Chapter 19
SECURE E-MAIL

Many e-mail products have cryptographic protection built in, and you can also install your own cryptographic programs. For example, Microsoft Outlook Express comes with S/MIME (Secure/Multipurpose Internet Mail Extension) support. But because some copies of Outlook Express S/MIME support only 512-bit public keys, some users might want to add PGP, which supports longer public keys.

Check that your e-mail program uses keys of the appropriate length.

There are many e-mail application programs; each one differs at least slightly from the others. Just as each e-mail application has particular characteristics, each cryptographic e-mail add-on has particular characteristics. Each secure e-mail program can offer unique choices for secret key methods, size of secret key, and so on. For example, some secure e-mail programs offer only 40-bit secret keys (recall that DES offers 56-bit keys).[1]

Generic Cryptographic E-mail Messages

Although there are differences among cryptographic e-mail add-ons, they all share a common characteristic: There's only a single message from e-mail sender to receiver.

All the cryptographic assurances and parameters—as well as the e-mail text itself—are contained within the one message. This is in contrast to SSL and IPsec (Chapters 20 and 21), in which sender and receiver exchange many messages just to establish cryptographic assurances and parameters.

In Figure 19-1, Alice prepares an e-mail for Bob including all possible assurances.

1. By now, you realize that 40-bit keys aren't particularly secure against a crypto-savvy snoop.

Generate one-time secret key (session key).

1. Alice generates a secret key for one-time use. This kind of secret key is called a *session key*. (Most e-mail programs use secret key cryptography for bulk data encryption. Recall that encryption with a secret key is much faster than encryption with a public key.)

Encrypt e-mail with session key.

2. Alice uses the session key to encrypt the e-mail. Many e-mail systems also encrypt a timestamp with the e-mail signature to thwart replay attacks (see Chapter 22).

Encrypt session key with receiver's public key.

3. Because Bob doesn't have Alice's newly created session (secret) key, Alice encrypts the session key with Bob's public key. This is an example of secret key exchange. (Encrypting a secret key with a public key is sometimes referred to as *enveloping* the session key.[2])

Sign with sender's private key.

4. Alice signs a digest of the e-mail plaintext and the timestamp.
5. Alice sends the e-mail package to Bob.

When Bob receives Alice's e-mail, his private key decrypts the encrypted session key, and the session key decrypts Alice's encrypted message. Bob uses his copy of Alice's public key to verify Alice's signed message digest.

Verifying the signed digest

As shown in Chapter 13, the message digest acts as a condensed, redundant copy of the e-mail message. The e-mail and signed digest ensure that BlackHat can't alter the e-mail plaintext without being detected (it thereby ensures the integrity of the message). When Bob verifies the signed digest (with Alice's public key), it also assures him of the authenticity (origin)[3] of the e-mail message and provides nonrepudiation (Alice can't deny signing the message).

A secure e-mail program should also give Bob the option to check whether Alice's digital certificate (and attached public key), used to verify her signed message digest, has been revoked. Because checking certificate revocation can be time-consuming, many e-mail programs allow the receiver to set whether and when the program will automatically check revocation lists.

2. Obviously, Alice must have Bob's public key. As discussed, digital certificates are the preferred way to deliver public keys. Different e-mail programs provide different ways to retrieve digital certificates. For example, S/MIME mostly uses X.509 digital certificates; PGP mostly uses PGP digital certificates. Fortunately, S/MIME, PGP, and other e-mail vendors provide online (real-time) access to their subscribers' certificates. If Alice doesn't already have a verified copy of Bob's public key, she can get one from an online server.
3. In cryptography lingo, the data origin.

