CD	37	3F	5B	D1
D	53	39	84	3C	41	A2	6D	47	14	2A	9E	5D	56	F2	D3	AB
E	44	11	92	D9	23	20	2E	89	B4	7C	B8	26	77	99	E3	A5
F	67	4A	ED	DE	C5	31	FE	18	0D	63	8C	80	C0	F7	70	07

The result of the multiplication is simply the result of a lookup of the L table, followed by the addition of the results, followed by a lookup to the E table. The addition is a regular mathematical addition represented by +, not a bitwise AND.

All numbers being multiplied using the Mix Column function converted to HEX will form a maximum of 2 digit Hex number. We use the first digit in the number on the vertical index and the second number on the horizontal index.

If the value being multiplied is composed of only one digit we use 0 on the vertical index. For example if the two Hex values being multiplied are AF * 8 we first lookup L (AF) index which returns B7 and then lookup L (08) which returns 4B.

Once the L table lookup is complete we can then simply add the numbers together. The only trick being that if the addition result is greater then FF we subtract FF from the addition result.

For example AF+B7= 166.

Because 166 > FF, we perform: 166-FF which gives us 67.

The last step is to look up the addition result on the E table. Again we take the first digit to look up the vertical index and the second digit to look up the horizontal index.

For example E(67)=F0

Therefore the result of multiplying AF * 8 over a Galois Field is F0

Two last exceptions are that:

- Any number multiplied by one is equal to its self and does not need to go through the above procedure. For example: FF * 1 = FF
- Any number multiplied by zero equals zero

VI. Mix Column Example

During Encryption

Input = D4 BF 5D 30

Output(0) = (D4 * 2) XOR (BF*3) XOR (5D*1) XOR (30*1)

= E(L(D4) + L(02)) XOR E(L(BF) + L(03)) XOR 5D XOR 30

= E(41 + 19) XOR E(9D + 01) XOR 5D XOR 30

= E(5A) XOR E(9E) XOR 5D XOR 30

= B3 XOR DA XOR 5D XOR 30

= **04**

Output(1) = (D4 * 1) XOR (BF*2) XOR (5D*3) XOR (30*1)

= D4 XOR E(L(BF)+L(02)) XOR E(L(5D)+L(03)) XOR 30

= D4 XOR E(9D+19) XOR E(88+01) XOR 30

= D4 XOR E(B6) XOR E(89) XOR 30

= D4 XOR 65 XOR E7 XOR 30

= **66**

Output(2) = (D4 * 1) XOR (BF*1) XOR (5D*2) XOR (30*3)

= D4 XOR BF XOR E(L(5D)+L(02)) XOR E(L(30)+L(03))

= D4 XOR BF XOR E(88+19) XOR E(65+01)

= D4 XOR BF XOR E(A1) XOR E(66)

= D4 XOR BF XOR BA XOR 50

= **81**

Output(3) = (D4 * 3) XOR (BF*1) XOR (5D*1) XOR (30*2)

= E(L(D4)+L(3)) XOR BF XOR 5D XOR E(L(30)+L(02))

= E(41+01) XOR BF XOR 5D XOR E(65+19)

= E(42) XOR BF XOR 5D XOR E(7E)

= 67 XOR BF XOR 5D XOR 60

= **E5**

During Decryption

Input **04 66 81 E5**

Output(0) = (04 * 0E) XOR (66*0B) XOR (81*0D) XOR (E5*09)

= E(L(04)+L(0E)) XOR E(L(66)+L(0B)) XOR E(L(81)+L(0D)) XOR E(L(E5)+L(09))

= E(32+DF) XOR E(1E+68) XOR E(58+EE) XOR E(20+C7)

= E(111-FF) XOR E(86) XOR E(146-FF) XOR E(E7)

= E(12) XOR E(86) XOR E(47) XOR E(E7)

= 38 XOR B7 XOR D7 XOR 8C

= D4

Output(1) = (04 * 09) XOR (66*0E) XOR (81*0B) XOR (E5*0D)

= E(L(04)+L(09)) XOR E(L(66)+L(0E)) XOR E(L(81)+L(0B)) XOR E(L(E5)+L(0D))

= E(32+C7) XOR E(1E+DF) XOR E(58+68) XOR E(20+ EE)

= E(F9) XOR E(FD) XOR E(C0) XOR E(10E-FF)

= E(F9) XOR E(FD) XOR E(C0) XOR E(0F)

= 24 XOR 52 XOR FC XOR 35

= BF

Output(2) = (04 * 0D) XOR (66*09) XOR (81*0E) XOR (E5*0B)

= E(L(04)+L(0D)) XOR E(L(66)+L(09) XOR E(L(81)+L(0E)) XOR E(L(E5)+(0B))

= E(32+EE) XOR E(1E+C7) XOR E(58+DF) XOR E(20+68)

= E(120-FF) XOR E(E5) XOR E(137-FF) XOR E(88)

= E(21) XOR E(E5) XOR E(38) XOR E(88)

= 34 XOR 7B XOR 4F XOR 5D

= 5D

Output(3) = (04 * 0B) XOR (66*0D) XOR (81*09) XOR (E5*0E)

= E(L(04)+L(0B)) XOR E(L(66)+L(0D)) XOR E(L(81)+L(09)) XOR E(L(E5)+L(0E))

= E(32+68) XOR E(1E+EE) XOR E(58+C7) XOR E(20+DF)

= E(9A) XOR E(10C-FF) XOR E(11F-FF) XOR E(FF)

= E(9A) XOR E(0D) XOR E(20) XOR E(FF)

= 2C XOR F8 XOR E5 XOR 01

= 30

CHAPTER 3

METHODOLOGY

Generally, this chapter will give a brief explanation about the methodology used for developing Data Transmission Using AES Encryption via SMS and Email will be provided. This chapter also will describe the detail about the rapid application development besides software and hardware specification that are needed for this project development.

3.1 Introduction of Methodology

Methodology is a set of recommended practices. What is recommended practices? This term of practices refers to practices which are widely used across an industry or scientific discipline, the techniques used in a particular research study, or techniques used to accomplish a particular project. In this chapter, methodology will be focused on several processes or techniques that will be used to develop Data Transmission Using AES Encryption via SMS and Email. It documented a set of procedures and guidelines for one or more phases of the software life cycle such as analysis and design. Many methodologies include a diagramming notation for documenting the result of the procedure and an objective (ideal quantified) set of criteria of determining whether the result of the procedure is acceptable quality.

There are many methodologies that help the developers develop their system such as Software Development Life Cycle (SDLC), Rapid Application Development (RAD), Agile methodologies and Extreme Programming (EP). The developer should choose the suitable methodology based on their case study or project. The suitable methodology is very important in developing a system because the developers should know how their project from beginning until end and how to maintaining their system when problems occur when developing the system.

3.2 Project Methodology

In order to develop this project, Rapid Application Development (RAD) has been chosen as the flow process model for developing the solution. RAD is a development lifecycle designed to give much faster development and higher quality than the traditional lifecycle. It is designed to take advantage of powerful development software. RAD is used for building large Information System applications of the kind which occur in every large business. In developing Email Cryptography System, RAD methodology has been choosing as a framework in developing this system. RAD is very suitable methodology for developing this system. RAD also have shortened and combined the analysis, design, build and test phase in the traditional SDLC into iterative process produce smoother development technique. RAD consists of four important phase which are:

i. Requirements Planning

The requirements planning phase requires that high level or knowledgeable end users determine what the functions of the system should be. It should be a structured discussion of the business problems that need to be solved. It can often be done quickly when the right users and executives are involved. Once the specific systems have been identified, the planning decided together with the end users.

ii. **User Design**

The user design phase requires the users to participate strongly in the nontechnical design of the system. The user needs to participate with the developer in creating the prototype of the system. This prototyping is used to help in requirement specification and design.

iii. **Construction**

The construction phase is a phase that the design that created in User Design Phase will be coded. Code optimizer may be used to improve the performance of the generated code. Several tools can be implementing to generate the code into the prototype design. In this phase, end user is closely involved in the construction phase where the testing will occur throughout the process of construction.

iv. **Cutover**

The cutover phase is a combination of implementation and maintenance phase. A variety of action needed which is a comprehensive testing, end user training, organizational changes and operation in parallel with the previous system until the new system settle in.

