P- 1706

KCT LIBRARY COIMBATORE-641 006

SECURE MAIL TRANSACTION USING PGP

By

AKHIL. V
Reg. No. 71203621002
Of

KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE

A PROJECT REPORT
Submitted to the

FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING

In partial fulfillment of the requirements
for the award of the degree
of

MASTER OF COMPUTER APPLICATIONS
JUNE 2006

Kumaraguru College of Technology

Coimbatore – 641006.

Department of Computer Applications

Bonafide Certificate

Certified that this project report titled **Secure Mail Transaction using PGP** is the bonafide work of Mr. **AKHIL. V** who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Project Guide

Head of Department

Submitted for the University Examination held on 30-6-06

20/6/06
Internal Examiner

30/6/06
External Examiner

DOEACC CENTRE CALICUT
(A Unit of DOEACC Society)
An Autonomous Body of Department of Information Technology,
Ministry of Communications and Information Technology, Government of India
POST BOX No.5, P.O.NIT CAMPUS, CALICUT-673601, KERALA, INDIA
☎ (0495) 2287266, Fax # (0495) 2287168, Email: info@cedtic.com, Web: www.cedtic.com

CERTIFICATE

This is to certify that

Akhil V

*of Final Year MCA,Kumaraguru College of Technology, Coimbatore, has successfully completed his project work entitled **"Secure Mail Transaction using PGP"**, at the **Information Technology Group, DOEACC CENTRE CALICUT,** as part of his MCA project work under the guidance of **Mrs. Saniya. A**, Senior Designer Engineer, during the period January '06 to May '06.*

Saniya A
30/May/06
External Guide
DOEACC CENTRE CALICUT

30/5/'06
Student Project Coordinator
DOEACC CENTRE CALICCUT

DOEACC CENTRE CALICUT * NIT Campus Post

ABSTRACT

Unauthorized access of messages and data over the network is an important problem faced by many at present. Usage of encryption has satisfied this problem to a great extent. But still most of the encryptions are susceptible to be broken by hackers and intruders. This project is aimed at developing a mailing system that could withstand illegal view or access to a great extent. It ensures both confidentiality as well as authentication thereby guaranteeing a high degree of reliability.

Even with the usage of encryption algorithms the messages are susceptible to attacks such as Trojan Horse Attack, Brute Force Attack etc. So great care should be taken in designing and selecting an algorithm that would ensure the needs specified i.e. confidentiality

Mainly aimed at sending and receiving confidential messages, this project is developed with the view of usage for normal, not so-confidential mails as well. Hence importance is to be given to other critical issues such as time, end user friendliness, user interface design etc.

ACKNOWLEDGEMENT

I express my deepest thanks to **Dr. Joseph. V. Thanikal**, Principal and **Dr. K. K. Padmanabhan**, former Principal, Kumaraguru College of Technology, Coimbatore for extending their help by providing all the necessary facilities within the college.

I thank **Dr. Gururajan**, Head of the Department, Dept. of Computer Applications for the support offered throughout the project.

I am greatly indebted to **Dr. G. M. Ajit**, Director, DOEACC Centre, Calicut for permitting me to do the project in the esteemed institution and providing me with all the required facilities for the successful completion of the project.

I would like to express my deepest and sincere gratitude and respect to my external guide **Mrs. Saniya. A**, Senior Design Engineer, DOEACC Centre, Calicut for the suggestions and guidance rendered throughout the project.

I extend my sincere thanks to **Mrs. V. Geetha**, Assistant Professor and my guide **Mr. Jayakanthan**, Lecturer, Dept of Computer Applications, Kumaraguru College of Technology, Coimbatore for guiding me with valuable suggestions and support for the successful completion of this project. .

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

PGP	Pretty Good Privacy
AES	Advanced Encryption Standard
SHA1	Secure Hash Agorithm
RSA	Rivest-Shamir-Adelman
MIME	Multipurpose Internet Mail Extension
SMTP	Simple Mail Transfer Protocol
POP	Post Office Protocol
H	Hash Function
EP	Public Key Encryption
EC	Symmetric Key Encryption
DP	Public Key Decryption
DC	Symmetric Key Decryption
Ks	Session Key for Symmetric Encryption
KR	Private Key of Mail Sender/Receiver
KU	Public Key of Mail Sender/Receiver

CHAPTER 1

INTRODUCTION

The requirements of information security either within an organization or between individuals have undergone widespread changes in the last several decades. Network security measures are needed for data either as plain text messages or as binary files during their transmission over any network. Security is critical to mail center operations, both large and small. Lack of security can result in theft of supplies, postage, mail, and valuable information about your business contained in sensitive mail.

