UNIT-4 : CRYPTOGRAPHY

LEARNING OBJECTIVES:

Upon completion of this material, you should be able to:

- Describe the most significant events and discoveries from the history of cryptology .
- Understand the basic principles of cryptography
- Understand the operating principles of the most popular tools in the area of cryptography
- List and explain the major protocols used for secure communications
- Understand the nature and execution of the dominant methods of attack used against cryptosystems.

INTRODUCTION

The science of cryptography is not as enigmatic as you might think. A variety of techniques related to cryptography are used regularly in everyday life. For example, open your newspaper to the entertainment section and you'll find the daily cryptogram, which is a word puzzle that makes a game out of unscrambling letters' to find a hidden message. Also, although it is a dying art, many secretaries still use stenography, a coded form of documentation, to take rapid dictation from their managers. Finally, a form of cryptography is used even in the hobby of knitting, where directions are written in a coded form, in such patterns as KIPI (knit I, pearl I), that only an initiate would be able to understand. Most of the examples above demonstrate the use of cryptography as a means of efficiently and rapidly conveying information. These aspects are only one important element of the science of cryptography. For the purposes of this chapter, the discussion of cryptography will be expanded to include the protection and verification of transmitted information.

In order to understand cryptography and its uses, you must become familiar with a number of key terms that are used across the information technology industry. The science of encryption, known as **cryptology**, encompasses *cryptography* and *cryptanalysis.* **Cryptography**, which comes from the Greek words *kryptos,* meaning "hidden," and *graphein,* meaning "to write," is the process of making and using codes to secure the transmission of information. Cryptanalysis is the process of obtaining the original message (called the **plaintext**) from an encrypted message (called the **ciphertext**) without knowing the algorithms and keys used to perform the encryption. **Encryption** is the process of converting an original message into a form that is unreadable to unauthorized individuals-that is, to anyone without the tools to convert the

encrypted message back to its original format. **Decryption** is the process of converting the ciphertext into a message that conveys readily understood meaning.

The field of cryptology is so complex it can fill many volumes. *As* a result, this textbook seeks to provide only the most general overview of cryptology and some limited detail on the tools of cryptography. The early sections of this chapter, namely "A Short History of Cryptology,"Principles of Cryptography," and "Cryptography Tools," provide some background on cryptology and general definitions of the key concepts of cryptography, and discuss the usage of common cryptographic tools. Later sections discuss common cryptographic protocols and describe some of the attacks possible against cryptosystems.

A Short History of Cryptology

The creation and use of cryptology has a long history among the cultures of the world.

Table 8.1 provides an overview of the history of cryptosystems.

TABLE 8-1 History of Cryptology

1900 B.C	Egyptian scribes used nonstandard hieroglyphs while inscribing clay tablets; this is the first documented use of written cryptography.
1500 B.C	Mesopotamian cryptography surpassed that of the Egyptians. This is demonstrated in a tablet that was discovered to contain an encrypted formula for pottery glazes; the tablet used special symbols that appear to have different meanings from the usual symbols used elsewhere.
500 B.C	Hebrew scribes writing the book of Jeremiah used a reversed alphabet substitution cipher known as the ATBASH.
487 B.C	The Spartans of Greece developed the Skytale, a system consisting of a strip of papyrus wrapped around a wooden staff. Messages were written down the length of the staff, and the papyrus was unwrapped. The decryption process involved wrapping the papyrus around a shaft of similar diameter.
50 B.C	Julius Caesar used a simple substitution cipher to secure military and government communications. To form an encrypted text, Caesar shifted the letter of the alphabet three places. In addition to this monoalphabetic substitution cipher, Caesar strengthened his encryption by substituting Greek letters for Latin letters.

725 Abu 'Abd al-Rahman al-Khalil ibn Ahman ibn 'Amr ibn Tammam al Farahidi al-Zadi at Yahmadi wrote a text (now lost) of cryptography; he also solved a Greek cryptogram by guessing the plaintext introduction.

855 Abu Wahshiyyaan-Nabati, a scholar, published several cipher alphabets that were used for encrypted writings of magic formulas.

1250 Roger Bacon, an English monk, wrote Epistle of Roger Bacon on the Secret Works of Art *and* of Nature and Also on the Nullity of Magic, in which he described several simple ciphers.

1392 The Equatorie of the Planetis, an early text possibly written by Geoffrey Chaucer, contained a passage in a simple substitution cipher.

1412 Subhalasha, a 14-volume Arabic encyclopedia, contained a section on cryptography, including both substitution and transposition ciphers, and ciphers with multiple substitutions, a technique that had never been used before.

1466 Leon Battista Alberti is considered the Father of Western cryptography because on his work with polyalphabetic substitution; he also designed a cipher disk.

1518 Johannes Trithemius wrote the first printed book on cryptography and invented a steganographic cipher, in which each Jetter was represented as a word taken from a succession of columns. He also described a polyalphabetic encryption method using a rectangular substitution format that is now commonly used. He is credited with the introduction of the method of changing substitution alphabets with each letter as it is deciphered.

1553 Giovan Batista Belaso introduced the idea of the passphrase (password) as a key for encryption; this polyalphabetic encryption method is misnamed for another person who later used the technique and thus is called "The Vigenere Cipher" today.

1563 Giovanni Battista Porta wrote a classification text on encryption methods, categorizing them as transposition, substitution, and symbol substitution.

1623 Sir Francis Bacon described an encryption method by employing one of the first uses of steganography; he encrypted his messages by slightly changing the type face of a random text so that each letter of the cipher was hidden within the text's letters.

1780s 1homas Jefferson created a 26-letter wheel cipher, which he used for official communications while ambassador to France; the concept of the wheel cipher would be reinvented in 1854, and again in 1913.

1854 Charles Babbage appears to have reinvented Thomas Jefferson's wheel cipher.

1861-5 During the U.S. Civil War, Union forces used a substitution encryption method based on specific words, and the Confederacy used a polyalphabetic cipher whose solution had been published before the start of the Civil War.

1914-17 World War I: The Germans, British, and French used a series of transposition and substitution ciphers in radio communications throughout the war. All sides spent considerable effort in trying to intercept and decode communications, and thereby brought about the birth of the science of cryptanalysis. British cryptographers broke the Zimmerman Telegram, in which the Germans offered Mexico U.S. territory in return for Mexico's support. This decryption helped to bring the United States into the war.

1917 William Frederick Friedman, the father of U.S. cryptanalysis, and his wife Elizabeth, were employed as civilian cryptanalysts by the U.S. government. Friedman later founded a school for cryptanalysis in Riverbank, Illinois.

1917 Gilbert S. Vernam, an AT&T employee, invented a polyalphabetic cipher machine that used a non-repeating random key.

1919 Hugo Alexander Koch filed a patent in the Netherlands for a rotor-based cipher machine; in 1927, Koch assigned the patent rights to Arthur Scherbius, the inventor of the Enigma Machine, which was a mechanical substitution cipher.

1927-33 During Prohibition, criminals in the U.S. began using cryptography to maintain the privacy of messages used in criminal activities.

1937 The Japanese developed the Purple machine, which was based on principles similar to those of Enigma and used mechanical relays from telephone systems to encrypt diplomatic messages.

By late 1940, a team headed by William Friedman had broken the code generated by this machine and constructed a machine that could quickly decode Purple's ciphers.

1939-42 The fact that the Allies secretly broke the Enigma cipher undoubtedly

shortened World War II.

1942 Navajo *Windtalkers* entered World War II; in addition to speaking a language that was unknown outside a relatively small group within the United States, the Navajos developed code words for subjects and ideas that did not exist in their native tongue.

1948 Claude Shannon suggested using frequency and statistical analysis in the solution of substitution ciphers.

1970 Dr. Horst Feistelled an IBM research team in the development of the Lucifer cipher.

1976 A design based upon Lucifer was chosen by the U.S. National Security Agency as the Data Encryption Standard and found worldwide acceptance.

1976 Whitefield Diffie and Martin Hellman introduced the idea of public key cryptography.

1977 Ronald Rivest, Adi Shamir, and Leonard Adleman developed a practical public key cipher for both confidentiality and digital signatures; the RSA family of computer encryption algorithms was born.

1978 The initial RSA algorithm was published in the Communication of ACM.

1991 Phil Zimmermann released the first version ofPGP (Pretty Good Privacy); PGP was released as freeware and became the worldwide standard for public cryptosystems.

2000 Rijndael's cipher was selected as the Advanced Encryption Standard.

Principles of Cryptography

Historically, cryptography was used in manual applications, such as handwriting. But with the emergence of automated technologies in the 20th century, the need for encryption in the IT environment vastly increased. Today, many common IT tools use embedded encryption technologies to protect sensitive information within applications. For example, all the popular Web browsers use built -in encryption features that enable users to perform secure e-commerce applications, such as online banking and Web shopping.

