# A Holistic Approach Towards Secure and Privacy Enhanced Email System

By

**Haris Javaid**

**NUST201362771MSEECS63013F**

Thesis Supervisor

**Dr. Shahzad Saleem**

**Department of Computing**

A thesis submitted in partial fulfillment of the requirements for the degree
Of Masters of Science in Information Security (MS IS)

In

School of Electrical Engineering & Computer Science (SEECS)
National University of Sciences & Technology (NUST),
Islamabad, Pakistan
April, 2017.

# Approval

It is certified that the contents and form of the thesis entitled "**An Holistic Approach Towards Secure and Privacy Enhanced Email System**" submitted by **Haris Javaid** have been found satisfactory for the requirement of the degree.

**Advisor:** **Dr. Shahzad Saleem_______**

**Signature:** ________________________

**Date:** ________________________

**Committee Member 1:** **Mr. Fahad Satti__________**

**Signature:** ________________________

**Date:** ________________________

**Committee Member 2:** **Mr. Ubaid ur Rehman____**

**Signature:** ________________________

**Date:** ________________________

**Committee Member 3:** **Ms. Ayesha Kanwal______**

**Signature:** ________________________

**Date:** ________________________

# Dedication

To My Parents

Mr and Mrs. Javaid Akhtar

And

My Wife

# Certificate of Originality

I hereby declare that this submission **"An Hollistic Approach Towards Secure and Privacy Enhanced Email System"** is my own work and to the best of my knowledge it neither contains material previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at National University of Sciences & Technology (NUST), School of Electrical Engineering & Computer Science (SEECS) or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics which has been acknowledged.

**Author Name:** ______Haris Javaid___________

**Signature:** ______________________________

# Acknowledgement

Firstly I would like to thank ALLAH (SWT) for blessing me with the courage, strength and wisdom to complete my MS thesis. Secondly, this thesis would not have been possible without the continuous support of my supervisor ***Dr. Shahzad Saleem***. He has always been a great source of motivation for me. I have learnt a lot of things from him during my research phase. Struggle for best results, achieving high targets and dedication are few of them. I am looking forward to his support in future as well.

I would like to thank my committee members for their guidance and support. I am really gratefull to *Mr Ubaid ur Rehman* for his time and suggestions. He helped me a lot during this phase. Thanks to *Mr Fahad satti* for his thought provoking conversations. With the help of his questioning, I am able to have a deep understanding of Email system. Many thanks to *Miss Ayesha kanwal* for her comments and time during thesis write up phase. She reviewed my documentation in depth despite her hectic schedule.

And thanks to my family and friends who endured this long process with me, always offering support and love.

**Haris Javaid**

# Table of Contents

## Contents

# List of Abbreviations

| **Abbreviation** | **Stands for** |
|---|---|
| SPS | Sender Proxy Server |
| RPS | Receiver Proxy Server |
| CA | Certification Authority |
| SEMS | Standard Email Server |
| PGP | Pretty Good Privacy |
| S/MIME | Secure/Multipurpose Internet Mail extensions |
| AH | Anonymous Header |
| OH | Original Header |
| PKI | Public Key Cryptography |
| MS | Master Server |

# List of Figures

# List of Tables

# List of Test Cases

# Abstract

The right to privacy and freedom of expression is the basic right of every human being. In the digital communication the same rights should also be provided. This thesis elaborates the proposed design and protocol of a privacy enabled and secure email system. Many existing email solutions provide features like protection of email contents, security of address book and even encryption of IP address field, but privacy of users is still an issue. This is a serious concern for security engineers, perhaps they mostly argue that an interceptor can identify the communicating parties. After analyzing the current issues, we have proposed a new mechanism to secure the email address of sender and receiver in inter domain and intra domain communication. This mechanism is based upon layered approach. A secure proxy server and database server are used to achieve privacy of users. The system has features of signing and encryption of email messages, protection of email headers, privacy of original email addresses, authentication of original users, and management of symmetric and asymmetric keys in a closed environment. This system is based on a secure proxy server that handles the communication between email clients and standard email server. This system can be integrated with any email server. Public Key Cryptography is used for verification of users and server. After designing and implementing email system, its verification was done by using a verification tool, called Scyther. Results demonstrate that original email addresses are not leaked during communication. Besides using Scyther, various test cases were designed to analyze performance of email system. Results of test cases demonstrates, the security module slightly affects the performance of existing email system while keeping the privacy intact.

# *Chapter One*

## Introduction

In day to day life, people can communicate with each other in a formal and informal capacity. They can exchange their ideas with friends and family in an informal way or an email can be used to present idea to an employer in a formal way. There are several ways of formal and informal communication, some of them are Short Message Service (SMS), phone calls, VoIP, facial gestures etc. Email is one of the mediums used for communicating with others. It is used for expressing ideas and exchanging of documents. Social contact is not the only purpose of this network application, but it is also used as a courier service to deliver critical business messages, financial documents and medical reports. Being a medium of communication, it becomes important to evaluate the privacy and security of existing email systems.

A good quantity of work has been done on the security of Email contents. Many solutions and email clients have been proposed and designed by researchers and commercial companies to keep the secrecy of emails. As headers of an email are in clear text, that attracts attacker to launch attacks on privacy of emails. Now a days, (Secure/ Multipurpose Internet Mail Extension) S/MIME protocol is deployed and is used for encryption and signing of letters [1] . This standard provides security to email contents but privacy is not addressed in S/MIME [1]. In email systems, security refers to the confidentiality and integrity of email contents while privacy is related to the identification of sender and receiver [2]. Email headers reveal a lot of information about the communicating parties [2]. By using these, extraction of IP information and tracking is possible. Medical field is one of the many fields where Email application is being used. In medical field, it is very vital to ensure the privacy of users. In hospitals, a patient disclose his/her social security number, a brief medical history and house address etc. If an attacker gets control of a system then he may impersonated himself as patient. In another case, if hospital staff wants to share the medical reports with a patient and they share it through email then email headers reveal information about patient. Law does not permit to reveal the information of such patient but loopholes of system may allow intruders to extract the information of patient. If an attacker gets all the information about a patient then it is quite possible that he may get unauthorized access to patient's financial accounts.

Several issues exist in current Email systems [1], some of them are mentioned below

- Privacy of users is not protected, most of the email clients send IP headers in clear text [2].
- Leakage of information about participants
- Contacts are stored in text form which can leak information about email addresses.
- Some Email clients provide security feature like content encryption, but it requires configuration and most of the users are not expert so it is not commonly used.
- Users receive messages without their consent. [3]
- Many Email clients do not support smart card option. [4]

It may be observed from literature that some work has already been done on the increasing issue of privacy [5] in Email systems. Currently few solutions exist that claim to provide complete anonymity and privacy but suffer from issues: Leakage of information about participants, burden of using two email addresses, storage of clear text address book are issues of previously proposed solutions [1] [4]. This research proposes a compact solution to user's privacy and secrecy by using secure proxy servers and smart cards. After intensive literature review, following 5 features are identified that could provide a secure and privacy enhanced email system.

- **Privacy of Location**
  Users' location should be hidden. Email headers must not reveal any information about the sender or receiver's location.
- **Privacy of Email Headers**
  Privacy of email headers refers to encryption of email addresses. During communication original email addresses must not be shown.
- **Protection of Address Book**
  In email clients, addresses of contacts are stored in clear text form. These addresses are then used to send spam messages to contacts. So address book must be encrypted.
- **Protection of Email Contents**
  This feature refers to encryption of email messages i.e. its body.
- **Smart Cards to access client**
  Most of the email clients are accessed by username and password. But an extra feature of smart card may be used to make client more secure.

We have analyzed existing solutions and the findings are as follow

# Analysis of the State of the Art

| Sr. No | Paper Name/ Email Systems | Privacy of Location | Privacy of Email Addresses | Protection of Address Book | Protection of Email Contents | Smart cards to access client |
|---|---|---|---|---|---|---|
| **1** | Scramble Email System [6] | ✓ | ✓ |  | ✓ |  |
| **2** | Opaque Email system [7] |  |  |  | ✓ |  |
| **3** | Gmail/SMIME plugin [8] |  |  |  | ✓ |  |
| **4** | Crypto Net: Design and Implementation of Secure Email System [4] |  |  | ✓ | ✓ | ✓ |
| **5** | A Smart Card Mediated Mobile Platform for Secure E-Mail communication [9] |  |  |  | ✓ | ✓ |
| **6** | Neomail Box System [10] |  |  |  | ✓ |  |
| **7** | Secure and Privacy Enhanced Email System as a Cloud Service [11] |  | ✓ | ✓ | ✓ |  |
| **8** | Secure and Privacy-enhanced E-mail System based on the Concept of Proxies [12] | ✓ |  |  | ✓ |  |
| **9** | The anonymous email remailer service [13] |  | ✓ |  | ✓ |  |
| **10** | Solutions for Anonymous Communication on the Internet [14] | ✓ | ✓ |  | ✓ |  |
| **11** | **Proposed Solution** | ✓ | ✓ | ✓ | ✓ | ✓ |

Table 1.1-Analysis of Email Systems

## 1.1 Objectives

The objective of our research work is to design and implement a secure and privacy enhanced email system to ensure

- The privacy during the exchange of messages
- The privacy of stored contact at mobile device
- The privacy and security of email body
- Zero configurations on client end to manage security credentials

## 1.2 Motivation

Right to privacy and freedom of expression is the basic right of every human being. In digital era, the same right exists as well. But on the other hand there are attackers, observers and intruders who want to access the information of people, it does not only involve in leaking out sensitive information, but privacy of participants is also violated. Email is the significant source of communication and its open nature invites attacks. There is a gap for a complete secure and privacy enhanced email system. Focus of this research is to provide such a system to each individual. This research aims to secure and enhance privacy in the emails.

In this era most of the people have at least one email account. Email system enjoy the status of being the most popular source of communication. Privacy of each individual user is very important. It is also used in the field of business, medical systems, educational institutes, media groups, and government organizations etc. This significant use requires a secure and privacy enabled email system which will be available at any level for the use of everyone. Government organizations need a system by which they can share secret documents without revealing sender and receiver information. Medical staff need a system that will help them in sharing the reports of patients without violating the privacy of patient. Journalists require anonymity in their work. Many countries impose ban on journalists so they demand that their identity should remain intact. By using our email system they can report without showing their location and identity. So this email system will be the solution that will provide privacy to each and every user.

## 1.3 Contributions

As nowadays the deployment infrastructural paradigm has been shifted to cloud computing environment from conventional arrangements, where all resources are shared, so organizations and institutions have more concern about their private data. Above explained problems have been solved by taking an entirely different approach in this research work. The proxy based architecture is designed to provide standard security services in email along with extended and novel features. Some of the extended features, extracted from literature survey and analysis of existing email systems are: (a) Security of email headers, (b) anonymous identities for participating entities, (c) privacy against tracking of user's identities. The proposed system is implemented in the form of a layered architecture that can be easily deployed without any complex configurations.

We have designed and implemented a completely different approach. As email system is very prevalent application used by approximately every institution and organization, so social security engineers have major concerns about the privacy vulnerabilities caused by the use of these email systems. From literature review, it was found that no protocol has been devised for the aforementioned issues.

