IMPLEMENTATION OF HYBRID ENCRYPTION METHOD USING CAESAR CIPHER ALGORITHM

CHAROMIE A/L TAT WI

A thesis submitted in partially fulfillment of the requirements for the award of degree of Bachelor of Computer Science (Computer Systems & Networking)

Faculty of Computer System & Software Engineering
Universiti Malaysia Pahang (UMP)

APRIL 2010

ABSTRACT

In daily life, we use internet or accessing information from various sources. Some of the procedure requires us to sending our own information. The objective of this project is to explore the way of doing encryption, improve certain aspect of the existing algorithm and to find out a way to create better security. In this project, encryption will be implemented in information on a web that makes it hard to be readable and secure. For that matter, encryption method known as Caesar cipher, one of the simplest and most widely used encryption techniques. In this encryption, it uses the substitution cipher in which each letter in the plaintext is replaced by some fixed number of position down the alphabet. This method is names after Julius Caesar who using this method to communicate with his generals. The result from this project is a data which is encrypted and be decrypted to its readable form. As a conclusion, Caesar cipher algorithm can be implemented in hybrid encryption project to make data secure and better.

ABSTRAK

Di dalam kehidupan seharian, kita menggunakan internet atau mengakses maklumat dari pelbagai sumber. Terdapat beberapa prosedur yang memerlukan kita untuk menghantar maklumat peribadi. Objektif dalam melaksanakan projek ini adalah untuk meneroka cara untuk melakukan penyulitan, memperbaharui aspek-aspek di dalam algoritma dan mencari jalan untuk menghasilkan sistem yang lebih selamat untuk pengguna. Di dalam projek ini, penyulitan akan diimplementasikan di dalam maklumat dalam web untuk menjadikannya susah untuk difahami dan lebih selamat. Bagi yang berkenaan, kaedah penyulitan yang dikenali sebagai Caesar Cipher , satu kaedah paling mudah dan paling meluas digunakan. Di dalam kaedah ini, penggunaan cipher penggantian iaitu menggantikan setiap huruf di dalam teks dengan susunan posisi tertentu. Kaedah ini dinamakan sempena Julius Caesar yang menggunakan kaedah ini untuk berkomunikasi dengan jeneralnya. Hasil daripada projek ini adalah data yang telah disulitkan dan dinyahsulitkan untuk kembali ke keadaan asal. Sebagai kesimpulannya, penyulitan Caesar boleh diimplementasikan di dalam penyulitan hibrid untuk menjadikan projek ini lebih baik dan selamat.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Security is one of the important aspects in computing. In data transfer, security must be considered as one of the method implemented to ensure secure data transfer. Data transfer is transferring information from a location or host to another host, or server. To have a secure data transfer, few method can be applied, and one of them is encryption of data, prepare it to be transferred in encrypted way and decrypted when the data want to be used.

The art and science of keeping messages secure is called cryptography. It is practiced by cryptographers, and cryptanalyst are practitioners of cryptanalysis, the art and science of breaking the cipher text. A message is plaintext or clear text. The branch of mathematics encompassing both cryptology and cryptanalysis is cryptology and cryptologists practice it.

Encryption is a process of transforming information that usually are plaintext using an algorithm (known as cipher) to make it unreadable to anyone except those who have the special knowledge known as the key. The output from the process is referring as cipher text in cryptography. In the reverse, the process is called decryption to make the information readable.

In this project, Caesar cipher algorithm will be implemented to encrypt the data from the user on a web based to prepare it for a safe transfer. This will turn the plaintext of data information unreadable unless without the knowledge of the key. The data will have the security to avoid intruder or hacker to steal and read this information.

Caesar cipher is one of the popular methods used in encryption. Other than this, existing system uses MD5, Blowfish or RSA to secure data. Usually, encryptions were applied on log in system, data, disk drives, and database and password generators.

To create a hybrid encryption, the algorithm will be enhanced with two more algorithms to encrypt the data exactly at three times using stack method and stenography.