Basic Encryption Definitions

To understand the fundamentals of cryptography, you must become familiar with the following definitions:

- **Algorithm:** The programmatic steps used to convert an unencrypted message into an encrypted sequence of bits that represent the message; sometimes used as a reference to the programs that enable the cryptographic processes

- **Cipher or cryptosystem:** An encryption method or process encompassing the algorithm, key(s) or cryptovariable(s), and procedures used to perform encryption and decryption
- **Ciphertext or cryptogram:** The unintelligible encrypted or encoded message resulting from an encryption
- **Code:** The process of converting components (words or phrases) of an unencrypted message into encrypted components
- **Decipher:** To decrypt or convert ciphertext into the equivalent plaintext
- **Encipher:** To encrypt or convert plaintext into the equivalent ciphertext
- **Key or cryptovariable:** The information used in conjunction with an algorithm to create the ciphertext from the plaintext or derive the plaintext from the ciphertext; the key can be a series of bits used by a computer program, or it can be a passphrase used by humans that is then converted into a series of bits for use in the computer program
- **Keyspace:** The entire range of values that can possibly be used to construct an individual key
- **Link encryption:** A series of encryptions and decryptions between a number of systems, wherein each system in a network decrypts the message sent to it and then reencrypts it using different keys and sends it to the next neighbor, and this process continues until the message reaches the final destination
- **Plaintext or cleartext:** The original unencrypted message that is encrypted; also the name given to the results of a message that has been successfully decrypted
- **Steganography:** The process of hiding messages-for example, messages can be hidden within the digital encoding of a picture or graphic
- **Work factor:** The amount of effort (usually in hours) required to perform cryptanalysis on an encoded message so that it may be decrypted when the key or algorithm (or both) are unknown

Cipher Methods

A plaintext can be encrypted through one of two methods, the bit stream method or the block cipher method. With the bit stream method, each bit in the plaintext is transformed into a cipher bit one bit at a time. In the case of the block cipher method, the message is divided into blocks, for example, sets of 8-,16-,32-, or 64-bit blocks, and then each block of plaintext bits is transformed into an encrypted block of cipher bits using an algorithm and a key. Bit stream methods most commonly use algorithm functions like the exclusive OR operation (XOR), whereas block methods can use substitution, transposition, XOR, or some combination of these operations, as described in the following sections. As you read on, you should note that most encryption methods using computer systems

will operate on data at the level of its binary digits (bits), but SOme operations may operate at the byte or character level.

Elements of Cryptosystems

Cryptosystems are made up of a number of elements or components. These are usually algorithms and data handling techniques as well as procedures and process steps, which are combined in multiple ways to meet a given organization's need to ensure confidentiality and provide specialized authentication and authorization for its business processes. In the sections that follow, you will first read about the technical aspects of a number of cryptographic techniques, often called ciphers. The chapter will continue with an exploration of some of the tools commonly used to implement cryptographic systems in the world of business. The discussion will then proceed to the security protocols used to bring communications security to the Internet and the world of e-commerce. Finally, the chapter will conclude with a discussion of the attacks that are often found being used against cryptosystems. Along the way, you will also encounter a number of Technical Details boxes that cover advanced material. Be sure to check with your instructor about how your course will include the Technical Details material.

Substitution Cipher

When using a **substitution cipher,** you substitute one value for another. For example, you can substitute a letter in the alphabet with the letter three values to the right. Or, you may substitute one bit for another bit that is four places to its left. A three-character substitution to the right would result in the following transformation of the standard English alphabet:

Initial alphabet ABCDEFGHIJKLMNOPQRSTUVWXYZ

yields

Encryption alphabet DEFGHIJKLMNOPQRSTUVWXYZABC

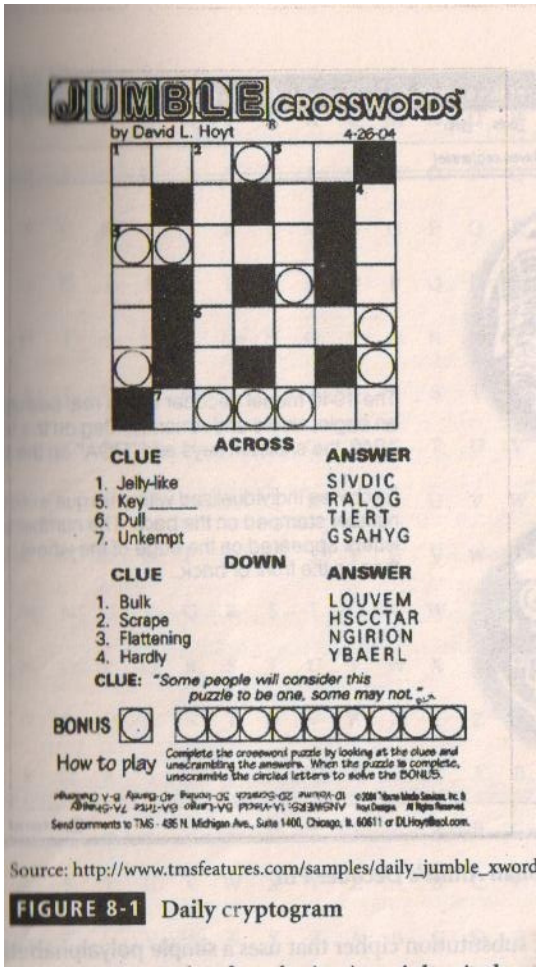
Within this substitution scheme, the plaintext MOM would be encrypted into the ciphertext PRP.

This is a simple enough method by itself but very powerful if combined with other operations. Incidentally, this type of substitution is based on a monoalphabetic substitution, since it only uses one alphabet. More advanced substitution ciphers use two or more alphabets, and are referred to as polyalphabetic substitutions.

To continue the previous example, consider the following block of text:

Plaintext = ABCDEFGHIJKLMNOPQRSTUVWXYZ 1st row
Substitution cipher 1 = DEFGHIJKLMNOPQRSTU~YZABC *2nd*
Substitution cipher 2 = GHIJKLMNOPQRSTUVWXYZABCDEF 3rd
Substitution cipher 3 = JKLMNOPQRSTUVWXYZABCDEFGHI 4th
Substitution cipher 4 = MNOPQRSTUVWXYZABCDEFGHIJKL 5th

The first row here is the plaintext, and the next four rows are four sets of substitution ciphers, which taken together constitute a single polyalphabetic substitution cipher. To encode the word TEXT with this cipher, you substitute a letter from the second row for the first letter in TEXT, a letter from the third row for the second letter, and so on-a process that yields the ciphertext WKGF. Note how the plaintext letter T is transformed into a W or a F, depending on its order of appearance in the plaintext. Complexities like these make this type of encryption substantially more difficult to decipher when one doesn't have the algorithm (in this case, the rows of ciphers) and the key, which is the method used (in this case the use of the second row for first letter, third for second, and so on). A logical extension to this process would be to randomize the cipher rows completely in order to create a more complex operation.



JUMBLE CROSSWORDS®
by David L. Hoyt 4-26-04

ACROSS

CLUE	ANSWER
1. Jelly-like	SIVDIC
5. Key _____	RALOG
6. Dull	TIERT
7. Unkempt	GSAHYG

DOWN

CLUE	ANSWER
1. Bulk	LOUVEM
2. Scrape	HSCCTAR
3. Flattening	NGIRION
4. Hardly	YBAERL

CLUE: "Some people will consider this puzzle to be one, some may not."

BONUS

How to play Complete the crossword puzzle by looking at the clues and unscrambling the answers. When the puzzle is complete, unscramble the circled letters to solve the BONUS.

Send comments to TMS · 435 N. Michigan Ave., Suite 1400, Chicago, IL 60611 or DLHoyt@aol.com.

Source: http://www.tmsfeatures.com/samples/daily_jumble_xword

FIGURE 8-1 Daily cryptogram

One example of a substitution cipher is the cryptogram in the daily newspaper (see Figure 8-1); another is the once famous *Radio Orphan Annie decoder pin* (shown in Figure 8-2), which consisted of two alphabetic rings that could be rotated to a predetermined pairing to form a simple substitution cipher. The device was made to be worn as a pin so one could always be at the ready. As mentioned in Table 8-1, Caesar reportedly used a three-position shift to the right to encrypt his messages (so A became D, B became E, and so on), thus this particular substitution cipher was given his name-the *Caesar Cipher.*

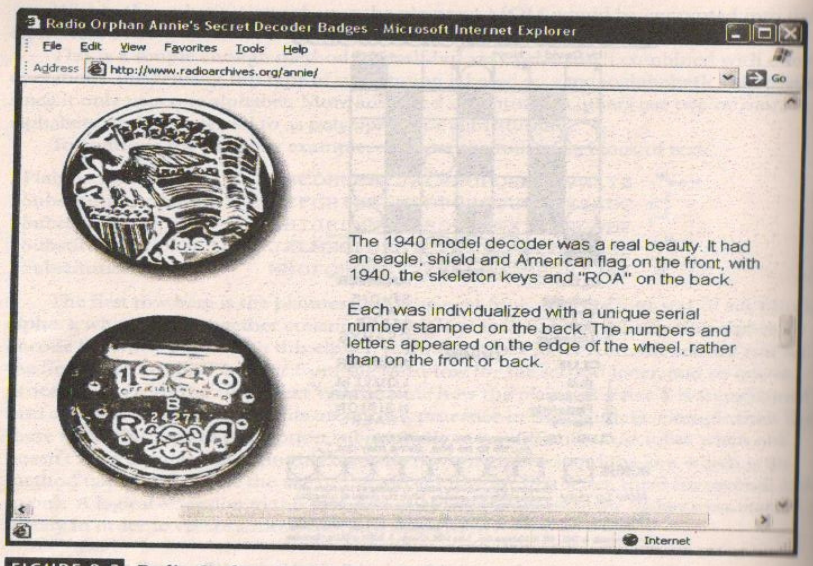


FIGURE 8-2 Radio Orphan Annie's Decoder Pin

An advanced type of substitution cipher that uses a simple polyalphabetic code is the Vigenere cipher. The cipher is implemented using the Vigenere Square. which is made up of 26 distinct cipher alphabets. Table 8-2 illustrates the setup of the Vigenere Square. In the header row, the alphabet is written in its normal order. In each subsequent row, the alphabet is shifted one letter to the right until a 26 X 26 block of letters is formed. There are a number of ways to use the Vigenere square. You could perform an encryption by simply starting in the first row and finding a substitute for the first letter of plaintext, and then moving down the rows for each subsequent letter of plaintext. With this method, the word SECURITY in plaintext would become TGFYWOAG in ciphertext.

