So, in this case, the parameters are

K = PMK

A = the text string "Pairwise key expansion"

B = a sequence of bytes formed by concatenating the two MAC addresses and the two nonces

Len = 384 bits

Similarly, a nonce is generated by

```
Nonce = PRF (Random Number, "InitCounter", MAC || Time, 256)
```

where **Time** is a measure of the network time known to the nonce generator.

The group temporal key is generated by

```
GTK = PRF (GMK, "Group key expansion", MAC || Gnonce, 256)
```

Figure 7.11 illustrates the function PRF(*K, A, B, Len*). The parameter *K* serves as the key input to HMAC. The message input consists of four items concatenated together: the parameter *A*, a byte with value 0, the parameter *B*, and a counter *i*. The counter is initialized to 0. The HMAC algorithm is run once, producing a 160-bit hash value. If more bits are required, HMAC is run again with the same inputs, except that *i* is incremented each time until the necessary number of bits is generated. We can express the logic as

```
PRF (K, A, B, Len)
    R ← null string
    for i ← 0 to ((Len + 159)/160 – 1) do
    R ← R || HMAC-SHA-1 (K, A || 0 || B || i)
    Return Truncate-to-Len (R, Len)
```

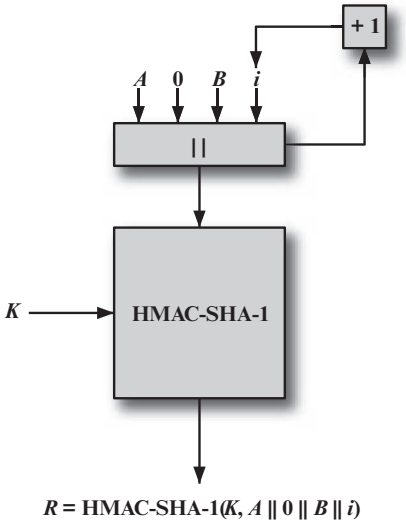

Figure 7.11 IEEE 802.11i Pseudorandom Function

7.5 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

Key Terms

4-way handshake	independent BSS (IBSS)	pseudorandom function
access point (AP)	logical link control (LLC)	Robust Security Network (RSN)
basic service set (BSS)	media access control (MAC)	Temporal Key Integrity Protocol (TKIP)
Counter Mode-CBC MAC Protocol (CCMP)	MAC protocol data unit (MPDU)	Wi-Fi
distribution system (DS)	MAC service data unit (MSDU)	Wi-Fi Protected Access (WPA)
extended service set (ESS)	message integrity code (MIC)	Wired Equivalent Privacy (WEP)
group keys	Michael	Wireless LAN (WLAN)
IEEE 802.1X	pairwise keys	
IEEE 802.11		
IEEE 802.11i		

Review Questions

7.1 What is the basic building block of an 802.11 WLAN?
7.2 List and briefly define threats to a wireless network.
7.3 List and briefly define IEEE 802.11 services.
7.4 List some security threats related to mobile devices.
7.5 How is the concept of an association related to that of mobility?
7.6 What security areas are addressed by IEEE 802.11i?
7.7 Briefly describe the five IEEE 802.11i phases of operation.
7.8 What is the difference between TKIP and CCMP?

Problems

7.1 In IEEE 802.11, open system authentication simply consists of two communications. An authentication is requested by the client, which contains the station ID (typically the MAC address). This is followed by an authentication response from the AP/router containing a success or failure message. An example of when a failure may occur is if the client's MAC address is explicitly excluded in the AP/router configuration.
- **a.** What are the benefits of this authentication scheme?
- **b.** What are the security vulnerabilities of this authentication scheme?

7.2 Prior to the introduction of IEEE 802.11i, the security scheme for IEEE 802.11 was Wired Equivalent Privacy (WEP). WEP assumed all devices in the network share a secret key. The purpose of the authentication scenario is for the STA to prove that it possesses the secret key. Authentication proceeds as shown in Figure 7.12. The STA sends a message to the AP requesting authentication. The AP issues a challenge, which is a sequence of 128 random bytes sent as plaintext. The STA encrypts the challenge with the shared key and returns it to the AP. The AP decrypts the incoming value and compares it to the challenge that it sent. If there is a match, the AP confirms that authentication has succeeded.
- **a.** What are the benefits of this authentication scheme?
- **b.** This authentication scheme is incomplete. What is missing and why is this important? *Hint:* The addition of one or two messages would fix the problem.
- **c.** What is a cryptographic weakness of this scheme?