Mailing systems has become one of the most popular mediums for sending and receiving messages in the current world. With the emergence of e-mails, large numbers of messages are sent daily through mailing services provided by commercial mailing service providers. A survey conducted by experts In June 2001 revealed that one out of every 1000 emails was infected with a computer virus. Normally sending normal messages does not need a high degree of attention. But today that is not the case.

Many organizations, institutions and even individuals rely on mailing services for sending and receiving highly important and confidential messages. In-between retrieval or unauthorized access or view of such messages could lead to undesirable events. The use of local mailing services or web-based email poses security challenges that every organization or individual should consider. In most cases, the risk of allowing users to access mails while at work will outweigh any potential benefit. Without realizing it, users are bypassing many of the information security measures of an organization by using mail accounts.

Employing Encryption Algorithms in conjunction with mails to ensure the integrity of their confidential information is the best possible solution to this problem. But the challenge lies in which algorithm one chooses and how it is applied since it's a cumbersome process for normal users to encrypt their text before sending it, via local or web based email. If an organization is planning to allow its users to send encrypted email via a web based service, end user training needs to be developed to teach users to do it properly.

CHAPTER 2

BASICS OF MAIL SECURITY

Security involving communication and networks is not as simple as it might first appear. The requirement seems to be straight forward such as

- Confidentiality
- Authentication
- Integrity etc

But the mechanisms used to meet those requirements can be quite complex, and understanding them may involve rather subtle reasoning. In developing a security system one must consider potential attacks on those security features. Having designed the various security mechanisms, it is necessary to decide on where to use them especially when one is working for security over a network medium.

Security mechanisms usually involve more than a particular algorithm or protocol. They also require that the participants be in possession of some secret information (e.g. an encryption key), which raises question about the creation, distribution, and protection of that secret information.

Basically Information Security can be viewed as three individual aspects.

Security attack : Any action that compromises the security of information owned by an organization or individual

Security mechanism : A mechanism that is designed to detect, prevent or recover from a security attack.

Security Service : A service that enhance the security of the data processing systems and the information transfers of an organization. The service providers make use of one or more security mechanisms to provide the service.

Attacks on mails containing plain text or binary files can be broadly classified into two:

- Passive Attacks
- Active Attacks

While passive attacks can only observe communication or data, active attacks can actively modify communications or data. Mail Forgery, mail modification and Session hijacking are popular examples for active attacks.

Figure 2.1 : Passive Attack

Figure 2.2 : Active Attacks

2.1 SECURITY MECHANISMS

One or more security mechanisms are combined to provide a security service. The main three building blocks are

- Encryption is used to provide confidentiality, can provide authentication and integrity protection
- Digital signatures are used to provide authentication, integrity Protection and non-repudiation
- Checksums/hash algorithms are used to provide integrity Protection, can provide authentication

2.2 SECURITY SERVICES

From the OSI definition:

- Access control: Protects against unauthorized use
- Authentication: Provides assurance of someone's identity
- Confidentiality: Protects against disclosure to unauthorized identities
- Integrity: Protects from unauthorized data alteration
- Non-repudiation: Protects against the originator of communications later denying it

2.3 INTRUDERS

A significant security problem for networked systems is the hostile, or at least unwanted, trespass by users or software. Software trespass can take the form of a virus, worm or Trojan horse. A user with access to a local terminal may attempt trespass without using an intermediate network. A virus or a Trojan horse may be easily introduced to a system by simple means. Only the worm is a uniquely network phenomenon.

One of the most publicized threats to security is the intruder(the other is viruses), generally referred to as hacker or cracker. There are three classes of intruder:

- **Masquerader :** An individual who is not authorized to use the computer but penetrates a system's access controls to exploit a legitimate user's account.
- **Misfeasor :** A legitimate user who accesses data, programs, or resources for which such access is not authorized.

- **Clandestine User :** An individual who seizes supervisory control of the system and uses this control to evade auditing and access controls or to suppress audit control.

The general objective of the intruder is to gain access to a system or to increase the range of privileges accessible on a system. This requires intruders to acquire information that should have been protected. In most case, this information is in the form of a user password.

Typically a system must maintain a file that associates a password with each authorized user. If such a file is stored with no protection, then it is an easy matter to gain access to it and learn the passwords. These password files can be protected using one of the two ways :

- **One-way encryption :** The system stores only an encrypted form of the users password. When the user presents a password, the system encrypts that password and compares it with the stored value. In practice, the system usually performs a one-way transformation in which the password is used to generate a key for the encryption function and in which a fixed length output is produced.
- **Access Control :** Access to the password file is limited to one or a very few accounts.

If one or both these countermeasures are in place, some effort is needed for a potential intruder to learn passwords.