So, proxy based layered architecture has been proposed to tackle these privacy concerns. Every layer has some key components with some responsibilities. The basic idea to overcome the leakage of user identities of participating entities is to introduce the anonymous identity for each user. By using the system, user can send email with his/her anonymous id (AID) rather than original one. While sending an email, sender gets AID from proxy server which maintains the records of anonymous identities against every original email id of user. By using this system, user can send email messages to different domains. User's private information is exchanged in secure and transparent manner. Master level server is deployed in top layer of architecture. Moreover, Master server (MS) acts like source to delegate trust to proxy servers. It is assumed that this entity is in secure environment and all the domains trust this server. Its purpose is to provide a closed environment. Furthermore this server makes sure that if receiving domain is using proposed system or not i.e. if that domain is not registered than sending side will not forward mail to any server and message will be discarded.  In order to develop a trust between two domains standard cryptographic techniques are used, some of which are: digital signature, verification of certificates, time stamps and support for public key infrastructure.

Finally Sender side proxy server sends email to standard email server (SEMS) with anonymous header.  SEMS directly sends an email to the receiver side proxy server at recipient side. Moreover, user’s original id is embedded in the content of an email and at recipient side, system extracts this original id and recipient gets email with conventional format including sender’s and receiver’s original email id. All inter and intra system communication is made secured using standard cryptographic techniques.

Now, by using the devised system user can send email to any recipient and no intermediaries can see his/her original email id. Additionally an authorization policy is also implemented which categorize users in guest and local users and only local user can send email from a particular domain. At the end we have verified our claims and objectives with the automated verification tool; Scyther. An output of scyther has been shown in verification chapter.

## 1.4 Summary

This chapter described the overview and features of the current email systems along with the problem statement evolved by analyzing theses email systems. It also covers the privacy threats caused by the transmission of private information about the sender and receiver in clear text with email header over the internet. The major objectives deducted from research problem and the motivation behind this research activity is defined by section 1.1 and 1.2 respectively. The approach taken for this research is discussed under the section of research methodology. This section also includes the division of chapters according to the phases of research approach. At the end in section 1.5, abstract level summary of proposed approach and the detail of architecture are given.

# *Chapter Two*

## Research Methodology

According to the Cambridge dictionary [15], research can be defined as “a detailed study of a subject, especially in order to discover (new) information or reach a (new) understanding”. Through research, we can understand or discover either new or specific information related to any field of study. The entire process of researching can be broken down into a number of steps [16] [17] [18] [19] [20]:

- Identifying the research problem/topic
- Reviewing the existing literature
- Formulating a hypothesis
- Coming up with a research plan
- Collecting and analyzing data
- Concluding and verifying or rejecting the hypothesis
- Presenting a solution to the problem (if applicable)

Following the steps mentioned above ensures a systematic and logical research.

### 2.1 Research Methods and research Methodology

Research methodology refers to the science of solving a research related problem systematically. It is a broad term encompassing various research methods and techniques as well as the logic and reasoning behind choosing the specific method. It is the science of employing the most-suitable procedures specific to the research field and to conduct them properly. This chapter of the presented thesis deals with the research methods adopted to carry out the research systematically, as well as the logical reasoning behind choosing the particular methods in the given scenarios. Listed below are the steps that were followed to compile and compose this thesis:

- Gathering thorough knowledge regarding contemporary email systems.
- Identifying security requirements of email users.
- Developing the hypothesis after extensively studying and analyzing the existing literature
- Designing a solution based on hypothesis
- Implementing the designed solution
- Verifying the proposed solution through evaluation techniques.

## 2.2 Various Types of Research Methods

There are several types of research methods, each suitable for different sort of research areas.

*Analytical research* [21] is a research method in which the researcher aims to answer *why or how* instead of *what*. It involves a lot of critical thinking and analysis of existing facts to explore a topic in depth determining cause and effect.

*Descriptive research*, [22] [23] unlike analytical research, aims to answer *what.* Through this method, the researcher establishes facts about a population or phenomenon but is not concerned about the cause or effect of the established facts.

*Applied research* [24] focuses on practical application of science to solve problems. Instead of focusing on forming theories or generalizations, it aims at providing a practical solution for an immediate problem at hand. The problems can be business-related, industrial, economic or social.

*Design Science* [25] approach is quite contrary to the traditional natural science approach which focuses on accepting or rejecting hypotheses explaining natural phenomenon. Design Science focuses on creating and evaluating IT artifacts designed to solve prevalent problems.

*Deductive research* [26] is a top-down approach for carrying out research. The researcher, first explores an existing theory, forms a hypothesis based on that theory and then, designs a strategy to test that hypothesis. The phases of deductive research include: theory, hypothesis, observation and verification.

*Inductive research*, [27] on the other hand, is a bottom-up approach in which the aim is to draw generalized theories from data. Beginning from observations, the researcher forms a hypothesis, explores it and concludes with a generalized theory.

*Quantitative research* [28] is usually used to verify an existing theory using numeric data collected from surveys, interviews and existing records, and applying statistical methods on them. It aims at generalizing the results derived from a large sized sample population

*Qualitative research*, [29] involves detailed information to explore reasons, opinions and motivations of a population. The information is gathered through several methods such as surveys, interviews and group discussions.

*Conceptual research* [30] is mostly used in the field of social science, philosophy or psychology. It deals with studying concepts and theories explaining a particular phenomenon and helps in developing new theories or reshaping existing theories.

*Empirical research* [31] deals with facts instead of abstract ideas and uses experience or observations to make conclusions. A hypothesis is developed first which is later verified through facts and figures.

### 2.3 Thesis Research Methodology

For thesis, I have followed the Design Science Approach which, as stated earlier, is directed towards creating IT artefacts to solve organizational problems [32]. Through this research, the aim was to study the underlying security problems in the current email systems and also, the shortcomings in a range of existing solutions, and designing a single, comprehensive solution that caters to all those issues.

During the research, the concept of Design Science was followed which states that aim of the outputs (constructs, methods, models, or instantiations) must be clarified first. The next step was developing and evaluating the artifacts. [33]

The four outcomes [same as above] of this design science approach in regards to this thesis are mentioned below:

a) **Constructs:** are the terms or vocabulary specific to a particular domain. These are used to define the problem as well as to describe the proposed solution. They are used to describe the problem and to specify its solution in a specific paradigm.
The proposed and implemented protocols, tools and techniques to secure emails were studied, explored and evaluated to understand the features and shortcomings of the email security systems.

b) **Models:** are statements used to depict the relationship between constructs. They propose what things do or what they should be doing.
After developing an understanding of the email security concepts and features, models were built to provide better security and to preserve the integrity of emails.

c) **Methods:** are set of steps or guidelines to perform a task in order to solve a particular problem. They are used to manipulate the constructs so that the proposed solution statement could be realized.
To deal with the problem at hand, I've suggested that each domain needs to be registered with a master server. Additionally, a proxy server will receive email from client and it will perform cryptographic functions on these emails to preserve the privacy of the users as well as the email contents.

d) **Instantiation:** is based on constructs, models and methods and is basically a complete realization of an artifact in its true environment. It is usually considered a material artifact while constructs, models and methods are of a more abstract nature. The final output of my research is the proposed architecture and a developed application based on that architecture.

Steps of The Design Science Research Methodology followed in this research are discussed in the next part of this chapter.

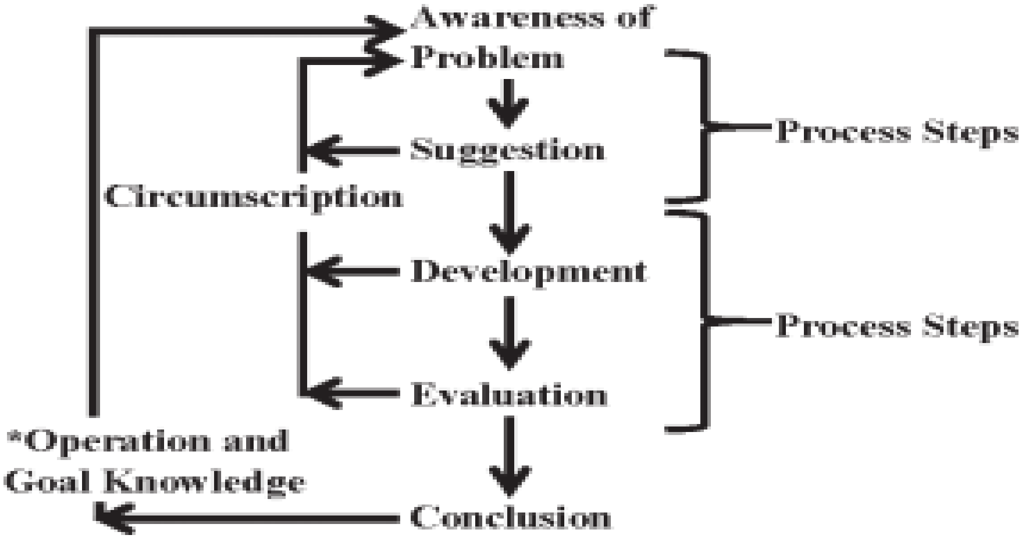


Figure 2.1--Research Process [34]

### 2.3.1 Awareness of Problem

The first step was to determine the area of research, which, for me was Email Security. The next step had been to survey the existing and proposed email security enhancing systems and protocols. After thoroughly surveying the solutions proposed by esteemed researchers in the field of Information Security, it was concluded that most of the systems targeted either one or other issue pertaining to the emailing systems. None of the proposed protocols fully implemented the idea of a privacy-enabled, secure and non-invasive email communication system. The survey further helped me in understanding the security trade-offs and also in formulating a comprehensive problem statement and hypothesis on the basis of which a worthwhile protocol could be devised and tested.

Some of the concerns regarding the privacy and security of conventional email systems and the shortcomings of the proposed solutions are listed below in the form of problem statements.

1. The right to privacy and freedom of expression are basic human rights; the current email systems overlook the fundamental right to privacy and security.
2. Some email clients provide security as an additional feature not that requires complicated configurations and hence, the feature is mostly ignored by non-technical users. Furthermore, the privacy of users is compromised since almost all of the Email clients send email headers in plain text.
3. Although a number of solutions claiming security and privacy enhancement exist; leakage of information about participants [4], burden of using two email addresses [11], storage of clear text address book are some of the issues of previously proposed solutions.
4. None of the proposed solutions tackle the issue of privacy and security, both at the same time.
5. Some of the protocols and suggestions [1] to deal with the issue of email privacy or security seem to be too complex for everyday use or for inexpert users. Usability and security appeared to be a trade-off.

### 2.3.2 Suggestion

In this phase of the research, a lot of hard work was put in to come up with a protocol which not only provided privacy and security to the email users but also was convenient enough for laymen. Following things were kept in mind while formulating a protocol design:

- Email clients send IP headers in clear text.
- A privacy enhanced email system was also required for light weight devices
- Security of the address book is extremely critical.
- Complex configurations will not be useful for unskilled email users.
- Spam messages are also undesirable.
- Most of the email clients do not support smart cards.

### 2.3.3 Development

During this phase, the suggestions made in the Suggestion phase are materialized in the form of an artifact. Any shortcomings found in this stage are to be dealt with by following the Design Science cycle all over again. The designed protocol was implemented so that it could be evaluated for its functionality and further improvements could be suggested.