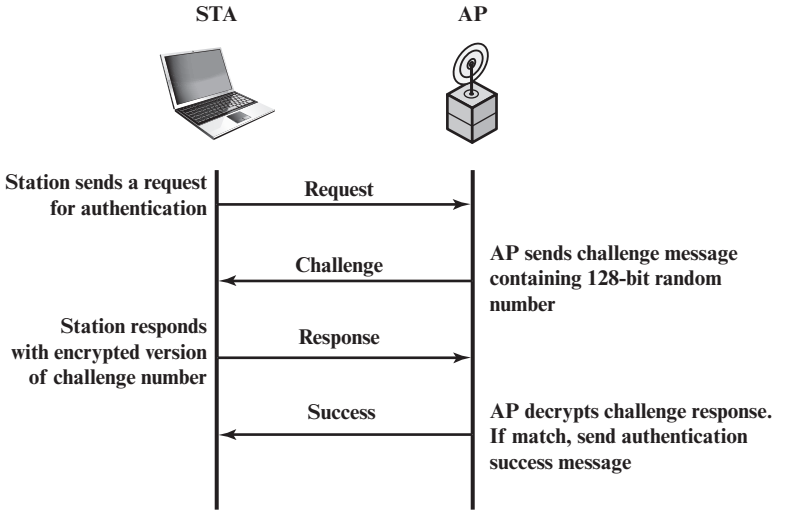

Figure 7.12 WEP Authentication; refer to Problem 7.2

7.3 For WEP, data integrity and data confidentiality are achieved using the RC4 stream encryption algorithm. The transmitter of an MPDU performs the following steps, referred to as encapsulation:

1. The transmitter selects an initial vector (IV) value.
2. The IV value is concatenated with the WEP key shared by transmitter and receiver to form the seed, or key input, to RC4.
3. A 32-bit cyclic redundancy check (CRC) is computed over all the bits of the MAC data field and appended to the data field. The CRC is a common error-detection code used in data link control protocols. In this case, the CRC serves as a integrity check value (ICV).
4. The result of step 3 is encrypted using RC4 to form the ciphertext block.
5. The plaintext IV is prepended to the ciphertext block to form the encapsulated MPDU for transmission.
   a. Draw a block diagram that illustrates the encapsulation process.
   b. Describe the steps at the receiver end to recover the plaintext and perform the integrity check.
   c. Draw a block diagram that illustrates part b.

7.4 A potential weakness of the CRC as an integrity check is that it is a linear function. This means that you can predict which bits of the CRC are changed if a single bit of the message is changed. Furthermore, it is possible to determine which combination of bits could be flipped in the message so that the net result is no change in the CRC. Thus, there are a number of combinations of bit flippings of the plaintext message that leave the CRC unchanged, so message integrity is defeated. However, in WEP, if an attacker does not know the encryption key, the attacker does not have access to the plaintext, only to the ciphertext block. Does this mean that the ICV is protected from the bit flipping attack? Explain.

CHAPTER 8

Electronic Mail Security

LEARNING OBJECTIVES

After studying this chapter, you should be able to:

- Summarize the key functional components of the Internet mail architecture.
- Explain the basic functionality of SMTP, POP3, and IMAP.
- Explain the need for MIME as an enhancement to ordinary e-mail.
- Describe the key elements of MIME.
- Understand the functionality of S/MIME and the security threats it addresses.
- Understand the basic mechanisms of STARTTLS and its role in e-mail security.
- Understand the basic mechanisms of DANE and its role in e-mail security.
- Understand the basic mechanisms of SPF and its role in e-mail security.
- Understand the basic mechanisms of DKIM and its role in e-mail security.
- Understand the basic mechanisms of DMARC and its role in e-mail security.

In virtually all distributed environments, electronic mail is the most heavily used network-based application. Users expect to be able to, and do, send e-mail to others who are connected directly or indirectly to the Internet, regardless of host operating system or communications suite. With the explosively growing reliance on e-mail, there grows a demand for authentication and confidentiality services. Two schemes stand out as approaches that enjoy widespread use: Pretty Good Privacy (PGP) and S/MIME. Both are examined in this chapter. This chapter concludes with a discussion of DomainKeys Identified Mail.