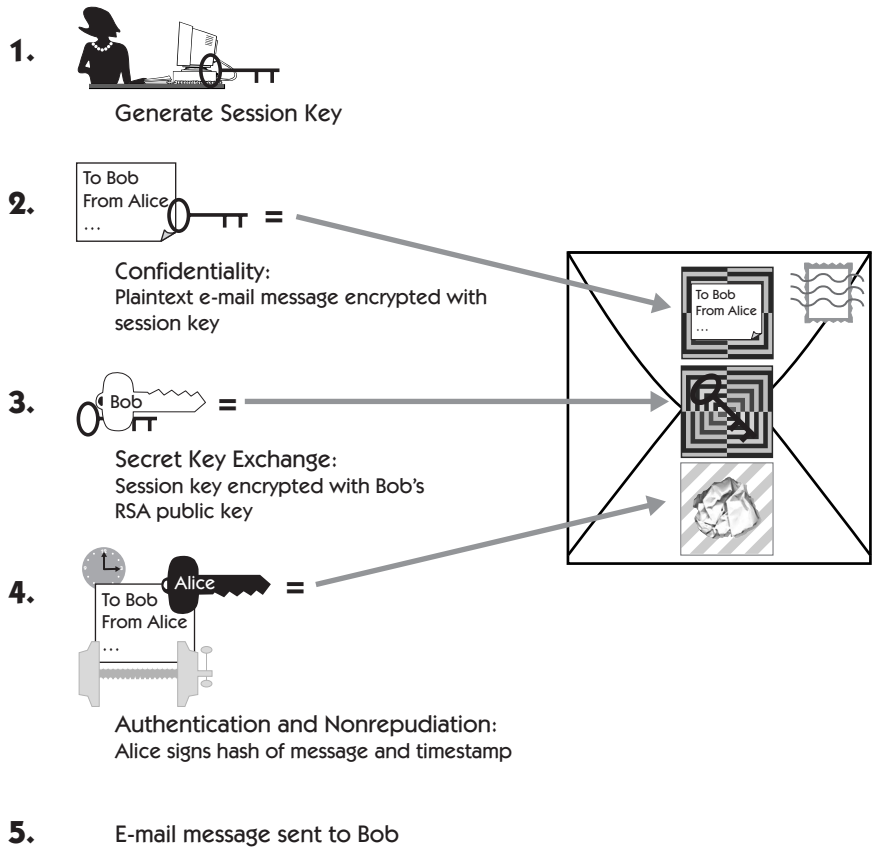

Figure 19-1 Alice prepares a secure e-mail message for Bob.

Invoking Cryptographic Services

Users must invoke cryptography.

Users of secure e-mail must invoke cryptographic services from most e-mail packages; they are not automatic. Figure 19-2 shows a screen shot of S/MIME and PGP services offered with Microsoft Outlook Express.

Although the top of Figure 19-2 shows that both S/MIME and PGP services are available, none has been selected; the message will be sent as unsigned plaintext, neither encrypted nor signed. At the bottom of Figure 19-2, PGP encryption and signing are requested, so the message will be encrypted and signed.

S/MIME and PGP signing and encryption offered but none selected

User must have private key and recipient's public key.

