Secure and Privacy Enhanced Email System as a Cloud Service

By

Amna Joyia

2010-NUST-MS PhD-IT-24

Thesis Supervisor

Dr. Abdul Ghafoor

Department of Computing

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

In

Department of Computing (DoC)
School of Electrical Engineering & Computer Science (SEECS)
National University of Sciences & Technology (NUST),
Islamabad, Pakistan
(2013)

Approval

This thesis has been submitted in partial fulfillment of requirements for the Master of Information Technology at National University of Sciences & Technology.

It is certified that the contents and form of the thesis entitled “**Secure and Privacy Enhanced Email System as Cloud Service**” submitted by **Amna Joyia** have been found satisfactory for the requirement of the degree.

Advisor: **Dr. Abdul Ghafoor**

Signature: ______________________________

Date: ______________________________

Committee Member 1: **Prof. Saed Muftic**

Signature: ______________________________

Date: ______________________________

Committee Member 2: **Dr. Madiha Shehzad**

Signature: ______________________________

Date: ______________________________

Committee Member 3: **Mr. Qasim Rajpoot**

Signature: ______________________________

Date: ______________________________

Dedication

To My Parents

Mr and Mrs. Zafar Iqbal Joyia

And

My Elder Sister

Maryam Joyia

Certificate of Originality

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials 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.

Amna Joyia

Signature: ______________________________

Acknowledgement

This thesis would not have been possible without the continuous support of my supervisor **Dr. Abdul Ghafoor**..

Also thanks to my committee members, **Prof. Saed Muftic**, **Dr. Madiha Shehzad**, and **Mr. Qasim Rajpoot**, for their guidance and support.

And finally, thanks to my family specially my elder sister Maryam Joyia and numerous friends who endured this long process with me, always offering support and love.

Contribution

Following is the research paper published as conference paper:

Amna Joyia, Abdul Ghafoor, Maryam Sajjad, and Qaisar Chaudary, “Secure and Privacy Enhanced Email System as a Cloud Service”, published in the proceeding of 8th IEEE International Conference on Digital Information Management (ICDIM-2013), pp. 73-78, Islamabad, Pakistan, September 10-12,2013.

Table of Contents

Contents

List of Abbreviations

Abbreviation	**Stands for**
PSA	Proxy Server at Domain A
PSB	Proxy Server at Domain B
IEMS	Infrastructure Email Server
TCA	Top Certification Authority
SEMS	Standard Email Server
TCP	Transmission Control Protocol
PGP	Pretty Good Privacy
S/MIME	Secure/Multipurpose Internet Mail extensions
AH	Anonymous Header
OPH	Original protected Header
AID	Anonymous Identity

List of Figures

Abstract.

Email system is very important source for organizations and individuals to exchange information between employees, colleagues and friends. Currently available email standards provide protection of email letters using standard cryptographic techniques and formats like PGP, S/MIME. All these standards focus on the protection of email contents rather than the header of email. Clear transmission of headers is the source of privacy leakage and most of social security engineers have very serious concerns about this. They argue that the intruders can easily establish the link between the sender and receiver after intercepting their emails. Furthermore, if someone can extract our email addresses then he/she can send spam messages which are the main cause of information flooding and garbage in the inboxes. Considering the current problems, we have analyzed the existing email systems and found that the current systems do not provide the feature to exchange anonymous emails between users belonging to two different domains (inter-domains).In addition to that, these systems do not enforce source and destination authentication policies which are main cause of spamming.

With the shifting of deployment infrastructural paradigm from conventional arrangements to cloud computing environment, email users and organizations have more concerns about their privacy and personal data. To solve these problems, a completely different approach has been taken in this research activity to design a complete privacy enhanced secure email system. The system is based on proxy architecture to provide standard email services along with extended and innovative features. Some of the extended features are: (a) protection of email headers using standard cryptographic format, (b) transparent handling of anonymous identities belonging to different domains, (c) protection of inboxes from unauthorized emails. The designed system is implemented in the form of a service using standard techniques so it can be deployed easily in the cloud environment as a service. The system also supports cross domain exchange of email letters. It transparently and securely exchanges user’s private information across the domain after developing infrastructure level trust between them. After designing and implementing, we have verified our system using automated verification tool; Scyther. We found that the original email ids of sender and receiver both are secured along with the aliveness and secrecy of the system.

1. Introduction

Email is a popular application which is being used by organizations to exchange their valuable information. An Internet email message comprises of three main constituents, the message envelope, the message header, and the message body. The header of message contains piece of information that actually controls email transportation; it includes an initiator's email address and one or more recipient addresses. Additionally some descriptive information such as a subject header field and a message submission date/time stamp are also enclosed in an email header. In the current email systems like Yahoo or Gmail, different cryptographic standards and techniques are being used to protect the email letters. Some of them are PGP [12], which provides cryptographic privacy and authentication for data communication and S/MIME [13], which is responsible for public key encryption and signing of MIME data. All of these standards focused on the protection of the email letters cryptographically but did not address the privacy concern of the users. Current email users can be easily tracked as most of the header information is in clear text [1] including *"To"* and *"From"* fields of Email header. These fields can be used by the intruder to disclose who is sending the email to whom. Through email headers, not only the source of the message can be traced but also the list of every point along which the mail has traveled can be drawn out. Moreover, some of current web based email service providers [2] also extract the user's private information like sender's and recipients' email address and uses this information to generate spam messages. In order to make spam protectors, email verification methods are used to prepare messages with certifiable information adequate for the receiver to distinguish the nature of arriving messages. The Transmission Control Protocol (TCP) and the IP address registries aid recipients of emails to attest the IP addresses of the sender. Banning or blacklisting attempts to isolate IP addresses of spammers who intend to breach email privacy. However, as a result of the use of dynamic IP addresses, blacklisting fails to become a perfect plan to fight against spamming issue.

As per last discussion the existing email systems have neither proposed any mechanism which protects email headers using standard cryptographic format. In the same way most of them address privacy issues of email system arises from intercepting email content only, but they have not focused on protection of user's privacy violating information of email header. Consequently, now there is a need to intend and implement an email system which protects the information of identities of communicating parties and helps to avoid spam generated as a result of extraction of this vital information. The second main purpose is that this system should be interoperable with the existing standards and formats. Like sending and receiving mechanisms of email messages should support current procedures of transference of an email.

1.1 Objectives

As discussed in previous section, current email systems have some cons which are the cause of violation of user’s privacy. To unfold these privacy vulnerabilities, we defined some major objectives for this research activity which provided assistance in designing and developing the secure and privacy enhanced email system. The following are the goals and objectives which have been drawn by keeping in view the highlighted problems in existing email systems.

- Protection of Email contents.
- Inter and intra domain exchange of email messages.
- Protection of email headers using standard cryptographic format.
- Transparent handling of anonymous identities belonging to different domains.
- Protection of inboxes from unauthorized emails.

1.2 Motivation

While developing and deploying a system, there are two factors for which security engineers have to take care about i.e. security and privacy. The first factor, security has been covered by current email system by using standard cryptographic techniques but the second factor of privacy is violating because of transmission of email header in clear text. So, keeping in view this privacy vulnerability in existing email systems, we have defined a layered architecture explained in chapter 3. This architecture performs some distinct steps at specific layers to solve the raised problem.

Secondly, in our country, a very less importance is given to the security and privacy of users so our system will help to develop a culture in country to consider the importance of email security.

In addition, it will also help in professional organizations and government agencies to use this system to exchange their confidential information without any threat of revealing their identities.

1.3 Research Methodology

A research is a systematic study of phenomenon and sources in order to establish facts and reach new conclusions. There are two main approaches for scientific research known as deductive research and inductive research. Deductive research approach works from the more general to the more specific, also known as top-down approach. On contrary inductive research approach works from specific observation to broader generalizations and theories, informally called as bottom-up approach [15]. Figure 1 demonstrates the normal cycle of research follows in deductive research approach.

