ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 1, January 2013

Electronic Mail Security

Dr. Ajit Singh, Meenakshi Gahlawat

***Abstract*— In today's electronic world the most popular service provided by the Internet is electronic mail. E-mail, the word itself implies sending messages from one user to another electronically via Internet. It is been growing as the main method for individuals and organizations to communicate. We all receive and send mail but the most important issue which arises is that whether the mail which we send are secure, been received by the indented receiver or not, and also not been misused. By looking at the widespread perception and popularity of email, the security of email messages has become an extremely important issue. This paper provides basic introduction about email and looking at its widespread popularity, it further emphasis on the security of the email messages which are been send. For this purpose, the paper describes about the two most important email security protocols PGP and S/MIME, their working, type of algorithms supported by them, various types of management functions been supported by them along with various security features which they provide. The purpose of these protocols is to provide confidentiality and authentication services for safe and intact sending and receiving of emails along with several others security measures which provides proof of delivery of messages along with several features such as access control, priority labels etc.**

***Index Terms*—PGP, S/MIME, Key rings, Web of Trust.**

I. INTRODUCTION

Electronic mail commonly known as e-mail is perhaps the most widely used application on the internet. Using email, an Internet user can send a message (and these days pictures, videos, sound, etc) to other Internet user(s). Modern email operates across the Internet or other computer networks. Some early email systems required that the author and the recipient both be online at the same time, in common with instant messaging. Today's email systems are based on a store-and-forward model. Email servers accept, forward, deliver and store messages. Neither the users nor their computers are required to be online simultaneously; they need to connect only briefly, typically to an email server, for as long as it takes to send or receive messages. [1], [4]

Historically, the term *electronic mail* was used generically for any electronic document transmission. For example, several writers in the early 1970s used the term to describe fax document transmission. As a result, it is difficult to find the first citation for the use of the term with the more specific meaning it has today. [4]

An Internet email message consists of three components, the message *envelope*, the message *header*, and the message *body*. The message header contains control information, including, minimally, an originator's email address and one or more recipient addresses. Usually descriptive information is also added, such as a subject header field and a message submission date/time stamp. Originally a text-only (7-bit ASCII and others) communications medium, email was extended to carry multi-media content attachments, a process standardized in RFC 2045 through 2049. Collectively, these RFCs have come to be called Multipurpose Internet Mail Extensions (MIME).Electronic mail predates the inception of the Internet, and was in fact a crucial tool in creating it, but the history of modern, global Internet email services reaches back to the early ARPANET. Standards for encoding email messages were proposed as early as 1973 (RFC 561). Conversion from ARPANET to the Internet in the early 1980s produced the core of the current services. An email sent in the early 1970s looks quite similar to a basic text message sent on the Internet today. Network-based email was initially exchanged on the ARPANET in extensions to the File Transfer Protocol (FTP), but is now carried by the Simple Mail Transfer Protocol (SMTP), first published as Internet standard 10 (RFC 821) in 1982. In the process of transporting email messages between systems, SMTP communicates delivery parameters using a message *envelope* separate from the message (header and body) itself. [4]

We generally see that in virtually all distributed environments email is most heavily used network-based application. It is also the only distributed application that is widely used across all architectures and vendor platforms. Users expect to be able to, and do, send mail to others who are connected directly or indirectly to the Internet, regardless of host operating system or communications suite. [2]

With the explosively growing reliance on electronic mail for every conceivable purpose, there grows a demand for authentication and confidentiality services and consequently, the security of email messages has become an extremely important issue. So the need for email security arises. [4]

II. EMAIL SECURITY PROTOCOLS

To provide security to email the two main security protocols needed are PGP and S/MIME.

A. PGP

PGP stands for Pretty Good Privacy protocol. It is been created by Phil Zimmerman. The most crucial aspects of PGP are that it is been used to support the basic cryptographic requirements, it is quite simple to use and is completely free,

Manuscript received Jan 18, 2013.
Dr. Ajit Singh, Department of Computer science and Engineering, School of Engineering and sciences, BPSMV, Khanpur Kalan, Sonepat, India.
Meenakshi Gahlawat, Department of Computer science and Engineering, School of Engineering and sciences, BPSMV, Khanpur Kalan, Sonepat, India .

including its source code and documentation. The security features offered by PGP includes: [1]

- Encryption
- Non-repudiation
- Message integrity

1) PGP Working:

In PGP, the identifiers of the algorithm used in message, along with the value of keys are been included by the sender. For an email message to be sent, PGP generally provides four security options as follows: [1]

- Signature only (Steps 1 and 2)
- Signature and Base-64 encoding (steps1,2 and 5)
- Signature, Encryption, Enveloping and Base-64 encoding (Steps 1 to 5)

We will now discuss about all these five steps.