Some of the common basic techniques for cracking a password are

- Try default password with standard accounts that are shipped with the system.
- Exhaustively try all short passwords(Trial & Error).

- Try words in the system's online dictionary or a list of likely passwords.
- Try passwords based on information collected about the user, such as their full name, name of their children etc.
- Use a Trojan horse to bypass restrictions on access.
- Tap the line between a remote user and the host system.

CHAPTER 3

ANALYSIS OF THE PROBLEM

The proposed system has to be analyzed well for its merits and demerits. Our main aim is to make use of all its merits and to remove the demerits. Initially the proposed system is analyzed, and then the merits of the system are discussed. The requirement specification of the proposed system should also be analyzed so that the hardware and software requirements of the proposed system are known in order to execute our application.

The documentation of the project is made with the details that best describe the system. The graphical representation of the system and its activities are presented in order to make the users understand the working of the system. The records and description of the system elements are analyzed and maintained.

3.1 PROBLEM DEFINITION

Mails carrying confidential corporate information can create immense inconvenience and expense for a company that has not equipped its mailing system with the appropriate security measures.

The threat of information leaks

Organizations often fail to acknowledge that there is a greater risk of crucial data being stolen from within the company rather than from outside. Most electronic mail is notoriously UNPRIVATE. Sending an email is less secure and in many ways is

more dangerous than sending your personal or business message on a postcard. Various studies have shown how employees use e-mail to send out confidential corporate information. Be it because they are disgruntled and revengeful, or because they fail to realize the potentially harmful impact of such a practice, employees use e-mail to share sensitive data that was officially intended to remain in-house.

The threat of mail interception and tampering

Unsecured mails can fall prey to malicious software tools such as Sniffers, which automatically lie in wait for interesting information relayed through their system as e-mails are transferred from sender to recipient. Unknown to e-mail senders, Sniffers are placed in the path of all mail or e-mail messages going through a computer. The individual who has planted this device receives copies of all the messages passing through and is then in a position to read – and store – those deemed interesting or profitable information. The situation is more worrying when the intentions behind Sniffers are injurious, stemming from a desire to spy on competitors, enemies, the wealthy, or whatever. Apart from scanning and intercepting mail traffic, it is also possible for unauthorized people to tamper with mails so that the message reaching the recipient is not the one originally sent by its author.

A 1998 survey sponsored by the US Senate found that 12.6% of Fortune 1000 companies reported evidence of e-mail tampering. Security experts believe the actual incidence of tampering is even higher – a reasonable supposition, considering that over 200 million e-mail messages are sent daily.

Relying on the server to do all the security related issues would be impractical since the illegal retrieval of message contents might happen on the client-server path.

3.2 SYSTEM STUDY

The existing system has to be analyzed well for its merits and demerits. Our main aim is to make use of all the merits and to remove the demerits. First we must analyze the existing system and we have to find the drawbacks and difficulties faced by using this system. The new system is proposed and analyzed to find out whether it satisfies the requirements and studies are conducted to justify the feasibility of the system under various user environments.

3.2.1 Existing System And Its Drawbacks

- Almost all of the mailing systems today does not guarantee the security needed for transmitting highly confidential messages
- None of the commercial mail service providers employ encryption
- Confidentiality and Authentication of messages are not provided
- E-mails use only base64 encryption before sending mails. Base64 is only an encoding scheme(different representation) and hence does not ensure security

3.2.2 Proposed System

- Employ the best of encryption algorithms
- Use Pretty Good Privacy (**PGP**) security scheme
- Usage of Encryption technique on messages to be sent
- Both public key encryption and symmetric key encryptions are to be used to guarantee efficiency both in terms of time and reliability
- Security issues dealt at the client side itself thereby making the client-server path secure
- Ensure both Confidentiality as well as Authentication

3.2.3 Objectives Of The Proposed System

- The system proposes a real time solution for protecting the confidentiality and integrity of mails sent by an individual or organization
- Reduce the burden of encryption details from the end users
- Make the system highly effective in terms of time and cost factors

3.3 FEASIBILITY STUDY

Feasibility study refers to the overall idea of the package we are designing. Software feasibility has mainly three solid dimensions namely
Technical – Is the project technically feasible? Is it within the state of art? Can the existing defects be reduced to a level matching the application's needs?
Economical – Is the system economically feasible? Can development be completed at a cost the software organization, clients, or the market can afford?
Operational – This study tells about how this package is useful to the end users, its advantages and disadvantages, whether the system is cost effective or not, etc

3.3.1 Technical Feasibility

The system developed using java does not require any high level software for its effective working. Developed completely in java the system is guaranteed on most new popular operating systems. The system is even capable of working even on a moderate hardware configuration. Hence the proposed system is technically feasible.