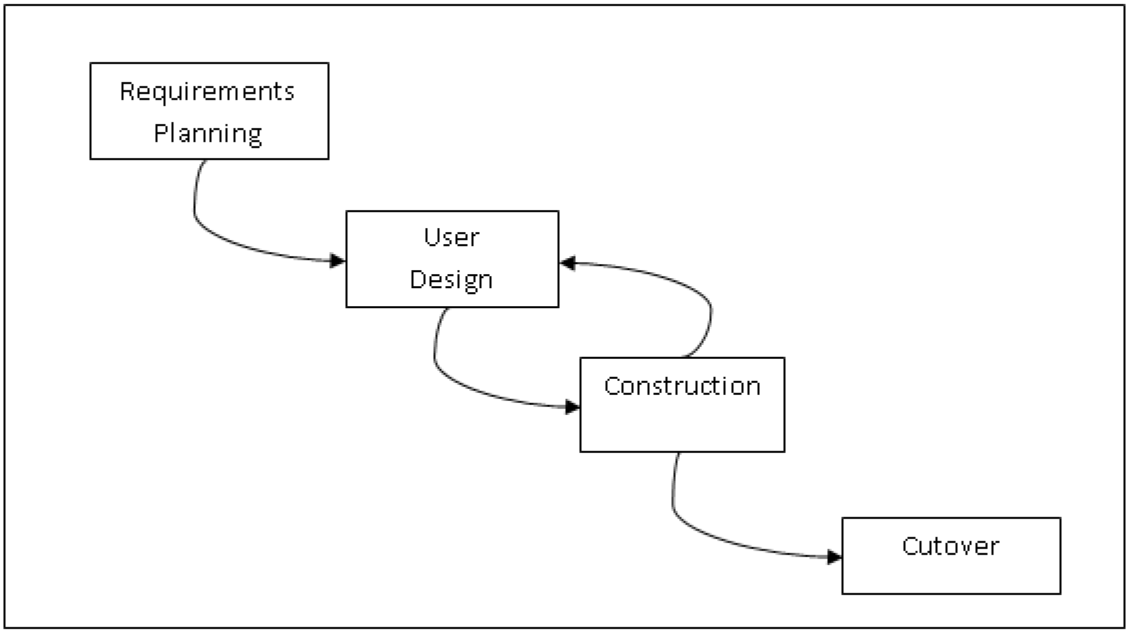
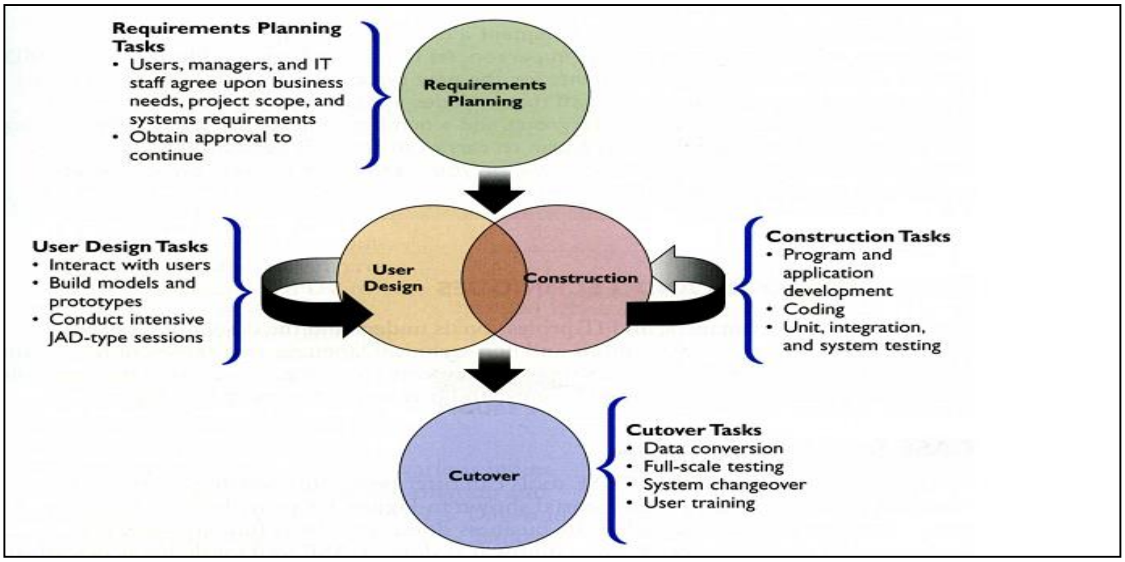


Figure 3.1: RAD Methodology


Requirements Planning Tasks
• Users, managers, and IT staff agree upon business needs, project scope, and systems requirements
• Obtain approval to continue
Requirements Planning
User Design Tasks
• Interact with users
• Build models and prototypes
• Conduct intensive JAD-type sessions
User Design
Construction
Construction Tasks
• Program and application development
• Coding
• Unit, integration, and system testing
Cutover
Cutover Tasks
• Data conversion
• Full-scale testing
• System changeover
• User training

.

Figure 3.2: The four phases of the RAD model. Notice the continuous interaction between the user design and construction phases.

3.3 Requirement Planning Phase

Requirement planning phase is the first phase in rapid application development methodology and the most important phase to the developers before they develops the system. This phase includes the entire project planning such as problem statements, objectives, scopes, Gantt chart and so on. The Gantt chart has been created in order to show the overall schedule of the project and make sure that the project will be delivered on time. The Gantt chart will illustrate a project schedule that helps to plan, manage and track a specific task in the project. In this phase, developers will collect all the information about modules in the prototype such as existing system and system requirements. The method and algorithm had been studied to develop it into the system. The information such as flow of modules and user are very important to illustrate the system design.

3.3.1 Planning

In this phase, Gantt chart has been created in order to show the overall schedule of the project and also to make sure that the project will be delivered on time. The Gantt chart will illustrate a project schedule that helps to plan, manage and track a specific task in the project.

3.3.2 Analysis Requirement

This phase will discuss about the scope of the project and the techniques. In this system, the scope includes sending Short Message Service (SMS), file encryption and decryption and sent email with attachment. The technique will be discussed in the next chapter.

3.4 User Design Phase

During the user design phase, the user interacts with the systems analysts and develops models and prototypes that represent all system process, outputs, and inputs. User design is a continuous, interactive process that allows users to understand, modify, and eventually approve a working model of a system that meets their needs. After all the information has been collected in planning phase, the analysis of the Data Transmission using AES encryption via SMS and Email (DTUA) prototype is being made. In RAD, the analysis and design phase is combining together to minimize the time.

3.4.1 Prototype Flowchart

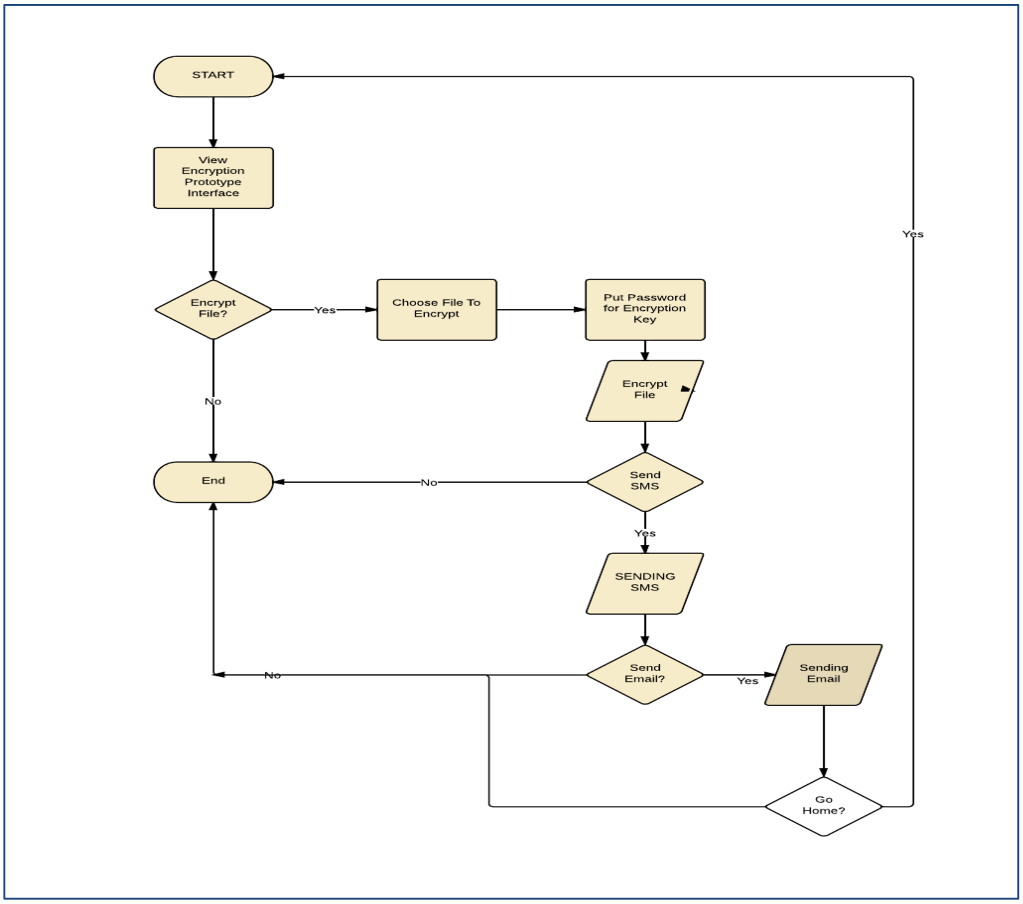


Figure 3.3: Flowchart

a) Sending Short Message Service (SMS) process

The process of the Sending Short Message Service (SMS). User needs to key in the receiver number telephone. This procedure is needed in order to send the message directly to the receiving mobile phone. After user write any remarks or note in the textbox given click the send button to execute the process.

