International Journal of Engineering and Technical Research (IJETR)
ISSN: 2321-0869 (O) 2454-4698 (P) Volume-7, Issue-5, May 2017

Survey and Analysis of Data Encryption Methods and Development of A Security Model to Encrypt/Decrypt Messages

Dr. Amit Mishra, Abdulrahman Abdulganiyu, Usman M Gana, Ayesha Awwal

Abstract— **Survey of the different encryption algorithms is carried out and also the analysis of those algorithms in other to determine their advantages and disadvantages. A message encryption and encryption architecture is being proposed and implemented for secure communication among the members of ibbu.edu.ng.**

Index Terms— **Encryption, Decryption, Security Model**

I. INTRODUCTION

It's no new news that we are gradually shifting into a digital world where all most everything is processed electronically. As good as it may sound; it will definitely have issues when it comes to unauthorized access to confidential data and private network. We need to provide a mechanism for protecting such data.

Although authentication is used to provide security to a certain extent but it takes far more than that when the network is been hijacked/compromised by hackers for personal interest. Encryption provides us with higher level of security because the data stolen will be of no use to the hacker because it's been converted to a cipher. A text, message or data that is converted into gibberish format using any the encryption algorithm is known as a cipher.

Encryption these days is used to protect unwanted access to data ranging from government confidential information, companies research, communication between two or more device (i.e. computers, smart-phones, tablets) and also our personal information and confidential data on our Personal computers.

Although encryption seems to provide the needed security on our data, definitely it has its own short comings because no system is perfect. A smart hacker can find such vulnerabilities and exploit the system so that he can have his hands on the data he needs. Most common problem of data encryption is: keeping key safe to ensure that only the intended receiver has the key to decrypt the message, having a reliable encryption algorithm that is difficult to hack.

This paper carries out Survey and analysis of the different encryption algorithms, and tries to find out the weakness in the commonly used data encryption algorithms, it also propose and implement software architecture for message encryption base on DES (data encryption standard) or AES (Advance encryption standards).

Dr. Amit Mishra, Department of Mathematics/Computer Science Ibrahim Badamasi Babangida University Lapai, Nigeria
Abdulrahman Abdulganiyu, Department of Mathematics/Computer Science Ibrahim Badamasi Babangida University Lapai, Nigeria
Usman M Gana, Department of Mathematics/Computer Science Ibrahim Badamasi Babangida University Lapai, Nigeria
Ayesha Awwal, Department of Mathematics/Computer Science Ibrahim Badamasi Babangida University Lapai, Nigeria

II. LITERATURES REVIEW

Cryptography is the discipline of writing a message in cipher text, usually by a translation from plaintext according to some (frequently changing) key text, with the aim of protecting a secret from adversaries, interceptors, intruders, interlopers, eavesdroppers, opponents or simply attackers, enemies. Professional cryptography protects not only the plaintext, but also the key and more generally tries to protect the whole cryptosystem

Cryptography is also the mathematics of making a system secure which is different from making the actual system secure (Schneier, 1995). It can also be defined as the science and art of securing your communication from unwanted person or adversary.

a. Encryption Schemes

The problem of providing secret communication over insecure media is the most traditional and basic problem of cryptography. The setting consists of two parties communicating over a channel that possibly may be tapped by a third party, called the wire-tapper or hacker. The parties wish to exchange information with each other, but keep the third party as ignorant as possible regarding the content of their communication. An encryption scheme is a protocol allowing two or more parties to communicate secretly with each other while keeping the third party as ignorant as possible regarding the content of their communication. This encryption scheme consists of a pair of algorithms.

One algorithm, called encryption, is applied by the sender while the other algorithm, called decryption, is applied by the receiver. Hence, in order to send a message, the sender first applies the encryption algorithm to the message and sends the result, called the ciphertext, over the channel. Upon receiving a ciphertext, the receiver applies the decryption algorithm to it and retrieves the original message called the plaintext.