3.3.2 Operational Feasibility

The system developed is highly user friendly. The complexities of advanced encryption and networking protocols are hidden from the user. The user interface is designed using Java Swing and it enables any user without much technical knowledge to use the system for sending messages.

CHAPTER 4

DESIGN AND DEVELOPMENT

The chapter describes how the Secure Mail Transaction system for sending and receiving mail is designed and how the user can make use of it, what are the processes taking place, the inputs and outputs to the system etc. The overall step-by-step process of design and how it is implemented, the hardware and software requirements, and the developing environment used is also specified.

4.1 SYSTEM DESIGN

Based on a study conducted as to what should be the mechanism employed to ensure security for the messages transmitted, it was concluded that neither a public key encryption nor a symmetric key algorithm alone could guarantee confidentiality as well as authentication.

Another important consideration is cost in terms of time. A symmetric key algorithm is many times faster than a public key algorithm, but one cannot rely on symmetric encryption alone since (i) it is comparatively easier to break a symmetric encryption as it deals with a single key (ii) symmetric keys does not ensure authentication (iii) extra measures have to be taken to keep the key secure.

Based on the above conclusions it was inferred that **P**retty **G**ood **P**rivacy (PGP) would be the most suitable algorithm to be employed in the given problem scenario. PGP provides a confidentiality and authentication service that can be used for electronic mail and file storage applications.

4.1.1 Pretty Good Privacy

PGP algorithm utilizes both symmetric key and public key algorithms for its working. Other important features of PGP includes

- Selected the best available cryptographic algorithms as building blocks
- Integrated these algorithms into a general-purpose application that is independent of Operating System and Processor.
- It is not developed by nor is it controlled by any governmental or standards organization.

Operational Description – Authentication

- The sender create a message
- SHA1 is used to generate a 160-bit hash code of the message
- The hash code is encrypted with RSA using senders private key, and the result is prepended to the message
- The receiver uses RSA with the sender's public key to decrypt and recover the hash code

Operational Description – Confidentiality

- The sender generates a message and a random 128-bit number to be used as a session key for the message.
- The message is encrypted using AES with the session key
- The session key is encrypted with RSA, using the recipient's public key and is prepended to the message
- The receiver uses RSA with is private key to decrypt and recover the session key.
- The session key is used to decrypt the message

PGP System - Encryption

The encryption part uses the following algorithms

▪ Hashing Algorithm	:	**SHA1**
▪ Signature Algorithm	:	**RSA**
▪ Encryption Algorithm	:	**AES**

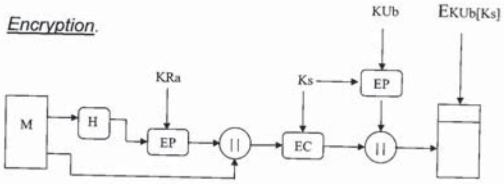


Figure 4.1 : PGP Encryption

PGP System – Decryption

Decryption

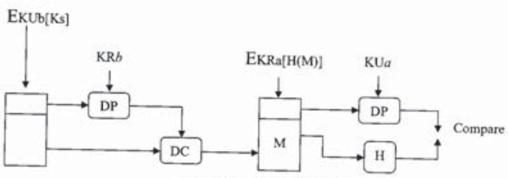


Figure 4.2 : PGP Decryption

4.2 MODULAR DESIGN

Dividing it into separately addressable components called modules that are integrated at the final stage to satisfy the problem requirements develops the software. By enforcing consistency and integrity, modular design makes the work not only easier, but better and understandable. The modules are presented below.

- Digest Generation
- RSA Signature Generator
- AES Cipher Encryption and Decryption
- RSA Signature Verification
- KeyStore Manager
- JavaMail SMTP Sender
- JavaMail POP3 Receiver
- GUI Design

4.3 DETAILED DESIGN

Detailed design is done to support the following :

- Make as many design decisions before coding starts
- Provide for developer design review
- Provide Programmer Documentation

Digest Generator

The module deals with creating a message digest for the input text message by applying a one-way hash function on it. A hash function is a public function that maps a message of any length into a fixed-length hash value, which serves as the authenticator.

SHA1 Logic is employed to generate the message digest for the message. The algorithm takes as input a message with a maximum length of 2^64 bits and produces as output a 160-bit message digest.

RSA Signature Generator

This module deals with the authentication process in the project. RSA being one of the most popular public key encryption algorithms, is employed for generating the digital signature. The module generates two keys(public key and private key) for each user. The private key is kept secret while the public key is made open to all users. Applying RSA to the message with the private key of the sender generates the Signature.

RSA uses mainly mathematical calculations for developing the digital signatures and generating the keys for developing the same. The RSA scheme is a block cipher in which the plaintext and cipher text are integers between 0 and n-1. A typical value for n is 1024 bits, or 309 decimal digits.

