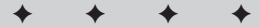


# E-Commerce Transaction Security

---

## EXAM OBJECTIVES

- ◆ Authentication and identification
- ◆ Encryption and decryption
- ◆ Certificates
- ◆ X.509v3
- ◆ Electronic commerce security myths



## CHAPTER PRE-TEST

1. What are the five purposes security serves in electronic commerce?
2. What is non-repudiation?
3. What is the difference between symmetric and asymmetric encryption?
4. What is a message digest?
5. What is the purpose of a Certifying Authority (CA)?
6. What are the four types of digital certificates?
7. What is SSL?
8. What is X.509?

**S**ecurity is vital to commerce of any kind, and systems designed to prevent fraud and theft are parts of every traditional and electronic commerce system. In this chapter, you get a broad overview of e-commerce security, learn why it's important, and discover how the different parts of e-commerce systems can be secured.

## Understanding E-Commerce Security

The Internet is probably one of the most complex entities ever created by humans. As a result, it's full of bugs, security holes, and vulnerabilities. The great thing about the Internet is that it is designed to keep running in spite of problems.

Individual sites on the Internet, however, are much less resilient. Web sites need to be protected from malicious attacks as well as from people who are just snooping around. As in the world outside of the Internet, where there's money, there are thieves. Where there are thieves, however, there are good locks and security systems.

Security on the Internet serves five purposes:

- ♦ Authentication and identification
- ♦ Access control
- ♦ Data confidentiality
- ♦ Data integrity
- ♦ Non-repudiation

In the following sections, we define and explain each of these purposes of security.

### Establishing identification and authentication

**Objective****Authentication and identification**

Identification and authentication ensure that someone is who he or she claims to be. As the following sections illustrate, these concepts, although similar, can have very different implications when dealing with a traditional versus an Internet application.

### By traditional means

When you write a check to your local grocery store, the cashier typically asks for your driver's license or another form of identification. This practice serves the following purposes:

- ♦ To verify that you are the person whose name is on the check
- ♦ To allow the cashier to record your driver's license number, which helps the merchant locate you if the check bounces

Similarly, when you go to a doctor's office, you go through a process of identification and authentication to make sure that your doctor is actually a doctor and not someone merely pretending to be one. Some of the steps involved in this process might include examining the degrees on the wall, comparing the office to other doctors' offices you've been in, and so forth. You may also have done some research in advance, such as asking a trusted family member or friend for a recommendation, or verifying the doctor's identity with an organization such as the American Medical Association.

### On the Internet

On the Internet, many of the methods we use for verifying identity and authenticity in the real world are meaningless. Anyone can display a picture of a supermodel on his or her free home page and claim to be that person. Although it's unlikely that a super model would create a *free* Web site, you would have a hard time proving or disproving this claim.

One form of proof would be to personally ask the super model. You could also ask someone you trust and who is a mutual friend of both you and the super model. If the owner of the site actually is who she claims to be, she may become tired of constantly affirming her identity to every visitor to the site.

To speed the process of verifying identification, the owner of a Web site can obtain a certificate from a third party that has done the research and verified the authenticity of the site. As long as the third party can be trusted, visitors to the site only need to check the certificate to confirm that the site belongs to the person claiming ownership. This, in fact, is how digital certificates work. You'll learn more about this in the "Understanding Digital Certificates" section later in this chapter.



**Exam Tip** It's important to know that digital certificates provide authentication.

## Managing access control

Access control is the process of limiting access to resources. In the physical world, access control is most often provided using locks and keys. Newer and more advanced access control systems may use unique identifying characteristics such as thumb prints or voice prints instead of keys. Either way, the concept is the same: Let in some people and keep others out.

In computer systems, and on the Internet, access control is provided by a variety of mechanisms, as follows:

- ♦ User names and passwords are the most common way of securing computing resources.
- ♦ You can also limit the times during which resources can be used (only during business hours, for example).
- ♦ You can further limit users by host name.

## Ensuring data confidentiality

Data confidentiality deals with the ability of parties to exchange information without it being read by anyone else. This is where cryptography comes in. *Cryptography* is the use of complicated mathematics to render data unreadable to prying eyes. The basic concepts behind cryptography are simple to understand. You'll explore these concepts in more detail in the "Understanding Cryptography" section later in this chapter.

## Ensuring data integrity

Data integrity deals with the validity of data. Examples of data integrity questions include:

- ♦ Is this e-mail the same as when it was sent?
- ♦ Are these the actual stock prices?
- ♦ Has the image on the television been modified?

Integrity is different from authentication. Authentication deals with who sent a message or who authorized a purchase, whereas integrity deals with whether the data in that message or the details of the purchase were modified after they were sent.

Integrity also does not deal with the accuracy of data. Accuracy tells how data relates to the world. Integrity deals with the data's relation to itself over time.

## Ensuring non-repudiation

*Non-repudiation* is defined as the method used to prevent the parties involved in a transaction from denying that they agreed to a sale or purchase. In the physical world, a signed credit card receipt from a store provides non-repudiation for both the merchant and the customer. The customer uses the receipt to prove that an item he or she is returning was actually purchased at that store. The merchant, on the other hand, can use its copy of the receipt in the event that the customer denies he or she purchased the item and demands a refund from the credit card company.

Non-repudiation is provided using authentication and auditing. We've already talked about authentication in the "Establishing identification and authentication" section. Auditing is generally done by monitoring a server's logs. Without auditing your secure system, you may have hackers running wild on your system and not even know it.

## Understanding Cryptography

### Objective

#### Encryption and decryption

Cryptography is the study of the development of methods of secret writing. Cryptography has been used for thousands of years to protect secrets. Hundreds of methods of secret writing have been designed, and hundreds of ways have been found to defeat the different methods of secret writing, which keeps the inventors of cryptographic systems busy coming up with new methods of secret writing.

### Cryptography terms

Today, cryptography is more widely used than ever, and an understanding of cryptographic techniques and technologies is vital to understanding e-commerce security. Before you continue, you need to understand the following terms:

- ♦ **Plaintext.** A message that can be read by humans.
- ♦ **Ciphertext.** Text that has been disguised to make it unreadable by humans.
- ♦ **Encryption.** The process of creating ciphertext from plaintext.
- ♦ **Decryption.** The process of restoring the plaintext from ciphertext.
- ♦ **Cipher.** A cryptographic algorithm used to encrypt and decrypt messages.
- ♦ **Key.** A value used by a cryptographic algorithm to encrypt and decrypt messages. Keys are the element of a cryptosystem that are unique to particular users of the system. As a result of the use of keys, the best encryption methods available today are open source. Anyone can find out how they work, but this doesn't aid in decryption, because keys are still kept secret. The same is true of real world keys and locks. Anyone can find out how a door lock works (they have pins that can be set to various positions), but to open the lock, you need the correct key.

### Evaluating encryption strength

The strength of encryption depends on the following three factors:

- ♦ **The strength of the algorithm.** Over the years, many algorithms that were thought to be unbreakable have proven to have flaws that could be exploited to crack the code.



The encryption standards we discuss in this chapter have all been thoroughly tested over time and have proven to be reliable. This is not to say that they're unbreakable, but the effort required to break them is so great that it outweighs the potential benefits of doing so.