### 2.3.4 Evaluation

The protocol design ensured secure and privacy enabled email system which followed S/MIME standard for cryptographic functions. In order to analyze and verify the proposed system a protocol verification tool called Scyther was used. Main claim of this system i.e. secrecy of email credentials was verified through Scyther. Moreover, it was also proved that man in the middle attack is not possible on this protocol because of PKI being used to ensure digital signatures, confidentiality and integrity of credentials.

### 2.3.5 Conclusion

In the final stage of Design Science, the entire research process was concluded and the artifact was finalized. The proposed solution not only provides confidentiality and integrity of email credentials but also enhances the security of the email contents.

# *Chapter Three*

## Related Work

### 3.1 Features of Secure Email System

A. Kapadia [1] has discussed a number of basic security mechanisms to make traditional emails more secure and reliable. He stressed the importance of secure email communications and proposed solutions for the issues and loopholes present in basic mechanisms that could be used to secure emails. The key points discussed in his case study are as follows:

- Digital Signatures could be used to ensure non-repudiation and data integrity. Non-repudiation means that none of the involved parties (sender or receiver) could deny sending or receiving the email whereas integrity means the data received at the other end should not be altered or tampered with in any way.
- Digital Signatures are created and managed through PKI (Public Key Infrastructure) and the major issue concerning digital signatures is ensuring that the keys are distributed securely. To deal with this problem, the author proposed using a reliable third party Certification Authority (CA), fingerprinting and hashing.

**Analysis:** The author has discussed solutions to the issues pertaining to security mechanisms for email content only. The study neither highlight the importance of ensuring maximum privacy for email users nor it highlights any means by which we could protect user's credentials and his personal information.

### 3.2 Privacy Issues

Another issue regarding email system is the privacy of users i.e. senders and receivers identities. Many solutions have been proposed but these do not provide end to end privacy.

*NEOMAILBOX* [35] is a well-known solution that emphasizes on the privacy and security of E-mail system. This technique provides security and secrecy by encrypting IP address field of the header. But there is a hindrance to the adoption of this mechanism, as encryption of IP address field only does not ensure privacy. By using this method, privacy of location can be achieved but it is not possible to completely hide communicating parties. Namely, there are other parameters that leak out information about sender and receiver. One of them is E-mail address that discloses the information about participants

### 3.2.1 Solution based on Two Email IDs

To ensure privacy of users, the concept of Proxy Server is implemented into another proposed solution (*Amna* et al [11]). Proxy server performs mapping of the user's email address into an anonymous E-mail account created by the user. All the domains participating in this solution need to be registered with a centralized Infrastructure Email Structure (IEMS). When a user sends email, Sender side proxy server identifies receiving domain and then asks domain to provide anonymous email address of receiver. As user has already created an account which is anonymous, it is registered with receiver proxy server as well. Domain replies with anonymous ID. Then proxy server sends email with a new header. In this way privacy is achieved but this approach has limitations, as user needs to create two accounts, which is a burden. It also depends on user's awareness, i.e. how anonymous accounts he/she would create. It is quite possible that user may create another account without keeping in mind the privacy, if this happens then system will not provide any kind of privacy. Their protocol in detail is as follow

Sender (Alice@hello.com) wants to send message to (Bob@hi.com)

- Step 1: In this step, Sender sends standard email with original email addresses. This email is sent to sender side proxy server

  Client A →Sending side server: {Original email IDs}

  **To:**Bob@hi.com **From:**Alice@hello.com

- Step 2: After receiving email from sender. Server will check whether user is registered or not. This is used to check whether user is local or guest.
  Sending Side Server: {Sender's original id is mapped with his/her Anonymous id}

  It is necessary that each user must have another account as an anonymous account. It will be checked whether Alice has an anonymous account or not

- Step 3: To prepare new header, now server requires anonymous ID of receiver for this server will find domain of receiving side. For this it will interrogate IEMS about IP address of receiver.

  Sending Side Server → IEMS: {Domain of receiver}

  In this step now server will send domain that is @hi.com
- Step 4: As all proxy servers are register with IEMS so it will reply back with domain and IP address. Purpose of this step is that Sender side can now interact with receiver side.

  IEMS→Sender side server: {IP address of receiving side server}

  IP address of hi.com will be sent back to Sender proxy server
- Step 5: In this step, sender server sends query to find out the anonymous ID of receiver. For that it will provide the original ID to receiving server. In reply, server sends the required anonymous id which is stored against the original one.

  Sending Side Server → Receiving Side Server: {Request for recipients anonymous id, Given its original one}

  Sending side server will query about anonymous id of bob against bob@hi.com
- Step 6: After receiving the anonymous ids of both sender and receiver side, Server sends all the information containing original IDs and anonymous IDs to sender client. It sends information in the form of an anonymous header and email contents. At this point, client sends an email to SEMS with the following format Client (Sender) →SEMS: {Email to SEMS with anonymous ids}

  a) Anonymous Header (AH): This header includes anonymous IDs of both sender and receiver

  b) Body part (Contents): It contains the encrypted mail message contents produced by usingS/MIME.

  c) Original Body part (original ids Information): It includes encrypted original ids of both sender and receiver attained from first step by using standard encryption mechanism.

  Now anonymous email will be generated using anonymous IDs of alice and bob and original headers will be encrypted to email will have sender as anonymous_alice@hello.com to anonymous_bob@hi.com

- Step 7: SEMS sends this email to receiver using the information in Anonymous Header (AH)

  SEMS→client: {email with above stated format}
- Step 8: Client simply gets the original sender and receiver from the body part of received email and places them in the position of "To" and "From" in the email viewed by receiver client.

  Receiving server will decrypt email and original headers i.e. alice@hello.com and bob@hi.com are revealed

### 3.2.2 Solution Based on Proxy Server

Another solution based on Proxy Servers is described by *Ioannis Kounelis* et al [12]. They have addressed the issue of privacy by proposing a solution based on web based proxy servers. Their solution does not require new email addresses as users can use existing email addresses. On the other hand, this does not ensure complete anonymity and also have some limitations. Both sender and receiver need to be in one domain. Besides this, security of key depends upon password policy. They have used X.509 certificates, where Distinguished Name (DN) is the email address of sender. Intruder can know about communicating parties by looking at DN name of certificate and it violates the privacy of participants. Email headers are transported in clear text form so any intruder can obtain the identities of users. Overview of their solution is as follow

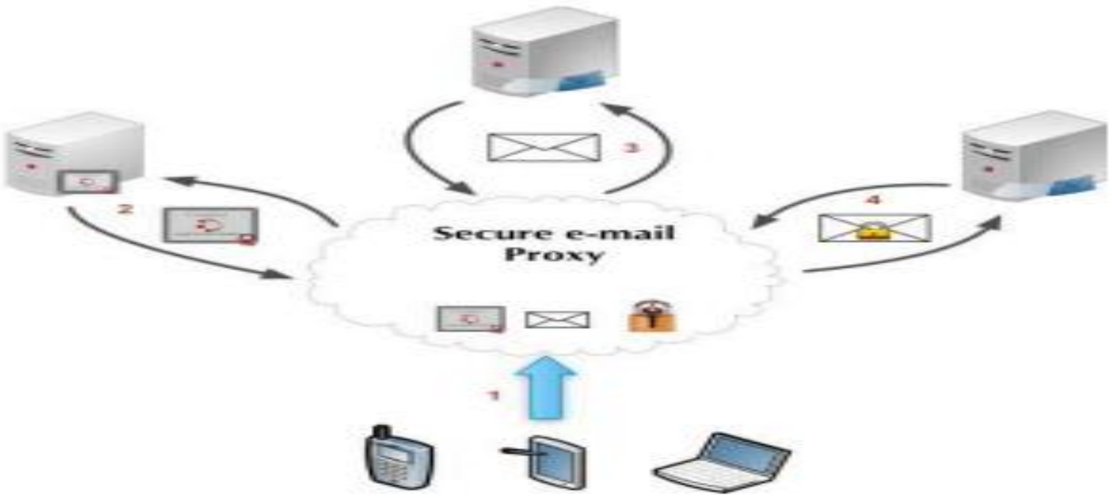


Figure 3.1-Secure Email by Ionnis

Step 1: Client uses his/her email credentials i.e. username and password to login to the secure email proxy. For this client uses native email.

Username: alice@hello.com

Password: 123456

Step 2: In this step, a key pair is generated for new users. Public key is sent to Certification Authority server. CA Server in return sends certificate of that particular user.

Private Key and public key for Alice are now accessible by email client

Step 3: In third step, proxy server uses IMAP to fetch emails from the relevant server. The proxy fetches via IMAP the e-mails from the corresponding e-mail server. User then double click to open emails. Upon click, cryptographic functions are used to display and process emails.

Now email client on the behalf of user fetches email because it has access to user's credentials so emails are shown to end user

Step 4: Finally user can digitally sign and encrypt emails before sending.

Now if user wants to sign or encrypt an email. It will simply click buttons in email client to perform these functions. On backend client will access private key or public key to sign and encrypt email letters.

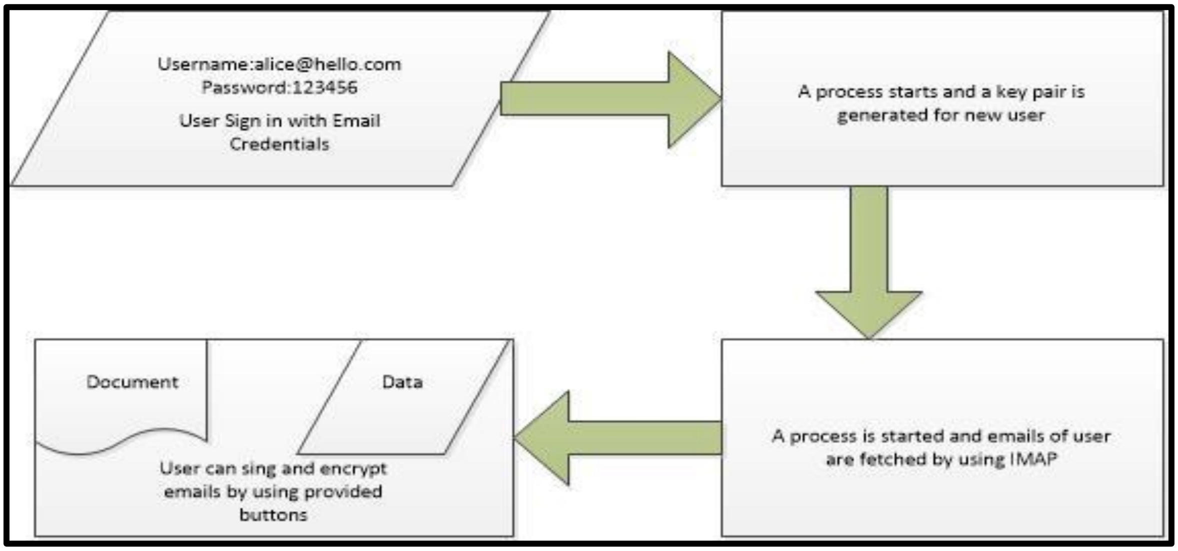

Username:alice@hello.com
Password:123456
User Sign in with Email Credentials
A process starts and a key pair is generated for new user
A process is started and emails of user are fetched by using IMAP
Document
Data
User can sing and encrypt emails by using provided buttons

Figure 3.2-Flow Chart

### 3.2.3 Solution Based on Remailers

*MailAnon* [36] is an E-mail service based on Type 0 remailer. Type 0 remailer is a way to provide anonymous communication. An intermediate relay server is used. In this technique sender sends E-mail to server, server removes identifying headers and sends E-mail to the recipient. But this solution does not protect against observer who observes communication between senders and relay server

*Anonymizer* [37] is a Type 2 remailer that removes the issues of Type 0 remailer, but it creates network performance issues. These type of remailers are also called mix masters, they relay email messages through different nodes. Each node encrypt email and then next node decrypts it. This relaying continues until last node decrypts it and send message to receiver in unencrypted form. In this technique, nested encryption is used. Email packet is encrypted several times. A receiving server can decrypt only one layer to identify the next hop. Then next hop can decrypt one more layer and it send packet to next relay server. In this mechanism, privacy is achieved throughout the path but it is not possible to hide information between senders and first relay server and then from last relay server to receiver. Keys are shared among all the relay servers. If any server is down then user may face denial of service. Secondly if a single key is updated then the complete keychain needs to be established again.