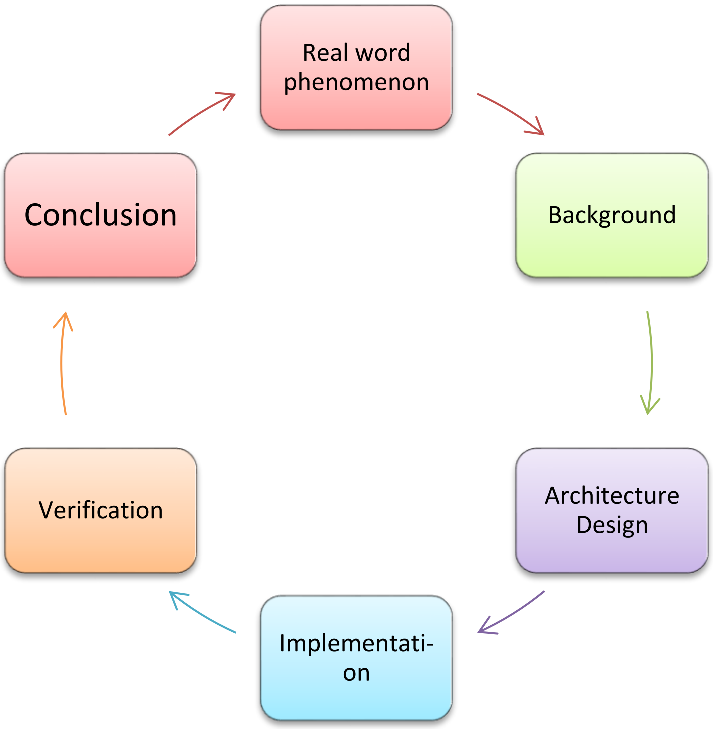


Figure 1: Deductive Research Approach

The aim of the current research activity is to describe the problem and then to draw the conclusion by narrowing down the focus. So, I have adopted deductive approach to solve this research problem. The deductive research approach comprises of four major methodologies i.e. a) Theory b) Hypothesis c) Observation and d) Confirmation. First hypothesis is derived from extensive literature review. Then the observations are made to approve or condemn the hypothesis.

At the end, verification of hypothesis has been done through automated verification tool; Scyther. I have described these different phases in the chapters mentioned in following table 1. In the first phase problem is identified through the deep literature survey. In second phase, first I designed the architecture of the system and then implemented it. Finally in third phase I have verified the claims and objectives designed for the system using a verification tool.

Phases	**Research Methods**	**Outputs**	**Chapter(s)**
1	Literature Survey	Identification of problem	2
2	System design and Implementation	Design and implementation	3,4
3	System verification	Verification	5

Table 1: Phases of Deductive Approach

1.4 Contributions

As now days 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 email services along with extended and innovative features. Some of the extended features are: (a) protection of email headers, (b) anonymous identities for participating identities, (c) privacy against tracking of user’s identities. The designed system is implemented in the form of a service using standard techniques so it can be deployed easily in the cloud environment as a service.

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 conceived idea to overcome the leakage of user identities of participating entities is to introduce the anonymous identity for each user. By using the implemented 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. The system also supports cross domain exchange of email letters. It transparently and securely exchanges user’s private information across the domain after developing infrastructure level trust between them using infrastructure email server (IEMS) deployed in top layer of architecture. Moreover, IEMS acts like source to delegate trust to proxy servers. 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 client at sender side sends the email with AIDs to standard email server (SEMS). SEMS directly sends an email to the recipient and the proxy server at recipient side is not involved in this process. 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, so no intruder attack is possible. 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.

.

1.5 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.

2. Related Work

2.1 Attributes of Secure Email System

[1] In a case study done by A. Kapadia on usability of secure emails, he emphasized that we should done communication on E-mails securely. He explained the basic mechanisms which make emails secure, issues regarding these mechanisms and proposed solutions for those issues, which are as follows:

- Digital Signature
    - Avoids non repudiation
        - Either sender or receiver cannot deny the sent and received message
    - Ensures Integrity
        - Receiver should receive exact e-mail content sent by sender
- Digital signature involves PKI (Public key infrastructure)
- Issues in PKI
    - Secure Key distribution
- Solution:
    - Trusted third party (CA) Certification authority
    - Finger printing (Hashing)

Analysis: This paper presents the basic cryptographic mechanisms by which we can make email system secure by protecting email content only, but he didn't talk about privacy of users involved in communication that how we will protect user's personal information and transparent handling of security credentials.

2.2 Privacy Issues

In **[2]** authors emphasize on the issue that how employer can detect the capability of employee to breach the policy of an organization via e-mails without violating the privacy rights of an employee. This paper mainly proposes a design and a protocol which give an employer the opportunity to monitor employee email in order to detect company policy violations. This can be achieved without violating the privacy of honest employees, while at the same time revealing evidence about the illegal actions of dishonest employees. Proposed protocol is divided into two parts:

1. **Sending a Message:**

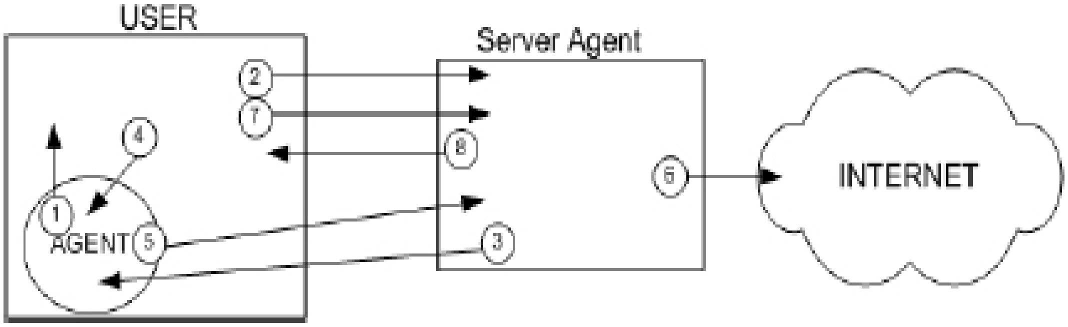


Figure 2- Message Sending Process

There are two agents

- Agent (installed at employee's machine)
- Server Agent (SA) (Installed at Employer side)

The first three steps in protocol will exchange session keys between Agent and SA which are as follows:

1. *Agent →User: "request keys"*
2. *User →SA: S USER_private-key ("request", NumKeys)*
3. *SA →Agent: S SA_private-key(ID, {key}K AGEN_Tpublic-key)*

Then in next steps user send data msg and crypto parameter (by which agent will encrypt message for SA) encrypted with user private key .

4. *User→Agent:S USER_private-key(DataMsg, CryptoPar)*

Then agent will check if message is encrypted, it stops processing the user request but if message is not encrypted then check the email content according to the policies of organization either the email content is valid or not and if it is valid, Agent sends it to the SA by encrypting message with CrptoPar received with data message otherwise it does not encrypt it and send the list of rules which have been violated with data message to SA:

5. *Agent →SA:SAGENTprivate-key({Msg, Flag}KSkey, ID,Version)*

The SA decrypts the Data and forwards the Msg to the destination, only in case the message is legal.

6. *SA →SERVER: Msg*

Although the SA is able to identify whether a message is malicious or not based on the Flag, the SA is not able to have access to the content of the message, in case the Message is encrypted. The SA will be able to have access to the content of the Message only if the message is malicious. The user asks the SA to provide evidence showing to the user that the exchanged messages did not violate the privacy of the user

7. *User→SA:SUSERprivate-key(ID)*

The SA sends the evidence

8. *SA →User:{S SA_private-key(key,ID,Version)}KUSERpublic-key*

The user can then verify whether the agent has sent (in step 5) private information to the SA. Also, the Flag must be zero if the message was considered legal. Otherwise, the Flag contains the IDs of the rules that the message violates. If the key given in step 8 can decrypt the [*{Msg, Flag}KS*key] from step 5 and the message is indeed the expected one, then the key (from step 8) is the one used. Otherwise, it is not the valid key.

2. **Receiving a message:**

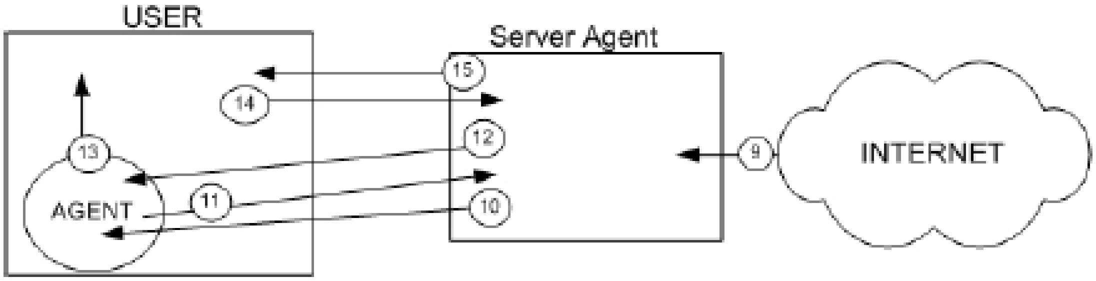


Figure 3-message Receiving Process

The SA receives a message from an Internet user. Based on the DestEmailAddress, the SA determines the destination of the email.

9. *ReceivedMsg=DestEmailAddress+ {ReceivedDataMsg}KUSER-AGENTpublic key*

Remote User →SA: ReceivedMsg

10) The SA encrypts, signs and forwards the received message to the appropriate agent through the related employee.