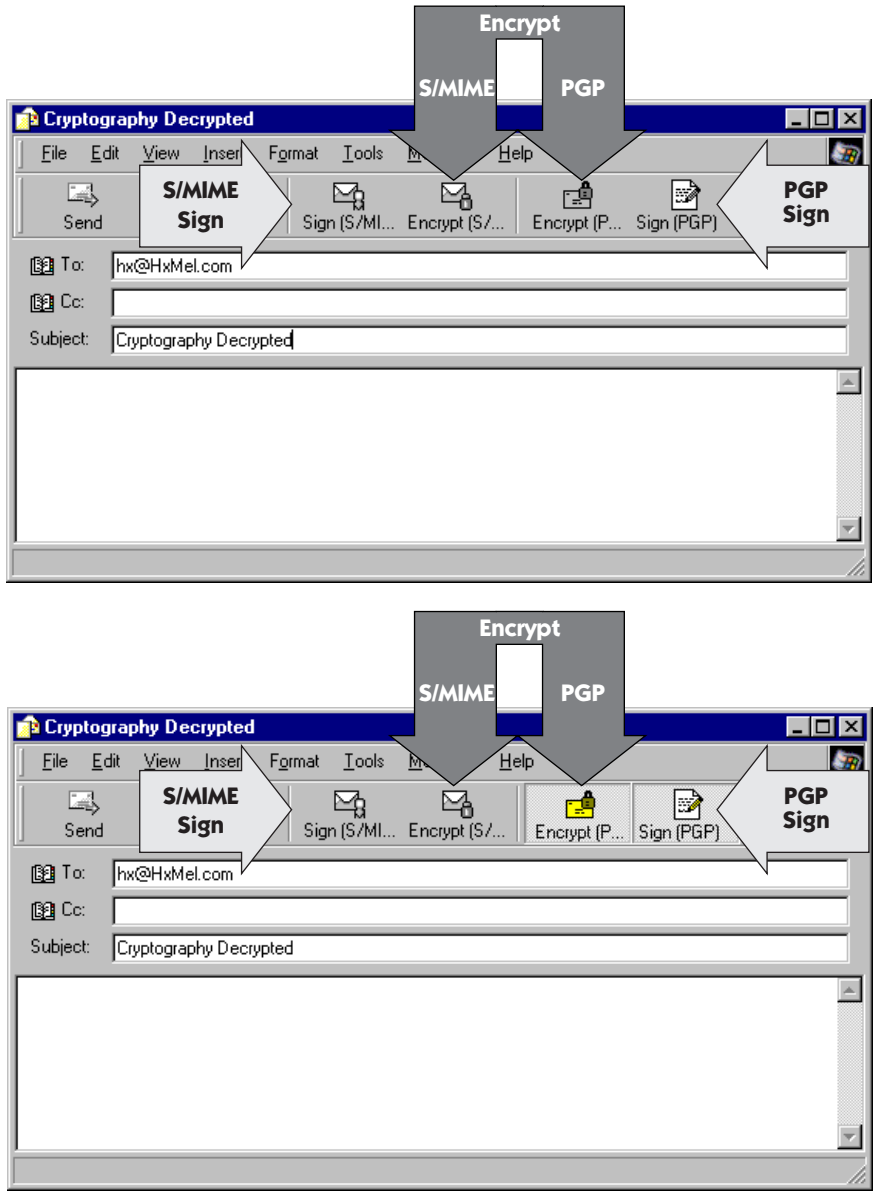

Figure 19-2 Cryptographic services available in Microsoft Outlook Express. Top: No cryptographic services invoked. Bottom: PGP encryption and signing services invoked.

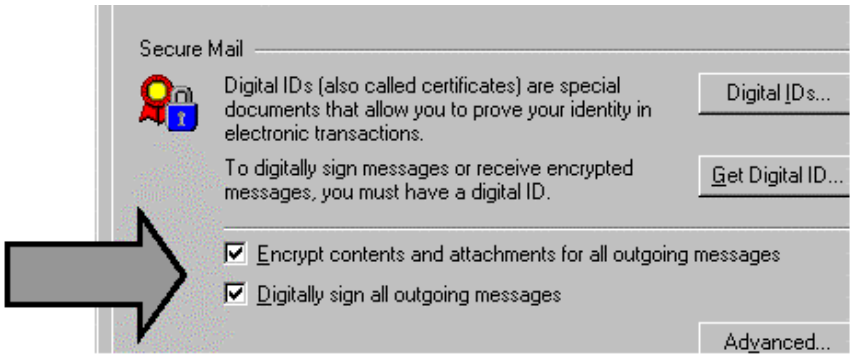

Figure 19-3 Setting default cryptographic services.

Although the need to select cryptographic services may seem obvious, at least one recent study of secure e-mail users found that many experienced Internet users were not properly invoking e-mail cryptographic services. They mistakenly believed that encryption and authentication are transparently and automatically invoked just because cryptographic e-mail icons appear on the menu bar.

Users can usually set their software to default to cryptographic protections.

Most secure e-mail programs, such as those bundled with Microsoft products or add-ons such as PGP, allow users to choose default settings. Figure 19-3 shows a screen shot in which encryption and signing have been set as default parameters.

Confidentiality and Authentication

Confidentiality and authentication must be applied appropriately if a user of secure e-mail is to achieve the desired cryptographic assurances.

Choosing Services

E-mail users must also know the difference between the encryption (confidentiality) and signing (authenticity) options. Usually, any message that is encrypted should also be signed. Encryption does not necessarily imply that the message can't be altered in transit.[4]

4. For an example, see Chapter 22.

Positioning Services

Some e-mail programs allow a user to choose the order in which confidentiality and authentication are invoked. The user can choose to first encrypt with secret key and then sign with private key, or vice versa. There are significant differences between these two approaches, and experts disagree on which is more desirable.

Encrypt and Then Sign

A message using this option (top, Figure 19-4) allows the recipient, or anyone else, to verify the signatures before decrypting the message. This option is useful in automated verification systems; public keys perform the verification process, and the encrypted message can be passed to some other service for decryption.