## 3.3 Management of Email Messages

### 3.3.1 Secure Solution using Proxy Server

Abbasi et al [4] deigned a secure E-mail system by using Proxy Servers. They have introduced many other features like encryption of address book, option for smart card and spam filtering. But to provide these features their Secure E-mail Client and Secure E-mail Servers interfere with the functionality of original email system. SEM server logically exists between SEM client and standard mail server. Initial connection is established between client and server. After successful authentication mail box is loaded. Both SEM server and client generates a symmetric key. Purpose of this key is to encrypt and decrypt address book. User can upload address book to server as well. In their system security of email contents is achieved by using S/MIME standards. Attachments are handled by SEM server. According to their system, attachments are uploaded to SEM server and then server sends a URL back to SEM client. Client embeds this URL with standard email packet and email is forwarded. Now on receiving side, receiver client breaks message and extracts URL and attachment is downloaded. When an attachment is downloaded it can be deleted from SEM server. Overview of their system is as follow

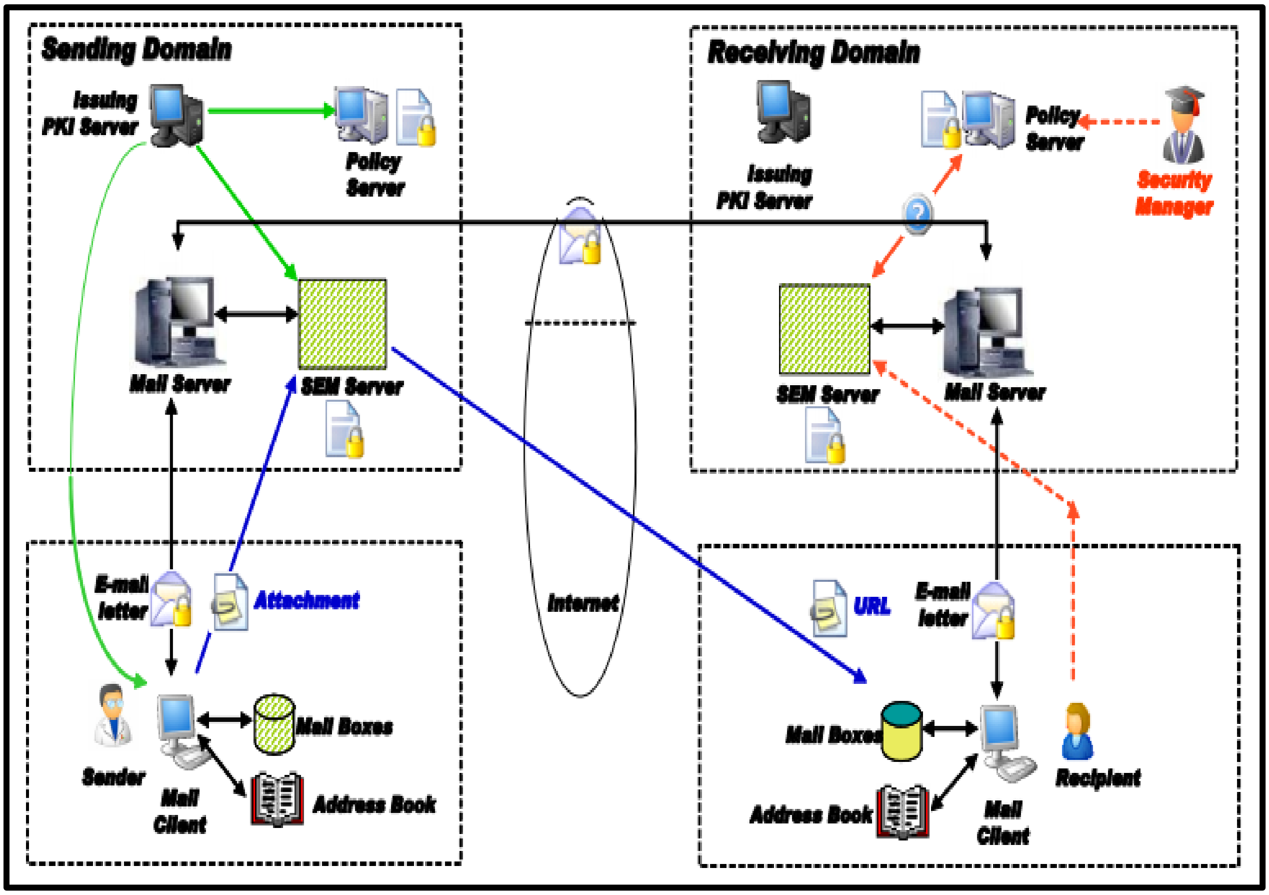


Figure 3.3-Secure Email System by A.G. Abbassi

Spam control is another feature which is provided by this system. It is achieved by using authorization polices defined by PDP (Policy decision point). Policy Enforcement Point is used to apply these policies. In policies, it is mentioned that a receiving side can receive emails from which domains and email account. PKI is used to sign and encrypt email messages. This system is designed to achieve security of email contents but as I have mentioned earlier that some components interfere with original Email system components. Besides this there is no mechanism to achieve privacy of users.

### 3.3.2 Scramble Email

Another proposed solution to provide the security in email system is Scramble [38]. This solution encrypts the emails letters. For this, a user has to create a new email account because the solution is not available as an add-on component, since Scramble uses an independent secure E-mail server. This is a significant obstacle to its use and popularity

### 3.3.3 Solution based on Smart Cards

G.Kardas et al [9] have proposed a solution based on the use of smart cards. User credentials are stored in a smart card. Whenever a user wants to send an Email, user credentials will be used for encryption and signing of E-mail letters. Private Key is used to sign the messages, system will read the signature from card and it will be used to sign the package. This will not only provide security to E-mail contents, but user credentials will also be saved in a secure environment. On the other hand, cost is an obstacle in adoption of this technique. Firstly, not every user possess smart card, and secondly smart card readers are also required in this case. Besides cost issue, privacy is not addressed in this solution.

## 3.4 Summary

In this chapter, the focus has been on the already existing research that has been done to enhance security protection of email clients. The chapter discusses the security concerns in the email system and highlights some of the solutions proposed in several case studies. It also contains analysis of these existing solutions and their shortcomings. Securing the email process and the email content and its effect on the privacy of users has been analyzed in the above chapter. The attributes of a secure email system and the solutions to shortcomings of the basic mechanisms of security have been discussed in the first part. Next part deals with a protocol to allow for maintaining check and balance of email content without disturbing the privacy of users and the analysis of its loopholes. The literature review has led to the conclusion that although a lot of solutions exit dealing with protection of the email content and communication level privacy, a lot of work needs to be done when it comes to protecting user's credentials and private data that is openly transmitted over the internet through email headers. There has to be a security design that protects the user's identity as well as securing the content at communication level.

# *Chapter Four*

## Proposed Architecture of the System

To encounter above mentioned problems, we have designed and proposed a secure and privacy enabled email system. This system is based on proxy servers and it is implemented in layered architecture, as shown in figure. The reason to follow layered methodology is that architecture should be easy to understand and design must be simple. Layers are formed on the basis of roles and responsibility. The architecture consists of following four layers.

1. Client layer
2. Proxy layer
3. Master layer
4. Standard Email layer

Each layer performs different functions and roles are assigned depending upon these functions. Different key components are enclosed in these layers. The details are discussed below

Components of Client layer are sender and receiver, residing on both sides of communication. Proxy layer consists of sender proxy server and receiver proxy server. Two different domains are deployed on these servers. Third layer, Master layer exists between standard email server and proxy layer. There are two main components at this layer i.e. Master server (MS) and Certification Authority (CA). Each layer has its own duty as Master Server provides a trust between all proxy servers. Certification authority certifies all clients. Standard email server routes email on internet. This is responsible for sending email to relevant domain.

Figure 3 is showing the interaction between different layers of architecture. It shows the steps of obtaining anonymous email IDs at sender side, then sending of email to standard email server. Standard email server sends email to receiving side server and finally email is delivered to recipient. Moreover, it also presents the exchange of messages between sender’s proxy server and master server. While certification authority (CA) is certifying all the communication of all clients.

So, all these components enclosed in the four layers of architecture have to perform their particular tasks. The details of these components are given in section 4.1 and the major protocol functionalities are described in next section.