AES Cipher Encryption and Decryption

The module is responsible for ensuring confidentiality to the message being transmitted. AES (Advanced Encryption Standard) is a symmetric key algorithm which employs only a single key for both encryption and decryption. AES uses a 128 bit key when compared to DES which had only a 56-bit key size. The block size for AES cipher is also 128-bits.

The main features of AES are:

- Resistance against all known attacks
- Speed (Symmetric key algorithms are generally faster than public key algorithms)

The AES encryption module acts at the sending end generating a symmetric key and converting the input to block AES cipher. The decryption module receives the cipher and key and does the reverse process.

RSA Signature Verification

Employing RSA algorithm the module verifies the signatures generated by the RSA Signature Generator using the public key of the sender who has signed the message. This is to ensure complete authentication since nobody other than the sender could have signed the message with his private key.

KeyStore Manager

KeyStore Manager is responsible for management of keys, both public and private. KeyTool class is used to manage KeyStore (database of private keys and their associated X.509 certificate chains authenticating the corresponding public keys). KeyTool is a key and certificate management utility. It enables users to administer their own public/private key pairs and associated certificates for use in self-authentication (where the user authenticates himself/herself to other users/services) or data integrity and authentication services, using digital signatures. It also allows users to cache the public keys (in the form of certificates) of their communicating peers. KeyTool stores the keys and certificates in a so-called KeyStore.

4.4 JAVA MAIL API

The JavaMail API is an optional package (standard extension) for reading, composing, and sending electronic messages. The Application Programmer Interface, offers a protocol-independent model for working with IMAP, POP, SMTP, MIME, and all those other Internet-related messaging protocols. Java Mail is used to send and receive encrypted messages over the network. This module deals with the communication aspects of the project, the protocol used to transfer the messages etc.

SMTP protocol is employed for sending the messages. It defines the mechanism for delivery of e-mail. In the context of the JavaMail API, the JavaMail-based program will communicate with the organization or Internet Service Provider's (ISP's) SMTP server. That SMTP server will relay the message on to the SMTP server of the recipient(s) to eventually be acquired by the user(s) through any appropriate protocol. This does not require your SMTP server to be an open relay, as authentication is supported, but it is your responsibility to ensure the SMTP server is configured properly. POP3 protocol is employed at the receiving end i.e. to receive the messages. It defines support for a single mailbox for each user.

Other important tasks in sending a message using JavaMail API are

- Retrieving a session
- Setting session properties
- Generating the addresses
- Setting the message content and type
- Setting the login authentication
- Sending/Receiving the message with the help of Transport class

4.5 GRAPHICAL USER INTERFACE

The user interface for the project is provided which ensures a user-friendly graphical interface for the users to interact with the system, thereby hiding all the technical and algorithm complexities from the user. The GUI is developed using Java Swing and provides a means of user interaction with the system.

The user screen is designed with immense care taking into account aspects like avoiding ambiguity, avoiding technical jargons, hiding mail API aspects etc from the user, making the software useful even to a common man.

The important tasks the end user has to perform are

- Enter user name and password for validation

- Enter the pass phrase for private/public keys
- Enter the text message

The mailing features provided to the end users are as follows

Mail Sender

- Insert address of the recipient and send text mail

Mail Receiver

- Receive incoming mails in inbox folder
- View the text messages
- Reply to the sender if required

User Interface Flow Chart

Sending End

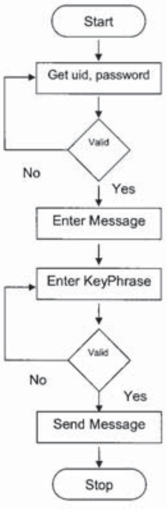


Figure 4.3 : Sender GUI

Receiving End

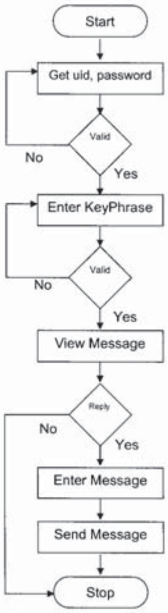
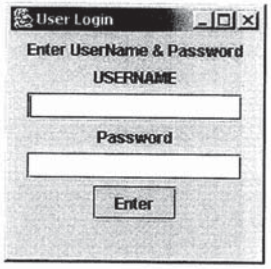


Figure 4.4 : Receiver GUI

4.6 SWING GUI SCREEN DESCRIPTION

The screens are designed and developed using java.Swing package. The Screens are designed as to give little ambiguity to the end user. Separate screens are provided for sender and receiver of the mail being sent.