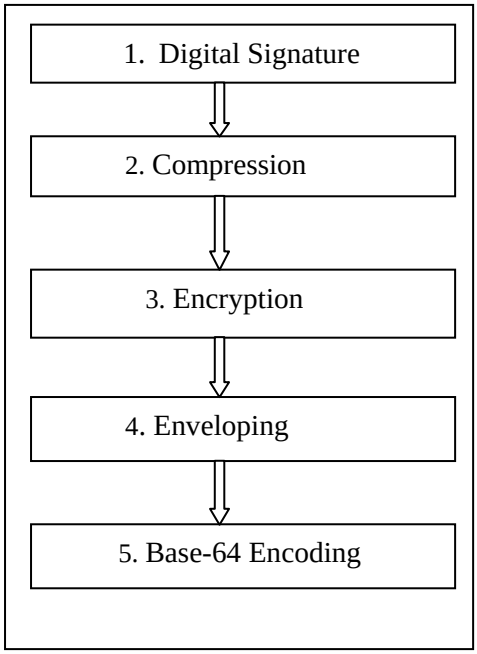

Fig. 1 "PGP Operations"

Step 1: Digital Signature: In this step message digest of the email message is been created using SHA-1 algorithm. The resulting message digest is then encrypted with the sender's private key to produce the sender's digital signature. [1]

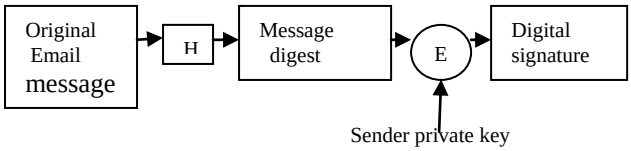

Fig. 2 "Digital Signature generation"

Step 2: Compression: In this step, the input message as well as the digital signature of the sender are both been compressed together in order to reduce the size of the final message which is to be transmitted. For this purpose ZIP program is used. ZIP is based on the Lempel-Ziv algorithm.

The Lempel-Ziv algorithm looks for repeated words or strings and stores them in variables. It then replaces the actual occurrence of the repeated word or string with a pointer to the corresponding variable. Since a pointer requires only a few bits of memory as compared to the original string, this method results in the data being compressed.

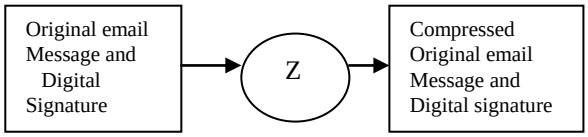

Fig. 3 "Message compression using ZIP technique"

Step3: Encryption: Here, the compressed output of step 2 i.e. the compressed from of the original email and the digital signature together are encrypted with the help of the symmetric key. For this, IDEA algorithm in CFB mode is used. [1]

Step 4: Digital Enveloping: In this, the symmetric key used for encryption in Step 3 is now been encrypted with receiver's public key. The output of Step 3 and Step 4 together form a digital envelope. [1]

Step 5: Base-64 encoding: This is the last step in PGP. The Base-64 encoding process transforms arbitrary binary input into printable character output. In this technique, the binary input is processed into blocks of 3 octects or 24 bits. These 24 bits are considered to be made up of 4 sets, each of 6 bits. Each such set of 6 bits is mapped into an 8-bit output character in this process. A mapping table is been used to map a 6 bit input block into an output 8-bit block.