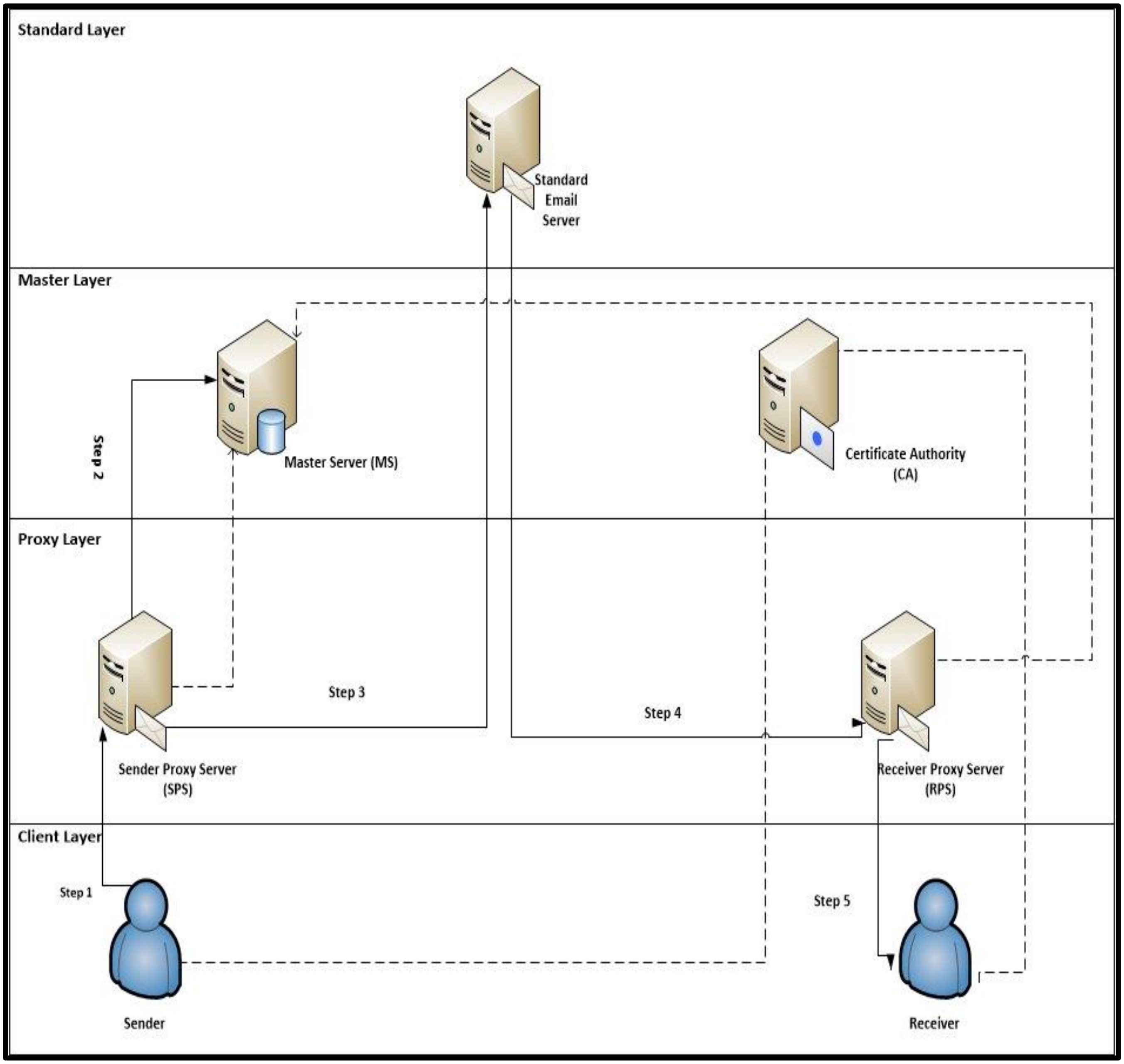


Figure 4.1-Overview of Proposed Architecture

## 4.1 Components of the System

As described in figure the proposed architecture consists of four layers and different critical components are enclosed in these four layers. Now, this section explains the major functionalities and responsibilities of these components according to protocol.

### 4.1.1 Client Layer

***Client:*** Main component of this layer is client. Client can be any standard email client that has functionality of sending and receiving emails. It will communicate with proxy server and email will be sent to proxy server. All the communication of client is secure using certificates.

### 4.1.2 Proxy Layer

***Proxy Servers:*** It is the most critical module of this system. All the pre-sending and post-sending functions are performed by this server. It will communicate with master server to find out receiving domain's anonymous email IDs and IP address. Then it will also perform mapping and creation of anonymous header. This server is also responsible for communicating with standard email server. Both these servers are domain level servers.

### 4.1.3 Master Layer

a) ***Master Server (MS):*** Master server is a kind of database server. It will store IP address of all domains and anonymous IDs for all domains. For this database to be populated, each proxy server needs to be registered with this server providing IP and anonymous ID. In this way it will provide IDs to proxy servers. Secondly, it also provide trust between different proxy servers, because if all will be registered with on master then they can trust each other.

b) ***Certification Authority (CA):*** Second component of this layer is a root lever certification authority. This component is responsible for providing x.509 certificate to all clients. These certificates will be used to secure communication.

### 4.1.4 Standard Email Server

This is a standard email server which will route email to other domain over the internet. It can be any server like Gmail, Hotmail, Yahoo, etc.

Analysis and implementation of this architecture is explained in section 3.2. Following is the step by step protocol of this architecture

## 4.2 Analysis and Implementation of the System

### 4.2.1 Protocol

Two conditions which are necessary for the execution of this protocol are as follow

- Every domain must have an anonymous id for all the users of that specific domain.
- Every proxy server must be registered with master server (MS) with its anonymous id and IP address for that domain.

**Step 1:** Firstly, client on sending side sends email to proxy server with original ids in “To” and “From” fields

Client →SPS: {Original “To” and “From” fields}

**To: Original  From: Original**

Figure 4.2-Client to SPS

**Step 2:** In this step, Sender side proxy server will check whether user is a local user of domain or not. After successful verification. Server will extract to and from fields from header. It will map sender side anonymous id with original from field as shown below.

SPS: {Sender’s original id maps to Anonymous id}

**Step 3:** Now Sender Proxy Server (SPS) needed to get anonymous id and IP address of receiver proxy server domain. To get this information, SPS sent query to the Master Server (MS). In query it sent domain, which it has extracted from original address.

SPS → MS: {Domain}

**Step 4:** As it is mentioned at start of protocol that each proxy server is registered with our Master Server (MS), So MS will reply to query with IP address and anonymous ID of that domain. This information will help in establishing trust between different proxy servers.

MS→SPS: {IP address of RPS, Anonymous Id of RPS}

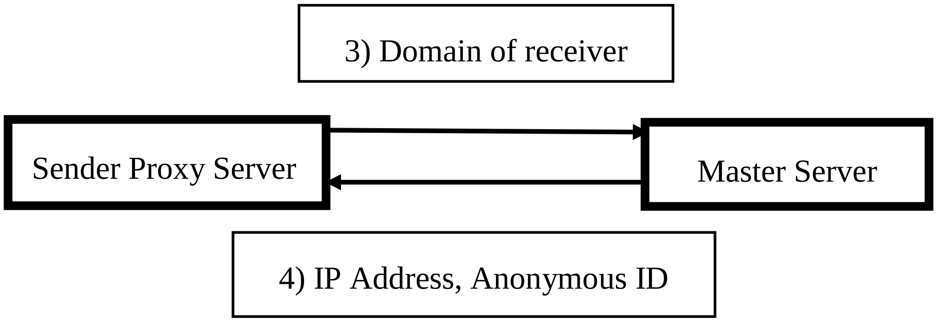


Figure 4.3-Communication between SPS and MS

**Step 5:** At this step, Sender proxy server now have anonymous Id and IP address of receiving side. It will map original “To” field to this anonymous ID.

SPS: {Original “To” ID maps to anonymous id provided by MS}

**Step 6:** After attaining the complete information i.e. anonymous email IDs for both sender and receiver, IP address of receiver proxy server, SPS will now create an anonymous header for routing of emails. Original email IDs will be concealed by using cryptographic functions. Original IDs will be made part of email body and complete body will be encapsulated with anonymous header. At this point, client sends an email to SEMS with the following format:

Client (Sender) →Standard Email Server

| | | |
|---|---|---|
| Anonymous Header | Original Header (Part of Body) | Email Content |
| | **Encrypted Part** | |

Figure 4.4-Anonymous Header

a) ***Anonymous Header (AH):*** This part includes sender and receiver’s anonymous email addresses for routing of emails over the internet. Anonymous header part is unencrypted because it will be used by standard email server to get information for transmission of messages.

b) ***Email Contents:*** This portion has encrypted email message that is intended for receiver. Encryption is done by following S/MIME [39] standards.

c) ***Original Header (Part of Body):*** This part contains the original header that has original email ids of sender and receiver. This header was sent to proxy server in step 1.

**Step 7:** When email will be received by standard email server, it will analyze header information and email will be sent to email address mentioned in “To” field.

Standard Email Server → client: {email message}

**Step 8:** When proxy server on receiver side will receive email, it will decrypt the message and get original email IDs. Then email will be forwarded to original email address which will be the user of receiving domain.

As mentioned earlier, these steps show the detailed flow of protocol and how messages are transmitted and received. It has also described that how original Ids are embedded into the body of email messages while adding a layer of anonymous header to route traffic. In this way, goals of maintaining privacy are achieved.

As Amna et el [11] had proposed a similar solution to this one but there is a major difference between these solutions. Former thesis had proposed solution based on two email accounts. These two email accounts would have been created by users. It was quite possible that they may create an email account with no anonymity in that case system fails to hide the identities of sender and receiver. Another difference is that in their solution email was sent back to client by proxy server and then client forward email to standard server (Gmail, Yahoo, etc.) but in our case email is forwarded by proxy server.

## 4.3 Interfaces

Following are the interfaces of compose email and inbox.

### 4.3.1 Compose Email

Figure 4.5 demonstrates the interface of email client. In compose email, user will have to provide his/her email credentials for login. There is also option of SMTP server address and SMTP port, these two fields will be used to access email server. Purpose of these fields is to make sure that this client works with any email server. User will have to know about settings of email server. Mostly SMTP port is 25 and server address is domain of user i.e. seecs.edu.pk or pgc.edu.

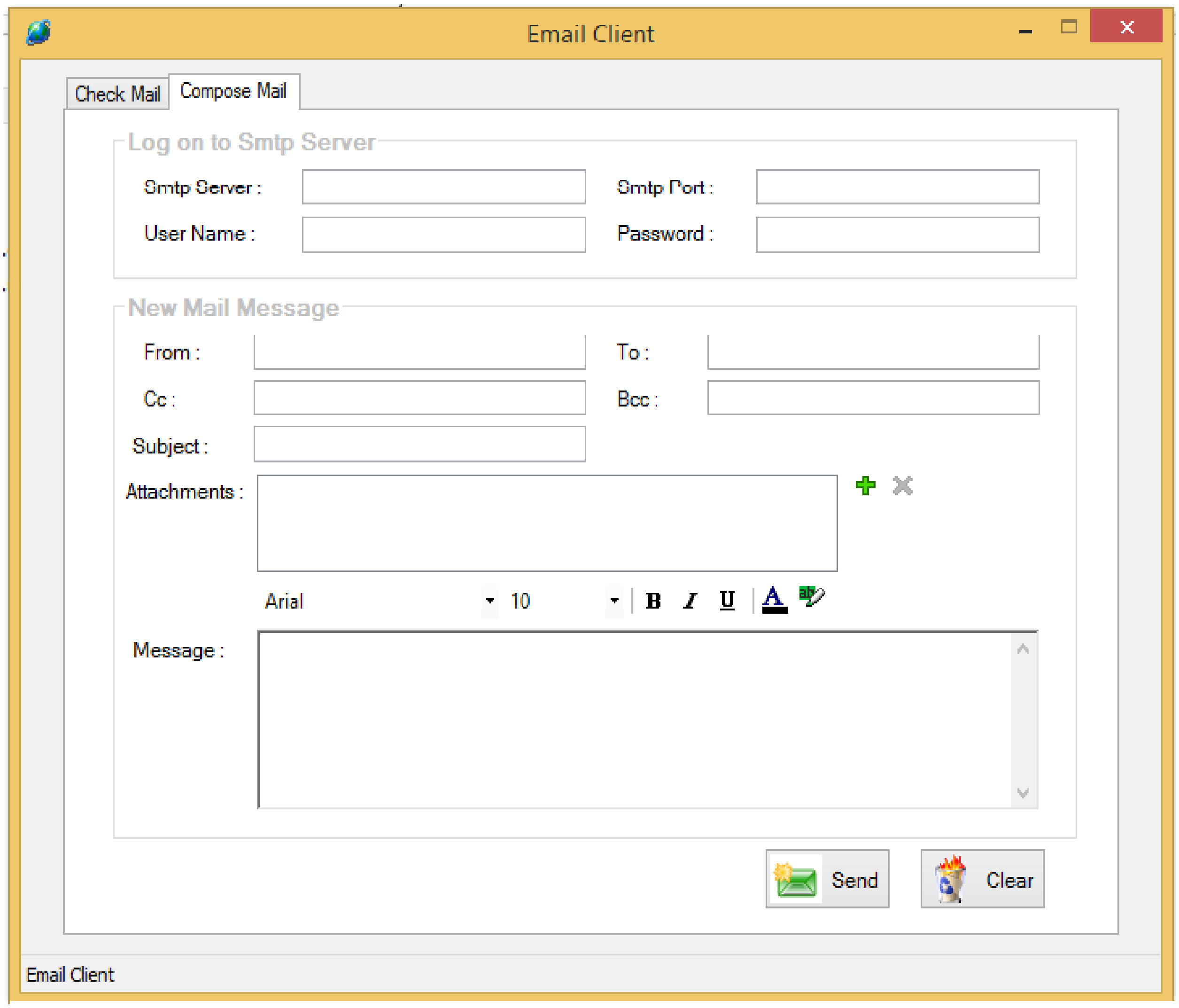


Figure 4.5-Compose Email

### 4.3.2 Inbox

Figure 4.6 shows interface of inbox. User will have to provide pop3 settings of email server for this he/she must know about server address and port number.  Mostly port number of POP3 is 110 and server address is domain name. User name and password are the email credentials. Any email server can be used to access emails through this email client.