b) Compose E-Mail message attachment process

The process of composing the message. User needs to attach the file that is already encrypted or decrypted to send to the receiver. After that, user key in the email data and the SMTP (Sending Message Transfer Protocol). This is to ensure which SMTP server that the user use for their email in order to complete the process. The user now can click the send button at the right bottom of the interface.

c) The Encryption Process

The process of encryption in this system. A file that is selected will be encrypted if the user puts the password. The password is needed in order to create a key for the encrypted file.

d) The Decryption Process

Figure 3.3 shows the process of decryption in this system. A file that is selected will be decrypted if the user puts the same password as the encrypted password. The password is needed in order to create a key to open encrypted file.

3.4.2 Use Case

The use case is a methodology used in a system analysis to identify, clarify and organize system requirements. It is made up of a set of possible sequences of interactions between systems and users in particular environments and related to a particular goal. Use case also contains all system activities that have significance to the users. The use case in Figure 3.4 is designed to the user. It illustrates the flaws of Data Transmission using AES encryption via Email And SMS. The user can encrypt/decrypt the file, send SMS and send email attachment file.

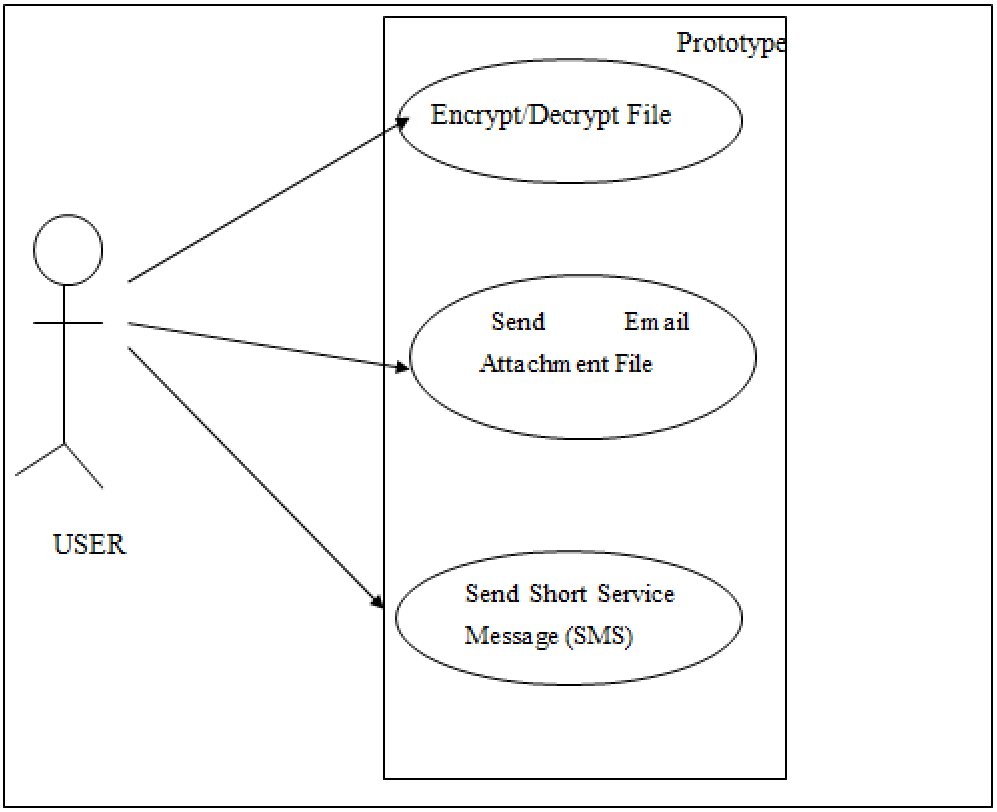


Figure 3.4 : Use Case Diagram

3.4.3 User Interface

This phase will provide some example of an interface that include in the Data Transmission using AES encryption via Email and SMS (DTUA) prototype. The interface will be developed by using Visual Basic Development(VB.net) that provide a suitable tool such as textbox, menu tab and lots of framework were provided when using VB.net. Below are the interfaces of DTUA prototype:-

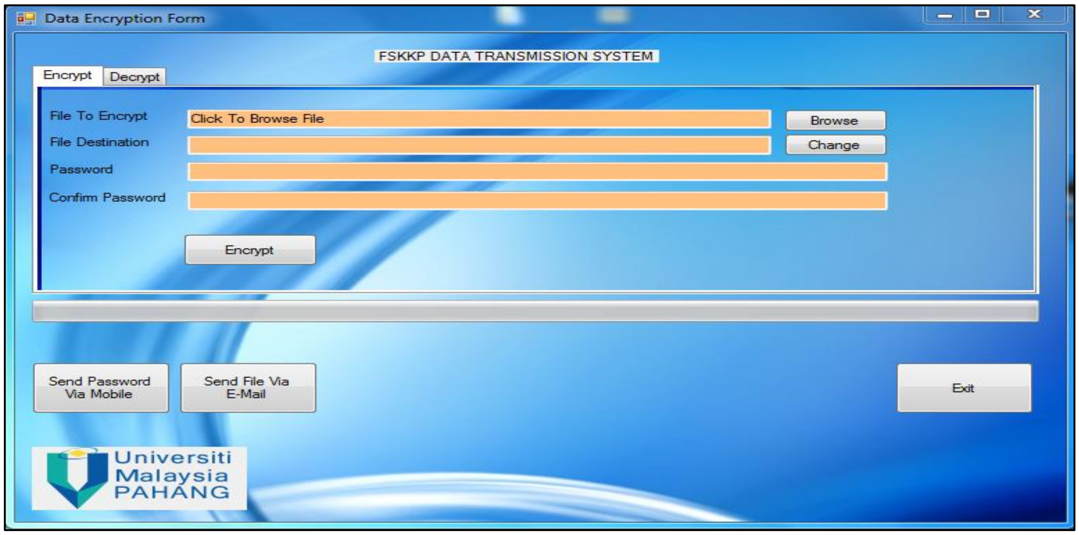
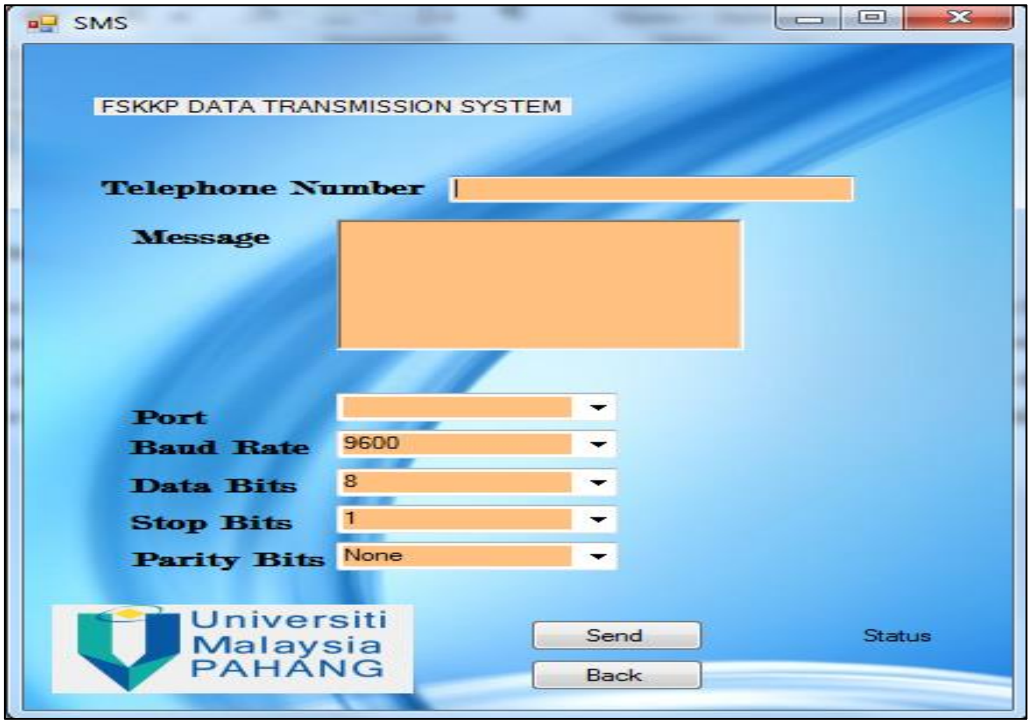


Figure 3.5: Data encryption and decryption interface.