8.1 INTERNET MAIL ARCHITECTURE

For an understanding of the topics in this chapter, it is useful to have a basic grasp of the Internet mail architecture, which is currently defined in RFC 5598 (*Internet Mail Architecture*, July 2009). This section provides an overview of the basic concepts.

E-mail Components

At its most fundamental level, the Internet mail architecture consists of a user world in the form of Message User Agents (MUA), and the transfer world, in the form of the **Message Handling Service (MHS)**, which is composed of Message Transfer Agents (MTA). The MHS accepts a message from one user and delivers it to one or more other users, creating a virtual MUA-to-MUA exchange environment. This architecture involves three types of interoperability. One is directly between users: messages must be formatted by the MUA on behalf of the message author so that

the message can be displayed to the message recipient by the destination MUA. There are also interoperability requirements between the MUA and the MHS—first when a message is posted from an MUA to the MHS and later when it is delivered from the MHS to the destination MUA. Interoperability is required among the MTA components along the transfer path through the MHS.

Figure 8.1 illustrates the key components of the Internet mail architecture, which include the following.

- **Message User Agent (MUA):** Operates on behalf of user actors and user applications. It is their representative within the e-mail service. Typically, this function is housed in the user's computer and is referred to as a client e-mail program or a local network e-mail server. The author MUA formats a message and performs initial submission into the MHS via a MSA. The recipient MUA processes received mail for storage and/or display to the recipient user.
- **Mail Submission Agent (MSA):** Accepts the message submitted by an MUA and enforces the policies of the hosting domain and the requirements of Internet standards. This function may be located together with the MUA or

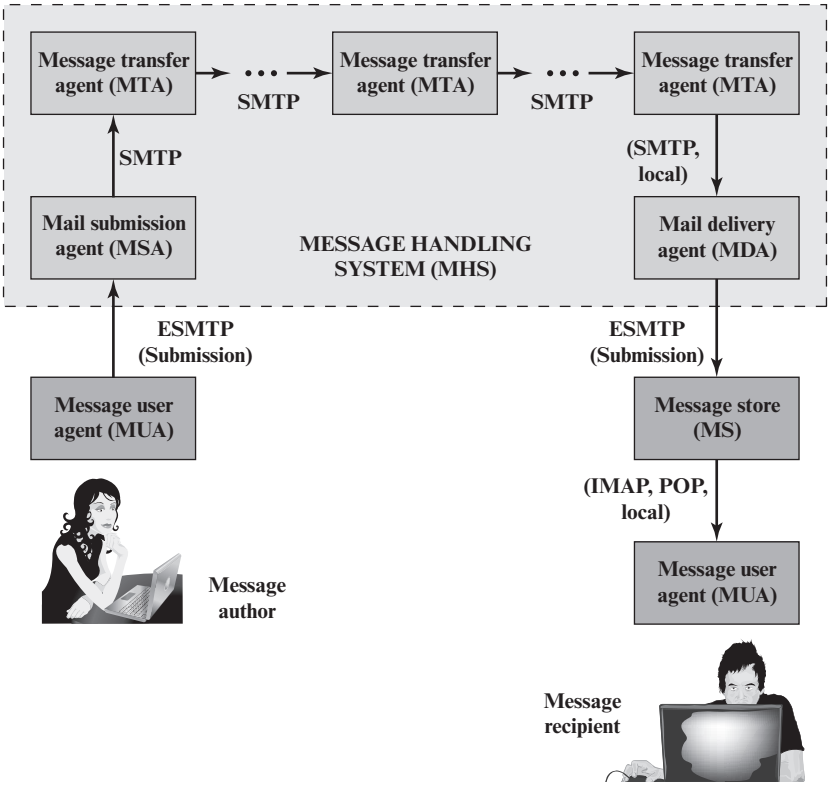

Figure 8.1 Function Modules and Standardized Protocols Used between them in the Internet Mail Architecture

as a separate functional model. In the latter case, the Simple Mail Transfer Protocol (SMTP) is used between the MUA and the MSA.