SA →Agnet: SSAprivate-key(ID, {ReceivedMsg}KSkey)

11) The agent uses the USER-AGENTpublic-key encrypting and accessing the ReceivedDataMsg. The agent decrypts the ReceivedDataMsg by using the USER-AGENTpublic-key and checks it if it is legal. The Counter is a number which is increasing by one each time

```
IF the ReceivedMsg is not illegal THEN
Comments= Counter
ELSE
Comments=Violated RuleID + SecurityParameters
END IF
```

If the agent detects an illegal message, the SA is informed about the violated rule as well as the necessary security parameters to help SA decrypt the encrypted ReceivedMsg.

Agent →SA: SAGENTprivate-key({Comments}KSkey, ID, Version)

12) The SA confirms that he received the comments. This is a necessary step because a malicious user could prevent the message from step 11 reaching the SA.

SA→Agent:SSAprivate-key({Comments}KSkey, ID, Version)

13) If the ReceivedMsg is not illegal, the agent will give the ReceivedMsg to the user. Otherwise, the agent will not give it to the user.

Agent →User: ReceivedMsg

14) The user asks the SA to provide evidence showing to the user that the exchanged messages between the agent and the SA didn't violate the privacy of the user

User →SA: SUSERprivate-key(ID)

15) The SA sends the evidence, where the user verifies whether the agent has violated (in step 5) the privacy of the user or not.

SA→User:{SSAprivate-key(key,ID,Version)}KUSERpublickey

Analysis: This protocol is handling the privacy issues at communication level but not at user level. So currently, most of the work has been done on privacy at communication level but there is lack of protocol which will handle extended privacy issues at user level other than revealing email content.

2.3 Management of Email Messages

[3] This paper addresses two main privacy issues in Email system which is as follows:

1. vulnerability of software-based keys used to protect email messages
2. Management of large copies of email messages held in the backup systems of intermediaries.

2.3.1 Protocol:

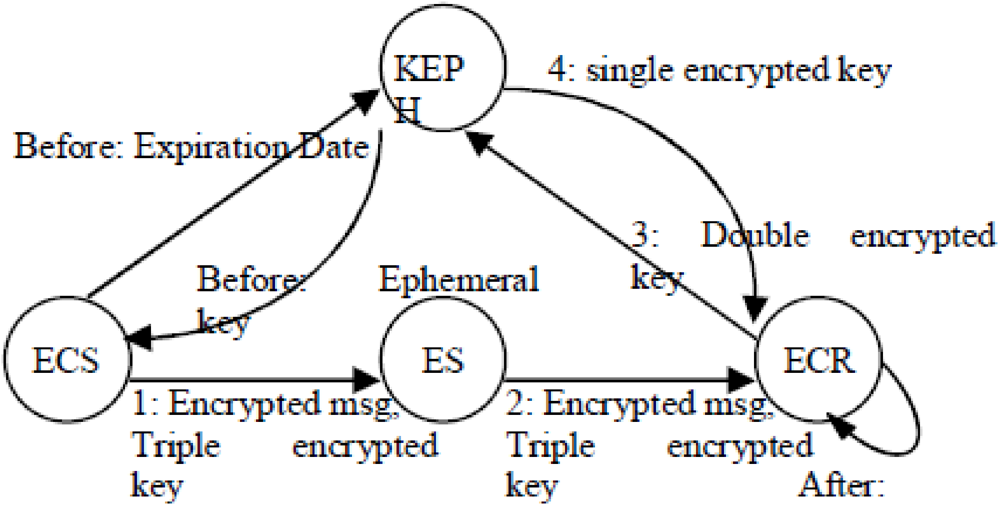


Figure 4-Flow of Protocol

Step 1: First, the ECS defines an expiry date for an email message and requests the KEPH to issue an ephemeral key, which returns the public part of the ephemeral key *KEPHpub* created for this particular email. Though not shown in the Figure, the ECS also receives the public part of an AIK key *AIKECRpub* from the ECR using the mutual attestation protocol described earlier as well as the public key of an Email Server *ESpub* from a public directory. The ECS creates an email message *msg* to send to the ECR and creates a secret key *s* to encrypt the message *{msg}s*. Since whoever holds the secret key *s* is able to decrypt the message, we protect the secret key using triple encryption. The secret key *s* is first

encrypted using the ECR'S AIK public key as *{|s|}AIKECRpub*. This will ensure that *s* can only be decrypted by a designated TPM-enabled platform that has a matching private part of AIK public key, namely the ECR. The secret key s is further encrypted with the public part of ephemeral key, denoted as *{|{|s|}AIKECRpub|}KEPHpub* to make sure only email messages with valid expiration date can be read. The final message sent from the ECR to an ES is: ECS-> ES: {|*{|{|s|}AIKECRpub|}KEPHpub, {msg}s|} ESpub*

Step 2: The ES routes the encrypted email message through many intermediaries whose public keys are available from a public key directory.

ES->ECR:*{|{|{|s|}AIKECRpub|}KEPHpub,{msg}s|} AIKECRpub*

Step 3: On receiving the encrypted message, the ECR sends the double encrypted secret key to KEPH.

ECR-> KEPH: *{|{|s|}AIKECRpub|}KEPHpub,*

Step 4: The KEPH selects the appropriate ephemeral key used for the received double encrypted secret key and checks the expiration date of the ephemeral key. If the expiration date has not expired, KEPH decrypts the requested blob and sends it back to the ECR. Otherwise the email is unrecoverable.

KEPH-> ECR: *{|s|}AIKECRpub,*

When the blob is received, the ECR decrypts the secret key *s* using the matching AIK private key. Subsequently, the ECR can now decrypt the encrypted email message *{msg}s* using the secret key *s*.

Analysis: These addressed issues involves privacy issues at communication level and handling of security credentials but again this protocol is not taking an account of user personal information hiding from server.

2.4 Email-based Social Network Trust

[4] This paper deals with Email-based Social Network Trust, It presents EMT email trust model which generates trust model on the basis of interactions between users. Trust is calculated through trust based proxy server which calculates trust by extracting user's email statistics information from email servers based on privacy mode. There are three types of privacy modes between user and trust calculating server

- ***Minimal privacy protection:*** User opts to trust EMT server. EMT server will communicate with email server and download email information.
- ***Maximal privacy protection:*** User relays email information to the EMT server. They do not want EMT server to manage the client's email credentials. They have the ability to control the data that is being transmitted to EMT server. This necessitates the usage of a

client application at the user’s side. The client application can filter the information and send the filtered results to the EMT server.

- ***Moderate privacy protection:*** A hybrid system. User can switch between the above two mode

EMT server has two tiers of trust checking, After two users exchange their email addresses, the TCA will perform trust checking based on each other’s email addresses through the EMT server. First, EMT server needs to perform statistical

analysis of the email header content and related information such as folders or labels and contact list with respect to the email address (*ID*) whose trust level has to be ascertained. They call this as tier-1 trust checking. If the tier-1 trust checking cannot discover the trust directly between two email addresses (i.e., the checking email address does not in the checker’s contact list), EMT needs to run *tier- 2 trust checking*, i.e., the checker evaluates the email trust through his/her trusted email contacts.

Analysis: This paper presents EMT email trust model which measures trust among social network users. Again this trust model is exploiting privacy of user in its minimal and moderate privacy protection because it calculates the trust by extracting communication statistics of user. So, here there is a need to design an architecture which can protect user’s personal information without violating the privacy of users.

2.5 Summary

This chapter presents the existing work in research domain. It has elaborated the privacy issues in email system with their analysis found in literature survey. First of all attributes of secure email system and the impact of these attributes on privacy of email user has been described in this section. Then the privacy issue and the deep analysis of its remedy protocol related to the email system of an organization are explained. Third part of this section is based on the technique for management of large number of copies of email messages on intermediate nodes in the network. In the end, EMT email trust model is defined which calculates trust among user of social network by extracting their interaction statistics which in cause raises privacy threats for communicating users. As a conclusion of this section, we have come up with the fact that there is a lack of protocol which protects user's private data in email header transmitted in clear text over the internet.

3. Designed Architecture of the System

Referring to the fore-mentioned problems, we have designed and implemented privacy enhanced email system based on proxy design following the layered architecture, shown in figure 5. The purpose to make it layered is to keep the design simple and understandable. The architecture comprises of three layers named as a) Client layer, b) Domain layer, c) Infrastructure layer. Each layer has its own roles and responsibilities according to the flow of procedure. Many key components are enclosed in these three layers.

Client layer has one main component that is client both at sender and receiver side. Domain layer consists of proxy server deployed at domain level separately at sender and receiver side respectively. The third infrastructure layer positioned at the top of the architecture has three significant components i.e. Infrastructure Email Server (IEMS), Top Certification Authority (TCA), Standard Email Server (SEMS). Every component performs its particular duty e.g. IEMS is a source to delegate trust between proxy servers, TCA is responsible to certify all communication between client and standard email server and SEMS actually routes the email over the internet.