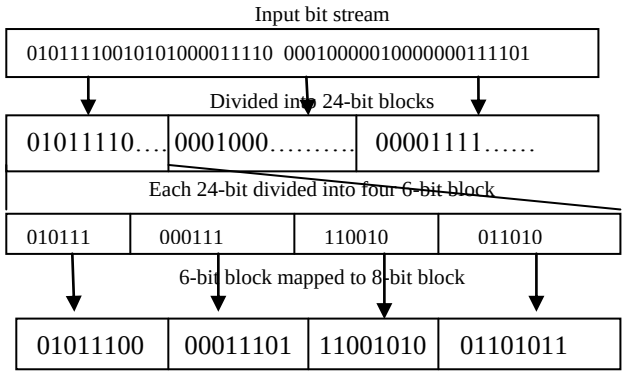

Fig. 4 "Base 64 encoding concept"

The above diagram is simply the rough representation to show Base-64 encoding, the values taken are been taken roughly. [1]

2) PGP Algorithms:

PGP is been used to support a number of algorithms. The most common are listed below: [1]

- Asymmetric key- RSA(Encryption and signing, Encryption only, Signing only)
- Message digest- MD5, SHA-1, RIPE-MD
- Encryption- IDEA, DES-3, AES

The use of these algorithms is explained as follows:

3) Key Rings:

When the message needs to be send to a single recipient, it can be send easily but in case of multiple recipients complexities are said to arise. For example, if Meenu wants

to communicate with 10 different people, she needs the public keys of all these 10 people. Hence, Meenu needs a key ring of 10 public keys. Additionally, PGP also specifies a ring of public-private keys. This is because Meenu may want to change her public-private key pair, or may want to use a different key-pair for different groups of user(e.g. one key pair when communicating with a business partner, another when communicating with friends, a third one when communicating with someone from family, etc.) Thus we see PGP user needs to have two sets of key rings: a ring of her own public-private key pairs and other one a ring of the public keys of the other users. [1]

The use of these keys can be explained by considering two possible situations:
1.Meenu needs to send a message to another user in the system. [1]

a. Meenu creates a message digest of the original message (using SHA-1) and then encrypts it using her own private key (via the RSA or DSA algorithm) from her own public-private key pairs.
b. She then creates a one-time symmetric key.
c. After that she uses the public key of the intended recipient by looking at the public keys of the PGP users which she holds in the system to encrypt the one-time symmetric key created earlier.RSA algorithm is used for this.
d. After doing so, she encrypts the original message with the one-time symmetric key (using IDEA or DES-3 algorithm).
e. She then encrypts the digital signature with the one-time symmetric key (using IDEA or DES-3 algorithm).
f. Lastly she sends the output of Steps (d) and (e) above to the receiver. What will the receiver supposed to do so is explained as follows: [1]

2. Now suppose that Meenu has received a message from one of the other users in the system. [1]

a. Meenu uses her private key to obtain the one-time symmetric key created by the sender.
b. She then uses the one-time symmetric key to decrypt the message.
c. Meenu computes a message digest of the original message (say MD1).
d. She now uses this one time symmetric key to obtain the original digital signature.
e. She then uses the sender's public key from the key ring that holds only the public keys of the PGP users in system to decrypt the digital signature and gets back the original message digest (say MD2).
f. Lastly she compares the message digests MDI and MD2. If they match, Meenu is sure about the message integrity and authentication of the message sender. [1]

4) PGP Certificates:

The public key of the user can only be trusted only when we have the user's digital certificate.PGP can use the certificates which are been issued by a CA, or it can also use its own certificate system. As we know that in X.509, there is a central authority (CA) that issues digital certificates, but in PGP there is no CA to issue certificates. In PGP anyone can sign a certificate belonging to anyone else in the ring. For example, Tia can sign certificate for Ria, Jia, Harsh and so on. We see that there is no hierarchy of trust or a tree like structure. This creates a situation where a user can have certificates issued by different users. For example, Jia can have a certificate signed by Tia and other one signed by Anita. And after all this if Harsh wants to verify Jia's certificate, he has two choices: Jia -> Tia, and Jia -> Anita. Harsh may have full trust on Tia, but not on Anita! So, there can be many choices in the line of trust from a fully or partially trusted authority to a certificate.
The equivalent of CA (i.e. a user who issues certificates) in PGP is known to be introducer. This whole process can be better understood with the help of three ideas given below: [1]

- Introducer trust
- Certificate trust
- Key legitimacy

Introducer Trust: It is been used to specify what level of trust the introducer wants to allocate to other user in the system. PGP provides multiple levels of trust. The number of levels depend on the decision of implementing PGP in a certain way. However for simplicity we implement three levels of trust to an introducer. These levels are called as none, partial and complete. For example, Tia says that she fully trust Jia, whereas Anita says she only partially trusts Jia. Jia, in turn, says that she does not trust Harsh. Harsh suggests that he partially trusts Anita in turn and so on. [1], [3]

Certificate Trust: In this case, when user A receives a certificate of another user B issued by a third user C, depending on the level of trust that A has in C, A assigns a certificate trust level to that certificate while storing it. It is normally the same as the introducer trust level that issued the certificate. This can be explained as: imagine there is a set of users in the system. Assume that Mahesh fully trust Naresh, partially trusts Ravi and Anmol and has no trust in Amit.