- ♦ **The secrecy of the key.** Naturally, if a third party intercepts the key, no message encrypted with that key is safe. This is actually the biggest problem with secret key encryption—it requires that all of the parties involved must securely exchange a key to exchange secure messages.



In a later section of this chapter, you'll look at public key encryption, or asymmetric encryption, which does not require the parties who wish to exchange secure messages to share a key.

- ♦ **The length of the key.** The longer the key, the more possible keys there are. The more possible keys that exist, the more difficult it is for someone to discover the key using a *brute force attack*. In a brute force attack, each possible key is tried until the correct one is found. This approach is not the most brilliant way to crack a code, but it works.

## Calculating security in key lengths

The length of a key is specified in bits. Because bits can have one of two possible values, the number of possible combinations of a key doubles for each additional bit in the key length and can be expressed as  $2^n$ , where  $n$  is the number of bits.

For example, a 2-bit key has  $2^2$ , or 4 possible combinations. A 10-bit key has  $2^{10}$ , or 1024 possible combinations. This is still extremely weak encryption. A 40-bit key has 1,099,511,627,776 possible values. Now we're getting somewhere. Still, a 40-bit key could be cracked in 1998 in 18 minutes. With the increasing speed of computers, 40-bit keys just don't cut it today.

Today, it's common for keys to be at least 128 bits. Using this key length, it would take today's fastest computers millions of years to try out every possible key combination. Because we cannot predict future computing advances, many experts today recommend key lengths of 1024 bits.



Key length is not the only factor that determines the security of a cryptosystem. It would also take a long time to try every possible combination on a padlock, or every possible key to a door lock, but this is not the most efficient way to defeat these security methods. It's much easier to cut the lock or break a window.

## Types of Cryptography

There are three types of encryption that are commonly used today:

- ♦ Symmetric encryption
- ♦ Asymmetric encryption
- ♦ One-way encryption

The next sections look at each of these encryption types in detail.



Knowing these three types of encryption, the difference between them, and how each uses keys is important and heavily emphasized on the exam.

### Symmetric encryption

Symmetric encryption is also known as private key encryption, session key encryption, shared key encryption, and secret-key encryption. In private key encryption, the parties sharing information must all have an identical, secret key. Symmetric key encryption is one of the oldest forms of secret writing.

### Block ciphers

One type of symmetric encryption algorithm is the *block cipher*, which uses a key to transform a block of plaintext into a block of ciphertext of the same length. Reversing the transformation that encrypted the plaintext decrypts the ciphertext.

A very simple example of symmetric encryption using a block cipher is as follows: Mr. Smith and Mr. Jones wish to exchange secret messages. During their daily meetings, Mr. Smith hands Mr. Jones a piece of paper containing a key. Today's key is the following:

TODAYSKEY

In addition to the key, both Mr. Smith and Mr. Jones must know the algorithm they are using. In this case, it's the one known as the modern Vigenère cipher. (This is referred to as the modern Vigenère cipher because it's a watered-down version of the cipher originally created by Blaise de Vigenère in the 16th century.)

This cipher works by encrypting plaintext using a key and a table, such as the one shown in Table 19-1.

**Table 19-1**  
**The Modern Vigenère Table**

<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>	<i>j</i>	<i>k</i>	<i>l</i>	<i>m</i>	<i>n</i>	<i>o</i>	<i>p</i>	<i>q</i>	<i>r</i>	<i>s</i>	<i>t</i>	<i>u</i>	<i>v</i>	<i>w</i>	<i>x</i>	<i>y</i>	<i>z</i>	
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

The alphabet across the top of the table is the plaintext alphabet. The one along the left side is the key alphabet. To begin encrypting a message using this key, you find the first plaintext letter in the top alphabet, and then trace down that column until it intersects with the row containing the first key letter. Repeat this process for each letter of the plaintext, repeating the key as necessary. For example, Listing 19-1 shows the plaintext, key, and ciphertext for a message encrypted using this method.

### Listing 19-1: The key, plaintext, and ciphertext for a message encrypted with the modern Vigenère cipher

<i>key</i>	TODAYSKEYTODAYSK
<i>plain</i>	hereisthemessage
<i>cipher</i>	ASUEGKDLCFVSYYO

Any one who intercepts the ciphertext will see only gibberish (ASUEGKDLCFVSYYO). A person who knows how to decode the message and who has the key is able to decrypt this message easily. For example, given the first letter in the ciphertext (A), and the first letter in the key (T), you simply locate the T in the key letter column (to the left), and then follow that row until you find the A. Follow that column up to the top to locate the first plaintext letter, which happens to be h.

Although this is a simple example of symmetric encryption, it illustrates the basic principles involved and the use of a key to make it possible for a publicly known algorithm to provide security. This particular cipher is fairly easy to crack (especially when a large amount of ciphertext is provided or when the key is short), and does not provide much security.

The following encryption algorithms use symmetric key encryption:

- ♦ **Data Encryption Standard (DES).** DES is the most widely used symmetric encryption algorithm. It was designed by IBM for the U.S. government in the 1970s and uses a 56-bit key. Today, it is widely considered to be obsolete and vulnerable to cracking.
- ♦ **Triple-DES.** Triple-DES encrypts data by running it through DES encryption three times — forwards, backwards, and then forwards again. During the backwards run, it uses a second 56-bit key. It is considered to be a good successor to DES, because it doesn't require new algorithms.
- ♦ **RC4.** Created by Ron Rivest, RC4 is a very fast encryption algorithm that is frequently used by the Secure Sockets Layer (SSL) protocol. It uses a variable key length.

**Caution**

RC4 was originally a trade secret of RSA Data Security, Inc, but the algorithm was anonymously posted to the Internet in September of 1994. RC4 has been cracked using brute force attacks and is beginning to show signs of age.

- ♦ **RC5.** RC5 uses a variable key length as well as variable block sizes and a variable number of rounds.

**In the Real World**

RC5 with a 32-bit key (RC5-32) was cracked in 1997 by an organization called distributed.net, which uses computers connected through the Internet to try possible keys. Since 1997, distributed.net has been trying to crack a message encoded in RC5-64. As of early 2001, the rate at which distributed.net is testing possible keys is 127,000,000,000 keys per second.

- ♦ **Skipjack.** Created by the National Security Agency (NSA), Skipjack uses an 80-bit key and a 64-bit 32-round block cipher.
- ♦ **International Data Encryption Algorithm (IDEA).** IDEA uses a 128-bit key to operate on 64-bit plaintext blocks in eight iterations.
- ♦ **Blowfish.** An encryption algorithm developed by Bruce Schneier in 1993 that has a variable key length from 32 to 448-bits.
- ♦ **Twofish.** Also designed by Bruce Schneier, Twofish was a finalist to become the new standard encryption of the U.S. government.
- ♦ **Advanced Encryption Standard (AES).** AES is the name for the new standard that will replace DES as the U.S. government's standard cipher.

## One-time pads

One-time pads are the simplest, and the most secure type of cryptographic algorithm. Unfortunately, they are very impractical for most purposes. The idea behind a one-time pad is that you have a pad containing key letters. To encrypt a message using this pad, you add each letter in the pad to a letter in the plaintext, and never use a key letter more than once. The result is a key that is the same length as the original message.