Figure 5 is also representing the interaction among layers of architecture in brief. It is showing the communication of client with proxy server at sender side for requesting the anonymous email identities of participating entities and with standard email server (SEMS) for sending email message to the recipient. Moreover, it also presents the exchange of messages between sender's proxy server and infrastructure email server. While Top certification authority (TCA) is certifying the communication between sender and standard email server SEMS.

So, all these components enclosed in the three layers of architecture have their specific tasks to perform. The detail of these components is given in section 3.2 and the major functionalities along with interaction among layers are explained in section 4.

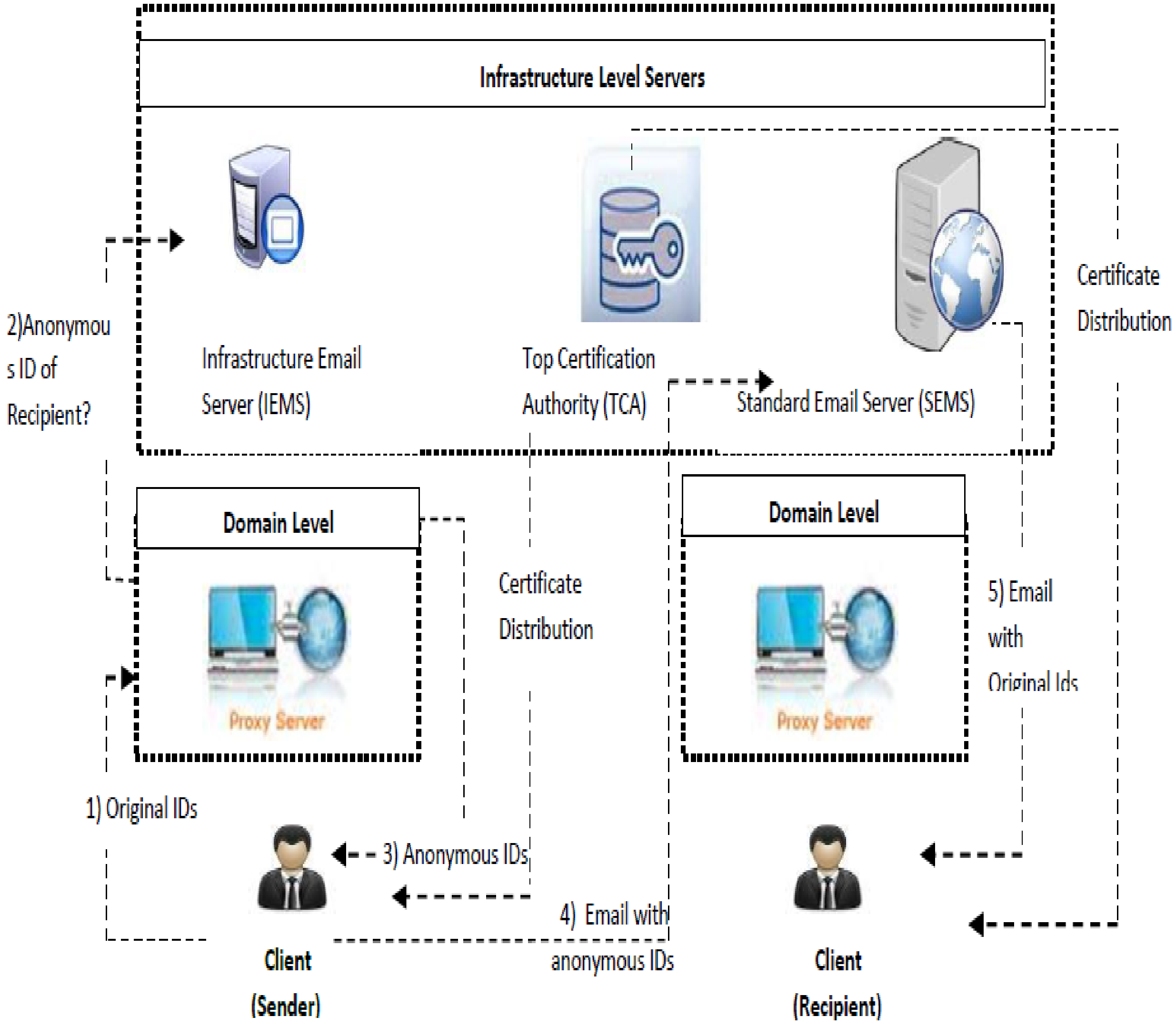

Infrastructure Level Servers
Infrastructure Email Server (IEMS)
Top Certification Authority (TCA)
Standard Email Server (SEMS)
Certificate Distribution
2)Anonymous ID of Recipient?
Domain Level
Proxy Server
Certificate Distribution
Domain Level
Proxy Server
5) Email with Original Ids
1) Original IDs
3) Anonymous IDs
4) Email with anonymous IDs
Client (Sender)
Client (Recipient)

Figure 5-Architecture of proposed solution

3.1 Components of the System

As shown in figure 5, the designed architecture comprises of three layers and many important components enclosed in these layers. This section describes the major components and their roles and responsibilities according to the implemented protocol.

3.1.1 Client Layer (Level-1)

Client: This is a standard email client who provides email sending and receiving functions. It directly interacts with proxy server to interrogate anonymous email identities of communicating parties and standard email server (SEMS) to send email message to receiver; All the communication between them is secure.

3.1.2 Domain Layer (Level-2)

Proxy Server: It is the most significant module of the middle layer. Its responsibility is to map original Id's to anonymous id's registered with standard email server SEMS. It is also responsible for interacting with the proxy server at receiving side to find out the anonymous email identity of recipient. The procedure of interaction between two proxy servers is explained in section 4.

3.1.3 Infrastructure Layer (Level-3)

a) ***Standard Email Server (SEMS):*** This is the major element of top layer which actually routes the email over the internet by extracting information from the email header. It can be any standard server like Gmail, yahoo, etc.

b) ***Top Certification Authority (TCA):*** It is a standard root level certification authority. The purpose of this component is to certify all the communication between client and SEMS. All the certificates issued by this authority are based on X.509 standard.

c) ***Infrastructure Email Server (IEMS):*** It is deployed at top level and every proxy server has to register with it by providing its domain and IP address. In our protocol its main purpose is to keep records of all proxy servers registered with it and to make available IP (address) of any proxy server belonging to a specific domain on demand. Secondly, it acts like a source to delegate trust to proxy servers from IEMS.

3.2 Summary

This chapter has described the designed proxy based layered architecture of secure and privacy enhanced email system. It also includes the purpose of each layer i.e. Client layer (layer-1), Domain layer (layer-2) and Infrastructure layer (layer-3). It has also provided the detail of components of every layer along with their major tasks and responsibilities. In this section, figure 5 of architecture is demonstrating the architecture and the interaction between the layers in brief but detail of this procedure is given in chapter 4.

4. Analysis and Implementation of the System

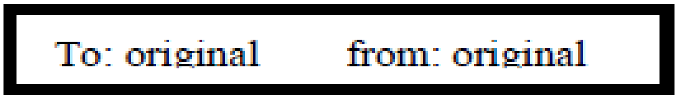
4.1 Protocol

Two things which are already assumed before execution of this protocol are as follows:

- Every client is registered to SEMS with its anonymous id through the respective proxy server.
- Every proxy server is registered to IEMS with its domain and IP address.

Step 1: First client, at sender side, sends original ids of “To” and “From” fields to proxy server A (PSA), positioned at sender side.

Client →PSA: { Original “to” and ”from”}



To: original from: original

Figure 6-Message Format

Step 2: Proxy Server performs a check before mapping original id of sender to anonymous id, whether it is a local user of domain or a guest user. This is because our system implements an authorization policy that determines who can send an email from a specific domain which will further stops spamming; Users from other domain are categorized as guest user and they are not authorized to send an email from particular domain.

PSA: {Sender’s original id ∞ Anonymous id}

Step 3: To get the anonymous id of receiver, the PSA finds out the domain of receiver from the recipient’s original address and sends query to infrastructure email server (IEMS).

PSA → IEMS: {Domain of recipient}

Step 4: It is assumed that every proxy server has been registered itself to the IEMS with its domain and IP address. Therefore, in response to step 3, IEMS sends an IP address of that particular domain to PSA. Then the proxy server A (PSA) can easily interact with proxy server B (PSB) positioned at receiver side to learn the anonymous id of recipient.