1. Naresh issues two certificates: one to Amrita (with public key K1) and another to Pallavi (with public key K2). Mahesh stores the public keys and certificates of Amrita and Pallavi in his ring of public keys with certificate trust level equal to full. [1]
2. Ravi issues a certificate to Uday (with public key K3). Mahesh stores the public key and certificate of Uday in his ring of public keys with certificate trust level equal to partial. [1]
3. Anmol issues two certificates: one to Uday (with public key K3), and another to Parag (with public key K4). Mahesh stores the public keys and certificates of Uday and Parag in his ring of public keys with certificate trust level equal to partial. Note that Mahesh now has two certificates for Uday, one issued by Ravi, and other issued by Anmol, both with partial level of certificate trust. [1]
4. Amit issues a certificate to Pramod (with public key K4). Mahesh stores the public key and certificate of Pramod in his ring of public keys with certificate trust level equal to none. Mahesh can also discard this certificate. [1]

Key Legitimacy: The purpose of using introducer and certificate trusts is to determine the legitimacy of a public key. Mahesh needs to know how legitimate are the public keys of Amrita, Pallavi, Uday, Parag, Pramod and so on. PGP defines a very clear procedure for determining key legitimacy. The level of the key legitimacy for a user is the weighted trust level of the user. For example, suppose we assign the following weights to certificate trust levels:
A weight of 0 to a nontrusted certificate.
A weight of ½ to a certificate with partial trust.
A weight of 1 to a certificate with full trust.
Then to trust an entity, Mahesh needs one fully trusted certificate or two partially trusted certificates. Thus, Mahesh can fully trust Amrita and Pallavi based on the certificates they had received from Naresh. Mahesh can also trust Uday, based on the two partially trusted certificates that Uday had received from Anmol and Ravi. Note that the legitimacy of a public key belonging to an entity does not have anything to do with the trust level of that person. Although, Naresh may be trusting Amit. Hence, Naresh can encrypt a message with the public key derived from Amit's certificate and can send the encrypted message to Amit. However, Mahesh will continue to reject certificates issued by Amit, since he does not trust Amit. [1], [3]

5) Web of Trust:

From the above discussion there arises a problem when nobody sends a certificate for a fully or partially trusted entity. For example, how can the legitimacy of Naresh's public key be determined if no one has sent a certificate for Naresh? In PGP, the key legitimacy of a trusted or partially trusted entity can be also determined by other methods.

1. Mahesh can physically obtain Naresh's public key. For example, Mahesh and Naresh can meet personally and exchange a public key written on a piece of paper or to a disk.
2. If Naresh's voice is recognizable to Mahesh, Mahesh can call him and obtain his public key on the phone.
3. A better solution proposed by PGP is for Naresh to send his public key to Mahesh by e-mail. Both Mahesh and Naresh make a 16-byte MD5 (or 20-byte SHA-1) digest from the key. The digest is normally displayed as eight groups of four digits (or 10 groups of four digits) in hexadecimal and is called a fingerprint. Mahesh can then call Naresh and verify the fingerprints on the phone. If the key is altered or changed during the email transmission, the two fingerprints do not match. To make it even more easy, PGP has created a list of words, each representing a four-digit combination. When Mahesh calls Naresh, Naresh can pronounce the eight words (or 10 words) for Mahesh. The words are carefully chosen by PGP to avoid those similar in pronunciation; for example, if sword is in the list, then word is not present.
4. In PGP, nothing prevents Mahesh from getting Naresh's public key from a CA in a separate procedure. He can then insert the public key in the public-key ring.