Plaintext is denoted by M, for message, or P, for plaintext. It can be a stream of bits, a text file, a bitmap, and a stream of digitized voice, a digital video image etc. As far as a computer is concerned, M is simply binary data. The plaintext can be intended for either transmission or storage. In any case, M is the message to be encrypted. Ciphertext is denoted by C. It is also binary data: sometimes the same size as M, sometimes larger. (By combining encryption with compression, C may be smaller than M. However, encryption does not accomplish this.) The encryption function E, operates on M to produce C. Or, in mathematical notation:

$$E(M) = C \quad \ldots (1)$$

In the reverse process, the decryption function D operates on C to produce M:
D(C) = M… (2)
Since the whole point of encrypting and then decrypting a message is to recover the original plaintext, the following identity must hold true:
D (E (M)) = M … (3)

b. Algorithms and Keys

A cryptographic algorithm, also called a cipher, is the mathematical function used for encryption and decryption. (Generally, there are two related functions: one for encryption and the other for decryption.)
If the security of an algorithm is based on keeping the way that algorithm works a secret, it is a restricted algorithm. Restricted algorithms have historical interest, but are woefully inadequate by today's standards. A large or changing group of users cannot use them, because every time a user leaves the group everyone else must switch to a different algorithm. If someone accidentally reveals the secret, everyone must change their algorithm. (Schneier, 1995)
Modern cryptography solves this problem with a key, denoted by K. This key might be any one of a large number of values. The range of possible values of the key is called the key space. Both the encryption and decryption operations use this key (i.e., they are dependent on the key and this fact is denoted by the k subscript), so the functions now become:
$E_K(M) = C$ … (4)
$D_K(C) = M$ … (5)
Those functions have the property that
$D_K(E_K(M)) = M$ …(6)
Some algorithms use a different encryption key and decryption key. That is, the encryption key, K_1, is different from the corresponding decryption key, K_2. In this case:
$E_{K1}(M) = C$ …(7)
$D_{K2}(C) = M$ … (8)
$D_{K2}(E_{K1}(M)) = M$ …(9)
All of the security in these algorithms is based in the key (or keys); none is based in the details of the algorithm. This means that the algorithm can be published and analyzed. Products using the algorithm can be mass-produced. It doesn't matter if an eavesdropper knows your algorithm; if she doesn't know your particular key, she can't read your messages.

c. Symmetric Algorithms

Symmetric algorithms is one of the two types of encryption algorithms, sometimes called conventional algorithms, are algorithms where the encryption key can be calculated from the decryption key and vice versa. In symmetric algorithms, they use the same key for both encryption and decryption.

Figure 2.1: Diagram illustrating symmetric encryption.

Encryption and decryption with a symmetric algorithm are denoted by:

$E_K(M) = C$ ……equation 10
D(C) = M……equation 11
Symmetric algorithms are divided into two main categories. Some operate on the plaintext/ciphertext a single bit (or sometimes byte) at a time; these are known as stream algorithms or stream ciphers.

i.AES/Rijndael encryption

Rijndael is a block cipher, designed by Joan Daemen and Vincent Rijmen as a candidate algorithm for the AES. AES stands for Advanced Encryption Standard. AES is a symmetric key encryption technique which will replace the commonly used Data Encryption Standard (DES). The Advanced Encryption Standard algorithm approved by NIST in December 2001 uses 128-bit blocks.

ii. Blowfish

Blowfish is a symmetric encryption algorithm designed in 1993 by Bruce Schneier as an alternative to existing encryption algorithms. Blowfish has a 64-bit block size and a variable key length - from 32 bits to 448 bits. It is a 16-round Feistel cipher and uses large key-dependent S-boxes. While doing key scheduling, it generates large pseudo-random lookup tables by doing several encryptions.

iii. CAST

CAST stands for Carlisle Adams and Stafford Tavares, the inventors of CAST. CAST is a popular 64-bit block cipher which belongs to the class of encryption algorithms known as Feistel ciphers. CAST-128 is a DES-like Substitution-Permutation Network (SPN) cryptosystem. It has the Feistel structure and utilizes eight fixed S-boxes. CAST-128 supports variable key lenghts between 40 and 128 bits.

iv. Data Encryption Standard (DES)

In 1977 the Data Encryption Standard (DES), a symmetric encryption algorithm, was adopted in the United States as a federal standard. Digital Encryption Standard (DES) is a symmetric block cipher with 64-bit block size that uses a 56-bit key.