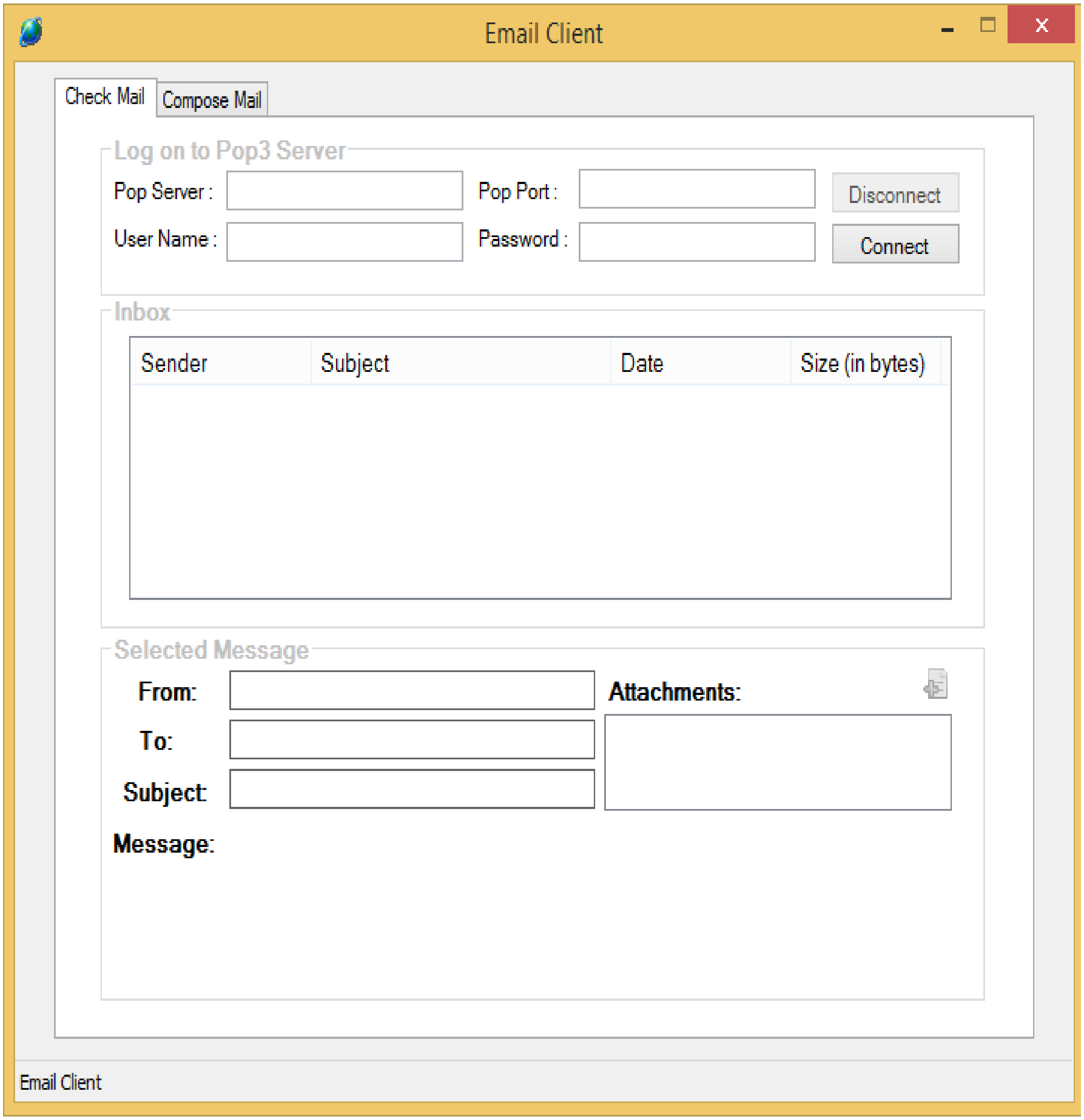


Figure 4.6-Inbox

# *Chapter Five*

## Verification of Protocol

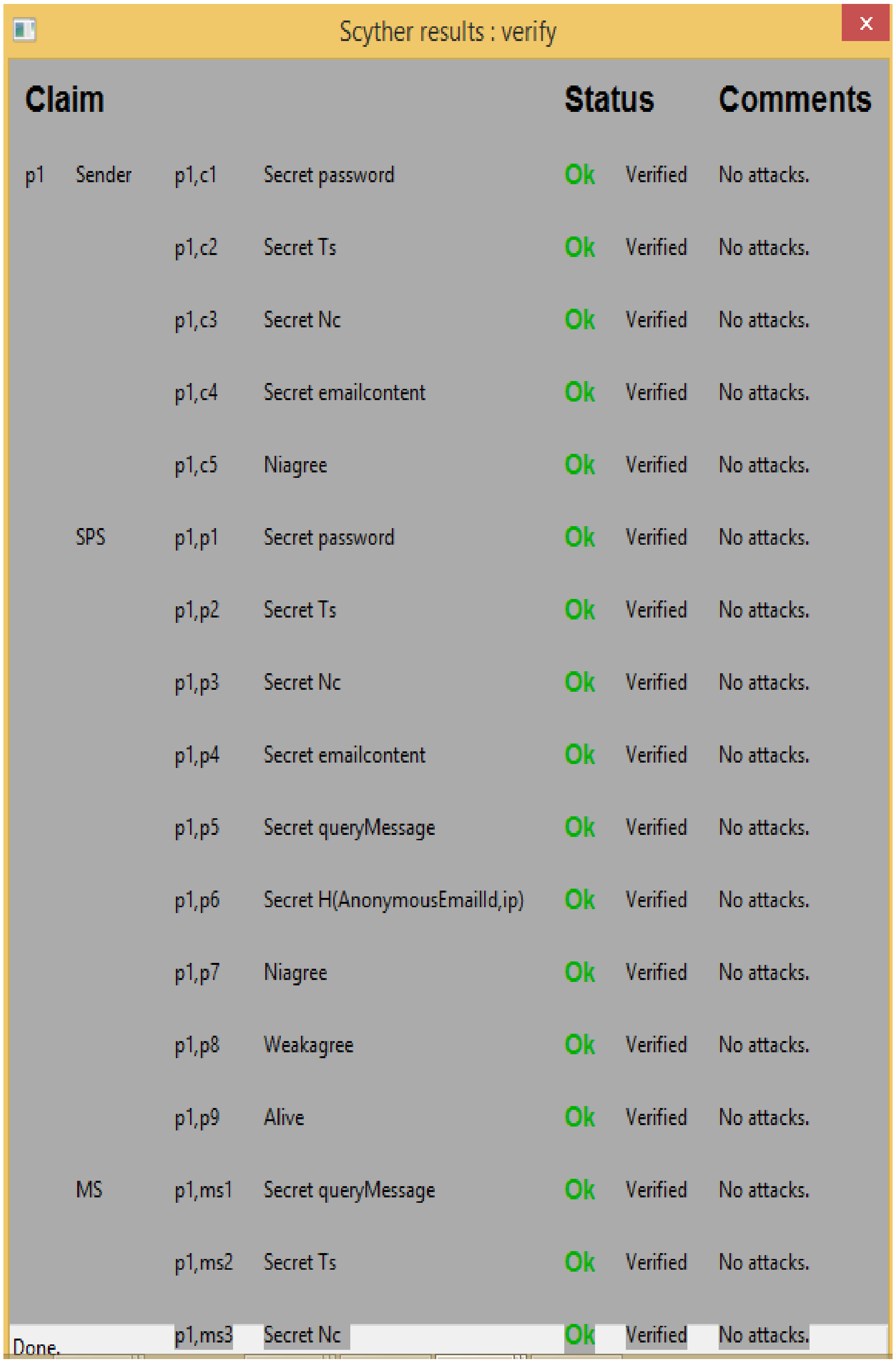
As we have discussed in previous chapter that how protocol works. Step by Step working of protocol is already explained. In this chapter, analysis and verification of protocol is discussed. I have used an automated security verification tool called Scyther [40]. This tool has verified that original email IDs of sender and receiver are not revealed at any stage. Besides this, it also verifies that aliveness and secrecy of the system is maintained throughout the communication. By aliveness [41], it is meant that when a receiver receives message, he/she should be sure that is sent by authentic sender and there is no intruder. To achieve this, sender needs to sign the message with his/her secret key. By secrecy [42], it is meant that when there is any encrypted communication then the key must be secret, there should not be any kind of information leakage about key. It is also ensured that if an attacker at any stage captures a packet, he/she will not be successful to find out communicating parties. Figure 5.1 shows the results of Scyther. Script is given in appendix 1 section.

Firstly variables are declared to describe the working of protocol. Using these variables, process of registration between client and proxy server has been carried out. After successful registration, communication is initiated. When a user sends email, proxy server interrogates the anonymous id for receiving side. This whole process is verified that it does not leak any information. After verification that email IDs are not shown during messages exchange, it is ensured that replay attacks are also encountered. Timestamp and nonce are used to avoid replay attacks and show aliveness of system. Other than this, roles of receiving proxy server and master server are also defined. Communication between proxy server and master server is also verified and claims have been defined in Scyther script

Claims that are verified include Niagree, Secrecy, Nisync and Aliveness. Aliveness and secrecy are already explained. Niagree [43] stands for non-Injective agreement. It means that the protocol is completed between the two intended parties and the exchange of messages are same while Nisync [44] stands for non-injective synchronization, it means that Niagree is achieved and also the messages are exchanged in order.

The result of Scyther has proved the claims of implemented protocol. Secrecy of email identities of both sender and receiver is the first priority and hence is the first claim of our system as well. Scyther has verified that this claim is achieved and IDs remain secure and unrevealed. It can be clearly observed from the figure 5.1 that privacy of credentials is achieved at each step irrespective of client side or server side. Furthermore these are protected from attackers as well. The second claim which we have verified is the aliveness of the system. In an email system there is exchange of messages between two parties, it is very critical to make sure that message are not sent by attacker and these messages are not being replayed. To avoid this attack digital signature and timestamp are used. By doing this confidentiality integrity and privacy are achieved for each role.