IPsec, discussed in Chapter 21 and Appendix B, encrypts and then signs. IPsec uses the principle that it's better to verify first so that unauthenticated packets can be discarded before time is wasted on decryption.

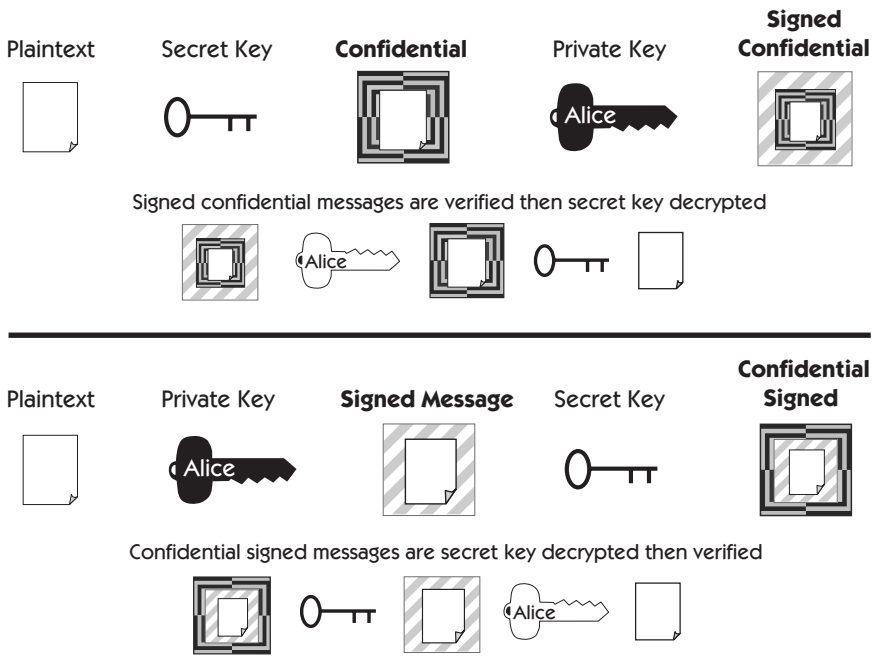

Figure 19-4 Comparing encryption and signing sequences. Top: encrypted and then signed. Bottom: signed and then encrypted. For simplicity, hash is not shown.

Sign and Then Encrypt

Some cryptographic experts recommend the use of this option (bottom, Figure 19-4). PGP and Secure Socket Layer (discussed in Chapter 20) sign and then encrypt; they use the principle that the signature should be over plaintext because plaintext is the essence of the message.

Encryption after signing hides private key signatories. But the receiver must first decrypt before authenticating and so can't quickly discard forged packets. This means that BlackHat can send forged packets to Bob, claiming they're from Alice. Bob must decrypt each one before any of them fails authentication.

Deterring E-mail Viruses

Signing deters viruses.

If you accept only signed e-mails, it's much more difficult for BlackHat to violate your computer. A signed e-mail is verified before most viruses can take control. If BlackHat intercepts Alice's e-mail to Bob and substitutes his forged e-mail, it will fail authentication when Bob verifies it with his trusted copy of Alice's public key. The forged e-mail will not get the chance to launch its destructive payload because it's simply dropped or quarantined.

Of course, if BlackHat can alter the cryptographic verification program or substitute a forged copy of Alice's public key, he can violate the authenticity check and fool Bob into accepting his e-mail.

Review

There are many secure e-mail packages, each with its own characteristics. Each can offer various choices for secret key methods, public key methods, message digest methods, and so on.

In general, secure mail encryption follows this pattern:

1. Alice generates a session (secret) key.
2. She encrypts the plaintext message with the secret key (to ensure confidentiality).
3. She encrypts the session key with Bob's public key (key distribution).
4. She signs a hash of the message and timestamp (to ensure authentication, integrity, and nonrepudiation).
5. She sends the secret key encrypted message, the public key encrypted session key, and the signed hash to Bob.

Most secure e-mail packages require that the sender invoke encryption and signing, either by setting the program's default or by selecting encryption and signing each time e-mail is sent. Knowing the difference between encryption and signing and the assurances provided by each method can help users to make appropriate choices when encrypting e-mail messages.