v. Triple DES

Triple DES encrypts data three times and uses a different key for at least one of the three passes giving it a cumulative key size of 112-168 bits. That should produce an expected strength of something like 112 bits, which is more than enough to defeat brute force attacks. Triple DES is much stronger than (single) DES; however, it is rather slow compared to some new block ciphers.

vi. Lucifer

Developed in the 1960s, by IBM initiated research program in computer cryptography, led by Horst Feistel and later led by Walt Tuchman. Lucifer is a substitution-permutation network, with building blocks similar to DES. In DES, the output of the function f is XORed with the input of the previous round to form the input of the next round. Table 2.1: Advantage and weakness in symmetric algorithms

Types	Advantages	Weakness
DES	Easy to be implemented. It is ideal for use on a special-purpose chip.	Susceptible to differential cryptanalysis and similar algorithms, linear cryptanalysis, brute force attack and improved davies attack
LUCIFER	Has a key schedule simpler than DES	Vulnerable to differential cryptanalysis and related key cryptanalysis
FEAL	Faster DES in encrypting plaintext	Vulnerable to cryptanalysis, brute force attack and linear cryptanalysis
CAST	A strong algorithm, not susceptible to linear and differential forms of cryptanalysis	Vulnerable to brute force attack
BLOWFISH	A strong algorithm, not susceptible to linear forms of cryptanalysis	Has a weak key which might make susceptible to differential attack

2.5 Public-Key Algorithms

Public-key algorithms (also called asymmetric algorithms) are designed so that the key used for encryption is different from the key used for decryption. Furthermore, the decryption key cannot (at least in any reasonable amount of time) be calculated from the encryption key.

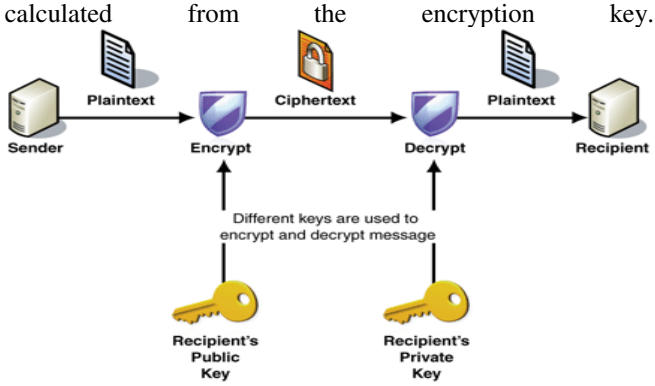


Figure 2.1: Diagram illustrating asymmetric encryption.

Encryption using public key K is denoted by:

$E_K(M) = C$ … (12)

Even though the public key and private key are different, decryption with the corresponding private key is denoted by:

$D_K(C) = M$ … (13)

Sometimes, messages will be encrypted with the private key and decrypted with the public key; this is used in digital signatures (see Section 2.6). Despite the possible confusion, these operations are denoted by, respectively:

$E_K(M) = C$ …(14)

$D_K(C) = M$ … (15)

2.5.1 Rivest Shamir Adleman (RSA)

Ron Rivest, Adi Shamir, and Len Adleman released the Rivest-Shamir-Adleman (RSA) public key algorithm in 1978. This algorithm can be used for encrypting and signing data. The encryption and signing processes are performed through a series of modular multiplications. The basic RSA algorithm for confidentiality can be explained as below.

Ciphertext = (plaintext)^e mod n

Plaintext = (ciphertext)^d mod n

Private Key = {d, n}

Public Key = {e, n}.

The basic RSA algorithm for authentication can be explained as below.

ciphertext = (plaintext)^d mod n

plaintext = (ciphertext)^e mod n

private key = {d, n}

public key = {e, n}

2.5.2 Elliptic Curve Cryptography (ECC)

Elliptic Curve Cryptography (ECC) provides similar functionality to RSA. Elliptic Curve Cryptography (ECC) is being implemented in smaller devices like cell phones.

2.5.3 El Gamal

El Gamal is an algorithm used for transmitting digital signatures and key exchanges. The method is based on calculating logarithms. Its algorithm is based on the characteristics of logarithmic numbers and calculations. The Digital Signature Algorithm (DSA) is based on El Gamal algorithm.