## 5.1 Scyther verification Output



| Claim | | | | Status | | Comments |
|---|---|---|---|---|---|---|
| p1 | Sender | p1,c1 | Secret password | Ok | Verified | No attacks. |
| | | p1,c2 | Secret Ts | Ok | Verified | No attacks. |
| | | p1,c3 | Secret Nc | Ok | Verified | No attacks. |
| | | p1,c4 | Secret emailcontent | Ok | Verified | No attacks. |
| | | p1,c5 | Niagree | Ok | Verified | No attacks. |
| | SPS | p1,p1 | Secret password | Ok | Verified | No attacks. |
| | | p1,p2 | Secret Ts | Ok | Verified | No attacks. |
| | | p1,p3 | Secret Nc | Ok | Verified | No attacks. |
| | | p1,p4 | Secret emailcontent | Ok | Verified | No attacks. |
| | | p1,p5 | Secret queryMessage | Ok | Verified | No attacks. |
| | | p1,p6 | Secret H(AnonymousEmailId,ip) | Ok | Verified | No attacks. |
| | | p1,p7 | Niagree | Ok | Verified | No attacks. |
| | | p1,p8 | Weakagree | Ok | Verified | No attacks. |
| | | p1,p9 | Alive | Ok | Verified | No attacks. |
| | MS | p1,ms1 | Secret queryMessage | Ok | Verified | No attacks. |
| | | p1,ms2 | Secret Ts | Ok | Verified | No attacks. |
| | | p1,ms3 | Secret Nc | Ok | Verified | No attacks. |

Figure 5.1-Output of Scyther

As it can be seen in figure 5.1, there are roles of Sender, Sender Proxy Server (SPS) and Master Server (MS). Ts represents timestamp, Nc represents and H shows hashing. Claims of secrecy, Niagree, alive are also shown. Details of these claims are explained in previous section.

## 5.2 Results in performance of Email System

In this part, I have analyzed performance of email system by carrying out different tests. Different tests are designed to evaluate the performance of system. To perform this testing, email accounts of 50 users are created on Hmail Server. Tests are divided into two categories, in first category I have sent a single message between two email accounts by using encryption module. In second category, a single message was exchanged between two clients when only two users were connected with email server. Then same message was exchanged when 50 users were connected to server. And then email messages were exchanged among them. Following are the details of test cases

In Test case 01. User sends email message, consisting of 2 lines of text carrying "*Hi Bob, How are you? Please call me when you are free*" First I sent this mail without using security module and noted down time taken by system, then in second phase I sent same email using security module and jotted down the difference of time.

| **Test Case Title** | **Real time performance Test on Single Domain** |
|---|---|
| Test Case ID | Test 01 |
| Test Case Objective | To measure the time taken by system to deliver email |
| Pre-Condition | Both users have email accounts |
| Post Condition | Receiver receives email |
| Procedure | 1. Sender successfully logs in to system<br>2. He sends email message to other ID. |
| Expected Result | Receiver successfully receives message |
| Actual Result | Received message |
| No of lines in Email message | 2 |
| Time taken in transmission of message (Without Security Module) | 3-4 Seconds |
| Time taken in transmission of message (With Security Module) | 5-6 Seconds |
| Status | Pass |

**Table 5.2-Test Case 01**

In above table 5.1 it is achieved that while using security module, email messages face a delay of 2 to 3 seconds. In second test, size of email message was same but this time email was exchanged between two users of different domains in order to obtain the time taken by system. This time email message was same as of test case 01. Following are the results of test

| **Test Case Title** | **Real time performance Test on Cross Domain** |
|---|---|
| Test Case ID | Test 02 |
| Test Case Objective | To measure the time taken by system to deliver email |
| Pre-Condition | Both users have email accounts |
| Post Condition | Receiver receives email |
| Procedure | 1. Sender successfully logs in to system<br>2. He sends email message to other ID. |
| Expected Result | Receiver successfully receives message |
| Actual Result | Received message |
| No of lines in Email message | 2 |
| Time taken in transmission of message (Without Security Module) | 5-6 Seconds |
| Time taken in transmission of message (With Security Module) | 7-8 Seconds |
| Status | Pass |

Table 5.2-Test Case 02

Table 5.2 shows that when email is exchanged between two different domains then delay increases as it is clear that firstly domain has to query to another domain about receiver and then it performs security operations so difference increases to 2 to 3 seconds.

In third test, size of email message was increased but email was exchanged on single domain. This time email message consists of four lines "*Hi Bob, How are you. Please call me when you are free. I need to talk to you about business plan. Thank You*" Following are the details of result

| **Test Case Title** | **Real time performance Test on Single Domain** |
|---|---|
| Test Case ID | Test 03 |
| Test Case Objective | To measure the time taken by system to deliver email |
| Pre-Condition | Both users have email accounts |
| Post Condition | Receiver receives email |
| Procedure | 1. Sender successfully logs in to system<br>2. He sends email message to other ID. |
| Expected Result | Receiver successfully receives message |
| Actual Result | Received message |
| No of lines in Email message | 4 |
| Time taken in transmission of message (Without Security Module) | 4-5 Seconds |
| Time taken in transmission of message (With Security Module) | 5-6 Seconds |
| Status | Pass |

Table 5.3-Test Case 03

Table 5.3 shows that when email message size was increased then delay was not increased much but difference was of only 1 second. In fourth test, message size was increased and this time is was sent in cross domain scenario. Message was "*Hi Bob, How are you. Please call me when you are free. I need to talk to you about business plan. Thank You*" Following are the results of test

| **Test Case Title** | **Real time performance Test on Cross Domain** |
|---|---|
| Test Case ID | Test 04 |
| Test Case Objective | To measure the time taken by system to deliver email |
| Pre-Condition | Both users have email accounts |
| Post Condition | Receiver receives email |
| Procedure | 1. Sender successfully logs in to system<br>2. He sends email message to other ID. |
| Expected Result | Receiver successfully receives message |
| Actual Result | Received message |
| No of lines in Email message | 4 |
| Time taken in transmission of message (Without Security Module) | 6-7 Seconds |
| Time taken in transmission of message (With Security Module) | 10-12 Seconds |
| Status | Pass |

Table 5.4-Test Case 04

Table 5.4 shows that in cross domain, delay increases with increase of email message size. Email was delayed by 3 to 4 seconds.

Results of above mentioned test cases are compiled in following table

| **Test Case ID** | **System Specifications** | **Domain** | **Length of Email Messages** | **Time taken Security Module Disabled** | **Time taken Security Module Enabled** | **Results** |
|---|---|---|---|---|---|---|
| 01 | Corei-5<br>2.3GHZ<br>8GB RAM<br>1 TB Hard Disk | Single | 52 Characters | 3-4 Seconds | 5-6 Seconds | 1-2 Seconds |
| 02 | Corei-5<br>2.3GHZ<br>8GB RAM<br>1 TB Hard Disk | Cross Domain | 52 Characters | 5-6 Seconds | 7-8 Seconds | 2-3 Seconds |
| 03 | Corei-5<br>2.3GHZ<br>8GB RAM<br>1 TB Hard Disk | Single | 52 Characters | 4-5 Seconds | 6-7 Seconds | 2 Seconds |
| 04 | Corei-5<br>2.3GHZ<br>8GB RAM<br>1 TB Hard Disk | Cross Domain | 52 Characters | 6-7 Seconds | 10-12 Seconds | 4-5 Seconds |

**Table 5.5 -Results Summary**

In second phase of performance testing, Selenium tool was used to test the efficiency of email system. In first test case two user logged in to system and exchanged a message "*A quick brown fox jumped over the lazy dog*". Following are results of that

| **Test Case Title** | **Real time performance Test on Single Domain** |
|---|---|
| Test Case ID | Test 05 |
| Test Case Objective | To measure the time taken by system to deliver email when only two users are logged in |
| Pre-Condition | Both users have email accounts |
| Post Condition | Receiver receives email |
| Procedure | 1. Sender successfully logs in to system<br>2. He sends email message to other ID. |
| Expected Result | Receiver successfully receives message |
| Actual Result | Received message |
| No of lines in Email message | Single line |
| Time taken in transmission of message | 5-6 Seconds |
| Status | Pass |

Table 5.6-Test Case 05

Table 5.6 shows that two email accounts were created on email server and email message was exchanged between them. The time taken in exchange of message was 5-6 seconds. In second test, as shown in Table 5.7, 50 email accounts were created and same message was exchanged between them. In this case it took 50-60 seconds to exchange message. So, performance was decreased.

| **Test Case Title** | **Real time performance Test on Single Domain** |
|---|---|
| Test Case ID | Test 06 |
| Test Case Objective | To measure the time taken by system to deliver email when 50 users are logged in |
| Pre-Condition | Both users have email accounts |
| Post Condition | Receiver receives email |
| Procedure | 1. Sender successfully logs in to system<br>2. He sends email message to other ID. |
| Expected Result | Receiver successfully receives message |
| Actual Result | Received message |
| No of lines in Email message | Single line |
| Time taken in transmission of message | 50-60 Seconds |
| Status | Pass |

Table 5.7-Test Case 06

## 5.3 Summary

This chapter describes the verification of claims and performance of proposed system. In first part ourput of Scyther is explained. Main claim was of secrecy of original email addresses of source and destination. It was proved by scyther. Besides, protocol is also verified against Niagree, Weakagree and result was successful. First claim of secrecy was achieved through PKI and anonymous header. As original header was encrypted using receiver domain's proxy server, so any intruder cannot look at original header. Second claim of Niagree is used to make sure that exchange of messages between two entities is complete and same content is received as was sent. For this digital signatures are used to verify sender and receiver. Third claim of making sure that no replay message is transmitted is tested. For this nonce and time stamps are used. In second part performance tests were carried out to analyze the time taken by system to exchange email messages. Results of tests are shared along with test cases. This chapter was divided into two parts, in first part we have explained the claims, and second part has test cases. Scyther code is attached in appendix for verification. Summary of results is as follow

| Test Cases | Time taken in original System | Time taken in proposed System | No of Users |
|---|---|---|---|
| 1 | 3-4 Seconds | 5-6 Seconds | 2 |
| 2 | 5-6 Seconds | 7-8 Seconds | 2 |
| 3 | 4-5 Seconds | 5-6 Seconds | 2 |
| 4 | 6-7 Seconds | 10-12 Seconds | 2 |
| 5 | 3-4 Seconds | 5-6 Seconds | 2 |
| 6 | 5-6 Seconds | 50-60 Seconds | 50 |

Table 5.8- Tests Summary

# *Chapter Six*

## Conclusion and Future Directions

### 6.1 Conclusion:

The exchange of sender and receiver's critical information in clear text form is violation of privacy. After analysis of existing email systems, a need of an email system that can hide the email addresses of communicating parties was felt. Therefore, we have proposed a new layer utilizing proxy server for privacy preservation. We have implemented an email system with features of security of email content and privacy of information related to email addresses. Authentication and authorization policies are implemented on end user's local machine. Our main goal was to protect email IDs and we have tried to achieve it by encrypting original IDs. Now if an attacker intercepts messages, he cannot identify the sender and receiver. Only domain can be identified. In second phase, we have verified our system by using verification software called Scyther. We have created roles of Client, Proxy servers, Master Server and Standard Email Server. All messages are secured using public key cryptography. By verification it is evident that addresses are not exposed at any point.