IEMS→PSA: {IP address of PSB}

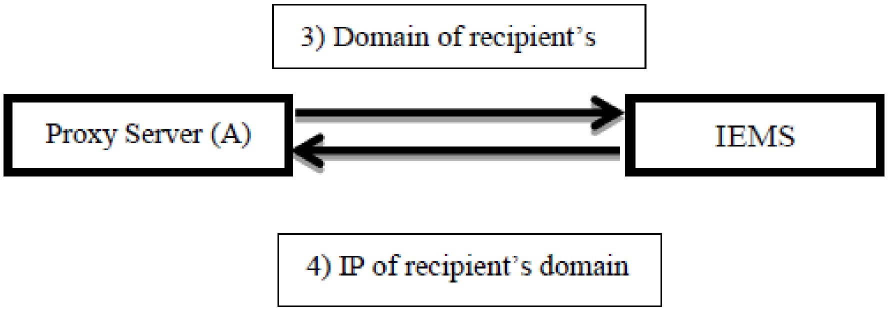


Figure 7-Communication between PSA and IEMS

Step 5: At this step, PSA directly sends request of anonymous id of recipient, providing its original id to PSB using IP sent by IEMS. In reply, PSB sends the required anonymous id which is stored against the original one.

PSA→PSB: {Query for recipients anonymous id, provided by its original one}

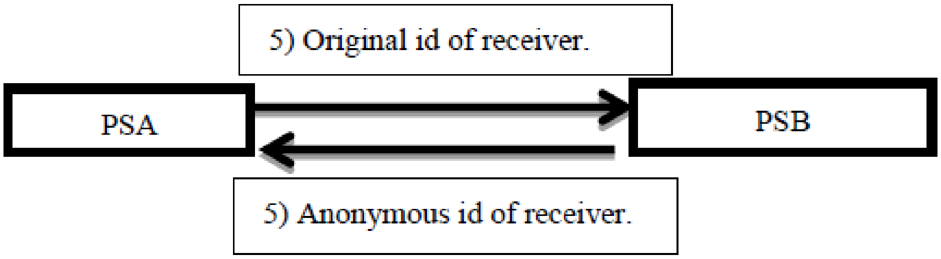


Figure 8-Communication between PSA and PSB

Step 6: After attaining the anonymous ids of both sender and receiver side, PSA sends anonymous ids (already registered with SEMS) and original ids protected with cryptographic function in the form of original protected header (OPH) to client (sender).

At this point, client sends an email to SEMS with the following format:

Client (Sender) →SEMS: { Email to SEMS with anonymous ids}

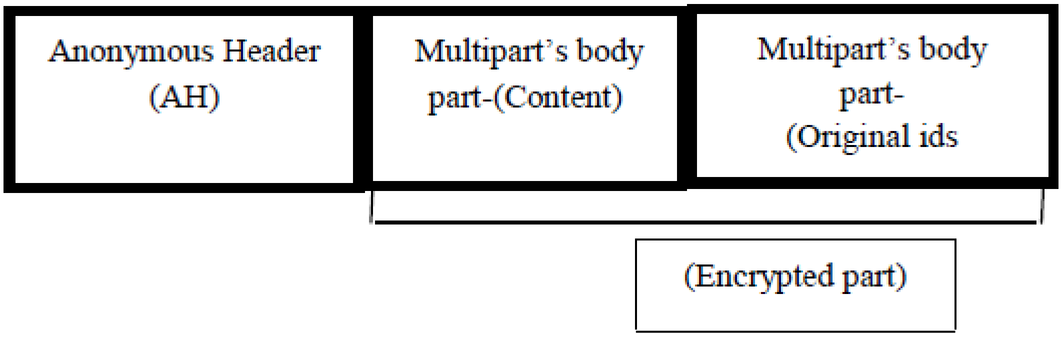

Anonymous Header (AH)
Multipart's body part-(Content)
Multipart's body part-(Original ids
(Encrypted part)

Figure 9-Final Message Format

a) ***Anonymous Header (AH):*** It includes anonymous ids of both sender and receiver registered with the standard email server SEMS to route the email properly. As it was assumed that every client is registered with SEMS with their anonymous identities so, this part is not encrypted and transmits in clear text which helps SEMS to actually route the email.

b) ***Multi part's Body part(Contents):*** It includes the encrypted mail message contents produced by using S/MIME.

c) ***Multi part's Body part (original ids Information):*** It includes encrypted original ids of both sender and receiver extracted from step-1 by using standard encryption techniques.

Step 7: After analyzing the header information, SEMS sends this email to receiver using the data in Anonymous Header (AH)

SEMS → client: {email with above stated format}

Step 8: When client at receiver side gets an email, it simply extracts the original source and destination from the body part of received email and places them in the position of "To" and "From" in the email viewed by recipient client.

Above explained steps are showing the detail of procedure that how email composes and then transmits to the standard email server. It is also explaining the technique we have used to protect user's personal information like original email identities of communicating parties. Similarly the goals defined in section 1.1 have also been achieved by implementing the system, following above steps. The major contribution is that the email sending and receiving mechanisms are preserving the existing standards and formats of email system. There is no change required in conventional format of email header travels over the internet. In next section, major interfaces of the implemented system are explained in detail.

4.2 Interfaces

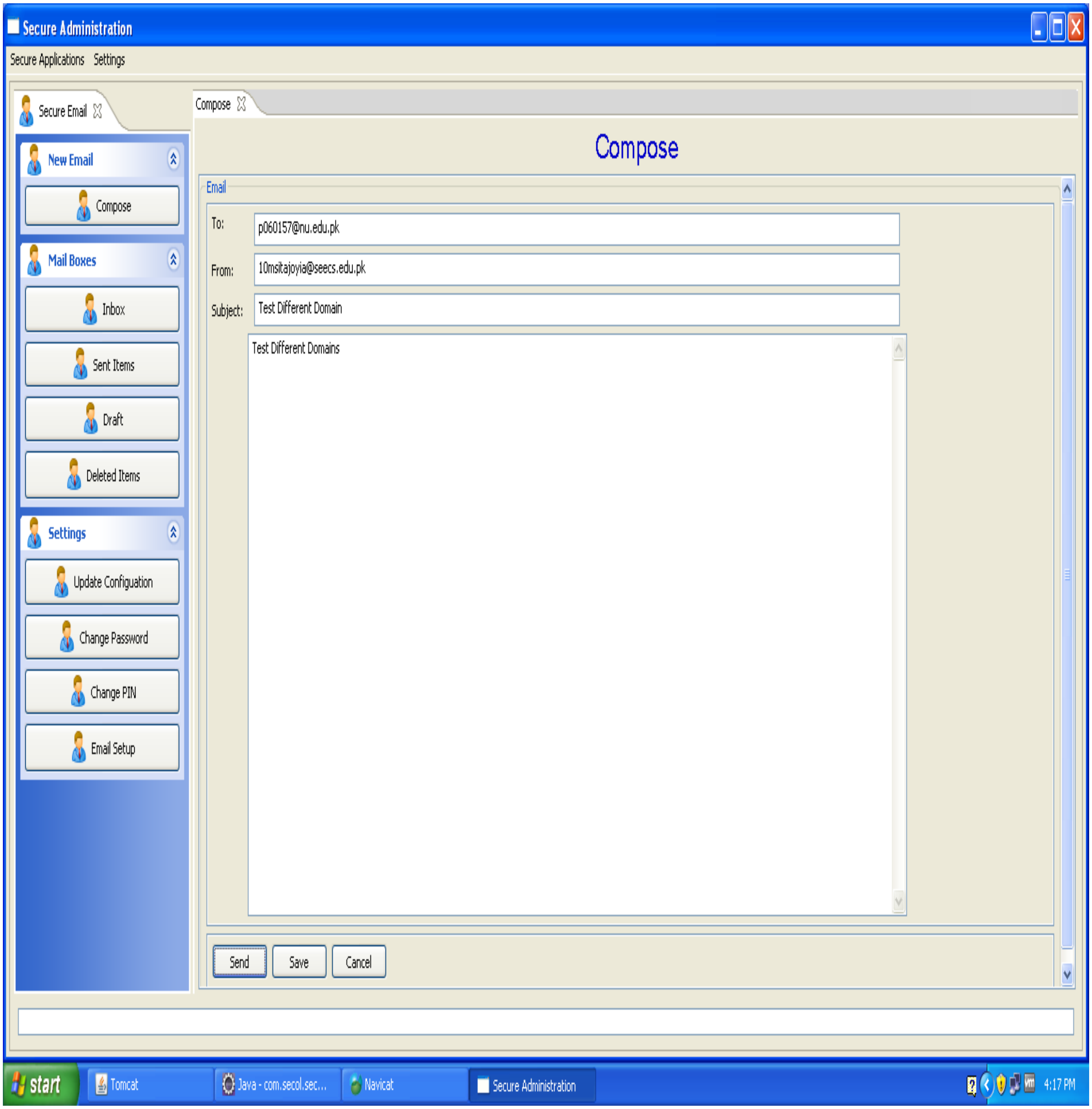
Following is the description of interfaces of Compose Email and Inbox.