Login Screen provides the user with a simple screen which asks him to specify his username and password at the correspondingly specified text boxes.


User Login

Enter UserName & Password

USERNAME

Password

Enter

Figure 4.5 : Login Screen

The values entered in the text fields are validated against the stored values. If both matches the user is given permission to enter his login, else the login screen appears and the process has to be continued. An upper bound can be kept for the number of attempts a user can make with an illegal username-password combination, which would stress the security of the project.

Once entered into the respective login, the user has the permission of setting up the text message that he is to send. A text field for typing the message appears along with a text box for specifying the address of the recipient.

Type your message here

To

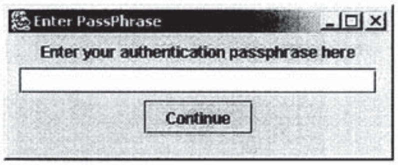
Enter text here

Send

Figure 4.6 : Mail Sender Screen

A button object – send - is provided at the bottom end which receives a click action as an argument and generates the next function. Once the user clicks the 'SEND' button he is again prompted for a passphrase. The passphrase is the key for retrieving the private key of the sender, encxrypted and stored in the Key Store, which is to be used for authentication purpose.

A validation is done with the passphrase to ensure its correctness. If correct the sender's private key is retrieved and the text message is digitally signed. Simultaneously the public key of the receiver is also retrieved using the username as the 'alias name'. This key is used to encrypt the symmetric key.


Enter PassPhrase

Enter your authentication passphrase here

Continue

Figure 4.7 : Passphrase Screen

After the validation is over and the passphrase is found to be correct, the message is ready to be sent. The rest of the working depends upon the java mail API reliability.

Correspondingly at the receiving end the user is validated with a user name and password before signing into his login. Once logged in he is asked for a passphrase, which retrieves his private key and the rest of the decryption process is carried out with the retrieved key. Once done with the passphrase gathering, key retrieval, decryption etc. the user is displayed with the text message he has received along with the address of the sender from whom the message came from.

The user has the option of sending a reply to the concerned mail sender if he wishes so. For ensuring this particular function, another button namely REPLY is also provided which when clicked generates another text editor screen where he can type his message and send it – method being same as the sending end process, the only difference being that he need not specify the 'to' address. Java Mail API provides functionalities for sending reply for a received mail.

Read Message
From
Message

Figure 4.8 : Receivers Text Screen

An exit button is also provided which closes the mail session retrieved when the mailing process had started. In addition a logout button can be employed if the system is to be used for general web based e-mail services. As of now working with an SMTP server the login and logout functionality relies wholly on the server settings. So a basic logout from the user work area would log him out of the mail session too.

4.7 DATA FLOW DIAGRAM

Context Diagram

Level 0

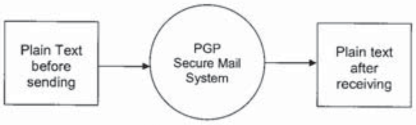


Figure 4.9 : Context Diagram

The level 0 DFD (Context Diagram) shows the basic input and output. In the system being developed the basic input is the plain text message to be sent by the mail sender while the output being the same plain text received by the recipient once it traverses through the mailing system. The 'PGP Secure Mail System' is responsible for the internal operations such as encryption, decryption, mail sending, mail receiving, user interface etc.

K C T
P-1706
LIBRARY
COIMBATORE

4.8 USE CASE DIAGRAM

A use case describes a sequence of actions that provide something of measurable value to an actor. An actor is a person, organization, or external system that plays a role in one or more interactions with your system. In this project there are only two possible actors - The mail sender and the mail receiver. An unauthorized intruder who tries to hack into the system may also be considered, but is not much relevant.

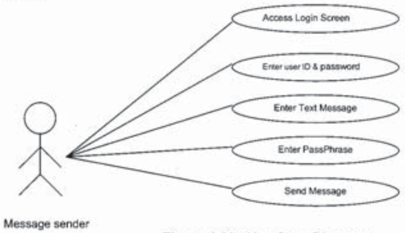


Figure 4.10 : Use Case Diagram I

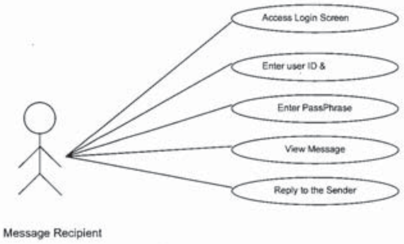


Figure 4.11 : Use Case Diagram II

4.9 CLASS DIGRAM

The purpose of a class diagram is to depict the classes within a model. The fundamental element of the class diagram is an icon that represents a class. In this project the important classes and their interconnections are given below in their respective class diagrams.