2.5.4 Digital Signature Algorithm (DSA)

The Digital Signature Algorithm (DSA) was developed by the United States government for digital signatures. Digital Signature Algorithm can be used only for signing data and it cannot be used for encryption. The DSA signing process is performed through a series of calculations based on a selected prime number. Although intended to have a maximum key size of 1,024 bits, longer key sizes are now supported. ElGamal digital signature scheme and borrows ideas from Schnorr digital signatures for reducing signature size. We describe a slight generalization of the algorithm that allows for an arbitrary security parameter, whereas the standard only supports a fixed parameter.

TABLE 2.2: Advantage and Weakness in asymmetric algorithms

Types	Advantages	Weakness
DES	Easy to be implemented. It is ideal for use on a special-purpose chip.	Susceptible to differential cryptanalysis and similar algorithms, linear cryptanalysis, brute force attack and improved davies attack
LUCIFER	Has a key schedule simpler than DES	Vulnerable to differential cryptanalysis and related key cryptanalysis
FEAL	Faster DES in encrypting plaintext	Vulnerable to cryptanalysis, brute force attack and linear cryptanalysis
CAST	A strong algorithm, not susceptible to linear and differential forms of cryptanalysis	Vulnerable to brute force attack
BLOWFISH	A strong algorithm, not susceptible to linear forms of cryptanalysis	Has a weak key which might make susceptible to differential attack

2.5 Cryptanalysis

The whole point of cryptography is to keep the plaintext (or the key, or both) secret from eavesdroppers (also called adversaries, attackers, interceptors, interlopers, intruders, opponents, or simply the enemy). Eavesdroppers are assumed to have complete access to the communications between the sender and receiver.
There are four general types of cryptanalytic attacks. Each of them with the assumptions that the cryptanalyst has complete knowledge of the encryption algorithm used in the system:

1. Ciphertext-only attack. The cryptanalyst has the ciphertext of several messages, in which all of them have been encrypted using the same encryption algorithm. The cryptanalyst's job is to recover the plaintext of as many messages as he can possibly recover, or to deduce the key or keys used to encrypt these messages, in order to decrypt other messages encrypted with the same keys.
Given: $C_1= E_k(P_1)$, $C_2= E_k(P_2)$,...$C_i= E_k(P_i)$
Deduce: Either P_1, P_2,...P_i; k; or an algorithm to infer $P_{i+1}$from $C_{i+1}= E_k(P_{i+1})$

2. Known-plaintext attack. The cryptanalyst has access not only to the ciphertext of several messages, but also to the plaintext of those messages. His job is to deduce the key (or keys) used to encrypt the messages or an algorithm to decrypt any new messages encrypted with the same key (or keys).
Given: P_1, $C_1= E_k(P_1)$, P_2, $C_2= E_k(P_2)$,...P_i, $C_i= E_k(P_i)$
Deduce: Either k, or an algorithm to infer $P_{i+1}$from $C_{i+1}= E_k(P_{i+1})$

3. Chosen-plaintext attack. The cryptanalyst not only has access to the ciphertext and associated plaintext for several messages, but he also chooses the plaintext that gets encrypted. This is more powerful than a known-plaintext attack, because the cryptanalyst can choose specific plaintext blocks to encrypt, ones that might yield more information about the key. His job is to deduce the key (or keys) used to encrypt the messages or an algorithm to decrypt any new messages encrypted with the same key (or keys).
Given: P_1, $C_1= E_k(P_1)$, P_2, $C_2= k(P_2)$,...P_i, $C_i= E_k(P_i)$, where the cryptanalyst gets to choose P_1, P_2,...P_i
Deduce: Either k, or an algorithm to infer $P_{i+1}$from $C_{i+1}= E_k(P_{i+1})$

4. Adaptive-chosen-plaintext attack. This is a special case of a chosen-plaintext attack. Not only can the cryptanalyst choose the plaintext that is encrypted, but he can also modify his choice based on the results of previous encryption. In a chosen-plaintext attack, a cryptanalyst might just be able to choose one large block of plaintext to be encrypted; in an adaptive chosen-plaintext attack he can choose a smaller block of plaintext and then choose another based on the results of the first, and so forth.
There are at least three other types of cryptanalytic attack.