TABLE 8-2 The Vigenère Square

	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
1	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
2	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
3	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
4	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
5	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
6	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
7	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
8	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
9	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
10	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
11	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
12	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
13	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
14	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
15	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
16	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
17	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
18	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
19	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
20	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
21	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
22	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
23	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
24	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
25	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

A much more sophisticated way to use the Vigenere Square would be to use a keyword to represent the shift. To accomplish this, you would begin by writing a keyword above the

plaintext message. For example, suppose the plaintext message was "SACK GAUL SPARE NO ONE" and the keyword was ITALY. We thus end up with the

Following :

ITALYITALYITALYITA
SACKGAULSPARENOONE

The idea behind this is that you will now use the keyword letter and the message (plaintext)letter below it in combination. Returning to the Vigenere Square, notice how the first column of text, like the first row, forms the normal alphabet. To perform the substitution of the message, start with first combination of keyword and message letters, IS. Use the keyword letter to locate the column, and the message letter to find the row, and then look for the letter at their intersection. Thus, for column "I" and row "S," you will find the ciphertext letter "JI". After you follow this procedure for each of the letters in the message, you will produce the encrypted ciphertext ATCVEINLDNIKEYMWGE. Curiously, one weakness of this method is that any keyword-message letter combination containing an *"N'* row or column will reproduce the plaintext message letter. For example, the third letter in the plaintext message, the C (of SACK), has a combination of AC, and thus is unchanged in the ciphertext. To minimize the effects of this weakness, you should avoid choosing a keyword that contains the letter "A."

Transposition Cipher

The next type of cipher operation is the transposition. Just like the substitution operation, the transposition cipher is simple to understand, but it can, if properly used, produce ciphertext that is complex to decipher. In contrast to the substitution cipher, however, the **transposition cipher** (or **permutation cipher)** simply rearranges the values within a block to create the ciphertext. This can be done at the bit level or at the byte (character) level. For an example, consider the following transposition key pattern.

Key pattern:

1-4, 2-8, 3-1, 4-5, 5-7, 6-2, 7-6, 8-3

In this key, the bit or byte (character) in position 1 (with position 1 being at the far *right)* moves to position 4 (counting from the right), and the bit or byte in position 2 moves to position 8, and so on.

The following rows show the numbering of bit locations for this key; the plaintext message 00100101011010111001010101010100, which is broken into 8-bit blocks for ease of discussion; and the ciphertext that is produced when the transposition key depicted above is applied to the plaintext:

Bit locations:	87654321	87654321	87654321	87654321
Plaintext 8-bit blocks:	00100101	01101011	10010101	01010100
Ciphertext:	00001011	10111010	01001101	01100001

Reading from right to left in the example above, the first bit of plaintext (position 1 of the first byte) becomes the fourth bit (in position 4) of the first byte of the ciphertext. Similarly, the second bit of the plaintext (position 2) becomes the eighth bit (position 8) of the ciphertext, and "so on.

To examine further how this transposition key works, let's see its effects on a plaintext message comprised of letters instead of bits. Replacing the 8-bit block of plaintext with the example plaintext message presented earlier, "SACK GAUL SPARE NO ONE," yields the following:

Letter locations:	87654321	87654321	87654321	87654321
Plaintext:	SACKGAUL	SPARENOO	N E	
Key:	Same key as above, but characters transposed, not bits.			

Ciphertext: UKAGLSCA ORPEOSAN E N

Here, reading again from right to left, the letter in position I of the first block of plaintext, "I.:', becomes the letter at position 4 in the ciphertext. In other words, the "L" that is the 8'h letter of the plaintext is the "L" at the 5th letter of the ciphertext. The letter in position 2 of the first block of plaintext, "U': becomes the letter at position 8 in the ciphertext. In other words, the "U" that is the 7th letter of the plaintext is the "U" at the 151 letter of the ciphertext. This process continues using the specified pattern.

In addition to being credited with inventing a substitution cipher, Julius Caesar was associated with an early version of the transposition cipher. As part of the Caesar block cipher, a courier would carry a message that when read normally would be unintelligible. However, the receiver of the message would know to fit the text to a prime number square (in practice, this meant that if there were fewer than 25 characters, the receiver would use a 5 x 5 square). For example,suppose you were the receiver and the ciphertext shown below arrived at your doorstep. Since it was from Caesar, you would know to make a square of 5 columns and 5 rows, and then to write the letters of the message into the square, filling the slots from left to right, top to bottom. Also, when you'd finished doing this, you'd know to read the message the opposite direction-that is, from top to bottom, left to right.

Ciphertext:

SGS-NAAPNECUAO KLR EO
S G S - N
A A P N E
C U A 0
K L R_- -
-_-_E O -

Reading from top to bottom, left to right reveals the plaintext "SACK GAUL SPARE NO ONE':

When mechanical and electronic cryptosystems became more widely used, transposition ciphers and substitution ciphers began to be used in combinations to produce highly secure encryption processes. To make the encryption even stronger (more difficult to cryptanalyze) the keys and block sizes can be made much larger (up to 64 or 128 bits in size), which produces substantially more complex substitutions or transpositions.

Exclusive OR

The **exclusive OR operation** (XOR) is a function of Boolean algebra in which two bits are compared, and if the two bits are identical, the result is a binary O. If the two bits are not the same, the result is a binary 1. XOR encryption is a very simple symmetric cipher that is used in many applications where security is not a defined requirement Table 8-3 shows a truth table for XOR with the results of all the possible combinations of two bits.

TABLE 8-3 XOR Truth Table

First Bit	Second Bit	Result
0	0	0
0	1	1
1	0	1
1	1	0

To see how XOR works, let's consider an example in which the plaintext we will start with is the word "CAT': The binary representation of the plaintext is "0 1110000 01100101 1000000". In order to encrypt the plaintext, a key value should be selected. In this case, the bit pattern for the letter "Y" (10000101) will be used and repeated for each character to be encrypted. Performing the XOR operation on the two bit streams (the plaintext and the key) will produce the

following result:

TABLE 8-4 Example XOR Encryption

CAT as bits	0	1	1	1	0	0	0	0	0	1	1	0	0	1	0	1	1	0	0	0	0	0	0	0
VVV as key	1	0	0	0	0	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0	1	0	1
Cipher	1	1	1	1	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	1

The row of Table 8-4 labeled "Cipher" contains the bit stream that will be transmitted; when this cipher is received, it can be decrypted using the key value of "y"'. Note that the XOR encryption method is very simple to implement and equally simple to break. The XOR encryption method should not be used by itself when an organization is transmitting or storing data that needs protection. Actual encryption algorithms used to protect data typically use the XOR operator as part of a more complex encryption pro<;ess, thus understanding XOR encryption is a necessary step on the path to becoming a cryptologist.

Often, one can combine the XOR operation with a block cipher operation to produce a simple but powerful operation. Consider the example that follows, the first row of which shows a character message "5E5+'" requiring encryption. The second row shows this message in binary notation. In order to apply an 8-bit block cipher method, the binary message is broken into 8-bit blocks in the. row labeled "Message Blocks."The fourth row shows the 8-bit key (01010101) chosen for the encryption; To encryptthe message, you must perform the XOR operation on each 8-bit block by using the XOR function on the message bit and the key bit to determine the bits of the ciphertext until the entire message is enciphered. The result is shown in the row labeled "Ciphertext': This ciphertext can now be sent to a receiver, who will be able to decipher the message by simply knowing the algorithm (XOR) and the key (01010101).

Message (text) : " 5E5+"

Message (binary): 00110010100010100110101001010111001010 1

Message blocks: 00110101 01000101 00110101 00101011 10010101

Key: 01010101 01010101 01010101 01010101 01010101

Ciphertext: 01100000 00010000 01100000 01111110 11000000

If the receiver cannot apply the key to the ciphertext and derive the original message, either the cipher was applied with an incorrect key or the cryptosystem was not used correctly.

Vernam Cipher

Also known as the one-time pad, the Vernam dpher, which was developed at AT&T, uses a set of characters only one time for each encryption process (hence, the name one-time pad). The pad in the name comes from-the days of manual encryption and decryption when the key values for each ciphering session were prepared by hand and bound into an easy-to-use form-i.e., a pad of paper. To peiform the Yernam cipher encryption operation, the pad values are added to numeric values that represent the plaintext that needs to be encrypted. So, each character of the plaintext is turned into a number and a pad value for that position is added to it. The resulting sum for that character is then converted back to a ciphertext letter for transmission. If the sum of the. two values exceeds 26, then 26 is subtracted from the total (Note that the process of keeping a computed number within a specific range is called a modulo; thus, requiring that all numbers be in the range 1-26 is referred to as Modulo 26. In Modulo 26, if a number is larger than 26, then 26 is repeatedly subtracted from it until the number is in the proper range.)

To examine the Yernam cipher and its use of modulo, consider the following example, which uses the familiar "SACK GAUL SPARE NO ONE" as plaintext. In the first step of this encryption process, the letter "S" will be converted into the number 19 (because it is the 19th letter of the alphabet), and the same conversion will be applied to the rest of the letters of the plaintext message, as shown below.

