Electronic Mail Security

Session 7

LEARNING OBJECTIVES

On completion of this session you should:

- Understand the security issues associated with email security
- Be familiar with secure email standards
- Understand the operation of PGP
- Understand how cryptography techniques are applied for secure email communication
- Understand the operation of Secure MIME
- Be familiar with DKIM

Contents

Reading

Prescribed readings

Reading 1: Read W. Stallings, Network Security Essentials - Application and Standard, 4th edition, Prentice Hall, 2011, pp. 236-243.

Reading 2: Read W. Stallings, Network Security Essentials - Application and Standard, 4th edition, Prentice Hall, 2011, pp. 244-248.

Reading 3: Read W. Stallings, Network Security Essentials - Application and Standard, 4th edition, Prentice Hall, 2011, pp. 248-250.

Reading 4: Read the online material PGP Tutorial by B.J. Poole. This tutorial that shows how to download PGP software, generate keys, distribute public key, send and receive secure email.

Reading 5: To gain an understanding on MIME read the on-line material "How MIME Works" by Heinz Tschabitscher.

Reading 6: Read W. Stallings, Network Security Essentials - Application and Standard, 4th edition, Prentice Hall, 2011, pp. 255-265.

Reading 7**:** Read W. Stallings, Network Security Essentials - Application and Standard, 4th edition, Prentice Hall, 2011, pp. 271-277.

7.0 Introduction

Due to the rapid growth of the Internet, electronic mail is perhaps the most widely used network-based application. It has become an essential tool for business as well as home users. Almost all organizations use e-mail for internal official communication and also for communicating with external customers. Unfortunately, the increasing popularity of the Internet has also

caused growing abuse of e-mail system. From security point of view, this indicates a demand for authentication and confidentiality services. In this study guide, we discuss two widely used approaches for email security: PGP and Secure MIME (S/MIME).

7.1 Electronic Mail Security

The basis for email over the Internet is the SMTP (Simple Mail Transfer Protocol) specified in RFC-821, the text and ASCII message syntax specified in RFC-822, and the MIME (Multipurpose Internet Mail Extension) specified in RFC 2045-2049. Neither SMTP nor the message syntax supports security services. It is not difficult to modify a mail's message content or forge the sender/recipient address or routing information. Anyone who can intercept an e-mail message may launch an active attack and tamper it. A small change, such as in financial data or invoice, can have disastrous consequences. This illustrates the need for security services as outlined in the OSI security architecture for email system. Confidentiality is required to ensure that the content is not revealed to unauthorized parties. Data integrity, authentication and non-repudiation is necessary to ensure that the transmitted message has not been altered in transit. Two main schemes that have emerged during the past years for email security are PGP and S/MIME.

7.2 Pretty Good Privacy

PGP, developed by Phil Zimmerman, provides confidentiality and authentication for email as well as file storage applications. Its popularity is boosted due to its free-of-charge availability, though a commercial version is also available from Network Associates, Inc. PGP uses symmetric and asymmetric algorithms, and integrates those into an easy-to-use general purpose application that operates independent of operating system and processor. The services supported by PGP are:

- Authentication
- Confidentiality
- Compression
- E-mail compatibilty
- Segmentation

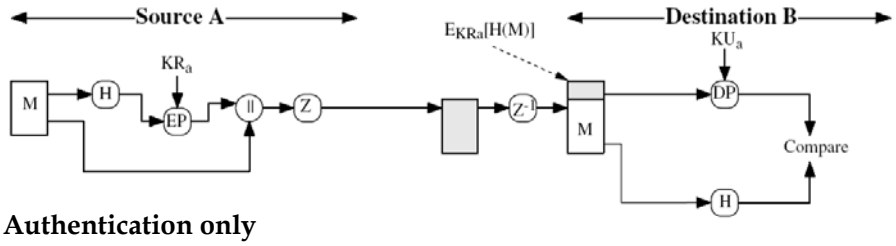
7.2.1 PGP Services

The operations of PGP with respect to above services are described below.

A. Authentication

Authentication service basically employs digital signature scheme. The operational sequence is as follows [1]:

1. The sender creates a message.
2. A hash code of the message is generated using SHA-1.
3. The sender signs the hash code by using his/her private key. The result is prepended to the message.
4. The receiver decrypts the hash code using sender's public key.
5. The receiver generates a new hash code from the message and compares with the decrypted hash code. If a match is found, it accepts the message as authentic.



Authentication only

In the above figure KR_a and KU_a are the private-public key pair of the sender, EP and DP are asymmetric key encryption and decryption repectively, Z denote a compression function. The combination of SHA and RSA (asymmetric encryption) is essentially a digital signature scheme.