The figure above shows users will browse/choose the file that need to encrypt or decrypt by clicking the browse button. The change button is where the user wants to change the file that was encrypted/decrypt in any directory that the users already choose. The interface also shows the option to the user to send the file encrypt/decrypt to send via email and send a password using mobile.



FSKKP DATA TRANSMISSION SYSTEM

Telephone Number

Message

Port

Baud Rate 9600

Data Bits 8

Stop Bits 1

Parity Bits None

Figure 3.6: Send password or any remarks, using SMS.

The figure above shows the interface that needs users to fill the receiver phone number and the message. About Port, Baud Rate, Data Bits, Stop Bits and Parity Bits the prototype will auto-detect which port is active in users' laptop or PC. The send button will initiate the Short Service Messages (SMS) to the number phone that fill by the user.

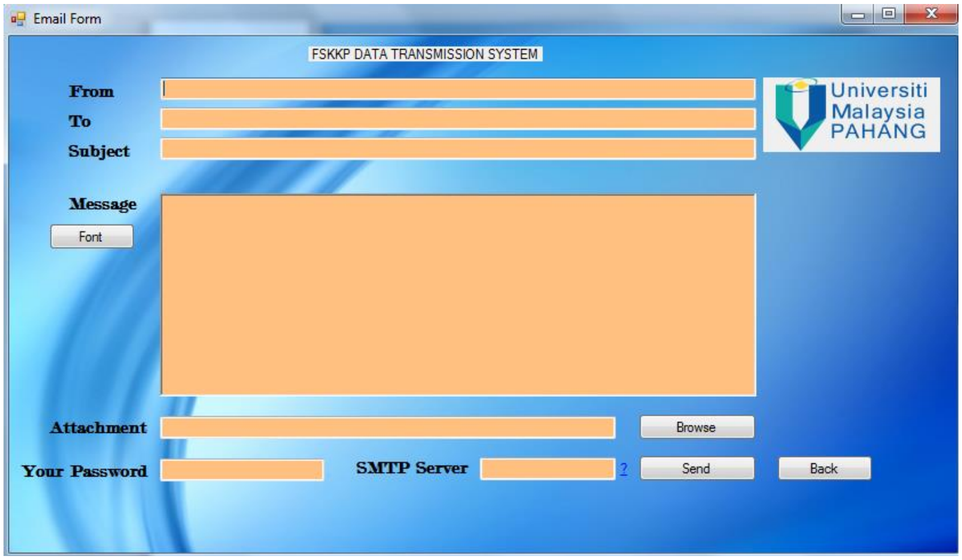


Figure 3.7: email platform with attachment interface.

The figure above shows the email platform that is provided in the DTUA prototype. The user needs to fill the textbox and the message that includes sender and receiver email address and send the password and SMTP server that is the server that sends email server using such as a famous server that is Yahoo or Gmail server. The user who does not know the server, the link label is provided to instruct the user which suitable email user should use. There is also an attachment file easier for the user to attach encrypted/decrypted file by using this email platform provided to avoid any time consume between the process in using the old method.

3.5 Construction Phase

In the construction phase, the process is focused on the prototype development where the purpose is to develop the system requirement. During this phase, the requirements turn into the working system that must be tested and being used. In addition, all of the coding, testing and installations are already done. The system requirement is important in order to provide the development system. There are two types of that need to be in this system such as software and hardware requirement.

Table 3.1: Hardware Requirement

NO	Item	Minimum Requirements / Specifications	Purpose
1	Laptop	Intel^{R} Core^{TM} Duo Processor P7350 Hard Disk : 500 GB RAM : 4GB DDR2 GSM MODEM HUAWEI	For documentation the system and development process.
2	Pen drive	Kingston 4GB	Backup data and files.
3	External Hard Disk	Seagate 500GB	Backup data and files.

Table 3.2: Software Requirement

NO	**Item Name**	**Purpose**
1	Notepad	Backup script
2	Microsoft Office • Microsoft Word 2007	Documentation and system report System Flow Chart
3	Avira Anti Virus	Protect from virus

4	Microsoft Window 7	Operating System that will be used for system development
5	WinRAR	Compressing the data and files
6	Visual Basic.Net	Editing the system pictures and designing the interface
7	Microsoft Visual Basic.Net	Develop system and design the interface

3.6 Cutover Phase

Cutover phase is where the system is being delivered to the end user. In here a variety of action is needed to be done which is a comprehensive testing and implement the system. The cutover phase is a combination of the implementation phase and the maintenance phase. In this phase, if there is any error occurred the system will be modified and corrected.

CHAPTER 4

IMPLEMENTATION

4.1 Introduction

The chapter covers the implementation phase of the Data Transmission Using AES encryption Via Email and SMS (DTUA). The implementation phase focuses on the development of the workable system activities. The implementation activities include the system coding, debugging and documenting. System coding and debugging are the main activities in this implementation phase. The coding includes the structure of the coding system that is used to run the functions in this system. During the implementation, the developer has to ensure that he has fulfilled the system requirements before implementing the system to avoid the system error or any complications. In this implementation stage, the system will be developed step by step based on function modules.

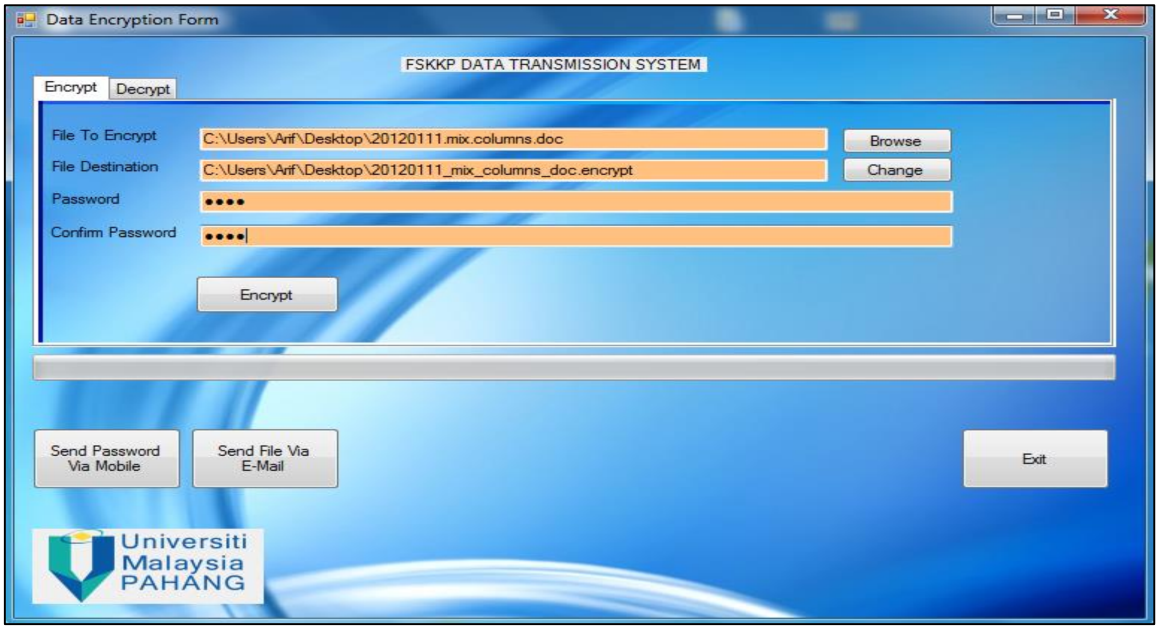
The implementation phase is an important phase in order to develop an effective system. In this phase, the developer will determine the tools that will be used to implement the system, starting from building interfaces to running the system with the free error until completing all functions that have been stated in the previous chapter. The programming is the main factor of this phase whereby the developer has to manage the coding properly to make sure the system run with free error and to determine the effectiveness of the system. The software development environment setup, software and the implementation status of the system will be discussed in this chapter.