5. Chosen-ciphertext attack. The cryptanalyst can choose different ciphertexts to be decrypted and has access to the decrypted plaintext. For example, the cryptanalyst has access to a tamperproof box that does automatic decryption. His job is to deduce the key.
Given: C_1, $P_1= D_k(C_1)$, C_2, $P_2= D_k(C_2)$,...C_i, $P_i= D_k(C_i)$
Deduce: k
This attack is primarily applicable to public-key algorithms. A chosen-ciphertext attack is sometimes effective against a symmetric algorithm as well. (Sometimes a chosen-plaintext attack and a chosen-ciphertext attack are together known as a chosen-text attack.)

6. Chosen-key attack. This attack doesn't mean that the cryptanalyst can choose the key; it means that he has some knowledge about the relationship between different keys. It's strange and obscure, not very practical.

7. Rubber-hose cryptanalysis. The cryptanalyst threatens, blackmails, or tortures someone until they give him the key. Bribery is sometimes referred to as a purchase-key attack. These are all very powerful attacks and often the best way to break an algorithm.

2.6 Security of Algorithms

Different algorithms offer different level of security; it all depends on how hard they are to break. If the cost required to break an algorithm is greater than the value of the encrypted data, then you're probably safe. Lars Knudsen classified these different categories of breaking an algorithm. In decreasing order of severity:

1. Total break. A cryptanalyst finds the key, K, such that $D_K(C) = P$.
2. Global deduction. A cryptanalyst finds an alternate algorithm, A, equivalent to $D_K(C)$, without knowing K.
3. Instance (or local) deduction. A cryptanalyst finds the plaintext of an intercepted
ciphertext.
4. Information deduction. A cryptanalyst gains some information about the key or plaintext.
This information could be a few bits of the key, some information about the form of the plaintext, and so forth.
An algorithm is unconditionally secure if, no matter how much ciphertext a cryptanalyst has, there is not enough information to recover the plaintext.
Cryptography is more concerned with cryptosystems that are computationally infeasible to break. An algorithm is considered computationally secure (sometimes called strong) if it cannot be broken with available resources, either current or future. Exactly what constitutes "available resources" is open to interpretation.

Complexity can be measured in different ways:

1. Data complexity. The amount of data needed as input to the attack.
2. Processing complexity. The time needed to perform the attack. This is often called the work factor.
3. Storage requirements. The amount of memory needed to do the attack.
Still, if you assume that you have enough computing speed to perform a million operations every second and you set a million parallel processors against the task, it will still take over 10^{19} years to recover the key. That's a billion times the age of the universe.

III. METHODOLOGY

A. SOFTWARE REQUIREMENT

The software needed by the new system are series of multi-user operating system that will support, control and co-ordinate the activities of the hardware. Therefore, the software requirements include the following:

1. Multi – user operating system like Window XP or later versions, Unix, Linux etc.
2. A webserver like Apache, IIS for running your PHP code.
3. Web browser like Mozilla, Internet explorer, Opera, Safari etc. to view the front end of the system.
4. A database management system, MySQL server preferably for keeping of needed record.
5. Jquery development library.
6. PHP 5 which is the language used in implementing our encryption and decryption model.

B. Program Design Methodology

The program for this system was designed using top down approach. The problem was broken down into simple units. The units were solved separately and later built up to form a whole system. The program was design under which each module performs a specific operation; the modules were later joined together using menu options for effective operation.

C. Encryption/Decryption Scheme

Rijndael/AES was used to in encrypting and decrypting the message. The AES is a symmetric block algorithm whose plaintext and ciphertext block size is 128 bits, and its keys can be 128, 192, or 256 bits long.

D. Proposed Database Design

The proposed database was designed in a way that it would suit the application flow and all the entities of the application. The database consists of the following main entities to record the application flow: users and messages.

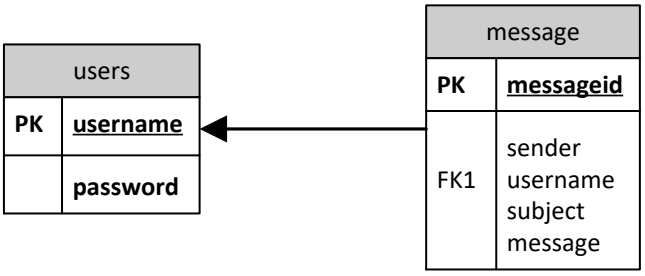


Figure 4.1: Database structure and entity relationship for the implementation of the project