4.2.1 Compose Email

Below in Figure 10, interface for compose mail is shown. There are four fields in the form, first is for the recipient's original email address field to which email will be sent. Second field is dedicated for the sender's original email id from which email will be generated to a specific recipient. Last field is for the content of an email message. After completing the entries of these fields when user press the send button, the backend interaction of client with proxy server and with other components as explained in protocol in section 4.1 will start and finally email has been sent to standard email server SEMS. Now SEMS routes the email over the internet by using information in email header to the destination at recipient's side.

Access control policy is an additional feature which is being implemented in this interface of compose mail. According to procedure explained in section 4.1, users are categorized as local and guest users. So, no guest user is allowed to send an email from a specific domain e.g. there are two users one is from the domain of NUST-SEECS with the domain name @seecs.edu.pk and the other none is from the domain of NUCES-FAST with the domain name @nu.edu.pk. For current implementation we are using he domain of NUST-SEECS, so email user with the domain of NUCES-FAST lays in the category of guest user and this user is not allowed to send an email from the current domain of NUST-SEECS.

So, by using this interface user can easily send an email to recipient belonging to any domain. In this way the goal of cross domain exchange of email letters is achieved. User sends an email with original email identities without taking care about the mapping of anonymous email ids. Now due to the secure and privacy enhanced email system, no intruder is able to extract original email identities or to create any communication link among specific email users. Furthermore, spamming is also avoidable because of hiding the original email identities using anonymous email ids.



To: p060157@nu.edu.pk

From: 10msitajoyia@seecs.edu.pk

Subject: Test Different Domain

Test Different Domains

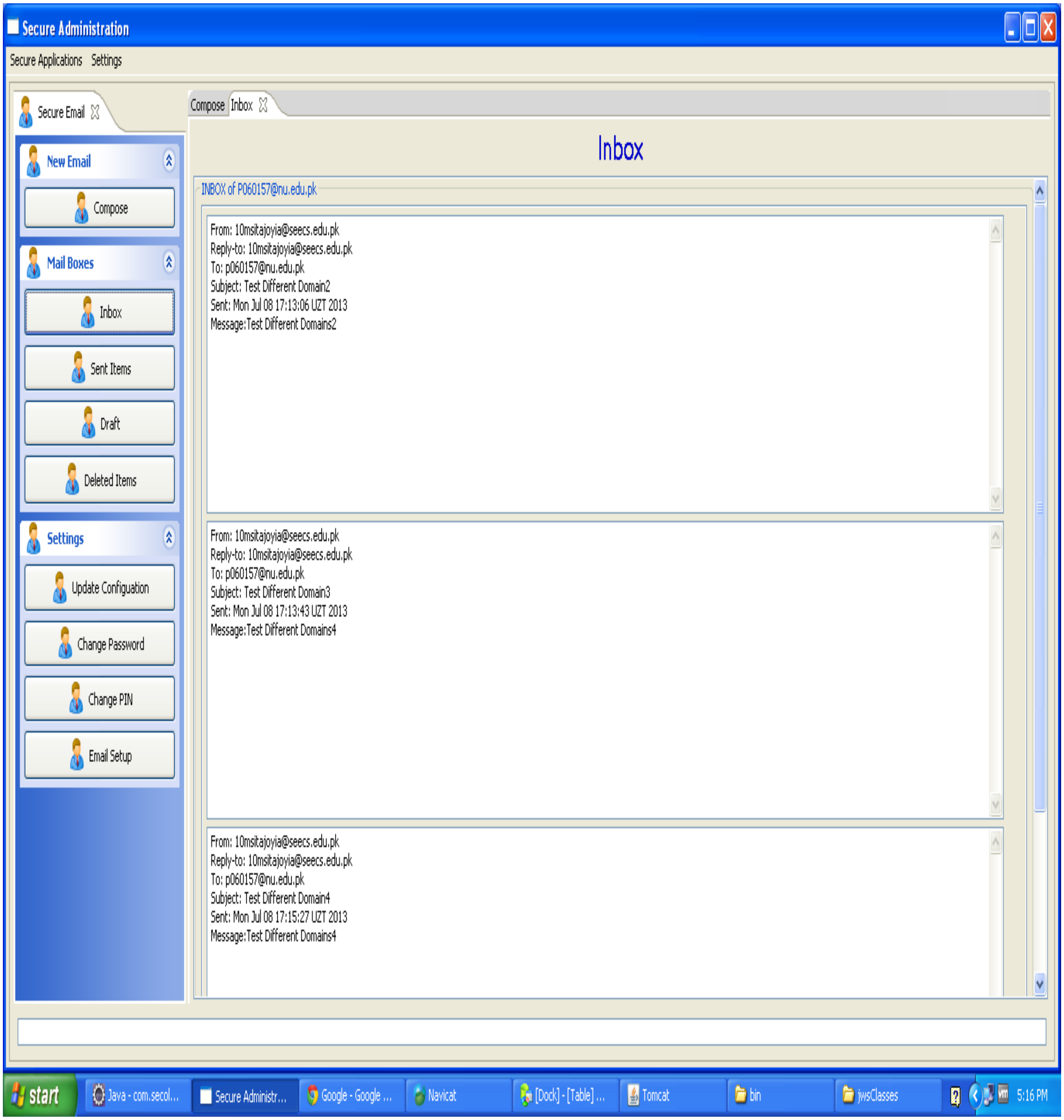
Figure 10-Compose Email Form

4.2.2 **Inbox**

Interface for the Inbox of an email system is shown in figure-11. This figure is showing the emails of a specific user and each message has a specific format. It includes *From* address; the user from whom email is received, *Reply-to*; same address of from, *To*; the address of recipient, *Subject*; the subject of an email, *Date;* the date on which email is received and finally the *Content*; message contained in an email message. As it was discussed in protocol that email routes over the internet with anonymous email identities, but we can see in email messages that email received at recipient side is showing original identities of sender and receiver. So, this thing is proving the fact that client at both sides of the system is ignorant of the mapping of original email ids to the anonymous one.

Standard email system SEMS directly sends an email to the recipient and the proxy server at recipient side is not involved in this process. System implemented at receiver side automatically extracts original email identities of *To* and *From* addresses and places them at their right places in the email viewed by client at recipient side.

So, in this way we have achieved our major objective of sending an email using existing email sending formats and standards without revealing the private information of email user transmitted in clear text over the internet. The information of original email identities of sender and receiver in email header is a source of privacy leakage of a user so by implementing this protocol, this threat has overcame.



Inbox

INBOX of P060157@nu.edu.pk

From: 10msitajoyia@seecs.edu.pk
Reply-to: 10msitajoyia@seecs.edu.pk
To: p060157@nu.edu.pk
Subject: Test Different Domain2
Sent: Mon Jul 08 17:13:06 UZT 2013
Message:Test Different Domains2

From: 10msitajoyia@seecs.edu.pk
Reply-to: 10msitajoyia@seecs.edu.pk
To: p060157@nu.edu.pk
Subject: Test Different Domain3
Sent: Mon Jul 08 17:13:43 UZT 2013
Message:Test Different Domains4

From: 10msitajoyia@seecs.edu.pk
Reply-to: 10msitajoyia@seecs.edu.pk
To: p060157@nu.edu.pk
Subject: Test Different Domain4
Sent: Mon Jul 08 17:15:27 UZT 2013
Message:Test Different Domains4

Figure 11-Inbox Form

4.3 Summary

This chapter basically represents the implementation of the system. The stepwise detail of protocol is explained in first part of the above section. How different components of the system interacts with each other and mechanism by which original email identities maps into anonymous one are the major topics of section 4.1. Every step of protocol is explained in detail and each communication between modules of the system is elaborated with pictorial representation. Section 4.2 is further divided into two sections, first is showing the figure of interface to compose the email message. With other features of the system, this section also describes the access control policy implemented at sender side in depth. The second part of section 4.2 is presenting the interface for the inbox of an implemented email system along with its major features.

5. Verification of Protocol

Previously discussed section 4 has explained implemented protocol in detail. In order to verify and analyze the implemented protocol, we have used an automated security verification tool called Scyther [14] [11]. Scyther has verified the claims of our protocol that the original email ids of sender and receiver both are secured along with the persistence of aliveness and secrecy of the system. This protocol also ensures that the intruder’s attack can never be successful as it was assumed that network is completely or partially under the control of intruder. Figure 12 in section 5.3 is showing the attributes by which we have verified our system using Scyther. The script of Scyther is provided in section 5.1.