4.2 Tools and Technologies

Data Transmission using AES encryption Via Email and SMS (DTUA) is developed using Microsoft Visual Basic.NET for the interface and function. Other than providing interfaces. The application runs by using the Visual Basic script. The software also uses GSM MODEM in order to provide the sending message SMS and also provide the data plan for sending email for the attachment.

4.2.1 Debugging and Running the System

After finishing the VB.NET coding, the developer needs to run or debug the system to test the running system if there any syntax error or error in the coding stage before. As we know, the system is running need to be run by using VB.Net debugging. To test whether the file is encrypted or not by changing the extension file to the previous state. If the file cannot read the content the file is completely encrypted. The test about the decrypt is how the file can be open into previous state by entering the correct password.



File To Encrypt: C:\Users\Arif\Desktop\20120111.mix.columns.doc

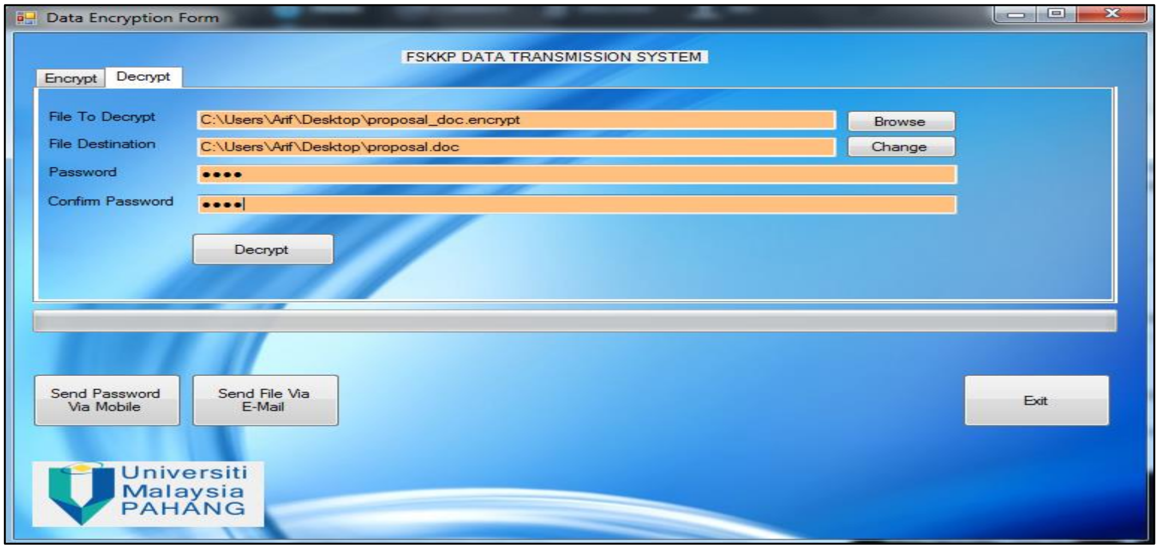
File Destination: C:\Users\Arif\Desktop\20120111_mix_columns_doc.encrypt

Password: ●●●●

Confirm Password: ●●●●

Figure 4.1: the running state of DTUA prototype using the VB.net debugger for the encryption process

The user chooses the data to be encrypted and put the password for the file to be the key for the encryption process.



File To Decrypt: C:\Users\Arif\Desktop\proposal_doc.encrypt

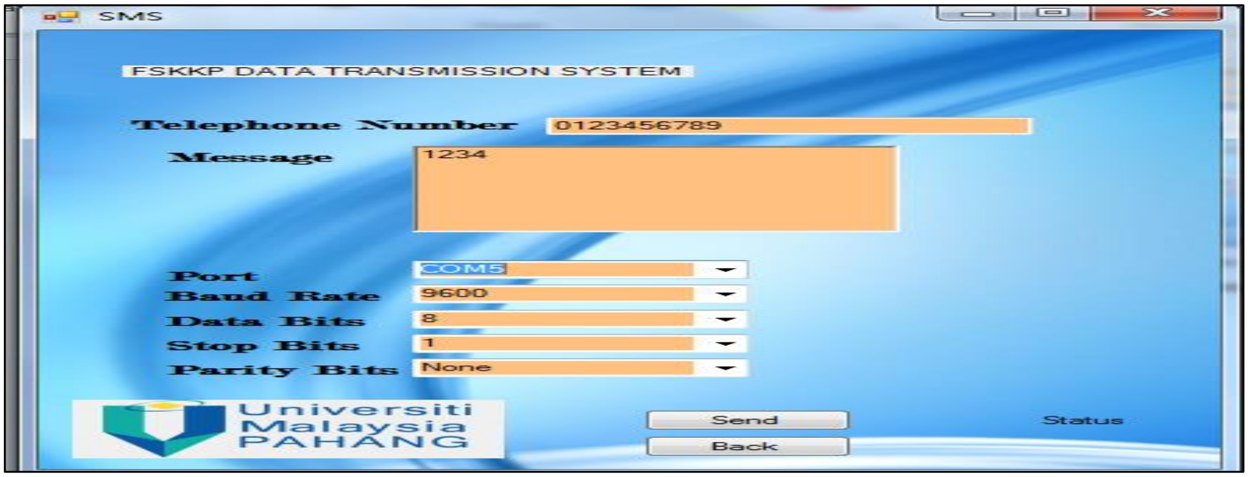
File Destination: C:\Users\Arif\Desktop\proposal.doc

Password: ●●●●

Confirm Password: ●●●●

Figure 4.2: the running state of DTUA prototype using the VB.net debugger for decrypt process.

The user chooses the data to be decrypted and put the password for the file to be the key for decryption process.



SMS

FSKKP DATA TRANSMISSION SYSTEM

Telephone Number 0123456789

Message 1234

Port COM5

Baud Rate 9600

Data Bits 8

Stop Bits 1

Parity Bits None

Figure 4.3: SMS platform

The user puts the receiving mobile phone number and message.

CHAPTER 5

RESULTS AND DISCUSSION

This chapter will briefly discuss the results of the proposed project. Then, the discussion on the result will also be stated out.

5.1 Expected Results

Based on the project proposal paper, a prototype that is going to help the users/sender to encrypt and decrypt the file and also manage to send the data and information to the receiver. This prototype is proposed to overcome the manual process that includes in the hand-to-hand transfer of confidential data file. Hopefully, the system can be accessed by the users and successfully execute smoothly. This system also is created to replace the existing manual process which does not carry out fully protected and lack of security aspect when comes to the confidential data transfer. The system will comprise encrypt/decrypt interface, sending SMS interface and email message interface. This system will be built in English language because English is an international language. Everyone can understand the language very well. Besides, the message or information that wanted to be send also can be in any or user language.

Transferring data via Email and SMS has considerable benefits over traditional paper based memo and portal systems. Through this prototype, the confidential data that has been decrypted or encrypt can be sent at any time across the world as easily to a group or people or a single recipient without the sender leaving their desk by using E-mail and SMS service wherever they are. Moreover, the recipient will receive the message directly without passing through any third-party.

5.2. Discussion

In order to make the prototype easy for the users, the interfaces that have been designed should be as simple as possible. The users will not have to spend more time in understanding the system before they started using the system's application. It is designed and implementation of high security. Furthermore, this system is very user friendly. The system designed for people should be easy to use, learn and more satisfying to use. So, the users will not have any difficulties in using the prototype. The prototype developed managed to execute the main function that is encrypting the confidential data successfully. The prototype also has shown that the data can be transferred when two users in different place and different time.

5.3 Project Constraint

Project constraint is the constraint in this development project. It consists of three components such as development constraint, software constraint and hardware constraint.

5.3.1 Development constraint

The e-mail delivery is depending on the internet speed. The prototype cannot implement On-Screen Keyboard from the prototype because of the compatibility of the windows and VB.Net platform is different. There are many types of programming language such as PHP, HTML and ASP.NET that also can develop DTUA prototype. In this case the developer should master in the language stated. VB.Net use would be reliable in using GSM MODEM but PHP, HTML, and ASP.NET can implement the gateway function easily that is more flexible and cheaper to developed sending SMS function.

5.4 Advantages and Disadvantages of the system.

The advantage of this system is to secure the data and information from being stolen or read by unauthorized persons and an attacker. This is because; the primary advantage of password or key cryptography is implanted to increase security and convenience. The user can send the encrypted data using the Email platform in the prototype. The graphic user interface (GUI) is understandable to the user. The user can send the password by using Short Message Services that provided by the prototype and this function is not carried out yet in any encryption existing system to protect the key/password of the confidential file because there is no way to retrieve it in case the owner or the user, if they lost it.

The disadvantage of this system is Rijndael Cipher is a highly secure algorithm that is used by the existing system nowadays, the only way to attack is to perform a brute-force attack on the modulus. This attack can be simply defeated by increasing the key size or bit size. The system uses the GSM Modem in order to send SMS. The user needs a GSM modem in order to execute the SMS function.