Plain Text:	S	A	C	K	G	A	U	L	S	P	A	R	E	N	O	O	N	E
Plain Text Value:	19	01	03	11	07	01	21	12	19	16	01	18	05	14	15	15	14	05
One-Time Pad text:	F	P	Q	R	N	S	B	I	E	H	T	Z	L	A	C	D	G	J
One-Time Pad Value:	06	16	17	18	14	19	02	09	05	08	20	26	12	01	03	04	07	10
Sum of Plaintext and	25	17	20	29	21	20	23	21	24	24	21	44	17	15	18	19	21	15

Pad:																		
After Substraction:	Modulo				03							18						
Ciphertext:		Y	Q	P	C	U	T	W	U	X	X	U	R	Q	O	R	S	U	O

Rows three and four in the example above show, respectively, the one-time pad textthat was chosen for this encryption and the one time pad value. As you can see, the padvalue is, like the plaintext value, derived by considering the position of each pad text letter in the alphabet, thus the pad text letter "F" is assigned the position number of 06. This conversion process is repeated for the entire one-time pad text. Next, the plaintext value and the one time pad value are added together-the first such sum is 25. Since 25 is in the range of 1 to 26, no Modulo- 26 subtraction is required. The sum remains 25, and yields the cipher text "Y'; as shown above. Skipping ahead to the fourth character of theplaintext, "K'; we find that the plaintext value for it is 11. The pad text is "R" and the padvalue is 18. Adding 11 and 18 will result in a sum of 29. Since 29 is larger than 26, 26 isSubtracted from it, which yields the value 3. The cipher text for this plaintext character will then be the third letter of the alphabet, "C”

Decryption of any cipher text generated from a one-time pad will require either knowledge of the pad values or the use of elaborate and (the encrypting party hopes)very difficult cryptanalysis. Using the pad values and the cipher text, the decryption process would happen as follows; "Y" becomes the number 25 from which we subtract the pad value for the first letter of the message, 06. This yields a value of 19, or the letter

“S”. This pattern continuous until the fourth letter of the cipher text where the cipher text letter is “c” and the pad value is 18. Subtracting 18 from 3 will give a difference of negative 15. Since modulo-26 is employed, it requires that all numbers are in the range of that fourth letter of the message is “K”

Book or Running Key Cipher

One encryption method made popular by spy movies involves using the text in a book asthe key to decrypt a message. The ciphertext consists of a list of codes representing thepage number, line number, and word number of the plaintext word. The algorithm is themechanical process of looking up the references from the ciphertext and converting eachreference to a word by usingthe ciphertext's value and the key (the book). For example,from a copy of a particular popular novel, one may send the message: 259,19,8; 22,3,8;375,7,4; 394,17,2. Although almost any book will work just fine, dictionaries andthesauruses are typically the most popular sources as they can guarantee having almostevery word that might be needed. Returning to the example, the receiver must first knowwhich novel is used-in this case, suppose it is the science fiction novel, *A Fire Upon theDeep,* the 1992 TOR edition. To decrypt the ciphertext, the receiver would acquire the book and begin by turning to page 259, finding line 19, and selecting the eighth word inthat line (which happens to be "sack"). Then the receiver would go to page 22, line 3, andselect the eighth word again, and so forth. For this example, the resulting message will be"SACK ISLAND

SHARP PATH". If dictionaries are used, the message would be made up of only the page number and the number of the word on the page. An even more sophisticated version might use multiple books, perhaps even in a particular sequence for each word or phrase

Hash Functions

In addition to ciphers, another important encryption technique that is often incorporated into cryptosystems is the hash function. Hash functions are mathematical algorithms that generate a message summary or digest (sometimes called a fingerprint) to confirm the identity of a ~pacific message and to confirm that there have not been any changes to the content While not directly related to the creation of a ciphertext, hash functions are used to confirm message identity and integrity, both of which are critical functions in e-commerce.

Hash algorithms are publicly known functions that create a hash value, also known as a message digest, by converting variable-length messages into a single fixed-length value. The message digest is a fingerprint of the author's message that is to be compared with the receiver's locally calculated hash of the same message If both hashes are identical after transmission, the message has arrived without modification. Hash functions are considered one-way operations in that the message will always provide the same hash value if it is the same message, but the hash value itself cannot be used to determine the contents of the message.

Hashing functions do not require the use of keys, but **a message authenticationcode(MAC),** which is a key-dependent, and one-way hash function, may be attached to a message to allow only specific recipients to access the message digest The MAC is essentially a one-wayhashvalue that is encrypted with a symmetric key. The recipients must possess the key to access themessage digest and to confirm message integrity.

Because hash functions are one-way, they are used in password verification systems to confirm the identity of the user. In such systems, the hash value, or message digest, is calculated based upon the originally issued password, and this message digest is stored

for later comparison. When the user logs on for the next session, the system calculates a hash value based on the user's inputted password. The newly calculated hash value is compared against the stored value to confirm identity.

The Secure Hash Standard (SHS) is a standard issued by the National Institute of Standards and Technology (NIST). Standard document FIPS 180-1 specifies SHA-l (Secure Hash Algorithm 1) as a secure algorithm for computing a condensed representation of a message or data file. SHA-I produces a 160-bit message digest, which can then be used as an input to a digital signature algorithm. SHA-I is based on principles modeled after MD4 (which is part of the MDx family of hash algorithms created by Ronald Rivest). New hash algorithms (SHA-256, SHA-384, and SHA-512) have been proposed by NIST as standards for 128, 192, and 156 bits, respectively. The number of bits used in the hash algorithm is a measurement of the strength of the algorithm against collision attacks. SHA-256 is essentially a 256-bit block cipher algorithm that creates a key by encrypting the intermediate hash value with the message block functioning as the key. The compression function operates on each 512-bit message block and a 256-bit intermediate message digest'

Cryptographic Algorithms

In general, cryptographic algorithms are often grouped into two broad categories-symmetric and asymmetric-but in practice, today's popular cryptosysterns use a hybrid combination of symmetric and asymmetric algorithms. Symmetric and asymmetric algorithms can be distinguished by the types of keys they use for encryption and decryption operations. The upcoming section discusses both of these algorithms, and includes Technical Details boxes that provide supplemental information on cryptographic notation and advanced encryption standards.

Symmetric Encryption. A method of encryption that requires the same secret key to encipher and decipher the message is known as **private key encryption or symmetric encryption.** Symmetric encryption methods use mathematical operations that can be programmed into extremely fast computing algorithms so that the encryption and decryption processes are done quickly by even small computers. As you can see in Figure 8- 3, one of the challenges is that both the sender and the receiver must have the secret key. Also, if either copy of the key falls into the wrong hands, messages can be decrypted by others and the sender and intended receiver may not know the message was intercepted. The primary challenge of symmetric key encryption is getting the key to the receiver, a process that must be conducted out of band (meaning through a channel or band other than the one carrying the cipher text) to avoid interception.

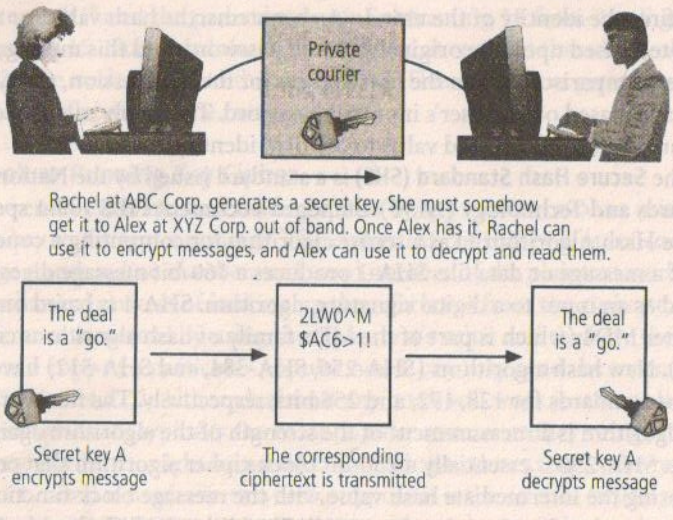


FIGURE 8-3 Example of Symmetric Encryption

Cryptographic Notation

The notation used to describe the encryption process varies, depending on its source. The notation chosen for the discussion in this text uses the letter M to represent the original message, C to represent the ending ciphertext, and E to represent the encryption process: thus, E(M) = C.[2] This formula represents the application of encryption (E) to a message (M) to create ciphertext (C). Also in this notation scheme, the letter D represents the decryption or deciphering process, thus the formula D[E(M)] = M states that if you decipher (D) an enciphered an message (E(M)), you should get the original message (M). This could also be stated as D[C]=M, or the deciphering of the ciphertext (remember that C=E(M)) results in the original message M. Finally the letter K is used the represent the key, therefore(M,K) = C suggests that encrypting (E) the message (M) with the key (K) results in the ciphertext (C). Similarly, D(C,K) = D[E(M,K),K] = M, or deciphering the ciphertext with key K results in the original plaintext message—or, to translate this formula even more precisely, deciphering with key K the message encrypted eith key K results in the original message.

To encrypt a plaintext set of data, you can use one of two methods: bit stream and block cipher. With the bit stream method, the message is divided into blocks, e.g., 8-,16-,32-, or 64-bit blocks, and then each block is transformed using the algorithm and key. Bit stream methods most commonly use algorithm functions like XOR, whereas block methods can use XOR, transposition, or substitution.