For example, assume that the key is:

TUSCKR

and the plaintext is:

monkey

To encrypt the plaintext, you would add the numerical values of the letters and subtract 26 if the result is greater than 26. In the previous example, the plaintext would be encrypted as:

$$\begin{aligned}
 20+13 &= 33 - 26 = 7 = g \\
 21+15 &= 36 - 26 = 10 = j \\
 19+14 &= 33 - 26 = 7 = g \\
 3+11 &= 14 = n \\
 11+5 &= 16 = p \\
 18+25 &= 43 - 26 = 17 = q
 \end{aligned}$$

or

ciphertext = gjgnpq

Starting with this ciphertext, it's impossible to figure out the plaintext without the key. Unfortunately, things aren't always so neat in the real world. Here's why:

- ♦ It's often impractical to exchange one-time pads.
- ♦ People make mistakes, which often compromise one-time pad systems.
- ♦ It's not realistic to encrypt large items using one-time pads, because the key must be just as large as the item being encrypted.

Because of these limitations, one-time pads are not used in Internet security.

## Asymmetric encryption

Asymmetric encryption does not require two parties who want to exchange encrypted data to share a key. In fact, using asymmetric encryption, two complete strangers can send each other encrypted messages that can only be read by the intended recipient.

Whereas the idea behind symmetric encryption is centuries old, asymmetric encryption is a very new technology. Whitfield Diffie and Martin Hellman first proposed it in 1976.

Asymmetric, or public key, encryption relies on algorithms that are easy to calculate in one direction but very difficult to calculate in the opposite direction. For example, it's easy to multiply two large prime numbers together to get a product, but it's very difficult to factor the large product to arrive at the two factors that were originally multiplied together.



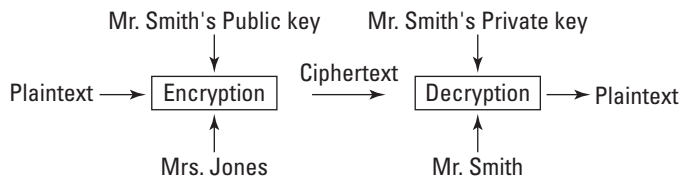
Although you don't need to understand the mathematical details of public key encryption systems to use them, you can find out these details easily on the Web. A place to begin is [www.rsasecurity.com](http://www.rsasecurity.com). Remember that keys make it possible for the actual algorithms to be public knowledge.

### The public key concept

Here's how public key encryption works: Mr. Smith uses software to create two keys: a public key and a private key. He keeps the private key in a safe place and doesn't share it with anyone. He does share the public key, however. There are vast

databases of public keys on the Internet where he can post his public key. He can also post his public key on his Web site, and even rent a billboard to display it to the world if he likes.

Mrs. Jones also has a public key and a private key. Mrs. Jones wants to send a secret message to Mr. Smith (who she has never met). While driving home from work one day, she stops to copy Mr. Smith's public key from the billboard she passes every day. (She could also just look at his Web site, or look him up in a directory.) Using Mr. Smith's public key, Mrs. Jones is able to encrypt a message to him. This message is only readable by Mr. Smith, because only his private key can decrypt it. Figure 19-1 illustrates this process.



**Figure 19-1:** Public-key (asymmetric) encryption

## How it works

Public key encryption has the following characteristics:

- ♦ The ciphertext is calculated by applying the algorithm with a public key on the plaintext. The ciphertext is decrypted by applying the algorithm with the private key.
- ♦ It's very difficult to calculate the private key from the public key.
- ♦ The public and private keys are easy to calculate, but not easy to guess.



In 1991, Phil Zimmerman released his public key encryption software, Pretty Good Privacy (PGP), as freeware. As a result, PGP quickly became the most widely used method for sending encrypted e-mail, and Zimmerman was the target of a three-year criminal investigation. The government held that strong cryptography was illegally exported to foreign countries as a result of PGP's publication. The case was eventually dropped and Network Associates, Inc. now owns and publishes PGP. PGP is still freely available for non-commercial uses at <http://web.mit.edu/network/pgp.html>.

The standard that is used by most public key encryption products, including PGP, is RSA. RSA was created by and named after Ron Rivest, Avi Shamir, and Rick Adleman in 1978 at the Massachusetts Institute of Technology (MIT). The key length used in RSA encryption is variable. Common key lengths in RSA security are 528-bit, 1024-bit, and 2048-bit.

## One-way encryption

One-way encryption is just what it sounds like: A way to encrypt data so it's not feasible to decrypt it. One-way encryption uses *hash functions*, which calculate a kind of digital fingerprint for a piece of digital data. The result of a hash function is much smaller than the data that was input into the function, but it's very difficult to create two different inputs into a hash function that would produce the same result or to derive the input from the result of the hash function. One-way hash functions are commonly used to store passwords, personal identification numbers, and so forth.

### How hash functions work

For example, assume that Laura's password is Cucumber23. When she first sets her password on a Web site or computer, the computer performs a one-way hash of this password and stores the result. For example, the result is t78dnkdur. The next time Laura returns to the site, she must enter the password. She's granted access if the password she enters, when run through the same hash function, matches the value stored on the computer.

Because it's nearly impossible to calculate the password from the one-way hash, anyone who manages to access the database where the passwords are stored would not be able to do anything with them.

One-way encryption also provides authentication and integrity. For example, if you send a letter to someone with a one-way hash of that letter, the recipient can hash the letter and compare the results to the hash you sent. If the two hashes match, you can be sure the letter has not been modified in transit. Applications of one-way encryption for providing authentication and integrity are sometimes known as *message digests*.

### Current sources

One-way encryption is a part of nearly every Internet protocol. Some of the hash functions in use today are:

- ♦ **Secure Hash Algorithm (SHA-1).** The U.S. government's standard hash function.
- ♦ **RIPEND-160.** The European one-way hash algorithm standard.
- ♦ **MD4.** An old, and obsolete (but still widely used), hash function developed by Ron Rivest.
- ♦ **MD5.** The latest in the series of hash functions developed by Ron Rivest. It involves several steps and results in a 128-bit message digest.

# Understanding Digital Certificates

**Objective****Certificates**

The biggest advantage to public key encryption is that it allows strangers to communicate with each other. This capability, however, also results in one of public key encryption's biggest weaknesses: people claiming to be people who they aren't.

In face-to-face transactions, we have safeguards to protect against identity theft or people who lie about their identities. We check drivers' licenses at liquor stores and we verify signatures at the bank. If you happen to be Little Red Riding Hood, you've learned to look at grandmother's teeth.

With public key encryption, however, anyone can create a public key, associate someone's (or some organization's) name with it, and claim to be that person or organization. Another person might then be fooled into giving confidential information or a basket of goodies to the wrong party. Public key encryption only works when users have a way to associate a public key with an identity.

As we mentioned earlier, one way to verify the identity of someone using a public key is to have a third party who is trusted by both parties certify that a user really is who he or she claims to be.

The trusted authority does this by creating a message, called a *certificate*, that contains the public key and the identity of its owner. The trusted authority then signs the certificate using its private key. As long as the trusted authority has done its job and verified the identity of the key owner, you can be sure that the owner of the certificate is who he or she claims to be.