B. Confidentiality

To achieve confidentiality PGP uses both symmetric and asymmetric encryptions. A 128-bit number is generated which acts as a session key. This is used to encrypt the original message. Remember symmetric key is computationally less expensive than asymmetric key. Asymmetric key (private and public key pair) is used to distribute this session key. Refer to the following figure, the operational sequence is as follows [1]:

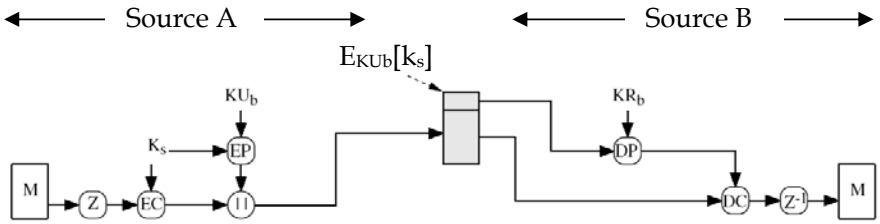
1. The sender generate a 128-bit session key.

2. The message is encrypted with the session key.

3. To distribute the session key, it is encrypted using the recipient's public key and prepended to the message.

4. The recipient recover the session by using his/her private key.

5. The recipient decrypt the message using the session key.



Confidentiality only

Ks is the session key, EP and DP are the symmetric key encryption and decryption, respectively.

C. Confidentiality and Authentication

A combination of the above is used to achieve both confidentiality and authentication at the same time. First a signature is generated and prepended to the message. (Message+signature) is zipped and then encrypted using the session key. As before, the session key is distributed using the recipient's public key and the zipped (message+signature) is encrypted by the session key. The receiver does the same things in reverse order to recover the message.

D. Compression

PGP compresses the message after applying the signature but before the encryption. This is preferable to applying compression after the encryption. The reasons are:

- The sender needs to store only the uncompressed message and the signature for future verification. Otherwise, compressed message needs to be store as well.

- There are different compression algorithms and different versions of the same algorithm. If compression is done after encryption, all

PGP implementation must use the same version of the same algorithm.

- It strengthens the security because cryptanalysis on compressed message is more difficult. Compression effectively gives an extra level of transformation of the message content.

E. E-Mail Compatibility

To ensure compatibility between different email systems, PGP uses radix-64 conversion to convert raw 8-bit binary stream to a stream of printable ASCII character. It also appends a CRC (Cyclic Redundancy Check) to detect transmission error. However, radix 64 expands the message by 33%, but it is compensated by the compression of the message.

F. Segmentation and Reassembly

Most email systems over the Internet are restricted to a maximum message length. To handle this, PGP breaks down a large message into smaller segments and reassembles those segments at the receiving end. Only the first segment carries the session key and the signature component. All other segments carry the email header only.

Reading 1:
Read W. Stallings, Network Security Essentials - Application and Standard, 4th edition, Prentice Hall, 2011, pp. 236-243.

7.2.2 PGP Message Format

PGP allows the use of multiple keys instead of a single key and introduces the concept of key rings. Think of key ring as a holder of multiple keys. The reason behind using multiple keys is that a user may wish to use different keys to interact with different people. Sometimes using different keys for different communication may give added security. This however raises one problem: how does the receiver know which key has been used? One solution to that is to transmit the public key with the message so that the corresponding private key can be used. This increases the overhead significantly. The better solution is to associate each private/public key pair with an identifier, called key ID, and attach that with the transmitted message. Each key ID must be unique for a particular user. PGP takes the least significant 64 bits of each key as the key ID.

A PGP message consists of the following three components:

- **Message:** contains data, filename and timestamp.
- **Signature:** contains signature timestamp, key ID of sender's public key, leading two octets of the message digest and 160-bit SHA-1 message digest encrypted with sender's private key. The message digest is derived over the signature timestamp concatenated with data portion of message component.
- **Session key:** encrypted session key and the key ID of recipient's public key used to encrypt the session key.

Reading 2:
Read W. Stallings, Network Security Essentials - Application and Standard, 4th edition, Prentice Hall, 2011, pp. 244-248.

7.2.3 PGP Message Generation and Reception

PGP message generation and reception basically use all the services and the message format that have been discussed so far. At the sending end, it mainly involves signing and encrypting the message. At the receiving end, it involves decrypting and authenticating the message. A simplified version, without compression or radix conversion, is shown in Fig. 7.5 and Fig. 7.6 of the textbook.

Reading 3:
Read W. Stallings, Network Security Essentials - Application and Standard, 4th edition, Prentice Hall, 2011, pp. 248-250.