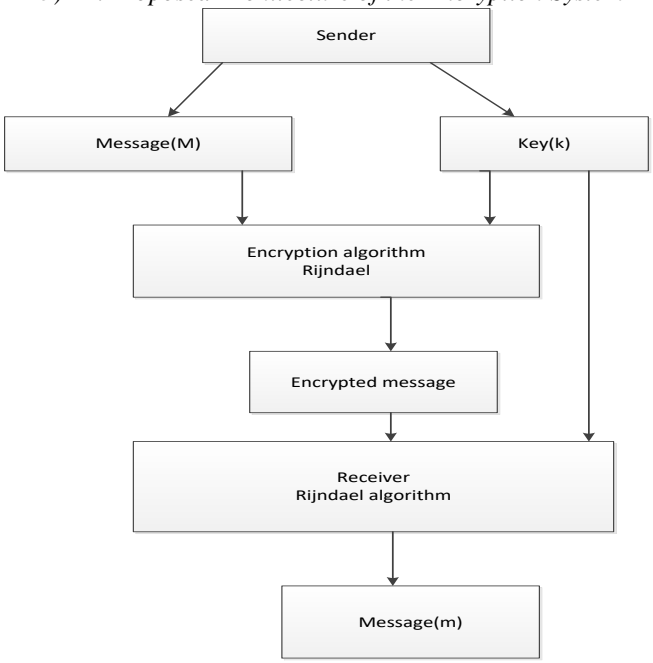
1) 1. Users

The users entity consist all the usernames and password of the users of the system. The username is used in establishing a session for security of the program.

2) 2. Message

This entity contains all the information that is related to a message and which belongs to a user. The messageid in this entity includes a unique ID allotted to each record in this table to uniquely identifying the message and username which specifies to whom the message belongs to, message subject, sender and date the message was sent. The original message is stored in encrypted format in the database. For encrypting the documents the keys are generated at the runtime by the sender of the message.

3) E. Proposed Architecture of the Encryption System



B. SYSTEM FLOW CHART

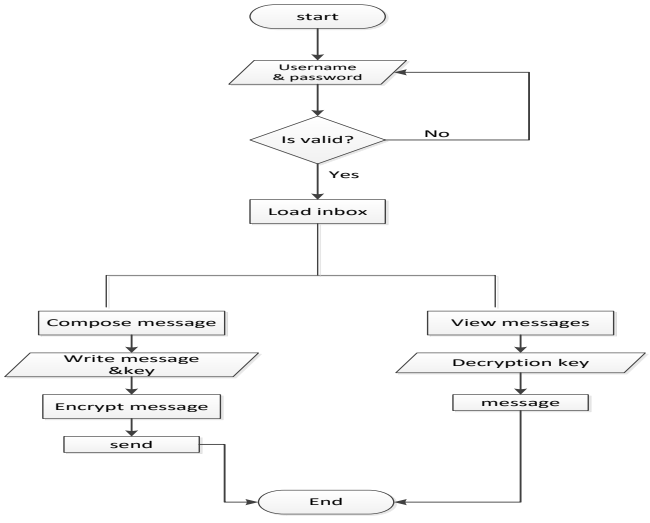
The encryption system is made up of file(s) sub-system namely.

i. Control subsystem: - this provides an interface between the user and the system and the various sub systems. It is the part that comes to play when the system is run.

ii. Inbox subsystem: - this displays each of the messages sent to that user. It displays the message sender, subject and date in a tabular form.

iii. Encryption subsystem: - this describes encryption and sending of messages to the desired receiver.

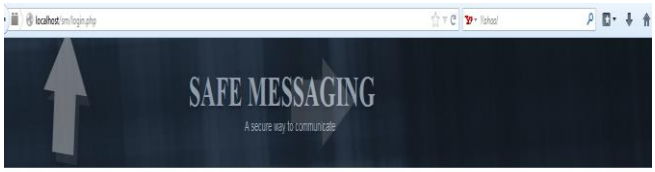
iv. Decryption subsystem: - this describes decryption and receiving of messages from the sender