## The X.509v3 standard

**Objective****X.509v3**

X.509v3 is the name of the standard that established the format and contents of the certificates that are most widely used on the Internet. The full name of the X.509v3 standard is the International Telecommunication Union Telecommunication Standardization Sector (ITU-T) Recommendation X.509v3. Table 19-2 shows the elements of certificates that are defined in the X.509v3 standard.

**Table 19-2**  
**X.509v3 Certificate Elements**

<i>Field</i>	<i>Description</i>
Version	The version number of the certificate. This is currently 1, 2, or 3.
Serial number	A unique serial number for the certificate file.
Signature algorithm ID	Indicates which message digest algorithm was used to sign the certificate file so it can be verified using the same message digest.
Issuer name	The name of the company that issued the certificate. For public certificates on the Internet, this is most often VeriSign or Thawte.
Validity period	The time during which the certificate is valid. The start date is the date the certificate was issued, and the end date is usually one year from the start date. Certificates cannot be used past their expiration dates.
Subject name	Contains the holder of the certificate's ID. For server certificates, the subject name might contain the name of the company and department that owns the Web site, the domain name of the company, the business city, state, and country.
Subject public key information	Contains the holder of the certificate's public key.
Issuer's unique identifier	Contains a unique number that identifies the issuer of the certificate.
Subject's unique identifier	Contains a unique number that identifies the holder of the certificate.
Extensions	Extensions were added to X.509 in version 3. Extensions can contain any type of information that the CA wants. This can include further information about the holder of the certificate, such as date of birth.
Signature	A cryptographic signature created using the contents of all of the previous fields. This is also known as the fingerprint or thumbprint.

## Types of certificates

There are four types of certificates:

- ♦ **Certificate authority certificates.** This can be thought of as the master certificate. This is owned by a trusted certificate provider, such as VeriSign or Thawte, and is used to sign other certificates.

- ♦ **Server certificates.** Server certificates are the certificates used to identify Web servers and their owners. Server certificates are necessary to use SSL.
- ♦ **Personal certificates.** Personal certificates are used to identify individuals.
- ♦ **Software publisher certificate.** Software authors use these certificates to sign software they distribute. A consumer of the software can then check the certificate to ensure that the software is from whom it claims to be from.

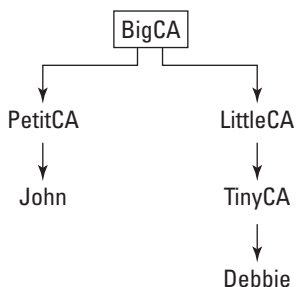
## Certifying Authorities

Third parties that issue digital certificates are called *Certifying Authorities (CAs)*. Anyone can be a CA and issue certificates. Certificates are often used within company networks to authenticate users. A network administrator, or anyone else for that matter, can issue these certificates. For certificates to be useful, however, the CA must be trusted by all of the parties involved in a transaction, and it must be possible for the signature on the certificate to be checked.

CAs can also issue certificates to other CAs. These CAs can then issue certificates to users. The end user only needs to include all of the CAs between himself or herself and the root in order to prove the legitimacy of the certificate.

### How CAs work

For example, assume that a CA, called BigCA, gives a certificate authority certificate to LittleCA. LittleCA may then give a certificate to TinyCA. TinyCA may then give a certificate to Debbie. BigCA may have also given a certificate to PetitCA, which may have then given a certificate to John. This relationship is shown in Figure 19-2.



**Figure 19-2:** A chain of certificates

Both Debbie and John share a common CA, BigCA. When Debbie gets a message from John, she can use BigCA's public key to verify PetitCA's identity, and then use PetitCA's identity to verify John's identity. This is called traversing the trust chain.

## Certificates on the Internet

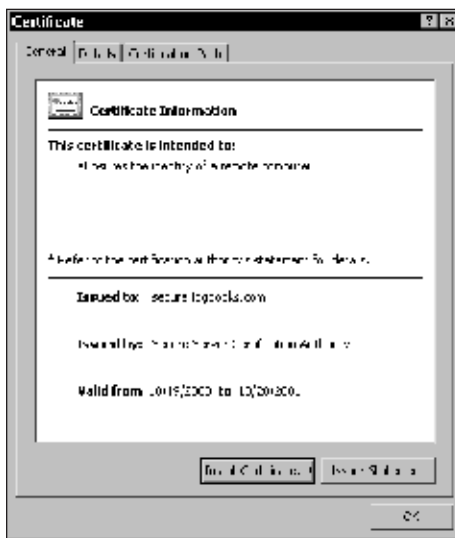
On the Internet, the vast majority of the server certificates are issued by VeriSign or Thawte Consulting. In December 1999, VeriSign acquired Thawte, but Thawte continues to issue certificates separately. The major Web browsers come with the public keys for both of these CAs. When you go to a secure Web site, your Web browser checks the site's certificate against the public keys it knows for the CA that issued the Web site's server certificate.

If everything checks out, you're shown a message or simply an icon, indicating that the connection is secure. If the public keys don't match or if the browser doesn't recognize the CA that issued the server certificate, you will get a different message that will indicate that something is not right about the certificate.

## Viewing certificate information

To view information about the certificate being used by a server, follow these steps:

1. Enter a secure area of an e-commerce site. You can usually get to a secure area by beginning the checkout process.
2. If you're using Internet Explorer, you see a lock icon in the lower right corner of your browser window. If you're using Netscape Navigator, the lock icon in the lower-left or lower-right corner of the screen is locked. In Internet Explorer, double-click the lock icon to open the certificate information window, as shown in Figure 19-3.



**Figure 19-3:** Internet Explorer's Certificate Information window

3. Use the tabs in the certificate information window to view the certificate details, and the certification path for a particular certificate, as shown in Figure 19-4.



Figure 19-4: Viewing the certification path

4. In Netscape Navigator, click the lock icon to view the security information window, as shown in Figure 19-5.

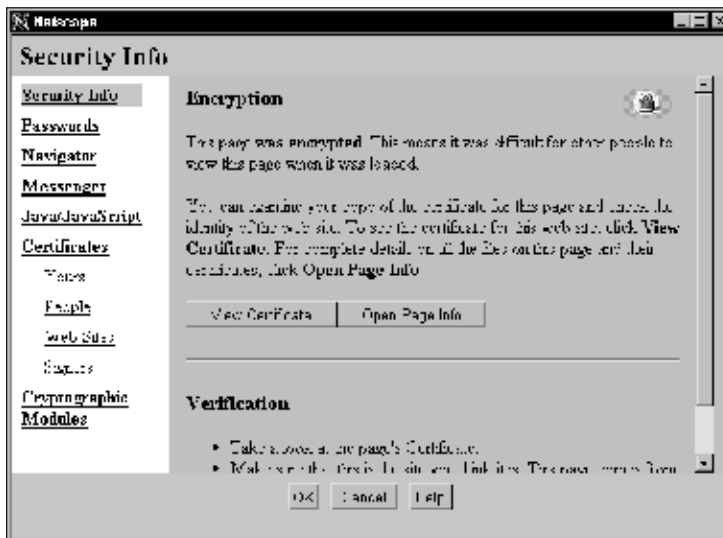


Figure 19-5: Netscape Navigator's Security Info window



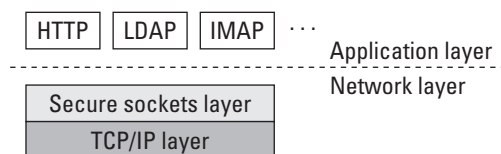
The SSL security protocol provides the following types of security for TCP/IP connections:

- ♦ Data encryption
- ♦ Server authentication
- ♦ Message integrity
- ♦ Optional client authentication

### How Internet data is transported

To understand SSL, it's important to understand how data is transmitted across the Internet. How data is transported and routed over the Internet is governed by the Transmission Control Protocol/Internet Protocol (TCP/IP). Other protocols, such as the Hypertext Transfer Protocol (HTTP), the Lightweight Directory Access Protocol (LDAP), the Internet Message Access Protocol (IMAP), and the File Transfer Protocol (FTP) run on top of TCP/IP. Protocols that operate on top of TCP/IP are said to be high-level protocols. The layer at which they operate is called the *Application layer*.