There are a number of popular symmetric encryption cryptosystem. One of the most widely known is the DATA ENCRYPTION STANDARS (DES)., which was developed by IBM and is based on the company's Lucifer algorithm, which uses a key length of 128 bits. As implemented , DES uses a 64-bit block size and a 56-bit key.DES was adopted by nIST in 1976 as a federal standard for encryption of non-classified information.With this approval, DES became widely employed in commercial applications as the encryption standard of choice. DES enjoyed increasing popularity for almost 20 years, until 1997, when users realized that using a 56-bit key size was no longer sufficient as an acceptable level of secure communications. And soon enough, in 1998, a group called Electronic Frontier Foundation (www.eff.org), using a specially designed computer, broke a DES key in less than three days(just over 56 hours, to be precise). Since then,it has been theorized that a dedicated attack supported by the proper hardware (thus , not even a specialized computer like that of Electronic Frontier Foundation) can break a DES key in less than four hours.

As DES became known as being too weak for highly classified communications, Triple DES (3DES) was created to provide a level of security far beyond that of DES. 3 DES was an advanced application of DES, and was in fact originally designed to replace DES. While 3DES did deliver on its promise of encryption strength beyond DES, it too was soon proven too weak to survive indefinitely-especially as computing power continued to double every 18 months. Within just few years , 3DES needed to be replced.

TRIPLE DES (3DES)

As it was demonstrated that DES was not strong enough for highly classified communication, 3DES was created to provide a level of security far beyond that of standard DES.(In between , there was a 2DES; however, it was statistically shown that the double DES did not provide

significantly stronger security than that of DES). 3DES takes three 64-bit keys for an overall key length of 192 bits.Triple DES encryption is the same as that of standard DES; however, it is repeated three times. Triple DES can be employed using two or three keys, and a combination of encryption or decryption to ontain additional security.The most common implementations involve encrypting and /or decrypting with two or three different keys, a process that is described below. 3DES employs 48 rounds in its encryption computation, generating ciphers that are approximately 2^{56}. (72 quadrillion) times stronger than standard DES ciphers but require only three times longer to process.

One example of 3DES encryption is illustrated here:

1. In the first operation, 3DES encrypts the message with key 1, then decrypts it with key 2, and then it encrypts it again with key 1. In cryptographic notation terms, this would be [E{D[E(M,K1)],K2},K1]. Decrypting with a different key is essentially another encryption, but it reverses the application of the traditional encryption operations.
2. In the second operation, 3DES encrypts the message with key 1, then it encrypts it again with key 2, and then it encrypts it a third time with key 1 again, or [E{E[E(M,K1)],K2},K1].
3. In the third operation, 3DES encrypts the message three times with three different keys; [E{E[E(M,K1)],K2},K3].This is the most secure level of encryption possible with 3DES.

The successor to 3DES is Advanced Encryption Standard (AES). AES is a Federal Information Processing Standard (FIPS) that specifies a cryptographic algorithm that is used within the U.S. government to protect information at federal agencies that are not a part of the national defense infrastructure. (Agencies that are considered a part of national defense use other, more secure methods of encryption, which are provided by the National Security Agency.) The requirements for AES stipulate that the algorithm should be unclassified, publicly disclosed, and available royalty-free worldwide. AES has been developed to replace both DES and 3DES. While 3DES remains an approved algorithm for some uses, its expected useful life is limited. Historically, cryptographic standards approved by FIPS have been adopted on a voluntary basis by organizations outside government entities. The AES selection process involved cooperation between the U.S. government, private industry, and academia from around the world. AES was approved by the Secretary of Commerce as the official federal governmental standard on May 26, 2002.

The AES implements a block cipher called the Rijndael Block Cipher with a variable block length and a key length of 128, 192, or 256 bits. Experts estimate that the special computer used by the Electronic Frontier Foundation to crack DES within a couple of days would require approximately 4,698,864 quintillion years (4,698,864,000,000,000,000,000) to crack AES. To learn more about the AES, See the Technical Details box entitled "Advanced Encryption Standard (AES).”

Advanced Encryption Standard (AES)

Of the many ciphers that were submitted (from across the world) for consideration in the AES selection process, five finalists were chosen: MARS, RC6, Rijndael, Serpent, and Twofish. On October 2, 2000, NIST announced the selection of Rijndael as the cipher to be used as the basis for the AES, and this block cipher was approved by the Secretary of Commerce as the official federal governmental standard as of May 26, 2002.

The AES version of Rijndael can use a multiple round based system. Depending on the key size, the number of rounds varies between 9 and 13: for a l28.bit key, 9 rounds plus one end round are used; for a 192-bit key, 11 rounds plus one end round are used; and for a 256-bit key, 13 rounds plus one end round are used. Once Rijndael was adopted as the AES, the ability to use variable sized blocks was standardized to a single l28-bit block for simplicity.

There are four steps within each Rijndael round, and these are described in "The

Advanced Encryption Standard (Rijndael)" by John Savard as follows:

1. The Byte Sub step. Each byte of the block is replaced by its substitute in an S-box

(Substitution box). [Author's Note: The S-box consists of a table of computed values, the calculation of which is beyond the scope of this text.]

2. The Shift Row step. Considering the block to be made up of bytes 1 to 16, these

bytes are arranged in a rectangle, and shifted as follows:

from				to			
1	5	9	13	1	5	9	13
2	6	10	14	6	10	14	2
3	7	11	15	11	15	3	7
4	8	12	16	16	4	8	12

Other shift tables are used for larger blocks.

3. The Mix Column step. Matrix multiplication is performed: each column is

multiplied by the matrix:

2 3 1 1

1 2 3 1

1 1 2 3

3 1 1 2

4. The Add Round Key step. This simply XORs in the subkey for the current round.

The extra final round omits the Mix Column step, but is otherwise the same as a

regular round."3

Asymmetric Encryption. Another category of encryption techniques is asymmetric encryption. Whereas the symmetric encryption systems are based on using a single key to both encrypt and decrypt a message, **asymmetric encryption** uses two different

but related keys, and either key can be used to encrypt or decrypt the message. If, however, Key A is used to encrypt the message, only Key B can decrypt it, and if Key B is used to encrypt a message, only Key A can decrypt it. Asymmetric ~ncryption can be used to provide elegant solutions to problems of secrecy and verification. This technique has its highest value when one key is used as a private key, which means that it is kept secret (much like the key of symmetric encryption), known only to the owner of the key pair, and the other key serves as a public key, which means that it is stored in a public location where anyone can use it. This is why the more common name for asymmetric encryption is **public key encryption.**

Consider the following example, illustrated in Figure 8-4. Alex at XYZ Corporation wants to send an encrypted message to Rachel at ABC Corporation. Alex goes to a public key registry and obtains Rachel's public key. Remember that the foundation of asymmetric encryption is that the same key cannot be used to both encrypt and decrypt the same message. So. when Rachel's public key is used to encrypt the message, only Rachel's private key can be used to decrypt the message and that private key is held by Rachel alone. Similarly, if Rachel wants to respond to Alex's message, she goes to the registry where Alex's public key is held, and uses it to encrypt her message, which of course can only be read by Alex's private key. This approach, which keeps private keys secret and encourages the sharing of public keys in reliable directories, is an elegant solution to the key management problems found in symmetric key applications.

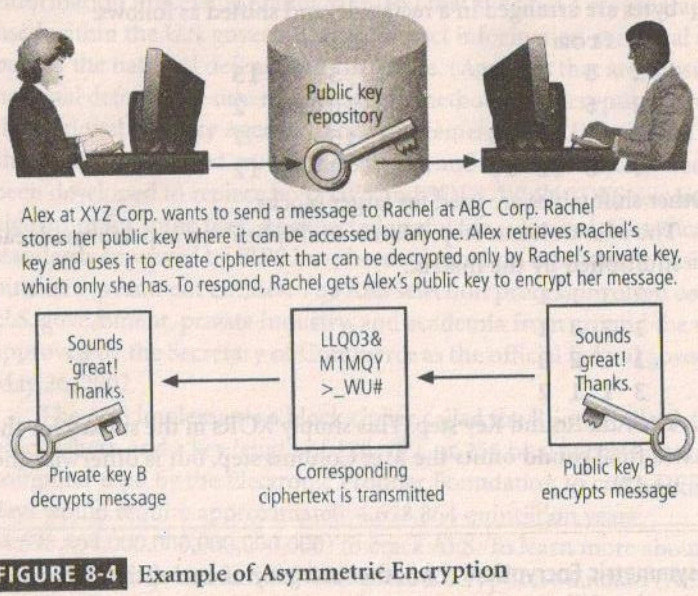


FIGURE 8-4 Example of Asymmetric Encryption

Asymmetric algorithms are based on one-way functions. A one-way function is simple to compute in one direction, but complex to compute in the opposite. This is the foundation of public-key encryption. Public-key encryption is based on a hash value, which, as you learned earlier in this chapter, is calculated from an input number using a hashing algorithm. This hash value is essential summary of the original input values. It is virtually impossible to derive the original values without knowing how the values were used to create the hash value. For example. if you multiply 45 by 235 you get 10,575. This is simple enough. But if you are simply given the number 10,575, can you determine which two numbers were multiplied to determine this number? Now assume that each multiplier is 200 digits long and prime. The resulting multiplicative product would be up to 400 digits long. Imagine the time you'd need to factor that out There is a shortcut, however. In mathematics, it is known as a trapdoor (which is different from the software trapdoor). A mathematical **trapdoor** is a "secret mechanism that enables you to easily accomplish the reverse function in a one-way function."4 With a trapdoor, you can use a key to encrypt or decrypt the ciphertext, but not both, thus requiring two keys. The public key becomes be the true key, and the private key is to be derived from the public key using the trapdoor.