### 6.2 Future Directions:

In future features of spam control and virtual smart cards can be embedded with existing email solution. TPM can be used to secure private keys of user and these keys will be used to protect passwords. Virtual Smart cards can be embedded with this email system as real smart cards are expensive and they require smart card readers. So, virtual smart card feature of windows operating system can be used. Spam messages is another area where a policy server can be used to reflect policies on this email system. Attachments can be inspected to prevent from viruses and Trojan horses. A mechanism can be developed to inspect attachments.

# References

[1] A. Kapadia, "A Case (Study) For Usability in Secure E-mail Communication," in *Security & Privacy,IEEE, vol. 5, pp. pp. 80-84.*, 2007.

[2] M. Banday, Design and Development o f Efficient Techniques fo r Securing E-mail System from threats, India, 01-May-2015.

[3] F. B. M. Mir, ""Control of Spam: A Comparative Approach with special reference to India," in *Journal of Information Technology Law, UK, 19(1), pp.22-59, ,* 2010.

[4] A. Abbassi, "CryptoNet: Design and implementation of the Secure Email System," in *1st International Workshop on Security and Communication Networks (IWSCN, 2009, pp. 1-6.*, 2009.

[5] L. Mitrou and M. Karyda, "Employees' privacy vs. employers' security: Can they be balanced?," in *Telematics and Informatics, vol. 23, pp. 164-178,*, 2006.

[6] [Online]. Available: https://dcposch.github.io/scramble/. [Accessed 14 February 2017].

[7] [Online]. Available: https://github.com/bertjohnson/OpaqueMail. [Accessed 14 February 2017].

[8] [Online]. Available: https://addons.mozilla.org/en-US/firefox/addon/gmail-smime/. [Accessed 13 February 2017].

[9] G. Kardas, "A Smart Card Mediated Mobile Platform for Secure E-Mail Communication," in *Fourth International Conference on Information Technology, 2007. ITNG' 07 Volume, Issue, pp. 925 – 928, April*, 2007.

[10] ""Neomailbox" [Online]. Available: https://www.neomailbox.com/. [Accessed: 16-MAY-2016]".. [Online]. Available: https://www.neomailbox.com. [Accessed 04 February 2017].

[11] A. Joiya, "Secure and Privacy Enhanced Email System as a Cloud Service," in *IEEE.*, 2013.

[12] I. Kounelis, "Secure and Privacy-enhanced E-mail System based on the Concept of Proxies," in *MIPRO 2014, 26-30 May, Opatija Croatia.*, 2014.

[13] [Online]. Available: https://www.anonymizer.com. [Accessed 02 January 2017].

[14] J. Claessens, "Solutions for Anonymous Communication on the Internet," in *EEE*, 1999.

[15] P. R. J. S. &. J. E. Daniel Jones, Cambridge Dictionary, Cambridge University Press , 2017.

[16] [Online]. Available: https://www.cmu.edu/ices/outreach/see/research-process.html. [Accessed 10 January 2017].

[17] [Online]. Available: http://guides.lib.umich.edu/c.php?g=283022&p=1885747. [Accessed 10 January 2017].

[18] [Online]. Available: https://olinuris.library.cornell.edu/content/seven-steps-research-process. [Accessed 10 January 2017].

[19] [Online]. Available: http://research-methodology.net/research-methodology/research-process/. [Accessed 10 January 2017].

[20] [Online]. Available: http://www.humankinetics.com/excerpts/excerpts/steps-of-the-research-process. [Accessed 10 January 2017].

[21] [Online]. Available: https://www.reference.com/business-finance/analytical-research-94534a536bf46028#. [Accessed 11 January 2017].

[22] [Online]. Available: http://www.csus.edu/indiv/y/yangy/145ch1.htm. [Accessed 11 January 2017].

[23] [Online]. Available: http://www.aect.org/edtech/ed1/41/41-01.html. [Accessed 11 January 2017].

[24] [Online]. Available: http://research-methodology.net/research-methodology/research-types/applied-research/. [Accessed 11 January 2017].

[25] A. R. Hevner, "Design Science in Information Systems Research," *MIS Quarterly,* vol. Vol. 28 No. 1, p. 77, 2004..

[26] [Online]. Available: http://research-methodology.net/research-methodology/research-approach/deductive-approach-2/. [Accessed 11 January 2017].

[27] [Online]. Available: http://www.socialresearchmethods.net/kb/dedind.php. [Accessed 11 January 2017].

[28] [Online]. Available: http://libguides.usc.edu/writingguide/quantitative. [Accessed 11 January 2017].

[29] [Online]. Available: https://www.snapsurveys.com/blog/what-is-the-difference-between-qualitative-research-and-quantitative-research/. [Accessed 11 January 2017].

[30] [Online]. Available: https://www.enago.com/academy/conceptual-vs-empirical-research-which-is-better/. [Accessed 11 January 2017].

[31] [Online]. Available: http://guides.libraries.psu.edu/emp. [Accessed 11 January 2017].

[32] A. R. Hevner, "Design Science in Information Systems Research," *MIS Quarterly ,* vol. Vol. 28 No. 1, p. 77, 2004.

[33] J. R. Venable, "The Role of Theory and Theorizing in Design Science Research," in *DESRIST*, 2006.

[34] S. Saleem, "Protecting the Integrity of Digital Evidence and Basic Human Rights During the Process of Digital Forensics," p. 14, 2015.

[35] [Online]. Available: https://www.neomailbox.com. [Accessed 16 December 2016].

[36] [Online]. Available: http://www.mailanon.com/. [Accessed 20 October 2016].

[37] [Online]. Available: https://www.anonymier.com/. [Accessed 20 October 2016].

[38] [Online]. Available: https://scramble.io. [Accessed 20 October 2016].

[39] [Online]. Available: http://en.wikipedia.org/wiki/S/MIME. [Accessed 15 September 2016].

[40] [Online]. Available: http://people.inf.ethz.chlcremersc/scyther/. [Accessed 14 February 2017].

[41] C. J. F. Cremers, "Scyther - Semantics and Verification," 2006, pp. 36-38.

[42] V. A. a. C. Diaz, "Computer Security," in *ESORICS 2011: 16th European Symposium on Research in Computer Security*, Belgium, September 12-14, 2011.

[43] N. R. P. N. H. N. N. Shetty, "Emerging Research in Computing, Information, Communication and Applications," in *ERCICA*, 2015.

[44] N. R. P. N. H. N. N. Shetty, "Emerging Research in Computing, Information, Communication and Applications," in *ERCICA*, 2015.

# Appendix 1

## Scyther Script

```
usertype EmailId,Domain,IP,SessionKey,
Packet,TimeStamp,Alphnum,EmailMessage,custom
Message;
const Fresh:Function;
protocol p1(SPS,Sender,RPS,Receiver,MS)
{
role Sender
{
fresh Nc: Nonce;
var nr: Nonce;
fresh Ts:TimeStamp;
fresh email:Packet;
hashfunction H;
fresh username:Alphnum;
fresh password:Alphnum;
fresh emailcontent:EmailMessage;
var skey:SessionKey;
// Registration between client and proxy server
send_1(Sender, SPS,{ H(
{username,password,Ts,Nc}pk(SPS)) } sk(SPS) );
send_2(Sender,
SPS,{H({emailcontent,Ts,Nc}pk(SPS))}sk(SPS) );
// claims at client side
claim_c1 (Sender, Secret,password);
claim_c2 (Sender, Secret,Ts);
```

```
claim_c3 (Sender, Secret,Nc);
claim_c4 (Sender, Secret,emailcontent);
claim_c5(Sender, Niagree);
}
role SPS
{
var packet:Packet;
var ip:IP;
var ni: Nonce;
hashfunction H;
fresh Nc: Nonce;
var nr: Nonce;
fresh Ts:TimeStamp;
fresh username:Alphnum;
fresh password:Alphnum;
fresh emailcontent:EmailMessage;
fresh queryMessage:customMessage;
fresh AnonymousEmailId:EmailId;
// Registration between client and proxy server
recv_1(Sender,
SPS,{H({username,password,Ts,Nc}pk(SPS)) }sk(SPS)
);
recv_2(Sender,
SPS,{H({emailcontent,Ts,Nc}pk(SPS))}sk(SPS) );
send_3(SPS,MS,{H({queryMessage,Ts,Nc}pk(MS))}sk(
SPS));
recv_4(MS,SPS,{H({AnonymousEmailId,ip,Ts,Nc}pk(S
PS))}sk(MS));
```

```
send_5(SPS,RPS,{H({emailcontent,Ts,Nc}pk(RPS))}sk(
SPS));
// claims at proxy server side
claim_ p1 (SPS,Secret,password);
claim_p2 (SPS, Secret,Ts);
claim_p3(SPS, Secret,Nc);
claim_p4(SPS,Secret,emailcontent);
claim_p5 (SPS,Secret,queryMessage);
claim_p6(SPS,Secret,H(AnonymousEmailId,ip));
claim_p7(SPS, Niagree);
claim_p8(SPS, Weakagree);
claim_ p9 (SPS,Alive);
}
role MS
{
fresh ip:IP;
fresh queryMessage:customMessage;
hashfunction H;
fresh AnonymousEmailId:EmailId;
fresh Nc: Nonce;
var nr: Nonce;
fresh Ts:TimeStamp;
recv_3(SPS,MS,{H({queryMessage,Ts,Nc}pk(MS))}sk(
SPS));
send_4(MS,SPS,{H({AnonymousEmailId,ip,Ts,Nc}pk(S
PS))}sk(MS));
claim_ms1 (MS,Secret,queryMessage);
claim_ms2 (MS, Secret,Ts);
```

```
claim_ms3(MS, Secret,Nc);
claim_ms4(MS,Secret,H(AnonymousEmailId,ip));
claim_ms5(MS, Niagree);
claim_ms6(MS, Weakagree);
claim_ ms7(MS,Alive);
}
role RPS
{
hashfunction H;
fresh emailcontent:EmailMessage;
fresh Nc: Nonce;
var nr: Nonce;
fresh Ts:TimeStamp;
recv_5(SPS,RPS,{H({emailcontent,Ts,Nc}pk(RPS))}sk(S
PS));
send_6(RPS,Receiver,{H({emailcontent,Ts,Nc}pk(Rec
eiver))}sk(RPS));
// claims at proxy server side
claim_sp1 (RPS, Secret,Ts);
claim_sp2(RPS, Secret,Nc);
claim_sp3(RPS, Niagree);
claim_sp4(RPS, Weakagree);
claim_ sp5 (RPS,Alive);
}
role Receiver
{
fresh email:Packet;
var skey:SessionKey;
```

```
hashfunction H;
fresh Nc: Nonce;
var nr: Nonce;
fresh Ts:TimeStamp;
fresh emailcontent:EmailMessage;
recv_6(RPS,Receiver,{H({emailcontent,Ts,Nc}pk(Rece
iver))}sk(RPS));
//claims
claim_r1 (Receiver, Secret,Ts);
claim_r2(Receiver, Secret,Nc);
} }
```