First of all some variables have been declared to represent the whole process of protocol. Using these variables, registration between proxy server and client has been done. After that communication between client and proxy server at sender side initiates to interrogate the anonymous identities of sender and receiver. Then the claims of *protection of original email identities of sender and receiver, niagree, nisynch, aliveness* and *weakagree* at sender side has been defined for which status of verified is shown in the output of script in figure-12. Similarly for all components, roles and claims have been defined one by one in the script and verified in figure-12.

The result of Scyther has proved the claims of implemented protocol. The first claim is the secrecy of the credentials i.e. the original email ids of both sender and receiver remains secure and unrevealed. It can be clearly seen from figure 12 that the attribute of secrecy of user’s credential is maintained at each level whether it is on client’s side or that of proxy servers’ side. Moreover, these credentials are also protected from intruder’s attacks.

The second claim which we have verified is the aliveness of the system; that in secure and privacy enhanced email system, there is always a reply or response from receiver at result of a message or event generated by sender. Results of Scyther are also showing that messages exchanged in communicating parties are not altered because they are digitally signed and time stamped. So in this way integrity and privacy of the system remains preserve and no intruder attack can be launched on the system.

5.1 Scyther Script

```
usertype EmailId,Domain,IP,SessionKey, Packet, TimeStamp;

const Fresh:Function;


protocolp1(ProxyServer1, Client1,ProxyServer2,Client2,ILServer, sems)

{

role Client1

    {

     fresh ni: Nonce; var nr: Nonce; fresh Ti: TimeStamp;

        fresh toOriginalEmailId:EmailId;

     fresh fromOriginalEmailId:EmailId;

        fresh email:Packet;

     hashfunction H;

        var skey:SessionKey;

        var toAnonymousEmailId:EmailId;

        var fromAnonymousEmailId:EmailId;

        var Tr: TimeStamp;


// Registration between client and proxy server

        send_1(Client1, ProxyServer1, (ni, Ti,{H(ni, Ti)}sk(Client1)));

        recv_2( ProxyServer1,Client1, (nr, Tr,{H(ni,nr, Tr)}sk( ProxyServer1)) );

        recv_3( ProxyServer1,Client1, {skey, Tr,{H(skey, Tr)}sk( ProxyServer1)}pk(Client1));

// process of getting anonymous email id
```

```
        send_4(Client1,ProxyServer1,({toOriginalEmailId,fromOriginalEmailId}skey));

        recv_9(ProxyServer1,Client1,toAnonymousEmailId,fromAnonymousEmailId);

            send_10(Client1,sems,{email,H(email)}skey;

// claims at client side

            claim_c1 (Client1, Secret, toOriginalEmailId);

            claim_c2 (Client1, Secret, fromOriginalEmailId);

            claim_c3(Client1, Niagree);

            claim_c4(Client1, Weakagree);

            claim_ c5 (Client1,Alive);

              claim_ c6 (Client1,Nisynch);

        }

role ProxyServer1

            {

        var skey1:SessionKey;

        var toOriginalEmailId:EmailId;

        var fromOriginalEmailId:EmailId;

            var packet:Packet;

        var ip:IP;

            var ni: Nonce;
```

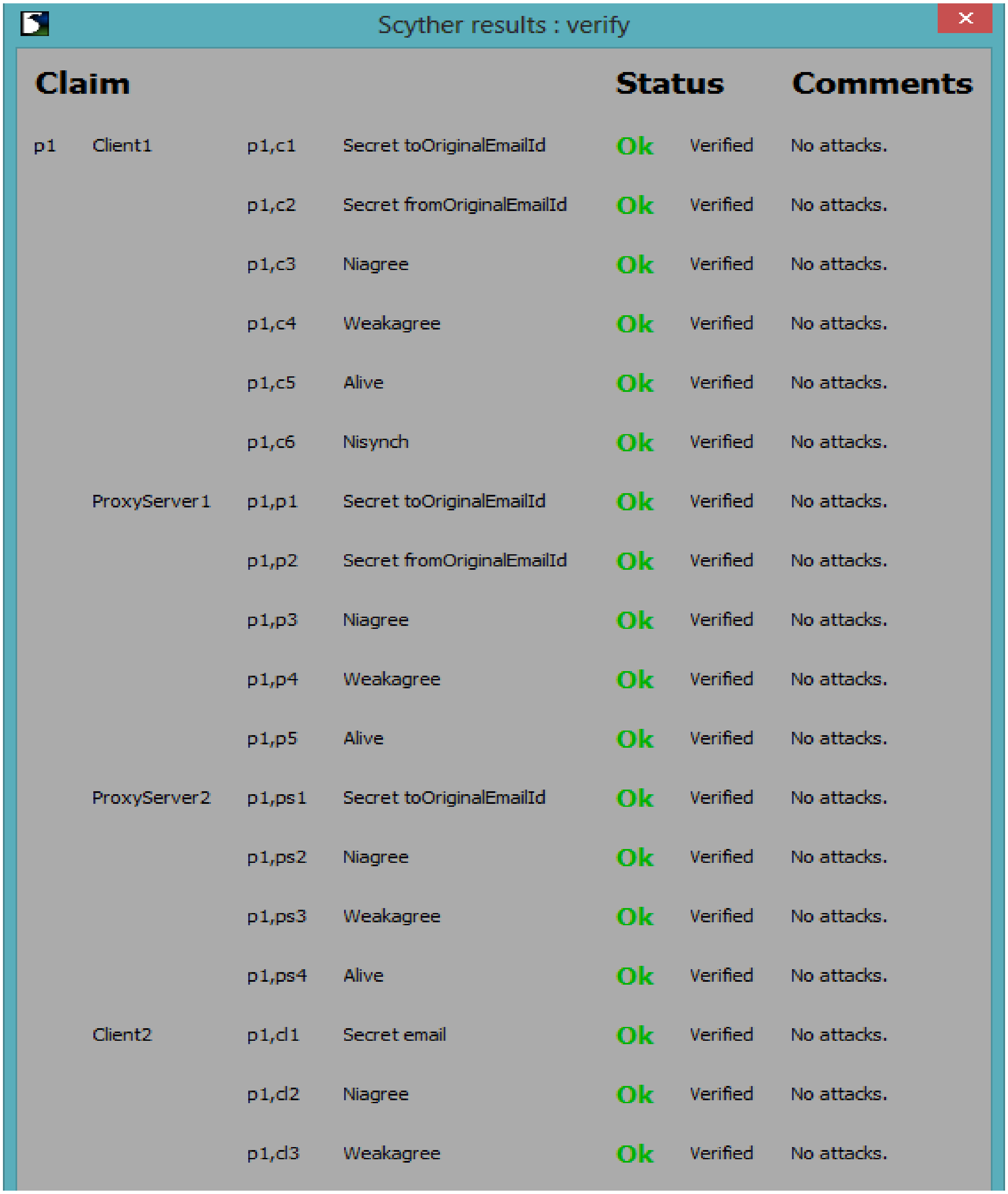
```
         var Ti: TimeStamp;
         var toAnonymousEmailId:EmailId;
         fresh fromAnonymousEmailId:EmailId;
      fresh domain:Domain;
         fresh nr: Nonce;
         fresh Tr:TimeStamp;
         fresh skey:SessionKey;
         hashfunction H;

// Registration between client and proxy server
         recv_1(Client1,ProxyServer1, (ni, Ti,{H(ni, Ti)}sk(Client1)));
         send_2(ProxyServer1,Client1, (nr, Tr,{H(ni,nr, Tr)}sk(ProxyServer1)) );
         send_3(ProxyServer1,Client1,{skey, Tr,{H(skey, Tr)}sk(ProxyServer1)}pk(Client1));
//process of getting anonymous email id
       recv_4(Client1,ProxyServer1,({toOriginalEmailId,fromOriginalEmailId}skey1));
          send_5(ProxyServer1,ILServer,domain);
          recv_6(ILServer, ProxyServer1,ip);
          send_7(ProxyServer1,ProxyServer2,({toOriginalEmailId,H(toOriginalEmailId)}skey));
          recv_8(ProxyServer2, ProxyServer1,toAnonymousEmailId);
//sending that anonymous id to client
```

```
    send_9(ProxyServer1,Client1,toAnonymousEmailId,fromAnonymousEmailId);

// claims at proxy server side

      claim_ p1 (ProxyServer1,Secret,toOriginalEmailId);

      claim_p2 (ProxyServer1,Secret,fromOriginalEmailId);

      claim_p3(ProxyServer1, Niagree);

      claim_p4(ProxyServer1, Weakagree);

      claim_ p5 (ProxyServer1,Alive);

      }


role ILServer

      {

      var domain:Domain;

      fresh ip:IP;

      recv_5(ProxyServer1,ILServer,domain);

      send_6(ILServer, ProxyServer1,ip);

      }


role ProxyServer2

      {

    var skey:SessionKey;

      var toOriginalEmailId:EmailId;
```

```
        fresh toAnonymousEmailId:EmailId;
        hashfunction H;
        recv_7(ProxyServer1, ProxyServer2,({toOriginalEmailId,H(toOriginalEmailId)}skey));
        send_8(ProxyServer2, ProxyServer1,toAnonymousEmailId);
// claims at proxy server side
        claim_ ps1 (ProxyServer2,Secret,toOriginalEmailId);
        claim_ps2(ProxyServer2, Niagree);
        claim_ps3(ProxyServer2, Weakagree);
        claim_ ps4(ProxyServer2,Alive);
         }

role sems
        {
        var email:Packet;
        var skey:SessionKey;
     hashfunction H;
        recv_10(Client1,sems,{email,H(email)}skey;
     send_11(sems,Client2,{email,H(email)}skey);
        }
```

```
role Client2
        {
         var email:Packet;
         var skey:SessionKey;
      hashfunction H;
//receving email
          recv_11(sems,Client2,{email,H(email)}skey);
//claims at receiving end
        claim_ cl1(Client2,Secret,email);
        claim_cl2(Client2, Niagree);
        claim_cl3(Client2, Weakagree);
        claim_ cl4(Client2,Alive);
        }
}
```