1.2 Problem Statement

Security is one of the issue often face by web developer and software engineer. In today's life, computer and networking is essential to today's communication and transferring of data is essential when doing online banking, data transferring, sharing and more. So in this project, Caesar cipher algorithm will be use as the base of the hybrid encryption. The next level of encryption is the cipher algorithm that can solve the weakness of the Caesar Cipher and the third level of the encryption method is to hide the original pattern of encryption.

1.3 Objective

The objectives of the research are to:

i. Implement hybrid encryption method using Caesar Cipher algorithm.
ii. Develop a prototype that helps to test the output from the encryption process.
iii. To compare the propose hybrid encryption method with Caesar Cipher.
iv. Create strong encryption by combining the different algorithm.

1.4 Scope

i. Caesar cipher algorithm is applied prototype, and the prototype will be enhanced by adding the other algorithm to increase the strength of security.
ii. Create encryption and decryption method for the propose hybrid encryption.

1.5 Thesis Organization

This thesis consists of six (6) chapters. Chapter 1 will discuss on introduction to system/research. Chapter 2 will discuss on the literature review on the Caesar cipher algorithm. Chapter 3 will be discussing on the methodology on this project. Chapter 4 will discuss on the implementation of the technique of Caesar cipher in the web based. Chapter 5 will show the result and discussion on the project and the technique implemented and lastly Chapter 6 will be the conclusion.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This literature review explains the concept of the ways to find out all information that will be used in order to develop this system. In this chapter, all the research that related with this system will be analyzed.

2.2 Communication

In general security is the quality or state of being secure and free from danger. In other words, building protection against adversaries, from those whose to do harm, intentionally or otherwise, it objective. Computer security can be defined as technological and managerial procedures applied to computer systems to ensure the availability, integrity, and confidentially, availability, and confidentiality of data and information from threat and vulnerabilities.

Confidentiality ensures that computer-related assets are accessed only by a authorized parties. That is, only who's who should have access to something will actually get that access. By "access", unauthorized user not only can read, but also

viewing, printing, or simply knowing that particular assets exist. Confidentially is sometimes called secrecy and or privacy.

Integrity means that assets can be modified only by authorized parties or only in authorized ways. In this context, modification includes writing, changing, changing status, deleting and creating. In distributed system, the traffic between clients and servers is a new point of attack for would-be intruders. Vulnerabilities introduced by insecure communication links can naturally counteract by services and mechanism fro communication security.

For this matter

- Data confidentiality: encryption algorithms hide the content of message
- Data integrity: integrity check functions provide the means to detect whether this document has been changed.
- Data origin authentication: Message Authentication codes in digital Signature Algorithms provide the means source and integrity of a message.

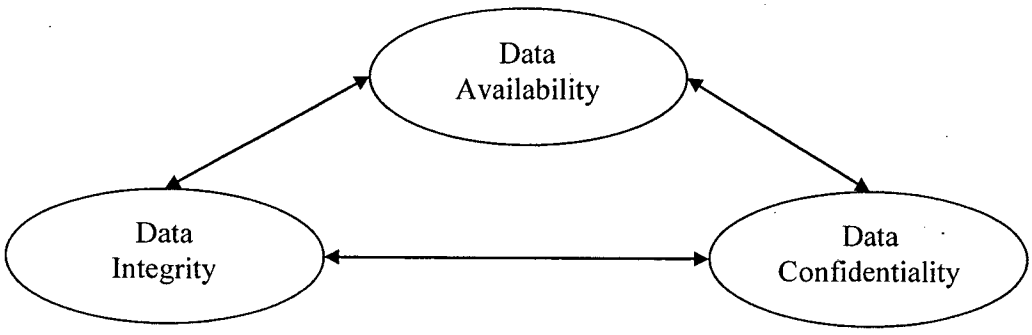

Figure 2.1 Relationship between confidentiality, integrity and availability.

The Figure 2.1 is the diagram that explains the relationship between confidentiality, integrity and availability. Data origin authentication includes data integrity. Message cannot be claim to be modified in the transit. Conversely, if the sender's address is part of the message, the message can be verify from the source when verifying the integrity. In such setting, data integrity, data integrity and data

origin authentication are equivalent concepts. A separate notion of data makes sense in other application, for example file protection in anti-virus software.