One of the most popular public key cryptosystems is RSA, whose name is derived from Rivest-Shamir-Adleman, the algorithm's developers. The **RSA algorithm** was the first public

key encryption algorithm developed (in 1977) and published for commercial use. It is very popular and has been embedded in both Microsoft's and Netscape's Web browsers to enable them to provide security for e-commerce applications. The patented RSA algorithm has in fact become the de facto standard for public use encryption applications. To see how this algorithm works, see the Technical Details box "RSA Algorithm."

.

TECHNICAL DETAILS BOX

RSA Algorithm

If you understand modulo If you understand modulo mathematics, you can appreciate the complexities of the RSA algorithm. The security of the RSA algorithm is based on the computational difficulty of factorin lar e composite numbers and computing the eth roots modulo, a composite number for a specified odd integer *e.* Encryption in RSA is accomplished by raising the message *M* to a nonnegative integer power *e.* The product is then divided by the nonnegative modulus *n* (n should have a bit length of at least *1024* bits), and the remainder is the ciphertext *C.* This process results in one-way operation (shown below) when n is a very large number.

$$C = Me / \mod n$$ I

In the decryption process, the ciphertext C is raised to the power d, a nonnegative

integer, as follows:

$$d = e^{-1} \mod((p-1)(q-1))$$

C is then reduced by modulo n. In order for the recipient to calculate the decryption key, the p and q factors must be known. The modulus n, which is a composite number, is determined by multiplying two large nonnegative prime numbers, p and q:

$$n=p \infty q$$

In RSA's asymmetric algorithm, which is the basis of most modern Public Key Infra- structure (PKI) systems (a topic covered later in this chapter), the public and private keys

are generated using the following procedure, which is from the RSA Corporation:

"Choose two large prime numbers, p and q, of equal length, and compute !

p x q = n, which is the public modulus. !

Choose a random public key, e, so that e and (p-1)(q-l) are relatively prime. I

Compute e x d = 1 mod (p -l)(q - 1), where d is the private key.

Thus d = e-lmod[(p - l)(q -1)]."

where "(d, n) is the private key; (e, n) is the public key. P is encrypted to generate ciphertext C as $C = p^e \bmod n$, and is decrypted to recover the plaintext, P as P = Cd

Essentially, the RSA algorithm can be divided into three steps:

1. Key *generation:* Prime factors p and q are statistically selected by a technique known as i probabilistic primality testing and then multiplied together to form n. The encryption exponent e is selected, and the decryption exponent d is calculated.
2. *Encryption:* M is raised to the power of e, reduced by modulo n, and remainder C is the ciphertext.
3. *Decryption:* C is raised to the power of d and reduced by modulo n.

The sender publishes the public key, which consists of modulus n and exponent e.

The remaining variables d, p, and q are kept secret.

A message can then be encrypted by:$C= M^e$(recipient) mod n(recipient)

Digitally signed by: $C= M'^d$(sender) mod n(sender)

Verified by: $M' = C'^e$(sender) mod n(sender)

Decrypted by: $M = C^d$(recipient) mod n(recipient)

Example Problems

Because this Technical Details box presents complex information, the following sections contain practice examples to help you better understand the machinations of the various algorithms.

RSA Algorithm Example: Work through the following steps to better understand

how the RSA algorithm functions:

1.Choose randomly two large prime numbers: P, Q (usually P, Q > 10^100)→t This means 10 to the power 100.

2.Compute:

N=PXQ
Z=(P-1)(Q-l)

3. Choose a number relatively prime with Z and call it *D.*

D < N; relatively prime means that *D* and Z have no common factors, except 1

4. Find number E, such that ->*EX D* = 1 mod Z;

5. The public key is: (N, E); the private Key is (N, D).

6. Create Cipher (Encrypted Text):

C=|TEXT|E(MOD N)

C→ Encrypted text -t this is the text that's transmitted

| TEXT |→ Plaintext to be encrypted (its numerical correspondent)

7. Decrypt the message:

D = Plaintext = CD (MOD N), C = Ciphertext from part 6.

Note that it is almost impossible to obtain the private key, knowing the public key, and it's almost impossible to factor N into P and Q.

RSA Numerical Example: 13 Work through the following steps to better understand RSA Numericals:

1. Choose P = 3, Q = 11 (two prime numbers). Note that small numbers have been chosen *for* the example, so that you can easily work with them. In real life encryption, they:are larger than 10^100.
2. N= P∞Q =3 ∞11 = 33; Z = (P-1)(Q-l) =2∞10 = 20
3. Choose a number *for D* that is relatively prime with Z, *for* example, *D* = 7→ (20 and 7 have no common divisors, except 1).
4. *E* = ? such as *E∞D* = I MOD Z (I MOD Z means that the remainder of E/D division is 1).
   E ∞ D / Z → E ∞7 /20→E=3
   Check E∞D/Z-+ *3007/* 20→2l/20→ Remainder = 1
5. So, the public key is (N,E) = (33,3) → This key will be used to encrypt the message.
   The private key is (N,D) = (33,7)→This key will be used to decrypt the message

English Alphabet and Corresponding Numbers *for* Each Letter:[7] In real life applications, the ASCII code is used to represent each of the characters of a message. For this example, the position of the letter in the alphabet is used instead to simplify the calculations:

A=01,B=02,etc…..Z=26.

Encrypt The Word "Technology" as illustrated in Table 8-5:[8] Now you can use the corresponding numerical and the previous calculations to calculate values for the public key (N,E) = (33,3) and the private key (N,D) = (33,7).

Table 8-5 Encryption

Plaintext	Text value	(Text)AE	(Text)AE MOD N = Ciphertext
T	20	8000	8000 MOD 33 = 14
E	05	125	125 MOD 33 = 26
C	03	27	27 MOD 33 = 27
H	08	512	512MOD33=17
N	14	2744	2744 MOD 33 = 05
O	15	3375	3375 MOD 33 = 09
L	12	1728	1728 MOD 33 = 12
O	15	3375	3375 MOD 33 = 09
G	07	343	343 MOD 33 = 13
Y	25	15625	15625 MOD 33 = 16

So, the cipher (encrypted message) is: 14262717050912091316. This is what is trans. mitted over unreliable lines. Note that there are two digits per letter. To decrypt the transmitted message we apply the private key *(AD)* and re-MOD the product, the result *of* which is the numerical equivalent *of* the original plaintext.

Table 8-6 Decryption

Ciphertext	(Cipher)AD	(Cipher)AD MOD N = |Text|	Plaintext
14	105413504	105413504 MOD 33 = 20	T
26	8031810176	8031810176 MOD 33 = 05	E

27	10460353203	10460353203 MOD 33 = 03	C
17	410338673	410338673 MOD 33 = 08	H
05	78125	78125 MOD 33 = 14	N
09	4782969	4782969 MOD 33 = 15	O
12	35831808	35831808 MOD 33 = 12	L
09	4782969	4782969 MOD 33 = 15	O
13	62748517	62748517 MOD 33 = 07	G
16	268435456	268435456 MOD 33 = 25	Y

As you can see in Table 8.6, although very small P and Q numbers were used, the numbers required for decrypting the message are relatively large. Now you have a good idea of what kind *of* numbers are needed when P and Q are large (that is, in the 10^{100} range).

If P and Q are not big enough for the cipher to be secure, P and Q must be increased. The strength of this encryption algorithm relies on how difficult it is to factor P and Q from N if N is known. If N is not known, the algorithm is even harder to break, of course.

The problem with asymmetric encryption, as is shown by the example ill Figure 8-4, is that holding a single conversation between two parties requires four keys. Moreover, if four organizations want to exchange communications frequently, each party must manage its private key and four public keys. In such scenarios, determining which public key is needed to encrypt a particular message can become a rather confusing problem, and with more organizations in the loop, the problem expands. This is why asymmetric encryption is sometimes regarded by experts as an inefficient endeavor. Compared to symmetric encryption, asymmetric encryption is also not as efficient in terms of CPU computations. Consequently, hybrid systems, such as those described in the section of this chapter titled "Public Key Infrastructure (PKI);' are more commonly used than pure asymmetric system.

Encryption Key Size

When using ciphers, one of the decisions that has to be made is the size of the cryptovariable or key. This will prove to be very important, because the strength of many encryption applications and cryptosystems is measured by key size. But does the size of the encryption key really matter? And how exactly does key size affect the strength of an algorithm? Typically, the length of the key increases the number of random selections that will have to be guessed in order to break the code. Creating a larger universe of possibilities that need to be checked increases the

time required to make guesses, and thus a longer key will directly influence the strength of the encryption.

It may surprise you to learn that when it comes to cryptosystems, the security of encrypted data is *not* dependent on keeping the encrypting algorithm secret; in fact, algorithms should be (and often are) published, so that research to uncover their weaknesses can be done. Instead the security of any cryptosystem depends on keeping some or all of the elements of the cryptovariable (s) or key(s) secret, and effective security is maintained by manipulating the size (bit length) of the keys and by following proper procedures and policies for key management.