- **Message Transfer Agent (MTA):** Relays mail for one application-level hop. It is like a packet switch or IP router in that its job is to make routing assessments and to move the message closer to the recipients. Relaying is performed by a sequence of MTAs until the message reaches a destination MDA. An MTA also adds trace information to the message header. SMTP is used between MTAs and between an MTA and an MSA or MDA.
- **Mail Delivery Agent (MDA):** Responsible for transferring the message from the MHS to the MS.
- **Message Store (MS):** An MUA can employ a long-term MS. An MS can be located on a remote server or on the same machine as the MUA. Typically, an MUA retrieves messages from a remote server using POP (Post Office Protocol) or IMAP (Internet Message Access Protocol).

Two other concepts need to be defined. An **administrative management domain (ADMD)** is an Internet e-mail provider. Examples include a department that operates a local mail relay (MTA), an IT department that operates an enterprise mail relay, and an ISP that operates a public shared e-mail service. Each ADMD can have different operating policies and trust-based decision making. One obvious example is the distinction between mail that is exchanged within an organization and mail that is exchanged between independent organizations. The rules for handling the two types of traffic tend to be quite different.

The **Domain Name System (DNS)** is a directory lookup service that provides a mapping between the name of a host on the Internet and its numerical address. DNS is discussed subsequently in this chapter.

E-mail Protocols

Two types of protocols are used for transferring e-mail. The first type is used to move messages through the Internet from source to destination. The protocol used for this purpose is SMTP, with various extensions and in some cases restrictions. The second type consists of protocols used to transfer messages between mail servers, of which IMAP and POP are the most commonly used.

Simple Mail Transfer Protocol SMTP encapsulates an e-mail message in an envelope and is used to relay the encapsulated messages from source to destination through multiple MTAs. SMTP was originally specified in 1982 as RFC 821 and has undergone several revisions, the most current being RFC 5321 (October 2008). These revisions have added additional commands and introduced extensions. The term Extended SMTP (ESMTP) is often used to refer to these later versions of SMTP.

SMTP is a text-based client-server protocol where the client (e-mail sender) contacts the server (next-hop recipient) and issues a set of commands to tell the server about the message to be sent, then sending the message itself. The majority of these commands are ASCII text messages sent by the client and a resulting return code (and additional ASCII text) returned by the server.

The transfer of a message from a source to its ultimate destination can occur over a single SMTP client/server conversation over a single TCP connection. Alternatively, an SMTP server may be an intermediate relay that assumes the role of an SMTP client after receiving a message and then forwards that message to an SMTP server along a route to the ultimate destination.

The operation of SMTP consists of a series of commands and responses exchanged between the SMTP sender and receiver. The initiative is with the SMTP sender, who establishes the TCP connection. Once the connection is established, the SMTP sender sends commands over the connection to the receiver. Each command consists of a single line of text, beginning with a four-letter command code followed in some cases by an argument field. Each command generates exactly one reply from the SMTP receiver. Most replies are a single-line, although multiple-line replies are possible. Each reply begins with a three-digit code and may be followed by additional information.

Figure 8.2 illustrates the SMTP exchange between a client (C) and server (S). The interchange begins with the client establishing a TCP connection to TCP port 25 on the server (not shown in figure). This causes the server to activate SMTP and send a 220 reply to the client. The HELO command identifies the sending domain, which the server acknowledges and accepts with a 250 reply. The SMTP sender is transmitting mail that originates with the user Smith@bar.com. The MAIL command identifies the originator of the message. The message is addressed to three users on machine foo.com, namely, Jones, Green, and Brown. The client

```
S: 220 foo.com Simple Mail Transfer Service Ready
C: HELO bar.com
S: 250 OK
C: MAIL FROM:<Smith@bar.com>
S: 250 OK
C: RCPT TO:<Jones@foo.com>
S: 250 OK
C: RCPT TO:<Green@foo.com>
S: 550 No such user here
C: RCPT TO:<Brown@foo.com>
S: 250 OK
C: DATA
S: 354 Start mail input; end with <crlf>.<crlf>
C: Blah blah blah . . .
C: . . . etc. etc. etc.
C: <crlf><crlf>
S: 250 OK
C: QUIT
S: 221 foo.com Service closing transmission channel
```

Figure 8.2 Example SMTP Transaction Scenario

identifies each of these in a separate RCPT command. The SMTP receiver indicates that it has mailboxes for Jones and Brown but does not have information on Green. Because at least one of the intended recipients has been verified, the client proceeds to send the text message, by first sending a DATA command to ensure the server is ready for the data. After the server acknowledges receipt of all the data, it issues a 250 OK message. Then the client issues a QUIT command and the server closes the connection.