SSL operates below the Application layer, but above TCP/IP. What this means is SSL secures all of the traffic that operates at the Application layer. The relationship between TCP/IP, SSL, and the Application layer is shown in Figure 19-7.



**Figure 19-7:** SSL runs above TCP/IP, but underneath high-level protocols.

### Establishing an SSL connection

The steps involved in establishing an SSL connection are as follows:

1. The client (Web browser) sends a request to the secure server.
2. The server sends its certificate to the client. These first two steps are known as the *handshake*.
3. The client checks the certificate to make sure that it was issued by a trusted CA. If it was, the connection continues. If there's a problem with the certificate, the connection may be terminated. Alternatively, the browser may ask the user if he or she would like to proceed without authenticating the server.
4. The CA validates the server to the client.

5. The client tells the server what types of encryption it supports.
6. The server checks the list of ciphers that the browser sent and chooses the strongest form they have in common. The server then informs the client of this choice.
7. The client uses this cipher to generate a session key. A *session key* is a symmetric encryption key that is used only for this transaction. The client encrypts the symmetric key using the server's public key and sends it to the server.
8. The server decrypts the session key. The client and the server now share a symmetric key that they can use to exchange data securely.

**Exam Tip**

SSL uses both symmetric and asymmetric encryption.

The previous steps illustrate the basic process that is followed every time you visit a secure Web site. An optional step in the SSL protocol, which isn't used very often with public sites, is that the server may authenticate the client. This step requires the client, or visitor to the site, to have a personal certificate that identifies him or her. Because most people do not have personal digital certificates, requiring this step hampers the ability of a site to conduct e-commerce (although this step increases the level of security somewhat).

**Cross-Reference**

See the lab exercise in this chapter to learn how to get a personal certificate.

## Requesting a digital certificate

Certificates can currently be requested from VeriSign or Thawte, and must be renewed (for a fee) every year. The price for a certificate depends on the type of certificate and from whom you purchase it, as follows:

- ♦ Server certificates from VeriSign currently range from \$1000 to \$1300 for a commerce site. (VeriSign also allows you to request a trial certificate, which is valid for 14 days.)
- ♦ Thawte currently charges \$125 for a regular certificate and \$300 for a *SuperCert*, which enables strong encryption for international transactions.
- ♦ Discount certificates can also be obtained from Entrust ([www.entrust.com](http://www.entrust.com)). Entrust's certificate authority certificate is signed by Thawte.

To request a digital certificate, you first need to use your Web server to generate a certificate service request (CSR). To generate a CSR using Microsoft IIS on Windows 2000, follow these steps:

1. Launch the Internet Services Manager.



See Chapter 13 for more information on the Internet Services Manager and IIS.

2. Right-click on your default Web site in the Internet Services Manager and select Properties from the pop-up menu to open the Default Web Site Properties window.
3. Click the Directory Security tab to bring the Directory Security properties sheet to the front, as shown in Figure 19-8.



**Figure 19-8:** The Directory Security properties sheet

4. Click the Server Certificate button to start the IIS Certificate Wizard.
5. Read the first screen of this wizard, which contains basic information about the IIS Certificate Wizard, and then click Next.
6. The next screen asks you if you want to create a new certificate, assign a certificate to the server, or import a certificate from a backup. Choose to create a new certificate and click Next.
7. The next screen may give you the option of only preparing a certificate request, or both preparing a certificate request and sending it to a CA. This screen is shown in Figure 19-9. Choose to only prepare the request.



**Figure 19-9:** Screen asking whether to only prepare a request or prepare and submit a request

8. The next screen asks you for a name for your certificate and the length of the key you want the certificate to have. Enter a name for the certificate and select a bit length (these values don't matter much for this exercise). Click Next.
9. The wizard then asks for information about your organization. This is where you enter the name of the business or organization that owns the server. The information here must uniquely identify your organization. After you've completed this step, click Next.
10. Enter the name of the site. This is the domain name of your site, if it's connected to the Internet, or the name of your site on your local intranet. Click Next.
11. Enter the requested geographical information: city (or locality), state (or province), and country. The name of your state must not be abbreviated. For example, type **California**, not **CA** or **Calif**. Click Next.
12. Choose where to save the certificate request on your computer and click Next.
13. You're shown a summary of the information in your certificate request. At this point, you can either go back and make changes to this information or click Next to generate the request. If you're happy with the information you entered, click Next to generate the request.
14. The next screen informs you that the request has been generated, as shown in Figure 19-10. You may click the link in this window to view a security site at Microsoft.



Figure 19-10: The end of the IIS Certificate Wizard

15. To view your certificate request, locate it and open it using a text editor. It should look something like the certificate request shown in Figure 19-11.



Figure 19-11: A certificate request

## Submitting a Certificate Server Request file

To submit your CSR to VeriSign for a free trial certificate, follow these steps:

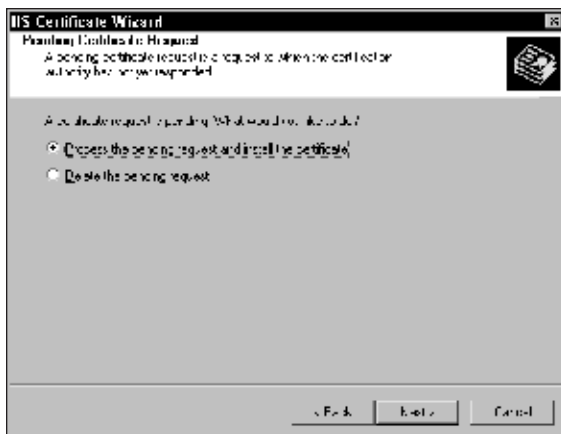
1. Go to <http://www.verisign.com>.
2. Select Get a server certificate from the drop-down menu or from the home page navigation. You should see a page called Web Site Trust Services.
3. Click the link to the free trial. This is currently labeled Try under the various available certificate packages. A form will appear.
4. After filling out this form, submit it.
5. When the Enrollment page is displayed, scroll to the bottom and click the green Continue button.
6. The next screen tells you to generate a CSR. You've already done this so click Next.
7. In the next screen, you're asked to paste your CSR into a form. To do this, open the file containing your CSR using a text editor and paste it into the form. After you've done this, click Continue.
8. If you generated a CSR with a key length of 512 bits, you see a warning message. You may ignore this warning or create a new CSR with a longer key length. When you are finished, click Continue.
9. You're then asked to verify the information from your CSR and to enter further information to complete the application. This screen also asks you to read the server agreement. Do this and then click Accept.
10. The next screen notifies you that you will be getting an e-mail containing the trial certificate.