For a simple example of how key size is related to encryption strength, suppose you have an algorithm that uses a three-bit key. You may recall from earlier in the chapter that keyspace is the amount of space from which the key can be drawn. Also, you may recall that in binary notation, three bits can be used to represent values from 000. to Ill, which correspond to the numbers 0 to 7 in decimal, and thus a keyspace of eight keys. This means that with an algorithm that uses a three-bit key you have eight possible keys to choose from (the numbers 0 to 7 in binary are 000, 001, 010, 011, 100, 101, 110,111). If you know how many keys you have to choose from, you can program a computer simply to try all the keys and see if it can crack the encrypted message.

The preceding statement presumes a few things: I) you know the algorithm, 2) you have the encrypted message, and 3) you have time on your hands. It is easy to satisfy the first criterion. The encryption tools that use the Data Encryption Standard (DES) can be purchased over the counter. Many of these tools are based on encryption algorithms that are standards, as is DES itself, therefore it is relatively easy to get a cryptosystem based on DES that would enable you to decrypt an encrypted message if you possess the key. The second criterion requires the interception of an encrypted message, which is illegal, but not impossible. As for the third criterion, the task required is a brute force attack, in which a computer randomly (or sequentially) selects possible keys of the known size and applies them to the encrypted text, or a piece of the encrypted text. If the result is plaintext-bingo! But as indicated earlier in this chapter, it can take quite a long time to exert brute force on the more advanced cryptosystems. In fact, the strength of an algorithm is determined by how long it takes to guess the key. Luckily, however, once set to a task, computers do not require much adult supervision, so you probably won't have to quit your day job.

But when it comes to keys, how big is big? From the example at the beginning of this section, you learned that a three-bit system has eight keys to guess. An eight-bit system has 256 keys to guess. Note, however, that if you use a 32-bit key, puny by modem standards, you have to guess almost 16.8 million keys. Even so, a modern PC, such as the one described in Table 8-7, could do this in mere seconds. But, as Table 8-7 shows, the amount of time needed to crack a cipher by guessing its key grows very quickly-that is, exponentially with each additional bit.

One thing to keep in mind here is that even though the estimated time to crack grows so rapidly with respect to the number of bits in the encryption key and the odds of cracking seem at first glance to be insurmountable, Table 8-7 doesn't account for the fact that computing power has increased (and continues to increase). Therefore, these days even the once-standard 56-bit encryption can't stand up to brute force attacks by personal computers, especially if multiple computers are used together to crack these keys. Each additional computer reduces the amount of time needed. Two computers can divide the possibilities and crack the key in approximately half the time and so on. Thus, two hundred and eighty five computers can crack a 56-bit key in one year, ten times as many would do it in a little over a month.

Encryption Key Power

Number of its In Key	Odds of Cracking:1 in	Estimated Time to Crack*
8	256	.000032 seconds
16	65,536	.008192 seconds
24	16,777,216	2.097 seconds
32	4,294,967,296	8 minutes 56.87 seconds
56	72,057,594,037,927,900	285 years 32 weeks 1 day
64	18,446,744,073,709,600,000	8,090,677,225 years
128	3.40282E+38	5,257,322,061,209,440,000,000 years
256	1.15792E+77	2,753,114,795,116,330,000,000,000,000,000,00, 000,000,000,000,000 years
512	1.3408E+154	608,756,305,260,875,000,000,000,000,000, 000,000,000,000,000,000,000,000,000,000, 000,000,000,000,000,000,000,000,000,000, 000,000,000 years

[Note]* Estimated Time to crack is based on a general purpose personal computer performing eight million guesses per second

Cryptography Tools

Public key Infrastructure

Public Key Infrastructure (PKI) is an integrated system of software , encryption methodologies protocols, legal agreements, and third party services that enable users to communicate security. PKI systems are based on public key cryptosystems and include digital certificates and certificate authorities (CAs)

Digital certificates are public key container files that allow computer programs to validate the key and identify to whom it belongs.PKI and the digital certificate registries they contain enable

the protection of information assets by making verifiable digital certificates readily available to business applications. This, in turn , allows the applications to implement several of the key characteristics of information security and to integrate these characteristics into business processes across an organization.

These processes include the following:

- Authentication: Individuals, organizations, and web servers can validate the identity of each of the parties in an internet transaction.
- Integrity: Content signed by the certificate is known to be unaltered while being moved from host to host or server to client.
- Privacy: Information is protected from being intercepted during transmission.
- Authorization: The validated identity of users and programs can be used to enable authorization rules that remain in place for the duration of a transaction; this reduces some of the overhead required and allows for more control of access privileges for specific transactions.

A typical PKI solution protects the transmission and reception of secure information by integrating the following components.

- A certificate authority (CA), which issues, manages, authenticates, signs, and revokes user's digital certificates, which typically contain the user's name,public key, and other identifying information.
- A registration authority (RA), which operates under the trusted collaboration of the certificate authority and can be delegated day-to-day certification functions, such as verifying registration information about new registrants, generating end-user keys, revoking certificates, and validating that users possess a valid certificate.
- Certificate directories , which are central locations for certificate storage that provide a single access point for administration and distribution.
- Management protocols, which organize and manage the communications between CAs, RAs, and end users. This includes the functions and procedures for setting up new users, issuing keys, recovering keys, updating keys, revoking keys, and enabling the transfer of certificates and status information among the parties involved in the PKI's area of authority.
- Policies and procedures that assist an organization in the application and management of certificates , the formalization of legal liabilities and limitations, and actual business practice use.

  Common implementations of PKI include: systems to issue digital certificates to users and servers; directory enrollment; key issuing systems; tools for managing the key issuan e; and verification and return of certificates. These systems enable organizations to apply an enterprise-wide solution that provides users within the PKI's area of authority the means to implement authenticated and secure communications and transactions.

Digital signatures

Digital signatures were created in response to the rising need to verify information transferred using electronic system. Currently, asymmetric encryption processes are used to create digital signatures. When an asymmetric cryptographic process uses the sender;s private key to encrypt a message, the sender's public key must be used to decrypt the message –when the decryption happens successfully, it provides verification that the message was sent by the sender and cannot be refued.This process is known as non-repudiation and is the principle of cryptography that gives credence to the authentication mechanism collectively known as a digital signature. Digital signatures are, therefore, encrypted messages that can be mathematically proven to be authentic.

The management of digital signatures has been built into most web browsers . As an example, the Internet Explorer digital management screen is shown in Figure 8-5.

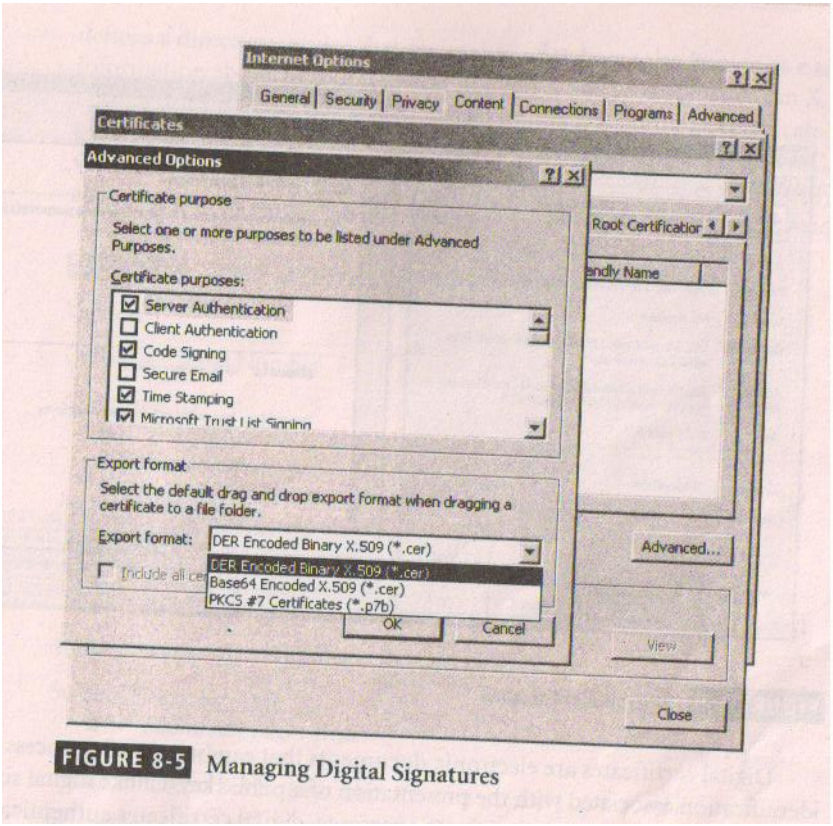


FIGURE 8-5 Managing Digital Signatures

Digital Certificates

Digital certificates are electronic documents that can be part of a process of identification associated with the presentation of a public key. Unlike digital signatures, which help authenticate the origin of a message, digital certificates authenticate the cryptographic key that is embedded in the certificate. When used properly these certificates enable diligent users to verify the authenticity of any organization's certificates. This is much like what happens when the Federal Deposit Insurance Corporation issues its "FDIC" logo to banks to help assure bank customers that their bank is authentic. Different client-server applications use different types of digital certificates to accomplish their assigned functions:

- The CA application suite issues and uses certificates that identify and establish a trust relationship with a CA to determine what additional certificates can be authenticated.
- Mail applications use Secure/Multipurpose Internet Mail Extension (S/MIME) certificates for signing and encrypting e-mail as well as for signing forms.
- Development applications use object-signing certificates to identify signers of object-oriented code and scripts.
- Web servers and Web application servers use Secure Socket Layer (SSL) certificates to authenticate servers via the SSL protocol (which is described in an upcoming section) in order to establish an encrypted SSL session.
- Web clients use client SSL certificates to authenticate users, sign forms, and participate in single sign-on solutions via SSL.