5.2 Scyther verification Output



Claim				Status		Comments
p1	Client1	p1,c1	Secret toOriginalEmailId	Ok	Verified	No attacks.
		p1,c2	Secret fromOriginalEmailId	Ok	Verified	No attacks.
		p1,c3	Niagree	Ok	Verified	No attacks.
		p1,c4	Weakagree	Ok	Verified	No attacks.
		p1,c5	Alive	Ok	Verified	No attacks.
		p1,c6	Nisynch	Ok	Verified	No attacks.
	ProxyServer1	p1,p1	Secret toOriginalEmailId	Ok	Verified	No attacks.
		p1,p2	Secret fromOriginalEmailId	Ok	Verified	No attacks.
		p1,p3	Niagree	Ok	Verified	No attacks.
		p1,p4	Weakagree	Ok	Verified	No attacks.
		p1,p5	Alive	Ok	Verified	No attacks.
	ProxyServer2	p1,ps1	Secret toOriginalEmailId	Ok	Verified	No attacks.
		p1,ps2	Niagree	Ok	Verified	No attacks.
		p1,ps3	Weakagree	Ok	Verified	No attacks.
		p1,ps4	Alive	Ok	Verified	No attacks.
	Client2	p1,cl1	Secret email	Ok	Verified	No attacks.
		p1,cl2	Niagree	Ok	Verified	No attacks.
		p1,cl3	Weakagree	Ok	Verified	No attacks.

Figure 12- Scyther verification of Protocol

5.3 Summary

This chapter is explaining the verification method used to validate the protocol. It also includes the list of attacks (Niagree, Weakagree, Nisynch and Aliveness) for which the system has been verified. First part of the above chapter is describing the tool Scyther; which is used for protocol verification purpose and the detail of scyther script i.e. how variables, roles and claims are defined for all components of the secure and privacy enhanced email system. The script implemented in environment of scyther is given in section 5.1. We can easily examine the flow of protocol and the secure communication among all key constituents of the email system from the scyther script. In last section, figure-12 is showing the output of scyther script with the status of each claim defined in script. Every claim is verified for each component of the system as well as no attacks are possible because all communication has made secure using different cryptographic techniques.

6. Conclusion and Future Directions

6.1 Conclusion:

The transmission of vital information of sender and receiver in clear text over the internet with email header is a clear threat to user's privacy. By analyzing the existing email systems, we came to know that no current email system is handling this privacy issue. So, there was a need to design and implement a system which should protect the user's personal information i.e. the original email identities of sender and receiver. The leakage of this personal information also leads the reception of spam emails which is a main cause of garbage in an inbox now days.

We designed and implemented a secure and privacy enhanced email system, which preserves the privacy of user's identity and avoids spamming by protecting the identity information of email user. It has also introduced the authorization policy of access control at sender side which allows authorized user only to send an email. Now no intruder can be able to trace out the original identities of participating entities. At recipient side, the receiver can easily recognize the sender by seeing the *From* field in an arrived email.

After designing and developing the protocol, we verified it using security verification tool Scyther. We found that the implemented protocol ensures the privacy of the user by protecting the information about user's original email identity. In chapter 5 all claims are shown which are verified for the system. The aliveness along with the secrecy of the system is verified and no attack is possible.

6.1 Future Directions:

As a result of shift of paradigm to the cloud computing environment, privacy and security has become the major concerns for users. Secure and privacy enhanced email system can be extended to protect other header fields too which are also the main source of privacy leakage for an end user. These fields includes date, subject, timestamp and majorly the IP address of the machine from which the email has originated. By extracting the IP address, any intruder can trace the sender of an email. So, to protect this field is also a significant matter for security engineers.

References

[1] Manali Oak. *Email Privacy Issues*. June 2012 http://www.buzzle.com/articles/email-privacy-issues.html

[2] Heinz Tschabitscher. *How to Clear Private Data, Empty Caches and Remove Cookies in Internet Explorer*. Retrieved on: June 2012 http://email.about.com/od/staysecureandprivate/qt/et_clear_ie.htm

[3] A. Kapadia, "A Case (Study) For Usability in Secure E-mail Communication" Security & Privacy, IEEE Volume 5, Issue 2, March-April 2007, pp. 80-84.

[4] Giannakis Antoniou, Udaya Parampalli, Lynn Batten, " Monitoring employees' emails without violating their privacy right" IEEE Eighth International Conference on Parallel and Distributed Computing, Applications and Technologies 2007.

[5] Julian Jang, Surya Nepal, John Zic "Trusted Email Protocol: Dealing with Privacy Concerns from Malicious Email Intermediaries" IEEE- CIT 2008

[6] Dijiang Huang and Vetri Arasan, "On Measuring Email-based Social Network Trust" IEEE Globecom 2010 proceedings

[7]M. Xie and H. Wang, "A collaboration-based autonomous reputation system for email services," in Preceedings of IEEE INFOCOM, 2010.

[8] S. Venkataraman, S. Sen, O. Spatscheck, P. Haffner, and D. Song, "Exploiting network structure for proactive spam mitigation," in Proc. USENIX Security 2007, 2007.

[9] Zhiyun Qian, Z. Morley Mao1, Yinglian Xie, Fang Yu, "Investigation of Triangular Spamming: a Stealthy and Efficient Spamming Technique" IEEE Symposium on Security and Privacy 2010

[10] Z.Qian, Z.M. Mao, Y.Xie, and F. Yu. "On network-level clusters for spam detection". In Proc. of NDSS, 2010.

[11] Casimier Joseph Franciscus Cremers .Scyther – Semantics and Verification of Security Protocols.

http://profs.info.uaic.ro/~cbirjoveanu/pagini/Sec_prot/References/Scyther%20%20Semantics%20and%20Verification%20of%20Security%20Protocols.pdf

[12] Pretty Good Privacy(PGP)

https://en.wikipedia.org/wiki/Pretty_Good_Privacy

[13] S/MIME (Secure/Multipurpose Internet Mail Extensions)

http://en.wikipedia.org/wiki/S/MIME

[14] Scyther: http://people.inf.ethz.chlcremersc/scyther/

[15] Deductive and Inductive research Approach:

http://www.drburney.net/INDUCTIVE%20&%20DEDUCTIVE%20RESEARCH%20APPROACH%2006032008.pdf

[16] S. Haber and W.S. Stornetta, “Howto Time-Stamp a Digital Document,” *J. Cryptology*, vol. 3, no. 2, 1991, pp. 99–111.

[17] C. Masone and S.W. Smith,“Towards Usefully Secure Email,”*IEEE Technology and Society Magazine*, to be published, Mar. 2007.

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

[19] S. Fleming, "Implicit Trust Can Lead to Data Loss," *Information Systems Security*, vol. 16, pp. 109 -113, 2007.

[20] “Email and webmail statistics,” Email Marketing Reports http://www.email-marketing-reports.com/metrics/email-statistics.htm, 2008.

[21] M. Xie and H. Wang, “A collaboration-based autonomous reputation system for email services,” in *Proceedings of IEEE INFOCOM*, 2010.

[22] R. Guimera, L. Danon, A. Diaz-Guilera, F. Giralt, and A. Arenas, “Selfsimilar community structure in a network of human interactions,” *Phys. Rev. E*, vol. 68, no. 6, p. 065103, 2003.

[23] Secure Multi-purpose Internet Mail Extensions (S/MIME) Working Group. www.imc.org/ietf-smime/.

[24] Oppliger, R. “Secure Messaging with PGP and S/MIME:” Artech House, Norwood, MA, 2001.