CHAPTER 6

CONCLUSION AND FUTURE WORKS

6.1 Conclusion

The purpose of this chapter is to make a conclusion of the research that has been done and to provide future suggestion to produce a better system. The prototype system that has been developed has to achieve all the objectives stated in this thesis as shown in chapter 5 result and discussion.

As for the conclusion, cryptography is a representation of readable text to a format that cannot be read. It is implemented in the DTUA prototype in order to make the data transfer and information is more secure. There is one technique that will be employed in this system. That is consists of Rijndael Cipher or Advanced Encryption Standard. This technique has a complex calculation and method that is included Subbyte phase, Shift Row phase, Mix-Column phase and AddRound- Key phrase.

The technique is met up the objective above in the Chapter 1 section. That is to implement the AES technique to the confidential data. The hand-to-hand method as we know can lead to the exposing of confidential data content to unauthorized personnel if the data was misplaced because of human-error. The email platform and message send to mobile phone is quite a good method to overcome the manual method in order to improvise and to avoid the human-error in the exposing of confidential data content to the wrong person.

The AES encryption which has been studied Rijndael cipher has been implemented successfully in the prototype as it can be seen that the file is not readable whether the file extension is changed to the previous state after the encryption process.

While the software development life cycle that is chosen in this project which is a Rapid Application Development (RAD) has also worked well in this project. As each phase of the RAD model provides high visibility and quality of the project. Each task is understood and done properly before moving on to the next phase.

6.2 Future Suggestion

In this thesis, it is found that the prototype system can be implemented with any encryption algorithm or technique. The prototype so far has proven itself by able to help the users to encrypt and decrypt the file using their own key/password and able to execute with the email platform service that's easier to the user to send the file directly to the receiver. However, throughout the development within the system a few constraints have come across.

The prototype can improve by executing the system in the mobile phone because all of users or people are using Smartphone that has access to their own email account, so that it is easy for the receiver to decrypt the file on the spot without having trouble to open the file. The prototype also can be embedded in any management system and or any system that involve in exchanging of confidential.

So, for better encryption system, maybe this system can be improved by enhancing the algorithm existing to the higher level and increase the key size. In addition, the prototype could be developed in mobile application to make use to the receiver of the file to instantly decrypt the encrypt file that was sent by the sender because lots of the user wear a Smartphone that can receive the email or attachment easily by phone. For further improvement also the prototype will be using gateway function to send SMS because the SIM card in the GSM Modem will be not effective in another 5-10 years ahead.

REFERENCES

1. Introduction to the Advanced Encryption Standard. (N.d.). Retrieved from http://cboard.cprogramming.com/c-programming/87805-[tutorial]-implementing-advanced-encryption-standard.html

2. Wikipedia, the free encyclopedia. “Advanced Encryption Standard” (2005), Online: http://en.wikipedia.org/wiki/AES

3. Coppersmith, D. 1994. The Data Encryption Standard (DES) and Its strength against attacks."IBM Journal of Research and Development, pp. 243-250.

4. Eskicioglu, A. M. 2001, Cryptography, IEEE Potentials , pp 36-38.

5. Kaufman, C., Perlman, R., and Speciner, M. 2002. Network Security, Private Communication in Public World. New Jersey: Prentice Hall.

6. Khate, Atul, 2009. Cryptography and Network Security. 2nd Edition, Tata McGraw Hill, pp. 87-2004.

7. Rayarikar, Rohan; Upadhyay, Sanket; Pimpale, 2012, Priyanka, SMS Encryption Using AES Algorithm on Android, International Journal of Computer Applications, vol. 50, issue 19, pp. 12-17.

8. Mare, S. F. , M. V. 2011. Decreasing change impact using smart LSB Pixel mapping and data rearrangement. Proceedings of the 11th IEEE International Conference on Computer and Information Technology. pp 269-275.
9. Berent, A. (2009). Advanced encryption standard by example (V 1.5) ABI Software Development.

10. Selent, D. (2010). Advanced Encryption Standard. 6(2),

11. Yang, Y., X. N. 2006. Symmetric Steganography Secure Against Chosen Message and Original Cover Attacks, First International Conference on Innovative Computing, Information and Control, IEEE, pp. 661-664.

12. Parikh, C, Patel, P, 2007. ‗Performance Evaluation of AES Algorithm on Various Development Platforms‘, *IEEE International Symposium on Consumer Electronics*, pp. 1 – 6.

13. NowSMS | SMS Gateway, SMS Server Software, MMS Gateway & MMSC (n.d.). Retrieved December 10, 2011, from What is a GSM Modem? | NowSMS: http://www.nowsms.com/faq/what-is-a-gsm-mode

14. Rouse, M. (2007 Feb). Definition: Rapid Application Development (RAD). Retrieved from http://searchsoftwarequality.techtarget.com/definition/rapid-application-development

15. Boesgaard, C. (2003). A short introduction to AES. Informally published manuscript, Department of Computer Science, University of Copenhagen, .

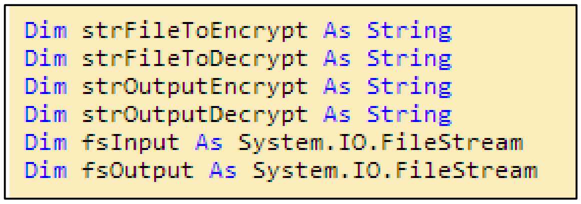
16. Top Ten Reviews. (Oct.2013.). Retrieved from http://encryption-software-review.toptenreviews.com

17. Peyravian, M., & Zunic, N. 2000. Methods for Protecting Password Transmission,Computers & Security, 19(5), Elsevier, pp 466 – 469.

18. Sakar, S., & Maitra, S. 2010. Cryptanalysis of RSA with more than one decryption exponent, Information Processing Letter, Elsevier, pp 336 – 340.