Your trial certificate should arrive within an hour. When it does, you can use the instructions in the following section to install it on your server.

## Installing a digital certificate

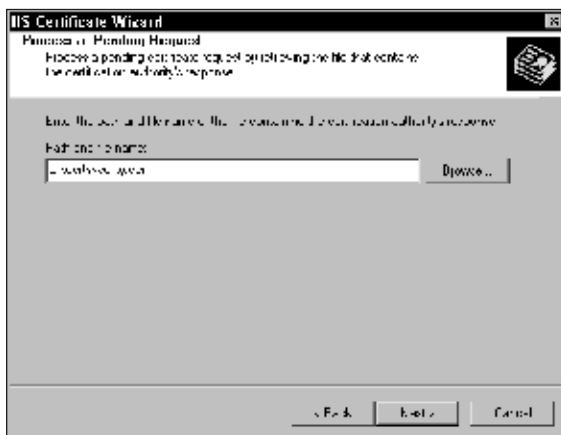
When the e-mail from VeriSign containing your trial certificate arrives, follow these steps to install it on your server:

1. Open the Web Site properties window and choose the Directory Security tab. Open the IIS Certificate Wizard.
2. Click Next. Because you already created a CSR and a key has been created on your server, you should see different choices on the second screen. These choices are Process the pending request and install the certificate, and Delete the pending request, as shown in Figure 19-12. Choose to install the certificate.



**Figure 19-12:** Install the certificate or delete the pending request.

3. Open the e-mail you received from VeriSign. Copy the certificate information from the e-mail and paste it into a text document. The certificate should start with a line that says START CERTIFICATE and end with a line that says END CERTIFICATE. Make sure that these lines are in the new text document you create. Save this document somewhere on your computer with the extension .cer.
4. In the IIS Certificate Wizard, locate the certificate file you just saved and select it, as shown in Figure 19-13.



**Figure 19-13:** Processing a pending certificate request

5. Click Next. You're shown the certificate information.
6. Click Next again to install the certificate and click Finish.

Your trial server certificate is now installed. To use this certificate, however, you need to install the Test CA Root on each browser you intend to use with the server on which you just installed the certificate. This is not required with regular certificates. To download and find out how to install the Test CA Root, go to <http://digitalid.verisign.com/server/trial/trialStep4.htm>.

## Securing a Directory or File in IIS

To secure a directory or file in IIS, follow these steps:

1. In the Internet Services Manager, open your Web site and highlight a directory or file you wish to secure or for which you want to create a new virtual directory.
2. Right-click on the directory or file, and select Properties. In the Properties window, bring the Directory Security Properties sheet to the front (or the File Security tab if you're securing a file).
3. Click the Edit button in the Secure Communications section of the window.
4. In the Secure Communications window, check the box next to Require secure channel (SSL), as shown in Figure 19-14.



**Figure 19-14:** Enabling SSL for a directory

5. Close the Secure Communications and Properties windows by clicking OK twice, and then close the Internet Services Manager to save the changes.

6. Attempt to access the directory or file you just created using `https://yourservername/directoryorfilename`. Note that you must use `https://` to access a directory secured with SSL.
7. If everything is correctly installed, you'll see an indication in your browser that the page you're viewing is secure.

## Using Secure Electronic Transactions

In 1995, an alliance of MasterCard, Netscape Communications, IBM, and others introduced the Secure Electronic Payment Protocol (SEPP). Several days later, Visa and Microsoft introduced the Secure Transaction Technology (STT). For a while, the two major credit card companies were each supporting different protocols for electronic payment.

In 1996, however, the two protocols were merged and became Secure Electronic Transactions (SET). The steps involved in an SET transaction are:

1. The card holder indicates that he or she wants to make a credit card purchase.
2. The merchant sends the buyer an invoice, the merchant certificate, and the merchant bank's certificate.
3. The card holder checks the certificates.
4. The card holder sends order information to the merchant, encrypted with the merchant's public key.
5. The merchant generates an authorization request and sends it to the merchant bank.
6. The merchant bank sends a request for payment authorization through an acquirer or other bank card channels.
7. The acquirer gets a response from the card holder's bank and sends a settlement response to the merchant's bank.
8. The card holder's bank authorizes (or denies) the payment and sends a response to the merchant.

It was thought that SET would eventually become the single payment standard on the Internet. However, SET never really caught on and, today, it's becoming apparent that it's not really needed.

**Exam Tip**

Although SET is not widely used, and it appears it won't ever be, there may still be a question or two about it on the exam.

## E-Commerce Security Issues

**Objective****Electronic commerce security myths**

To understand why security is so important to e-commerce, you need to understand the risks of doing business electronically. As an e-commerce merchant, you must protect your customers' personal and payment information from theft, and your site from attack. The survival of your company depends upon this.

Besides simply providing the best security you can, you also need to assure your customers that their information is safe on your site. Unfortunately, many Web users are extremely reluctant to reveal personal information on the Web. These fears, whether justified or not, can prevent visitors to your site from becoming customers.

To convince potential customers, you need an awareness of several commonly held beliefs about e-commerce security and the truths behind them. The following beliefs are discussed in detail in the following sections:

- ♦ Hackers can copy any credit card information transmitted across the Internet.
- ♦ The encryption used on the Internet can be easily broken.
- ♦ All you need to do to protect a Web site is install a digital certificate.
- ♦ It's impossible to secure a Web site.

**Exam Tip**

Prosoft's exam materials refer to the following statements as "Myths of Electronic Commerce."

### Hackers can copy any credit card information transmitted across the Internet

It's possible for a third-party to intercept and copy information transmitted between two computers on the Internet. This is done using programs called *packet sniffers*. Packet sniffers simply allow users to view all of the network traffic coming and going through their subnets. Essentially, packet sniffers allow you to see what everyone around you is viewing on the Internet. By filtering this information, hackers can locate particular pieces of information, such as credit card numbers or passwords.

However, a packet sniffer cannot specifically target any one person's credit card information for interception. A packet sniffer must be on the same wire as the target and must be listening at the right time. Because of the volume of data transmitted over the Internet, and because so much of it is not credit card information, this is not the best or most common way to steal information. In fact, the risk that a hacker will intercept your credit card information while it's in transit across the Internet is extremely small. Use of encryption for transmitting sensitive data easily defeats packet sniffers.

The biggest risk to users' personal information and payment information on the Internet is improperly protected databases residing on Web servers. Rather than wasting time filtering network packets looking for a few unencrypted passwords or credit card numbers, it's much easier for hackers to locate sites that don't properly secure their databases. Thousands of credit card numbers can be obtained from a single unsecured database.

## **The encryption used on the Internet can be easily broken**

The most common security protocol used on the Internet is Secure Sockets Layer (SSL). SSL 3.0 supports a variety of standard ciphers, including RC4, DES, and Triple DES, with key-lengths of between 40- and 168-bits. Given enough time, and enough computing power, any cipher (except a one-time pad) can be broken through a brute-force attack. However, the resources and time required to break any of the ciphers used by SSL are so great, it's not worth a criminal's time to try.