A significant security-related extension for SMTP, called STARTTLS, is defined in RFC 3207 (*SMTP Service Extension for Secure SMTP over Transport Layer Security*, February 2002). STARTTLS enables the addition of confidentiality and authentication in the exchange between SMTP agents. This gives SMTP agents the ability to protect some or all of their communications from eavesdroppers and attackers. If the client does initiate the connection over a TLS-enabled port (e.g., port 465 was previously used for SMTP over SSL), the server may prompt with a message indicating that the STARTTLS option is available. The client can then issue the STARTTLS command in the SMTP command stream, and the two parties proceed to establish a secure TLS connection. An advantage of using STARTTLS is that the server can offer SMTP service on a single port, rather than requiring separate port numbers for secure and cleartext operations. Similar mechanisms are available for running TLS over IMAP and POP protocols.

Historically, MUA/MSA message transfers have used SMTP. The standard currently preferred is SUBMISSION, defined in RFC 6409 (*Message Submission for Mail*, November 2011). Although SUBMISSION derives from SMTP, it uses a separate TCP port and imposes distinct requirements, such as access authorization.

Mail Access Protocols (POP3, IMAP) Post Office Protocol (POP3) allows an e-mail client (user agent) to download an e-mail from an e-mail server (MTA). POP3 user agents connect via TCP to the server (typically port 110). The user agent enters a username and password (either stored internally for convenience or entered each time by the user for stronger security). After authorization, the UA can issue POP3 commands to retrieve and delete mail.

As with POP3, Internet Mail Access Protocol (IMAP) also enables an e-mail client to access mail on an e-mail server. IMAP also uses TCP, with server TCP port 143. IMAP is more complex than POP3. IMAP provides stronger authentication than POP3 and provides other functions not supported by POP3.

8.2 E-MAIL FORMATS

To understand S/MIME, we need first to have a general understanding of the underlying e-mail format that it uses, namely, MIME. But to understand the significance of MIME, we need to go back to the traditional e-mail format standard, RFC 822, which is still in common use. The most recent version of this format specification is RFC 5322 (*Internet Message Format*, October 2008). Accordingly, this section first provides an introduction to these two earlier standards and then moves on to a discussion of S/MIME.

RFC 5322

RFC 5322 defines a format for text messages that are sent using electronic mail. It has been the standard for Internet-based text mail messages and remains in common use. In the RFC 5322 context, messages are viewed as having an envelope and contents. The envelope contains whatever information is needed to accomplish transmission and delivery. The contents compose the object to be delivered to the recipient. The RFC 5322 standard applies only to the contents. However, the content standard includes a set of header fields that may be used by the mail system to create the envelope, and the standard is intended to facilitate the acquisition of such information by programs.

The overall structure of a message that conforms to RFC 5322 is very simple. A message consists of some number of header lines (*the header*) followed by unrestricted text (*the body*). The header is separated from the body by a blank line. Put differently, a message is ASCII text, and all lines up to the first blank line are assumed to be header lines used by the user agent part of the mail system.

A header line usually consists of a keyword, followed by a colon, followed by the keyword's arguments; the format allows a long line to be broken up into several lines. The most frequently used keywords are *From*, *To*, *Subject*, and *Date*. Here is an example message:

```
Date: October 8, 2009 2:15:49 PM EDT
From: "William Stallings" <ws@shore.net>
Subject: The Syntax in RFC 5322
To: Smith@Other-host.com
Cc: Jones@Yet-Another-Host.com

Hello. This section begins the actual
message body, which is delimited from the
message heading by a blank line.
```

Another field that is commonly found in RFC 5322 headers is *Message-ID*. This field contains a unique identifier associated with this message.

Multipurpose Internet Mail Extensions

Multipurpose Internet Mail Extension (MIME) is an extension to the RFC 5322 framework that is intended to address some of the problems and limitations of the use of Simple Mail Transfer Protocol (SMTP) or some other mail transfer protocol and RFC 5322 for electronic mail. RFCs 2045 through 2049 define MIME, and there have been a number of updating documents since then.

As justification for the use of MIME, [PARZ06] lists the following limitations of the SMTP/5322 scheme.