Regardless of the mechanism, eventually this process of obtaining keys of other users and sending our own to others creates a **web of trust** between groups of people. This keeps the public key ring getting bigger and bigger, and helps secure the communication. [1], [3]

6) Key revocation:

It may become necessary for an entity to revoke his or her public key from the ring. This may happen if the owner of the key feels that the key is lost or just too old to be safe. To revoke a key, the owner can send a revocation certificate signed by herself. The revocation certificate must be signed by the old key and disseminated to all the people in the ring who use that public key. [3]

B. Secure Multipurpose Internet Mail Extensions (S/MIME)

S/MIME is a security enhancement to the MIME Internet email format standard based on technology from RSA Data Security. Although both PGP and S/MIME are on an IETF standards track it appears like that S/MIME will emerge as the industry standard for commercial and organizational use, while PGP will remain the choice for personal e-mail security for many users. S/MIME is defined in a number of documents, most importantly RFCs 3369, 3370, 3850, 3851. [2]
Before beginning with S/MIME, we need to have some knowledge of the underlying e-mail format that uses it named MIME. [2]

When the standards for Internet mail were created in 1982, they provided for only ASCII characters and a 1000-character line limit, and they had a message length restriction. As time passed and technology such as multimedia came of age, a new electronic mail standard was needed to handle the large number of new data types. The MIME standard was adopted 10 years later in 1992. [5]

Using the new MIME standard, a mail message can contain the following: [5]

Images
Audio
Video
Binary data
Multiple fonts
Additional character sets
Multiple objects in the same message
Unlimited line and message length

To handle these new data types, MIME defines the following header fields: [5]

- **MIME-Version**: Must have the parameter value 1.0. This field indicates that the message conforms to RFCs 2045 and 2046.

- **Content-Type**: Describes the data contained in the body of the message. The details provided are sufficient so that the receiver email system can deal with the received email message in an appropriate manner.

- **Content-Transfer-Encoding**: Indicates the type of transformation that has been used to represent the body of the message in a way that is acceptable for mail transport.
- **Content-ID**: Used to identify MIME entities uniquely in multiple contexts.
- **Content-Description**: Used when the body is not readable (e.g. video).

1) Content Types

The current MIME standard supports a number of top-level MIME types. Some of these types, such as multipart and message, apply only to electronic mail and might not have any Navigator plug-in applications. The current top level types are covered in the following sections. [5]

- **Text:** The text type is a human-readable format used to represent text in a number of different character sets. Examples of text subtypes are plain, richtext, and html. [5]
- **Multipart**: The multipart type allows a mail message to combine several different MIME types within a single message. This format has very little application for plug-ins. [5]
- **Application**: The application type is a catch-all for binary data. If your plug-in uses binary data that is not an image, audio, or video, this one's for you. Some current application subtypes are postscript, wordperfect5.1, mathmatica, and zip. [5]
- **Message:** The message type is used to encapsulate a mail message. Again, this is not of much use for plug-ins. [5]
- **Image**: The image type handles still image or picture data in many different formats. This type plays an important role in displaying graphics on a Web browser. Some examples are jpeg, gif, and tiff . [5]
- **Audio**: The audio type handles audio data, usually in PCM format. Most audio subtypes such as x-wav and x-aiff are still going through IANA approval. Approved audio subtypes are basic and 32kadpcm. [5]
- **Video:** The video type is used for transmitting video data. This video data can contain interleaved audio, depending on the video file format. Mpeg and quicktime are examples of video subtypes. [5]

2) S/MIME Functionality:

In terms of the general functionality, S/MIME is quite similar to PGP. Like PGP, S/MIME, provides for digital signatures and encryption of email messages. [1]

Its functions include:

- **Enveloped data**: It consists of encrypted content of any type and the encrypted content encryption keys for one or more recipients. [1]
- **Signed data**: A digital signature is formed by taking the message digest of the content to be signed and then encrypting that with the private key of the signer. The content and the digital signature are both Base-64 encoded. [1]
- **Clear-signed data**: It is similar to Signed data. However, only the digital signature is Base-64 encoded. [1]
- **Signed and enveloped data**: Signed-only and encrypted-only entities can be combined, so that the Enveloped data can be signed, or the Signed/Clear-signed data can be enveloped. [1]

3) Cryptographic Algorithms Used in S/MIME:

They are as follows:

- Digital Signature Standard (DSS) for digital signatures. [1]
- Diffie-Hellman for encrypting the symmetric session keys. [1]
- RSA for either digital signatures or for encrypting the symmetric session keys. [1]
- DES-3 for symmetric key encryption [1]

S/MIME defines two terms: must and should for describing the usage of the cryptographic algorithms. These two terms mean:

- **Must**: This word specifies that these cryptographic algorithms are an absolute requirement. The user systems of S/MIME have to necessarily support these algorithms. [1]
- **Should:** There could be reasons because of which algorithms in this category cannot be supported. However, as far as possible, these algorithms should be supported. [1]

4) S/MIME messages:

A MIME entity can be made secured in S/MIME with the use of signature, encryption or both. MIME entity means an entire message or a sub part of the whole message. The MIME entity is prepared as per the usual MIME rules. This is processed by S/MIME, along with the security-related data, such as identifiers of algorithms and digital certificates. The output of this process is called as a Public Key Cryptography Standard (PKCS) object. This PKCS object itself is been considered as a message content and is been wrapped inside MIME, with the addition of appropriate MIME headers. S/MIME processes the email message along with the other security-related data, such as the algorithms used and the digital certificates to produce a PKCS object. [1]

5) Key Management Functions:

S/MIME user performs three key management functions as given below:

- **Key generation**: The user with some administrative capabilities must be able to create Diffie-Hellman

and DSS key pairs and should be able to create RSA key pairs. [1]

- **Registration:** A user's public key must be registered with a CA to receive an X.509 digital certificate. [1]
- **Certificate storage and retrieval**: A user needs digital certificates of other users to decrypt incoming messages and validate signatures of incoming messages. These must be maintained by a local administrative entity. [1]

6) S/MIME Additional Security Features:

S/MIME provides three additional security features:

- **Signed receipts**: This message can be used as acknowledgement for the original message been sent. This act as the delivery proof of the message to the original sender. The recipient signs the entire message (including the original message sent by the sender, the signature of the sender and the acknowledgement) and creates S/MIME message out of it. [1]
- **Security labels**: A security label is been added to the message to identify its sensitivity (how confidential it is), priority (confidential, restricted, secret, etc), access control (who can access it). [1]
- **Secure mailing lists**: Whenever a sender sends a message to multiple users an S/MIME Mailing List Agent (MLA) is been created to take over the processing that is required per recipient. [1]

III. CONCLUSION

In this paper an overview of Email is been presented first. After it the two security protocols needed to provide email security namely: PGP and S/MIME are been discussed. In PGP we discussed about its working, various algorithms supported by it, the use of key rings and the PGP certificate and various types of trust related to it. After it we discussed about S/MIME, functionality provided by it, various algorithms used by it, key management functions related to it and lastly the security features which it supports.

.Acknowledgment

I would like to give my sincere gratitude to my guide Dr. Ajit Singh who guided me throughout, to complete this topic.

.

REFERENCES

[1] Atul Kahate (2009), *Cryptography and Network Security*, second edition, McGraw-Hill.
[2] William Stallings (Fourth Edition), Cryptography and Network Security.
[3] Behrouz A Forouzan (Fourth Edition), Data Communications and Networking.
[4] en.wikipedia.org/wiki/**Email.**
[5] podgoretsky.com/ftp/docs/internet/netscape%20plug-ins/ch3.htm.

Dr. Ajit Singh is presently working as Chairperson of School of Engineering & Sciences in BPSMV, Khanpur Kalan (Sonepat). He is also having the additional charge as a Director of University Computer Center (UGC). He posses qualifications of B.Tech, M.Tech, Ph.D. He is a member of BOG (Board of Governors) of Haryana State Counselling Society, Panchkula and also member of academic council in the University. He published approximate 20 papers in National/ International journals and conferences and holds a teaching experience of approximate 10 years. He holds the membership of Internal Quality Assurance cell, UG-BOS &PG-BOS and the NSS advisory committee. He is also an associate member of CSI & IETE. His research interests are in Network Security, Computer Architecture and Data Structure.

Ms. Meenakshi Gahlawat has completed her B.Tech degree in Computer Science from Maharishi Dayanand University, Rohtak in year 2011. She is pursuing M.Tech in Computer Science from BPSMV, Khanpur Kalan from June 2011. Her research interests are in Network Security and Computer Networks.