Reading 4:
Read the online material PGP Tutorial by B.J. Poole. This tutorial that shows how to download PGP software, generate keys, distribute public key, send and receive secure email

7.3 Secure MIME

MIME stands for Multipurpose Internet Mail Extension and refers to an official Internet standard that specifies how messages must be formatted so that they can be exchanged between different email systems. MIME is a very flexible format and allows to include virtually every type of file or document in an email message. MIME extends the format of the Internet mail to allow non-ASCII textual messages, non-textual messages, multipart message bodies, and non-ASCII information in message headers. Typically e-mail messages consists of 7-bit ASCII text whereas multimedia applications like audio, video etc. generates data stream of arbitrary binary 8-bit pattern. This means, we need to convert these 8-bit patterns into printable characters that can be handled by various email systems and reconvert them into binary format at the receiving end. MIME does this conversion using base64 encoding and provides a general structure for the content type of an email message and allows extensions for new content types.

Reading 5:
To gain an understanding on MIME read the on-line material "How MIME Works" by Heinz Tschabitscher.

Secure MIME provides signing and encrypting functionality. In that respect it functions similar to PGP. It utilizes the cryptographic techniques covered in study guide 2 and 3. S/MIME specifies a number of hash and encryption algorithms that MUST be supported and recommends a number of algorithms that SHOULD be supported. For example, to calculate message digest, SHA-1; to create digital signature, DSS; and to encrypt the session key Diffie-Hellman MUST be supported. Table 7.6 of the textbook gives a comprehensive list. This requires the sending email system to determine if the receiving email system is capable of handing a given encryption algorithm. If receiver can handle weak encryption only, then sender has to decide whether sending by weak encryption is acceptable.

S/MIME provides the following functions [1]:

Enveloped data: contains encrypted data of any type, recipient information and encrypted session key. The steps for creating enveloped data are:

1. Generate a random session key for symmetric encryption.
2. Encrypt the session key with recipient's public key.

3. For each recipient, a RecipientInfo block is prepared which contains an identifier of recipient's certificate, an identifier of the encryption algorithm and the encrypted session key.

4. Encrypt the message with the session key. RecipientInfo block followed by the encrypted message content constitute envelopedData entity. This is then encoded into base64 (another name of radix64).

5. The recipient strips of the base64 encoding and uses own private key to retrieve the session key. Once the session key is known, the content can be decrypted.

Sign data: contains encrypted data and digital signature. A signed data can only be viewed by a recipient with S/MIME capability. The steps for creating signed data are as follows.

1. Create a message digest of the content using either SHA or MD5 hash algorithm.

2. Encrypt the message digest with the signer's private key.

3. A SignerInfo block is created that contains: a) signer's public-key certificate, b) identifier of message digest algorithm, c) identifier of the encryption algorithm, and d) encrypted message digest. SingerInfo and encrypted message content constitute SignedData entity. SignedData is then encoded into base64.

4. The receiver uses signer's public key to decrypt message digest and verify the signature. The receiver recomputes the message digest from the received message and compares with the decrypted message digest. A match verifies the signature.

Clear-signed data: same as signed data but only the digital signature is encoded using base64. The message is not transformed, so it is sent 'in the clear'. This allows any recipient with MIME (without S/MIME) capability to view the message content, but can't verify the signature. It has two parts. The first part contains a detached signature and is processed in the same manner as signedData. The second part has a MIME 'content type' of the application.

Signed and enveloped data: This allows various ordering of signing and encrypting.

For certificate processing S/MIME uses public-key certificates that conform to X.509-version 3. S/MIME managers and/or users must configure each

client with a list of trusted keys and with certificate revocation lists. The key management functions that need to be performed by a user client are as follows:

- Key generation: Diffie-Hellman, DSS and RSA key pairs.
- Registration: user's public key must be registered with a CA.
- Certificate storage and retrieval: access to local list of certificates for signature verification and message encryption purpose.

Reading 6:
Read W. Stallings, Network Security Essentials - Application and Standard, 4th edition, Prentice Hall, 2011, pp. 255-265.

7.4 DomainKeys Identified Mail (DKIM)

DomainKeys Indentified Mail (DKIM) is a specification for cryptographically signing e-mail messages, whereby a signing domain claims responsibility for a message in the mail stream.

Reading 7:
Read W. Stallings, Network Security Essentials - Application and Standard, 4th edition, Prentice Hall, 2011, pp. 271-277.

7.5 Conclusion

With the growth of the Internet email has become ubiquitous and is becoming increasingly important for business organizations as well as private use. In recent times PGP has gained considerable acceptance among private users. Even though commercial version of PGP is available it has not gained strong acceptance among organizations because of its inability to scale adequately for large deployments and has not been adopted by major software vendors. S/MIME is supported by a number of leading industry vendors including Microsoft, Netscape, Lotus, ConnectSoft, SecureWare, VeriSign, and Novell. This will probably be the industry

standard in future. DKIM has been adopted by a range of e-mail providers, including corporations, government agencies, gmail, yahoo and many Internet Service Providers (ISPs).

7.5 References

[1] W. Stallings, Network Security Essentials - Application and Standard, 4th edition, Prentice Hall, 2011.