## **All you need to do to protect a Web site is install a digital certificate**

Digital certificates enable Web servers to use SSL. SSL provides authentication and encryption to data in transit over the Internet. It does not, however, protect data stored on the Web server.

As mentioned earlier, user data is in more danger once it arrives at an e-commerce site than it is when it's traveling across the Internet. How data is stored and used by an e-commerce merchant is at least as important as how it's transmitted. Unfortunately, typical users see a lock or key icon on their browsers (the indication that a site is using SSL) and assume it's safe to share personal data with the vendor. This is not always the case, as demonstrated by the many high-profile sites that have had their databases breached.

## **It's impossible to secure a Web site**

Absolute security cannot be achieved. You can, however, achieve a level of security high enough to make the effort required to penetrate it far outweigh the potential benefits of doing so.

Security can be expensive, but it's important to remember how much you stand to lose if your security is breached. The right security measures can be more than adequate against the types of attacks to which a site is likely to be subjected.

Some attacks, however, are harder to guard against. Most Web servers would not survive a direct attack from a terrorist wielding a hand grenade, although an attack from a dreaded insider or disgruntled former employee is more likely. This risk of physical damage to equipment and insider attack can be reduced with good policies and careful monitoring and auditing.

## Key Point Summary

In this chapter, you examined the important topic of e-commerce security. Security involves much more than just encryption and passwords. Making sure your e-commerce site is secure must be an on-going process involving authentication, integrity, cryptography, access control, auditing, and regular monitoring. Here's a review of the major points covered in this chapter:

- ♦ Security on the Internet serves five purposes: authentication and identification, access control, data confidentiality, data integrity, and non-repudiation.
- ♦ Authentication and identification are provided by digital certificates.
- ♦ Access control is most commonly provided using passwords.
- ♦ Data confidentiality is provided using cryptography.
- ♦ Data integrity is provided using message digests.
- ♦ Non-repudiation is provided using authentication with auditing.
- ♦ Encryption is the process of creating ciphertext from plaintext.
- ♦ Decryption is the process of creating plaintext from ciphertext.
- ♦ A cipher is a cryptographic algorithm used to encrypt and decrypt messages.
- ♦ Encryption strength depends on three factors: the strength of the algorithm, the secrecy of the key, and the length of the key.
- ♦ Symmetric encryption, or private key encryption, uses a single, shared key for both encryption and decryption.
- ♦ Examples of encryption algorithms that use symmetric encryption include: DES, RC4, RC5, Twofish, and AES.
- ♦ Asymmetric Encryption, or public key encryption, uses a public key for encryption and a private key for decryption. This allows strangers to exchange encrypted messages.
- ♦ One-way encryption is used to create digital fingerprints. One-way hash functions are a type of one-way encryption that is used to create message digests.
- ♦ MD4 and MD5 are examples of hash functions.
- ♦ Data can be intercepted on the Internet using packet sniffers. This is not the easiest or most common way to steal payment information, however.
- ♦ The goal of encryption is to make the resources and time required to break a code not worth the potential rewards for doing so.
- ♦ Web site security requires more than just installing a digital certificate.
- ♦ It's not impossible to achieve a high level of security.
- ♦ Certifying Authorities (CAs) act as trusted third parties.

- ♦ The standard that defines the format and contents of a digital certificate is X.509v3.
- ♦ The four types of digital certificates are certificate authority certificates, server certificates, personal certificates, and software publisher certificates.
- ♦ Certificates may be revoked. Certifying Authorities (CAs) maintain lists of revoked certificates.
- ♦ Secure Sockets Layer (SSL) is a protocol that provides data encryption, server authentication, message integrity, and optional client authentication for TCP/IP connections.
- ♦ To enable SSL on most Web servers, you just need to install a digital certificate.
- ♦ SSL uses symmetric and asymmetric encryption.
- ♦ Secure Electronic Transactions (SET), a protocol for conducting secure payments, was created as a joint effort between VISA and MasterCard.
- ♦ SET uses both Symmetric and Asymmetric encryption.



# STUDY GUIDE

---

E-commerce security is a large and complex subject. It's also heavily emphasized on the CIW E-Commerce exam. Make sure you know the answers to all of the assessment questions in the chapter and that you study the key points carefully.

## Assessment Questions

1. How is authentication provided in SSL?
  - A. Using digital certificates
  - B. Using symmetric encryption
  - C. Using access control
  - D. Using passwords
2. What is symmetric encryption?
  - A. A digital fingerprint
  - B. A type of encryption requiring both participants to share a secret key
  - C. A type of encryption requiring both participants to have a public key and a private key
  - D. A way to ensure a message is the same when it arrives as it was when it was sent
3. What is asymmetric encryption?
  - A. A digital fingerprint
  - B. A type of encryption requiring both participants to share a secret key
  - C. A type of encryption requiring both participants to have a public key and a private key
  - D. A way to ensure a message is the same when it arrives as it was when it was sent
4. What is a message digest?
  - A. A single message containing shortened versions of multiple other messages
  - B. An application of symmetric encryption
  - C. A type of encryption impossible to crack
  - D. An application of one-way encryption used to check data integrity

5. What type of digital certificate can be used to identify a person?
  - A. A certificate authority certificate
  - B. A server certificate
  - C. A personal certificate
  - D. A software publisher certificate
6. Which type of certificate can be used to sign other certificates?
  - A. A certificate authority certificate
  - B. A server certificate
  - C. A personal certificate
  - D. A software publisher certificate
7. Which security services are provided by Secure Sockets Layer (SSL)?
  - A. Encryption and credit card authorization
  - B. Encryption and access control
  - C. Auditing and authentication
  - D. Encryption and authentication
8. Which of the following best describes the steps involved in establishing an SSL connection?
  - A. Handshake, client checks the server's certificate, client and server agree on a symmetric encryption algorithm, client sends session key encrypted using server's public key, data is exchanged using the session key
  - B. Handshake, client and server agree on a symmetric encryption algorithm, client checks the server's certificate, the client sends a session key encrypted using the server's public key, data is exchanged using the session key
  - C. The client checks the server's certificate, handshake, the client sends a session key encrypted using the server's public key, client and server agree on a symmetric encryption algorithm, data is exchanged using the session key
  - D. The client sends a session key encrypted using the server's public key, client and server agree on a symmetric encryption algorithm, client checks the server's certificate, handshake, data is exchanged using the session key

9. Which of the following statements is not true?
- A. It's possible to achieve a good level of security for an e-commerce site.
  - B. Hackers can intercept and use passwords and credit card numbers that aren't encrypted.
  - C. Installing a digital certificate and using SSL guarantees that your e-commerce site is secure.
  - D. Most of the types of encryption used on the Internet cannot be easily broken by most people or organizations.

## Scenarios

1. Security of cryptography is partially based on the length of the key. As you learned in this chapter, the number of possible keys can be calculated using  $2^n$ , where  $n$  is the length of the key, in bits. This same formula works for any other type of key. For example, if a lock has 5 pins, and each pin has 10 possible settings, the lock has  $5^{10}$ , or 100,000 possible combinations. In this scenario, do the following:
- Calculate the number of possible combinations for several cryptographic algorithms. In the first column of Table 19-3, you find the name of an algorithm; in the second column, you find the key length (in bits).
  - Write the number of possible keys in the third column.