Availability means that the assts are accessible to authorized parties at appropriate times. In other words, if some of the person or system has legitimate access to a particular set of objects, that access should be prevented, for this reason, availability is sometimes known by its opposite, denial of service.

Security [25] is about the protection of the assets. A rough classification of protection measures distinguishes between prevention, detection and reaction. Prevention is take measure to prevent assets from being damage. Detection is the measures that allow detecting the asset have been damaged, how it damaged and who has cause the damage. Reaction is how a user or administrator or developer to recover from the damage of the assets. There are four elements of computer security. And these elements are related to each other. Those elements are application software, resource (object), hardware and user (subject).

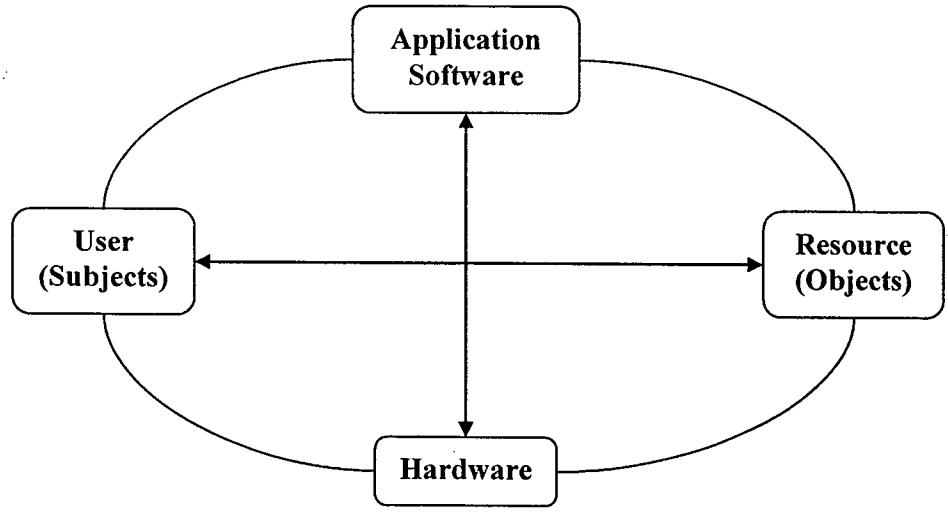

Figure 2.2 The Dimension of Computer Security

The Figure 2.2 is the dimension of computer security. Security evaluation is to aim for assurance that the system is secure. System assurance related to

functionality which is the security feature of the system, effectiveness of the mechanism used in the system, and assurance; the thoroughness of the evaluation.

Security in computing addresses these three goals. One of the challenges in building a secure system is finding the right balance among the goals, which often conflict. It is easy to preserve the particular objects confidentially in a secure system simply by preventing everyone from reading that object.

2.3 Cryptology, Encryption and Algorithm

Cryptology is the most commonly encountered area of cryptography, consisting of the science of understanding, implementing, and using information obfuscation techniques. These techniques are called cryptographic algorithms, codes, codebooks, cryptosystem, crypto algorithms or ciphers. It can be extremely complex, requiring advanced mathematics just to understand some of the basic principles, such as the RSA algorithm. The term encryption refers to taking information that is unobfuscated (the plaintext) and applying the cipher to acquire obfuscated data (the cipher text). Taking cipher text and deriving the plaintext is called decryption.

Cryptology or secrecy is the strongest tool for controlling against many kind of security threats. Well disguised data cannot be read, modified or fabricated easily. Cryptography is rooted in higher mathematics, group and field theory, computational complexity, and even real analysis, not to mention probability and statistics. Fortunately, it is not necessary to understand the underlying mathematics to be able to use cryptography.

The aim of cryptography is not to hide he existence of a message, but rather to hide its meaning, and this method is called encryption. To make a message unintelligible, it scrambled according to a particular algorithm, which is agreed upon beforehand between the sender and the intended recipient. Thud, recipient can reverse the scrambling protocol and make the message comprehensible. This reversal or scrambling is known as decryption. The advantage of using encryption and decryption is that, without knowing the scrambling protocol, the message is difficult

to recreate. Cryptography has its roots in communication security. Communication security is described in this figure.