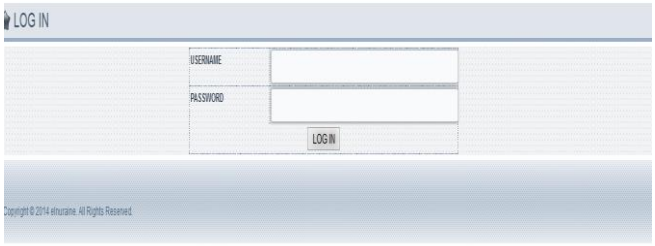


Flow diagram of the system

A. Login Interface

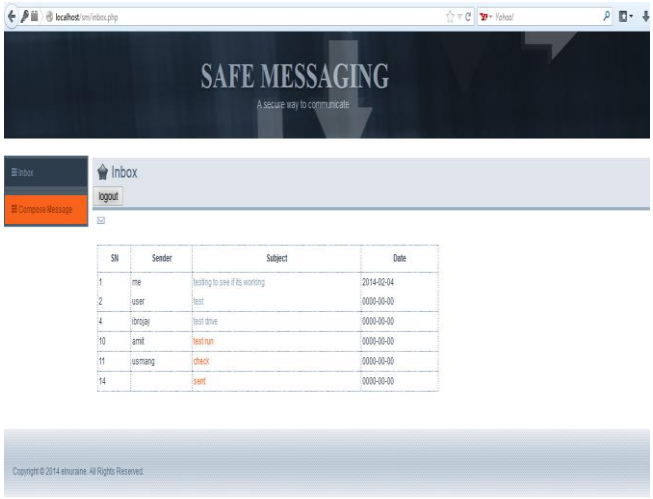
The login interface is the first interface you will come across in the system. It carries out the following; performs authentication of the users and create session for managing access





B. Inbox Interface

This displays each of the messages sent to that user. It displays the message sender, subject and date in a tabular form.

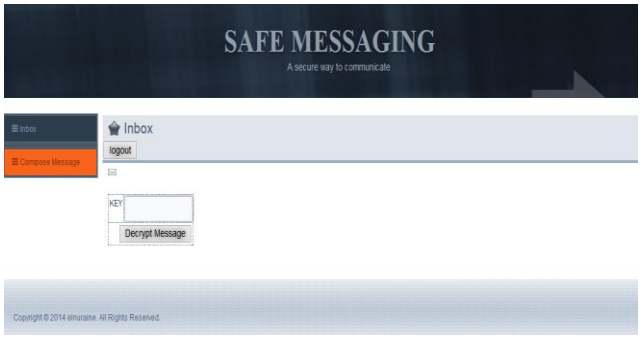


C. Encryption Interface

This is the interface where the message is being written and encrypted using the encryption key.

D. Decryption interface

This is where messages are being decrypted and used viewed using the decryption key.



REFRENCES

Asymmetric Algorithms. (n.d.). Retrieved january 13, 2013, from encryption and decryption: http://www.encryptionanddecryption.com/algorithms/asymmetric_algorithms.html

[1] Huang, J. Z.-z. (2010). A Method for Secure Real-Time Image Transmission Based on Optical Encryption. *international conference on the Intelligent Signal Processing and Communication Systems.*

[2] Kuo, D. X.-C. (2004). Enhanced multiple Huffman table (mht) . *IEEE Transactions*, 568-571.

[3] Mort Naraghi-Pour, V. S. (2011). Secure Detection in Wireless Sensor Networks Using a Simple Encryption Method . *IEEE transactions*.

[4] *public key encryption*. (n.d.). Retrieved january 13, 2013, from encryption and decryption: http://www.encryptionanddecryption.com/encryption/public_key_encryption.html

[5] Rajashekarappa, K. M. (2013). Study on Cryptanalysis of the Tiny Encryption. *International Journal of Innovative Technology and Exploring Engineering, 2*(3), 88-89.

[6] Schneier, B. (1995). *Applied Cryptography:Protocols, Algorithms, and source code in c.* Wiley.

[7] Suhaila O. Sharif, L. K. (2010). Classifying Encryption Algorithms Using Pattern Recognition techniques. *IEEE Transactions*, 1168-1172.

[8] *symmetric algorithms*. (n.d.). Retrieved January 13, 2014, from Encryption And Decryption: http://www.encryptionanddecryption.com/algorithms/symmetric_algorithms.html

[9] Tilborg, H. C. (2005). *Encyclopedia of Cryptography and secuirty.* Springer.