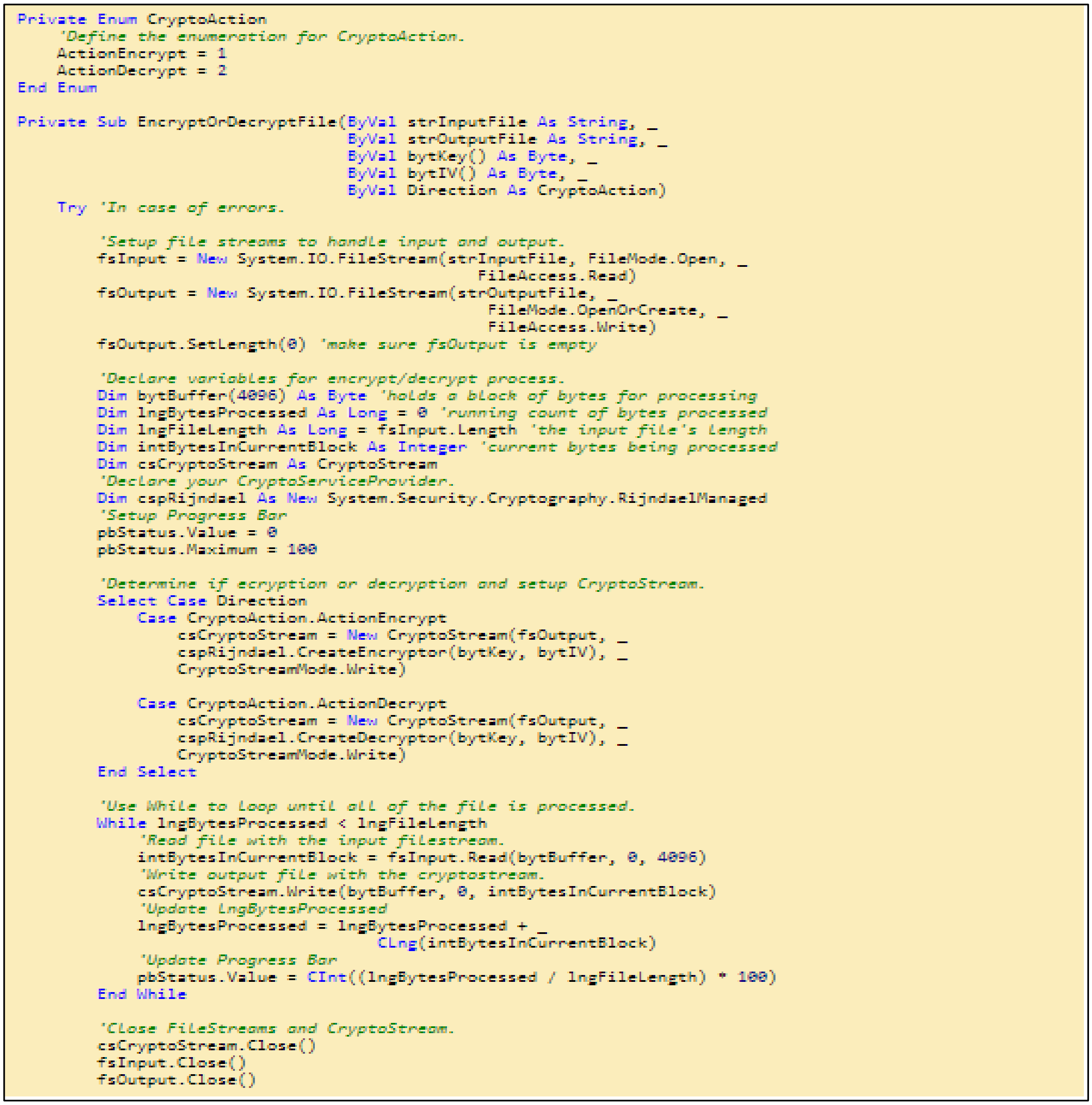
APPENDIX A

Function Code



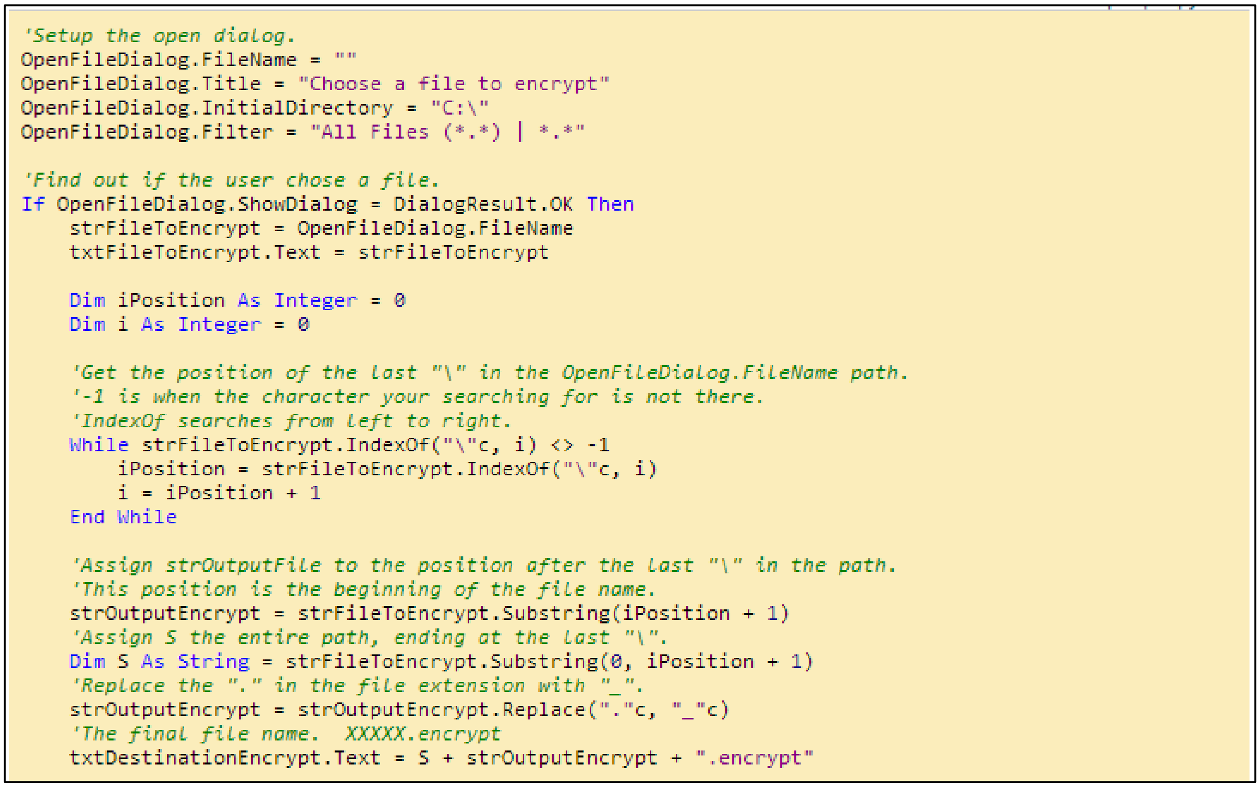
```
Dim strFileToEncrypt As String
Dim strFileToDecrypt As String
Dim strOutputEncrypt As String
Dim strOutputDecrypt As String
Dim fsInput As System.IO.FileStream
Dim fsOutput As System.IO.FileStream
```

Global Variable



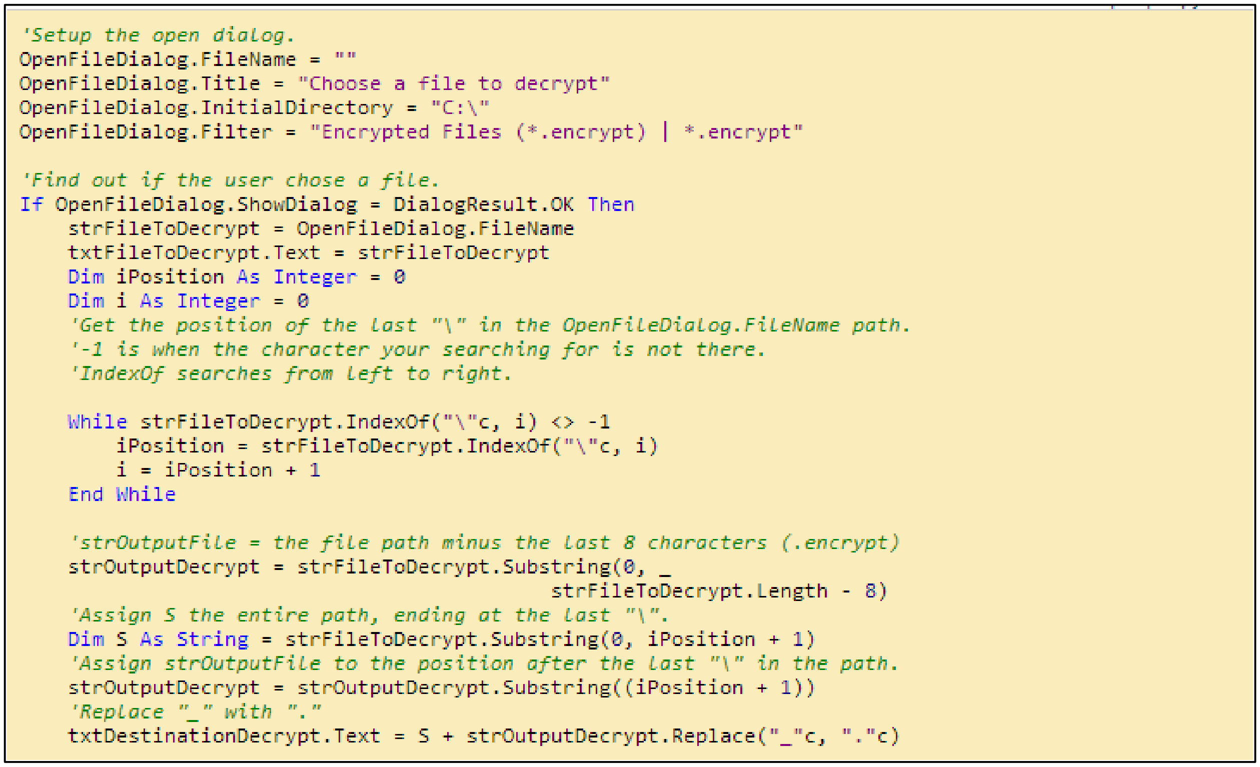
```
Private Enum CryptoAction
    'Define the enumeration for CryptoAction.
    ActionEncrypt = 1
    ActionDecrypt = 2
End Enum

Private Sub EncryptOrDecryptFile(ByVal strInputFile As String, _
                                 ByVal strOutputFile As String, _
                                 ByVal bytKey() As Byte, _
                                 ByVal bytIV() As Byte, _
                                 ByVal Direction As CryptoAction)
    Try 'In case of errors.

        'Setup file streams to handle input and output.
        fsInput = New System.IO.FileStream(strInputFile, FileMode.Open, _
                                              FileAccess.Read)
        fsOutput = New System.IO.FileStream(strOutputFile, _
                                               FileMode.OpenOrCreate, _
                                               FileAccess.Write)
        fsOutput.SetLength(0) 'make sure fsOutput is empty

        'Declare variables for encrypt/decrypt process.
        Dim bytBuffer(4096) As Byte 'holds a block of bytes for processing
        Dim lngBytesProcessed As Long = 0 'running count of bytes processed
        Dim lngFileLength As Long = fsInput.Length 'the input file's length
        Dim intBytesInCurrentBlock As Integer 'current bytes being processed
        Dim csCryptoStream As CryptoStream
        'Declare your CryptoServiceProvider.
        Dim cspRijndael As New System.Security.Cryptography.RijndaelManaged
        'Setup Progress Bar
        pbStatus.Value = 0
        pbStatus.Maximum = 100

        'Determine if ecryption or decryption and setup CryptoStream.
        Select Case Direction
            Case CryptoAction.ActionEncrypt
                csCryptoStream = New CryptoStream(fsOutput, _
                cspRijndael.CreateEncryptor(bytKey, bytIV), _
                CryptoStreamMode.Write)

            Case CryptoAction.ActionDecrypt
                csCryptoStream = New CryptoStream(fsOutput, _
                cspRijndael.CreateDecryptor(bytKey, bytIV), _
                CryptoStreamMode.Write)
        End Select

        'Use While to loop until all of the file is processed.
        While lngBytesProcessed < lngFileLength
            'Read file with the input filestream.
            intBytesInCurrentBlock = fsInput.Read(bytBuffer, 0, 4096)
            'Write output file with the cryptostream.
            csCryptoStream.Write(bytBuffer, 0, intBytesInCurrentBlock)
            'Update lngBytesProcessed
            lngBytesProcessed = lngBytesProcessed + _
                                    CLng(intBytesInCurrentBlock)
            'Update Progress Bar
            pbStatus.Value = CInt((lngBytesProcessed / lngFileLength) * 100)
        End While

        'Close FileStreams and CryptoStream.
        csCryptoStream.Close()
        fsInput.Close()
        fsOutput.Close()
```