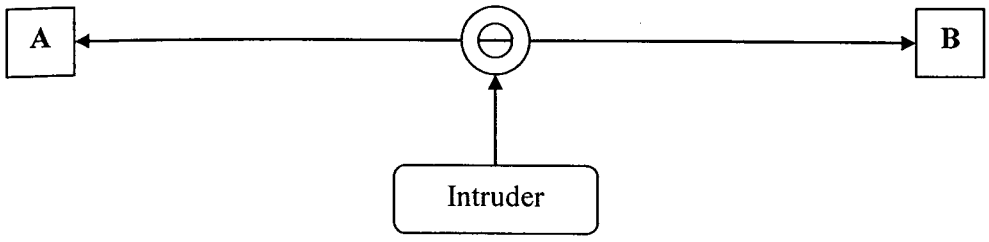

Figure 2.3 The Communication Security

The Figure 2.3 is the description of two entities that tries to communicate over an insecure channel. The antagonist is an intruder who has full control over this channel, being able to read their messages, delete messages and insert messages. The two entities A and B trust each other. They want a protection from the intruder. Cryptography gives them the means to construct a secure logical channel over an insecure physical connection.

Encryption is the formal name for scrambling program. The normal data, unscrambled, called plaintext or clear text and transform them so that unintelligible to the outside observer, the transformed data is called enciphered text or cipher text. Using encryption security professional can virtually nullify the value of an interception and the possibilities of effective modification and fabrication.

Encryption is clearly addressing the need for confidentially of data. Additionally, it can used to ensure integrity, that the data cannot be read generally cannot be easily changed in the meaningful manner. It is basis of the protocol that enables to provide security while accomplishing an important system or network task. A protocol is is an agreed-on sequence of actions that leads to desirable results. For example, some operating system protocols ensure availability of resources as different tasks and users request them. Thus, encryption can also be thought of as supporting availability. That is, encryption is at the heart of methods for ensur8ing all aspects of computer security.

Consider the steps involved in sending message from a sender, S to a recipient, if S entrusted the message to T, who then delivers it to R, T then becomes the transmission medium. If an outsider, O, wants to access the message (to read, change or even destroy it), O is called and interceptor or intruder. Anytime after S transmits it via T, the message is vulnerable to exploitation, and O might try to access, O maybe tries to block the message, intercept, modify and fabricate.

The cryptosystem involves a set of rules for how to encrypt the plaintext and how to crypt the cipher text. The encryption and decryption rules, called algorithms, often used a device call key, denoted by K, so that the resulting cipher text depends on the original plaintext message, the algorithm, and the key value. This dependence is written as C = E(K,P). Essentially, E is a set of encryption algorithms, and the key K selects one specific algorithm from the set.

Sometimes the encryption and decryption keys are the same, so P =D(K,E(K,P)). This from is called symmetric encryption because D and E are mirror-image processes. At other times, encryption and decryption keys come in pairs. then, a decryption key KD, inverts the encryption of key KE so that P = D(KD,E(KE,P)). Encryption algorithms key, KD, inverts the encryption of KE so that P= D(KD,E(KE,P)). Encryption algorithms of this form are called asymmetric because converting C back to P involves a series of steps and a key that are different from the steps and key of E.

A key makes the programmer flexibility in using and encryption scheme. Different key can be obtained by just changing the key. Moreover, using key provides additional security. If the encryption algorithm should fall into the inceptor's hand, the future message can still be kept secret because the interceptor will not know the value. An encryption scheme that does not require the use of a key is called keyless cipher.

Both a cryptographer and a cryptanalyst attempt to translate coded material back to its original form. Normally, a cryptographer works on behalf of a legitimate sender or receiver, where as cryptanalyst works on behalf of an unauthorized

interceptor. Finally, cryptology is the research into and study of encryption and decryption; it includes both cryptography and cryptanalysis.

A cryptanalyst's chore is to break an encryption. That is the cryptanalyst attempts to deduce the original meaning of the cipher text message. Better yet, he or she hopes to determine which decrypting algorithm matches the encrypting algorithm so that other messages encoded in the same way can be broken. For instance, suppose two countries are at war and the first country has intercepted encrypted messages of the second. Cryptanalysis of the first country wants to decipher a particular message so that the first country can anticipate the movement and resources of the second. But it is even better to discover the actual decryption algorithm, then the first country can easily break the encryption of all message sent by the second country.