**Table 19-3**  
**Numbers of Possible Key Combinations**

<i>Algorithm</i>	<i>Key length (bits)</i>	<i>Number of combinations</i>
DES	56	
RC5-32	32	
RC5-64	64	
RSA	528	

2. It's usually much more likely someone will guess a password, and thus gain access to a private key, than they will guess a key. Given this assumption, perform the following:
- Assume (for simplicity) that a password only uses lowercase letters, isn't necessarily a word, and is six characters long. Calculate the number of possible passwords.

- Even though this number shows that passwords are generally much less secure than the keys they protect, why is this number essentially meaningless?

## Lab Exercise

### Lab 19-1 Obtaining a personal digital certificate

1. Go to <http://www.thawte.com/certs/personal/contents.html>.
2. Click the Enroll button.
3. Read the terms of use carefully. If you agree with these terms, click the Next button at the bottom of the screen.
4. Enter the information about your identity that is requested on the next few screens.
5. On the screen that asks for your password, read the information carefully and choose a good password you won't forget.
6. When you finish the first step of the enrollment process, an e-mail will be sent to you. Wait for this e-mail.
7. When you get the e-mail from Thawte, go to the address it contains and enter the Probe and Ping values that were sent to you. You're now registered with Thawte and can obtain a personal digital certificate.
8. To obtain a certificate, log-in by clicking the Next button and entering your ID and password.
9. Select the software you want to use the certificate with from the list, and then click Get X.509 Certificate.
10. Click Next on the screen that asks about the Common name and Employment.
11. Select the e-mail addresses you wish to include on your certificate and then click Next.
12. Click Next until you get to the screen that asks if you'd like to configure the optional certificate extensions. Choose Accept Default Extensions.
13. Read the instructions for generating a private key.
14. Review the final information. If you're happy with it, click Finish. A certificate request is generated, and you're notified via e-mail when the certificate is ready. After you have the certificate, you can install it in your e-mail program or Web browser.

# Answers to Chapter Questions

## Chapter pre-test

1. The five purposes served by security in electronic commerce are:
  - Authentication and identification
  - Access control
  - Data confidentiality
  - Data integrity
  - Non-repudiation
2. Non-repudiation assures that every action is provable. It's provided using authentication and auditing.
3. Symmetric encryption requires participants to share a key. Asymmetric encryption doesn't require a shared key.
4. A message digest is the result of using a hash function on a plaintext message. It can be used to provide authentication and integrity.
5. Certifying Authorities (CAs) act as trusted third parties for digital certificates.
6. The four types of digital certificates are: certificate authority certificates, server certificates, personal certificates, and software publisher certificates.
7. SSL, or Secure Sockets Layer, is a protocol used to secure TCP/IP connections on the Internet.
8. X.509 is the standard that specifies the format and content of digital certificates.

## Assessment questions

1. **A.** SSL uses digital certificates. Digital certificates provide authentication and identification through the use of a trusted third party (see "Establishing identification and authentication").
2. **B.** Symmetric encryption is also called shared key encryption or secret key encryption (see "Symmetric encryption").
3. **C.** Asymmetric encryption does not require the use of a shared key. It requires a public and a private key (see "Asymmetric encryption").
4. **D.** Message digests are created using hash functions, which are used in one way encryption (see "One-way encryption").
5. **C.** Personal digital certificates identify individuals (see "Understanding Digital Certificates").

6. **A.** In order to issue certificates, you must have a certificate authority certificate (see “Understanding Digital Certificates”).
7. **D.** SSL provides encryption using symmetric and asymmetric encryption, and authentication using digital certificates (see “SSL”).
8. **A.** The first step in establishing an SSL connection between a server and a client is the handshake. During the handshake, the client sends a request to the browser (using https) and the server sends its certificate to the client (see “SSL”). The other steps are as follows:
  - The client then checks the certificate.
  - The client then informs the server of the types of symmetric encryption it supports. The server chooses the strongest type of encryption it has in common with the client and informs the client of this choice.
  - The client generates a session key using this type of encryption. The session key is sent to the server using the server’s public key.
  - The server decrypts the session key, and begins communicating with the client using this key.
9. **C.** There are no guarantees of security. Using SSL and digital certificates is only part of the solution. To be fairly sure your site is secure, you need to make sure the operating system is secure, passwords are secure, and the database is secure. You also need to regularly monitor your site and keep abreast of current security issues and new server vulnerabilities as they are discovered (see “E-Commerce Security Issues”).

## Scenarios

1. To give you an idea of just how large the numbers in Table 19-3 are, consider the number of possible keys for RC5-64 (see the completed table shown here). If you were able to try a billion keys per year, it would take you 50 million years to try all of them. Distributed.net (<http://distributed.net/>) is currently punching away at the possible keys at a rate much faster than that — at over 100 billion keys per second. At that rate, it will still take nearly six years for distributed.net to try all of the possible combinations.

<i>Algorithm</i>	<i>Key length (bits)</i>	<i>Number of combinations</i>
DES	56	$7.2 \times 10^{16}$
RC5-32	32	4,294,967,296
RC5-64	64	$1.8 \times 10^{19}$
RSA	528	$8.8 \times 10^{158}$

2. The number of possible combinations of 6 letters (where case doesn't matter) is  $26^6$ , or 308,915,776. This is much smaller than any of the numbers in Table 19-3 from the first scenario. Now, consider the following:
- The reason this number is irrelevant, however, is that people don't simply choose random letters for passwords. Most often, people choose passwords that mean something to them, such as their dog's name, a word, a date, a favorite place, or a favorite food. The easiest way to crack a password is simply to find out something about the person who created the password.
  - In addition, people often store their passwords close to, or on, their computers. Rather than writing a program to try each possible password to a system, it's often possible to simply look in a desk drawer or under the keyboard.

## For More Information . . .

- ♦ **Crypto-Gram Newsletter.** <http://www.counterpane.com/crypto-gram.html>. A free monthly e-mail newsletter from Bruce Schneier
- ♦ **SSL 3.0 Specification.** <http://home.netscape.com/eng/ssl3/index.html>
- ♦ **Generating a CSR and key pair with Windows 2000 / Internet Information Services 5.0.** [www.entrust.net/tech/miis50/csr.htm](http://www.entrust.net/tech/miis50/csr.htm)
- ♦ **W3C Security Resources.** [www.w3.org/Security/](http://www.w3.org/Security/)
- ♦ **Computer Security Resource Center (CSRC).** <http://csrc.nist.gov>
- ♦ **Microsoft Security.** [www.microsoft.com/security](http://www.microsoft.com/security)
- ♦ **CERT Coordination Center.** [www.cert.org](http://www.cert.org)
- ♦ **Rootshell.** [www.rootshell.org](http://www.rootshell.org)

# Managing E-Business Information

---

## EXAM OBJECTIVES

- ◆ Integrating information systems
- ◆ Oracle and SQL Server relational databases
- ◆ Order tracking

# 20

CHAPTER