Encrypt/Decrypt File



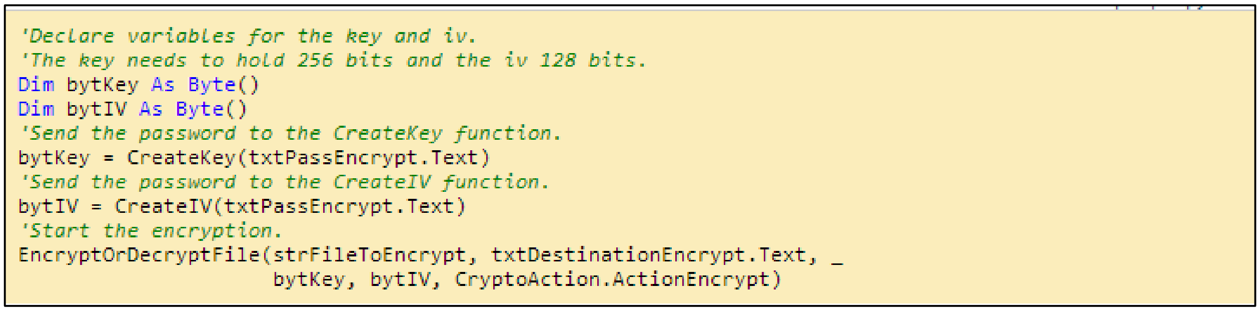
```
'Setup the open dialog.
OpenFileDialog.FileName = ""
OpenFileDialog.Title = "Choose a file to encrypt"
OpenFileDialog.InitialDirectory = "C:\"
OpenFileDialog.Filter = "All Files (*.*) | *.*"

'Find out if the user chose a file.
If OpenFileDialog.ShowDialog = DialogResult.OK Then
    strFileToEncrypt = OpenFileDialog.FileName
    txtFileToEncrypt.Text = strFileToEncrypt

    Dim iPosition As Integer = 0
    Dim i As Integer = 0

    'Get the position of the last "\" in the OpenFileDialog.FileName path.
    '-1 is when the character your searching for is not there.
    'IndexOf searches from left to right.
    While strFileToEncrypt.IndexOf("\"c, i) <> -1
        iPosition = strFileToEncrypt.IndexOf("\"c, i)
        i = iPosition + 1
    End While

    'Assign strOutputFile to the position after the last "\" in the path.
    'This position is the beginning of the file name.
    strOutputEncrypt = strFileToEncrypt.Substring(iPosition + 1)
    'Assign S the entire path, ending at the last "\".
    Dim S As String = strFileToEncrypt.Substring(0, iPosition + 1)
    'Replace the "." in the file extension with "_".
    strOutputEncrypt = strOutputEncrypt.Replace("."c, "_"c)
    'The final file name.  XXXXX.encrypt
    txtDestinationEncrypt.Text = S + strOutputEncrypt + ".encrypt"
```

Changing File Extension



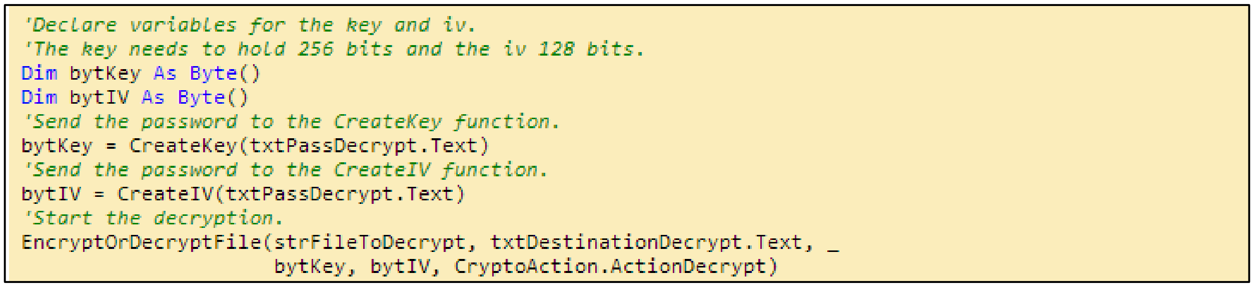
```
'Setup the open dialog.
OpenFileDialog.FileName = ""
OpenFileDialog.Title = "Choose a file to decrypt"
OpenFileDialog.InitialDirectory = "C:\"
OpenFileDialog.Filter = "Encrypted Files (*.encrypt) | *.encrypt"

'Find out if the user chose a file.
If OpenFileDialog.ShowDialog = DialogResult.OK Then
    strFileToDecrypt = OpenFileDialog.FileName
    txtFileToDecrypt.Text = strFileToDecrypt
    Dim iPosition As Integer = 0
    Dim i As Integer = 0
    'Get the position of the last "\" in the OpenFileDialog.FileName path.
    '-1 is when the character your searching for is not there.
    'IndexOf searches from left to right.

    While strFileToDecrypt.IndexOf("\"c, i) <> -1
        iPosition = strFileToDecrypt.IndexOf("\"c, i)
        i = iPosition + 1
    End While

    'strOutputFile = the file path minus the last 8 characters (.encrypt)
    strOutputDecrypt = strFileToDecrypt.Substring(0, _
                                      strFileToDecrypt.Length - 8)
    'Assign S the entire path, ending at the last "\".
    Dim S As String = strFileToDecrypt.Substring(0, iPosition + 1)
    'Assign strOutputFile to the position after the last "\" in the path.
    strOutputDecrypt = strOutputDecrypt.Substring((iPosition + 1))
    'Replace "_" with "."
    txtDestinationDecrypt.Text = S + strOutputDecrypt.Replace("_"c, "."c)
```

Changing Extension in decrypt to remove the (. encrypt)



```
'Declare variables for the key and iv.
'The key needs to hold 256 bits and the iv 128 bits.
Dim bytKey As Byte()
Dim bytIV As Byte()
'Send the password to the CreateKey function.
bytKey = CreateKey(txtPassEncrypt.Text)
'Send the password to the CreateIV function.
bytIV = CreateIV(txtPassEncrypt.Text)
'Start the encryption.
EncryptOrDecryptFile(strFileToEncrypt, txtDestinationEncrypt.Text, _
                     bytKey, bytIV, CryptoAction.ActionEncrypt)
```

Encrypt Button



```
'Declare variables for the key and iv.
'The key needs to hold 256 bits and the iv 128 bits.
Dim bytKey As Byte()
Dim bytIV As Byte()
'Send the password to the CreateKey function.
bytKey = CreateKey(txtPassDecrypt.Text)
'Send the password to the CreateIV function.
bytIV = CreateIV(txtPassDecrypt.Text)
'Start the decryption.
EncryptOrDecryptFile(strFileToDecrypt, txtDestinationDecrypt.Text, _
                     bytKey, bytIV, CryptoAction.ActionDecrypt)
```

Decrypt Button