Thus, a cryptanalyst can attempt to do any of all or six different things

a. Break the single message
b. Recognize patterns in encrypted messages, to be able to break subsequence ones by applying a straightforward decryption algorithm
c. Infer some meaning without even breaking the encryption, such as noticing an unusual frequency of communication or determining something by whether the communication was long or short.
d. Deduce the key, to break subsequent messages easily
e. Find the weaknesses in the implementation or environment of use encryption.
f. Find the general weaknesses in an encryption algorithm, without necessarily having interception any messages.

There are few popular algorithm uses in the encryption today. Those are

a. RSA,
b. DES,
c. Blowfish,
d. RC4
e. SHA1

f. SHA2
g. ROT13
h. Vernam
i. Substitution
j. Caesar Cipher.

2.3.1 The RSA Encryption Algorithm

RSA is a popular encryption method that uses very large prime numbers to generate public and private keys. This algorithm was programmed by Ronald Rivest, Adi Shamir and Leonard Adelman. This algorithm based on concept of factoring, making it easy to encrypt but hard to be decrypt. RSA is one of the hardest algorithm that robust, and difficult to crack of the encryption standards currently used by application for secure storage and transmission of data. RSA uses factors containing roughly 150 digits, making it not impossible to be calculated manually, even with a powerful computer, one cannot crack the encryption in a reasonable timeframe. RSA is a block encryption algorithm, which means when a chunk of data is encrypted. It is first broken into a number of blocks. Each block is treated as a sequence of bits, with the number of digits being just a little less. The RSA encryption is among the safest encryption algorithms currently used in application. The keys are everywhere between 1024-2048 bits long, making them quite difficult to crack with brute force. One of disadvantage of the RSA encryption, it is slower than other types of encryption.

In RSA algorithm, it works by multiplying each other by a third number, for example, the two number X and Y, it can be calculated by multiplying each by a third number, N. if N is known only by the person who knows it, other who try to break the code will have a difficult time calculating X and Y. in addition, on RSA, N is a very large number, making the calculation even more difficult.. Indeed, RSA uses factors roughly 150 digits, making it hard.

When the message is decrypted, the recipient uses another special number, K, where (KE-1) is divisible by (P-1) (Q-1). The value of k is chosen such that

multiplying the encrypted message by itself K times, and then dividing the result by N, gives the original message as the remainder. To find out K, then, the values P and Q should be known. The values E and N constitute the public key, which can be freely distributed. The value K forms the private key, which should be kept secret.

The RSA algorithm is among the safest encryption algorithm currently used by application. Typically, RSA keys are anywhere between 1024-2048bits long, making them quite difficult to crack with brute force. The only vulnerability in the RSA encryption algorithm is man-in-the-middle impersonation attacks during the key distribution stage. If the sender and destination systems are able to securely exchange the keys, then the RSA is quite secure.

Data communication is an important aspect of our living. So, protection of data from misuse is essential. A cryptosystem defines a pair of data transformations called encryption and decryption. Encryption is applied to the plain text i.e. the data to be communicated to produce cipher text i.e. encrypted data using encryption key. Decryption uses the decryption key to convert cipher text to plain text i.e. the original data. Now, if the encryption key and the decryption key is the same or one can be derived from the other then it is said to be symmetric cryptography. This type of cryptosystem can be easily broken if the key used to encrypt or decrypt can be found.

Any practical implementation of the RSA cryptosystem would involve working with large integers (in our case, of 1024 bits or more in size). One way of dealing with this requirement would be to write our own library that handles all the required functions. While this would indeed make our application independent of any other third-party library, we refrained from doing so due to mainly two considerations. First, the speed of our implementation would not match the speed of the libraries available for such purposes. Second, it would probably be not as secure as some available open-source libraries are. There were several libraries to consider for our application. There are three choices of libraries: the Big Integer library (Java), the GNU MP Arbitrary Precision library (C/C++), and the OpenSSL crypto library (C/C++). Of these, the GMP library (i.e. the GNU MP library) seemed to suit our needs the most.