Encryption

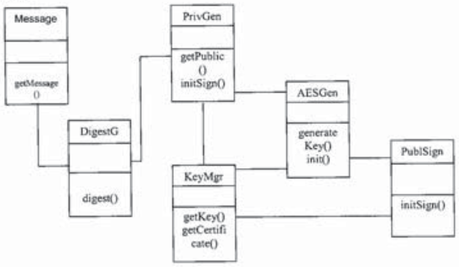


Figure 4.12 : Class Diagram I

Decryption

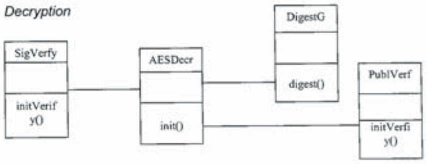


Figure 4.13 : Class Diagram II

4.10 ACTIVITY DIGRAM

Activity diagrams represent the business and operational workflow of a system. An Activity diagram is a dynamic diagram that shows the activity and the event that causes the object to be in the particular state. The activity diagram for this project gives an overall overview of the technical as well as the non technical factors included in the system. It highlights each and every activity associated with the design of this project, namely *encryption, decryption, mail API, user interface* etc

Encryption

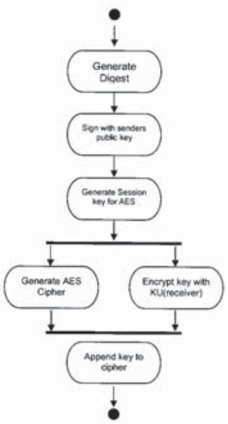

Generate Digest
Sign with senders public key
Generate Session key for AES
Generate AES Cipher
Encrypt key with KU(receiver)
Append key to cipher

Figure 4.14 : Activity Diagram I

Decryption

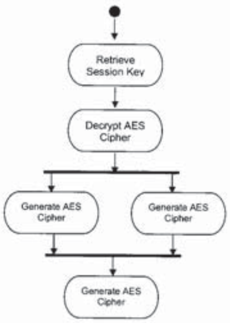


Figure 4.15 : Activity Diagram II

Since the Java Mail API deals with sending and receiving messages between sender and receiver a separate activity diagram is needed to represent the mail interface part of the project. The diagram depicts the activities involved in sending and receiving text at both the clients end.

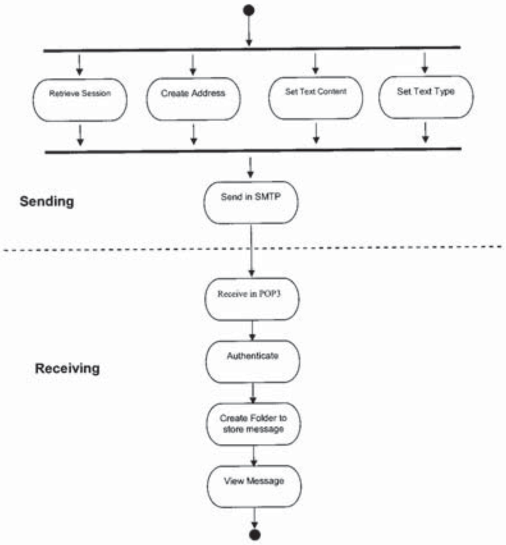


Figure 4.16 : Activity Diagram III

The activities such as retrieving a session, setting up the address, setting message text and type, etc does not depend on each other. They are independent processes which can take its own individual path of execution. Hence these activities are represented in the same level.

The outcome of these activities are needed to send the message, which on receipt at the receiver end is decrypted and made available to the recipient.

CHAPTER 5

TESTING

Testing is a process of high importance because, it decides whether the product is reliable or not. Hence it is done with an aim to break the system, finding maximum number of errors. These errors can then be corrected, thereby increasing the reliability of the system.

The developed system was tested at each and every stage for fixing possible errors and inefficiencies that would hinder the proper working of the system. Standard procedures were followed for testing the PGP mail software. Test cases were generated for each and every screens and other processes. These test cases include every possible input, events and constraints through which the system would go through.

The following testing phases were conducted on regular or orderly basis for ensuring maximum reliability.

- Unit Testing
- Integration Testing
- System Testing

Since the system is mainly aimed at dealing with messages of high importance, rigorous testing has to be conducted to ensure its security even at peak situations.

5.1 UNIT TESTING

Unit testing focuses verification effort on the smallest unit of software design – modules. The different modules in the developed project are DigestGenerator, AES Cipher Generator, Signature Generator etc, Each of them are tested individually at the early stages itself as soon as they are completely developed . All of these modules were seen to be working correctly, even when subjected to extreme cases. Extreme care was taken while dealing with issues like local data structures, boundary conditions, independent paths etc, since the system demands high level of security and reliability.