Timings for 1024-bit RSA (without compiler optimization) All the times recorded below have been measured on a 733 MHz Pentium class processor, using the time measurement functions offered by the C library on a GNU/Linux platform (kernel 2.4.20). Key generation: 0.465994 seconds (averaged over 5 samples) II-30.9 the following times were recorded while encrypting/decrypting a file with exactly 10,000 characters, shown in Table 2.1.

Table 2.1: Timing for 1024-bit RSA

Number of characters	Encryption time	Decryption time
1	5.068153	254.349886
25	0.219304	11.367492
50	0.128547	6.293279
75	0.096291	4.419277
100	0.078851	3.365591

By varying the constant representing the bit-strength, RSA module of other sizes may be used quite easily. The following times were recorded using the same input file (10,000 characters):

Table 2.2: Bit Strength in RSA module.

Bit Strength	Chars at once	Key Generation	Encryption	Decryption
512	50	0.057984	0.054362	0.903180
768	75	0.194653	0.065302	2.098904
1024	100	0.465994	0.078851	3.365591
1280	125	0.657473	0.089712	4.437929
1536	150	1.613467	0.105439	5.804798
1792	175	2.057411	0.116585	7.430849
2048	200	4.052181	0.126361	9.001286

The Table 2.2 shows the bit strength in RSA module. While the 512-bit RSA is definitely the fastest among the ones shown, it is not the most secure, providing marginal security from a concerted attack. The slowest (2048-bit RSA) should be used in critical situations since it offers the maximum resistance to attacks. In our opinion the 1024-bit modulus is a good balance between speed and security.

Encrypting sensitive data and managing the encryption keys can be a complex task in an enterprise environment. RSA has extensive experience in solving hard encryption and key management problems for companies across the globe. RSA is currently used for many applications like RSA SecureID, Digital Certificates, Smart Cards, etc. This algorithm is considered computationally unbreakable. i.e. it would take a very long time to break the code. Especially if we use large keys (1024 bits at least), it is almost impossible to find the private key to decode the cipher text. This is because the algorithm requires factoring two very large numbers.

There are few security protocols that uses RSA. Some of they are IPSEC/IKE for IP data security, TLS/SSL for transport data security, PGP for email security, SSH for terminal connection security and SILC, the conferencing service security.

2.3.2 The Data Encryption Standard Algorithm

Data Encryption Standard Algorithm or DES is and algorithm develops in 1977 by U.S government as the official standard for secure transmission of data within the system. It is often even suspected that the National Security Agency (NSA) is purposely left behind a backdoor in the DES algorithm in the interest of national security. DES algorithm is a block encryption system that transforms 64-bits data blocks under a 56-bit secret key system. It converts the fixed length input data into encrypted data with the help of a few complex functions. This algorithm is commonly used in various applications like SSL. Unfortunately DES algorithm has few issues which make it very susceptible to brute force cracking.

Unfortunately, even before the DES encryption algorithm was adopted as standard, a number of questions were raised related to its security. The length of the

keys, used by DES is only 56 bits, which makes it very susceptible to brute force cracking attacks. Even today, brute force cracking remains the most common attack against the DES standard. Once brute force cracking became popular, application developers came up with a quick fix in the form of Triple DES, where input text is passed through the DES algorithm three times in the following manner;

Input text =M

Triple DES Encrypted Text = DES (DES (DES (M)))

Triple DES successfully managed to increase the length of the key (from 56 bits to 168 bits) being used by the application and hence, was considered to be more secure. However, brute force takes a long time to crack data encrypted with the full 16 round of the DES algorithm. Many faster attacks, such as differential and linear cryptanalysis, have been proposed to crack the DES encryption algorithm, and they have been successful to varying extends. Unfortunately, both DES and Triple DES suffer from being quite slow and difficult to implement.

Already, security researchers around the world have recommended that alternate algorithms like RC5, blowfish, IDEA and AES be used instead of DES. The primary advantages of using AES instead opf DES is the fact that it uses a larger key (128, 192, and 256 bits) and a higher number of rounds for encryption. This makes AES more secure.