5.2 INTEGRATION TESTING

The objective of integration testing is to take unit tested components and build a program structure that has been designed. Modules that worked correctly individually might not work properly when put together. Integration test helps in eliminating these errors.

In the developed mail application, the individual modules once done with unit testing are integrated to form the entire program structure. Great care is taken since the modules do not resemble each other either in data structure or functionality. In the developed system even though each module does not depend much on the other in terms of functionality, the outputs from one module forms the input to the other.

5.3 SYSTEM TESTING

The software alone does not form the system. Other aspects such as the hardware, the user, the environment etc also forms part of the system being developed. System testing deals with testing the system as a whole.

The developed PGP system was tested for its compatibility with the hardware specification, the users involved with the system etc. It was found that the system works with all the specified requirements without any flaw or error.

As mentioned earlier the PGP system demands a high level of confidentiality and security. Tests are to be conducted that would ensure the actual security capabilities of the system. Security Sensitive areas in the design have to identified and measures have to be taken to confirm that such points would not deviate the system from proper working. Some of the sensitive points were identified such as Key Storing and Retrieval, KeyPass validation, StorePass Validation etc. Proper testing was conducted to ensure that these points do not deviate from actual working.

CHAPTER 6

IMPLEMENTATION

6.1 DEVELOPING ENVIRONMENT

The project is completely developed in Java on windows platform. Java with its unique and specific features for cryptography greatly supports the development of such an application. Java has a specific package called Java Cryptographic Extension (JCE) dedicated especially for cryptographic purposes. The inbuilt classes and interfaces in JCE are well suited for the project and moreover add to the efficiency of the application.

Apart from basic cryptographic providers provided by Sun Microsystems along with Java2 Standard Edition, the project requires the features supplied by some extra third party providers. The encryption and decryption part is entirely developed using Java2 Standard Edition and could be compiled and executed using a J2SE compiler and interpreter.

The third party providers used in developing the cryptographic module are specified below :

- Cryptix
- BouncyCastle
- SunRsaSign(Provided by Sun)

The mail API part demands a little more than the basic java compiler. An extra package namely Java Mail API is needed to develop the mailing interface (sending and receiving messages) modules. Java Mail API comes along with the Java2 Enterprise Edition (J2ee).

The GUI is developed completely using Java Swing. The event handlers and action listeners provided with Swing is most appropriate for developing an unambiguous, user friendly and efficient user interface. Moreover since the project is completely developed in Java it can be successfully run on any platform, Java being platform independent.

The only performance issue being the cost in terms of time (key generation for RSA takes time) it is assumed that the developed system would work efficiently even on a moderately configured PC. It is seen that AES algorithm would work efficiently on an 8bit processor, hence it is to be concluded that the entire system would work efficiently on an 8 bit processor or higher system.

CHAPTER 7

CONCLUSION AND FURTHER WORKS

7.1 BROAD CONCLUSION

Studies conducted during the analysis and design phases have shown that the system developed is well suitable and adaptable to general web based e-mail services. Usage of such a high security system in a local network such as a LAN would not be practical in most cases, since LAN being used within a local geographic area does not usually demand a high level of security, with certain exceptions existing.

7.2 FURTHUR WORKS

The project is developed mainly with the aim of sending and receiving text messages. Further enhancements to the project being

- Sending and Receiving attachments
- Sending and Receiving HTML pages
- Creating a database for storing and retrieving encryption (public, private and session) keys

The above specified comments and conclusions would add to the strength, usability and efficiency of the system being developed. The first two requirements are very

obviously clear. The third requirement of having a database designed for storing the keys is a great challenge. The design of a database for the system would greatly increase the usability of the system because then the users need not rely on the KeyStore object stores locally for storing and retrieving keys.

But the design of the database leaves us with a great challenge. The keys have to be stored in the database in an encrypted form or some other means have to be adopted to make the key secure such that an illegal access or hack do not result in a success.

REFERENCE

1. Eric Trombold, (2001) "The Security Implications of Web Based Email" GSEC Practical Assignment Version 1.2e
2. Jonathan B. Postel, Request for Comments – RFC82, SMTP
3. Marco Pistoia, Duane. F. Reller, Deepak Gupta, Milind Nagnur, Ashok. K. Ramani (1999) "Java 2 Network Security" Second Edition, Pearson Education Asia
4. R.L. Rivest, A. Shamir, and L. Adleman - A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, RSA Paper
5. Wiliam Stallings (2003) "Cryptography and Network Security - Principles and Practices" Third Edition, Pearson Education
6. ibm.com/developerWorks Paper "Fundamentals of Java Mail API"
7. Sun Microsystems Inc. Revision (01 August 1998) "Java Mail Guide for Service Providers "