Two popular certificate types in use today are those created using Pretty Good Privacy (PGP) and those created using applications that conform to International Telecommunication Union's (ITU-T) x.509 version 3. You should know that X.S09 v3, whose structure is outlined in Table 8-8, is an ITU-T recommendation that essentially defines a directory service that maintains a database (also known as a repository) of information about a group of users holding X.SOY v3 certificates. An X.S09 v3 certificate binds a **distinguished name (DN)**, which uniquely identifies a certificate entity, to a user's public key. The certificate is signed and placed in the directory by the CA for retrieval and verification by the user's associated public key. X.S09 v3 does not specify an encryption algorithm; however, RSA with its hashed digital signature is recommended.

Table 8-8 X.509 v3 Certificate Structure

Version

Certificate Serial Number

Algorithm ID

- Algorithm ID
- Parameters

Issuer Name

Validity

- Not Before
- Not After

Subject Name

Subject Public Key Info

- Public Key Algorithm
- Parameters

- Subject Public Key

Issuer Unique Identifier (Optional)

Subject Unique Identifier (Optional)

Extensions (Optional)

- Type
- Criticality
- Value

Certificate Signature Algorithm

Certificate Signature

Hybrid Cryptography Systems

Except in the case of digital certificates, pure asymmetric key encryption is not widely used. Asymmetric key encryption is more often used in conjunction with symmetric key encryption-thus, as part of a hybrid encryption system. The most common hybrid system is based on the **Diffie-Hellman Key Exchange method**, which is a method for exchanging private keys using public key encryption. With Diffie-Hellman, asymmetric encryption is used to exchange session keys. These are limited-use symmetric keys for temporary communications; they allow two organizations to conduct quick, efficient, secure communications based on symmetric encryption. Diffie-Hellman provided the foundation for subsequent developments in public key encryption. Because symmetric encryption is more efficient than asymmetric for sending messages, and asymmetric encryption doesn't require out-of-band key exchange, asymmetric encryption can be used to transmit symmetric keys in a hybrid approach. Diffie-HeIlman avoids the exposure of data to third parties that is sometimes associated with out-of-band key exchanges.

A hybrid encryption approach is illustrated in Figure 8-7, and it works as follows:

Alex at XYZ Corp. wants to communicate with Rachel at ABC Corp., so Alex first creates a session key. Alex encrypts a message with this session key, and then gets Rachel's public key. Alex uses Rachel's public key to encrypt both the session key and the message, which is already encrypted. Alex transmits the entire package to Rachel, who uses her private key to decrypt the package containing the session key and the encrypted message, and then uses the session key to decrypt the message. Rachel can then continue to use only this session key for electronic communications until the session key expires. The asymmetric session key is used in the much more efficient asymmetric encryption and decryption processes. After the session key expires (usually in just a few minutes) a new session key will be chosen and shared using the same process.

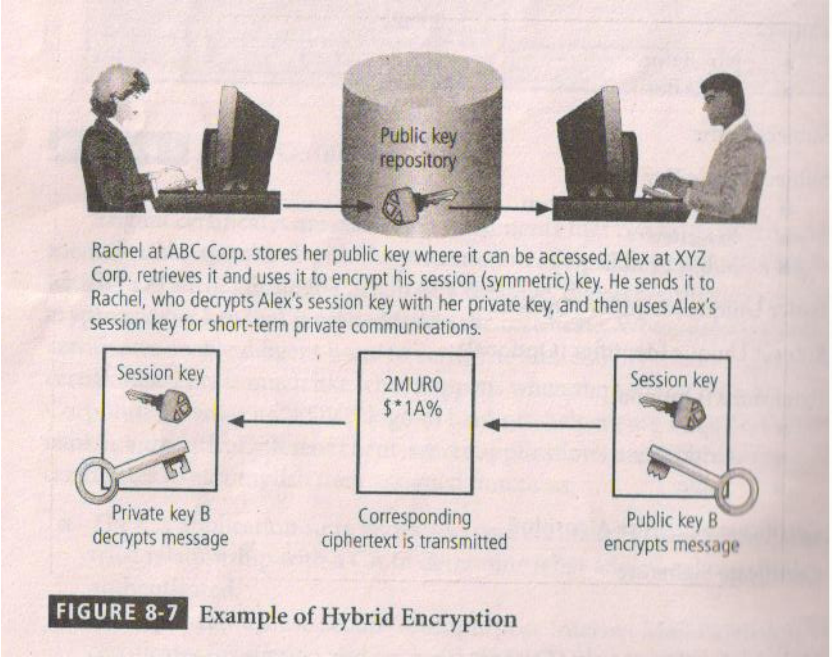


FIGURE 8-7 Example of Hybrid Encryption

Steganography

Steganography is a process of hiding information and has been in use for a long time. In fact the word "steganography" is derived from the Greek words *steganos* meaning "covered" and *graphein* meaning "to write." The Greek historian Herodotus reported on one of the first steganographers when he described a fellow Greek sending a message to warn of an imminent invasion by writing it on the wood beneath a wax writing tablet. If the tablet were intercepted, it would appear blank.[11] While steganography is technically not a form of cryptography, it is related to cryptography in that it ~ also a way of transmitting information so that the information is not revealed wl1i1e it's in transit. The most popular modem version of steganography involves hiding information within files that appear to contain digital pictures or other images.

To understand how modern steganography works in this specific case, you must first understand a little about how images are stored. Most computer graphics standards use a combination of three color values (red, blue, and green (RGB)) to represent a picture element, or pixel. Each of the three color values usually requires an 8-bit *code* for that color's intensity (e.g., 00000000 for no red and 11111111 for maximum red). Each color pixel of an image requires 24 bits to represent the color mix and intensity. Some image encoding standards use more or fewer bits per pixel, but for the purposes of this discussion, 24-bit color will suffice. When a picture is created (by a digital camera or a computer program), the number of horizontal and vertical pixels captured and recorded is known as the image's *resolution.* Thus, for example, if 1024 horizontal pixels are recorded and 768 vertical pixels are captured, the image has a I024x768 resolution and would commonly be said to have 786,432 pixels or three-quarters of a *megapixel.* Thus, an image that is I024x768 pixels contains 786,432 groups of 24 bits to represent the red, green, and blue data. The raw image size can be calculated as 1024x768x24, or 5.66 megabytes. There are plenty of bits in this picture data file in which to hide a secret message.

To the naked eye, there is no discernible difference between a pixel with a red intensity of 00101001 and another slightly different pixel with a red intensity level of 00101000. In other words, the two different values will result in pixels that do have a discernible difference. This inability to perceive difference on part of humans provides the steganographer with one bit per

color (or three bits per pixel) to use for encoding data into an image file. If a steganographic process uses three bits per pixel for all 786,432 pixels, it will be able to store 236 kilobytes of hidden data within the uncompressed image. Some steganographic tools can calculate the maximum size image that can be stored before being detectable. In addition to digital photos, messages can be hidden in any computer file that does not utilize all of its available bits. Some applications are capable of hiding messages in .bmp, .wav, .mp3, and .au files, as well as in unused storage space on CDs and DVDs. One program can take a text or document file and hide a message in the unused whitespace.

After the attacks of September II, 200 I, U.S. federal agencies were worried that terrorist organizations were "hiding maps and photographs of terrorist targets and posting instructions for terrorist activities on sports chat rooms, pornographic bulletin boards, and other websites" through the use of steganographic methods. No documented proof of this activity was ever publicized.[12]

Attacks on Cryptosystems

Historically, attempts to gain unauthorized access to secure communications have used brute force attacks in which the ciphertext is repeatedly searched for clues that can lead to the algorithm's structure. These attacks are known as ciphertext attacks, and involve a hacker searching for a common text structure, wording, or syntax in the encrypted message that can enable him or her to calculate the number of each type of letter used in the message. This process, known as frequency analysis, can be used along with published frequency of occurrence patterns of various languages and can allow an experienced attacker to crack almost any code quickly if the individual has a large enough sample of the encoded text. To protect against this, modern algorithms attempt to remove the repetitive and predictable sequences of characters from the cirhertext.

Occasionally, an attacker may obtain duplicate texts, one in ciphertext and one in plaintext, which enable the individual to reverse-engineer the encryption algorithm in a **known-plaintext attack scheme**. Alternatively, attackers may conduct a **selected- plaintext attack** by sending potential victims a specific text that they are sure the victims will forward on to others. When the victim does encrypt and forward the message, it can be used in the attack if the attacker can acquire the outgoing encrypted version. At the very least, reverse engineering can usually lead the attacker to discover the cryptosystem that is being employed.

Most publicly available encryption methods are generally released to the information and computer security communities for testing of the encryption algorithm's resistance to cracking. In addition, attackers are kept informed of which methods of attack have failed. Although the purpose of sharing this information is to develop a more secure algorithm, it has the danger of keeping attackers from wasting their time--that is, freeing them up to find new weaknesses in the cryptosystem or new, more challenging means of obtaining encryption keys.

In general, attacks on cryptosystems fall into four general categories: man-in-the-middle, correlation, dictionary, and timing. Although many of these attacks were discussed in Chapter 2, they are reiterated here in the context of cryptosystems and their impact on these systems.

Man-in-the-Middle Attack

A man-in-the-middle attack, as discussed in Chapter 2, is designed to intercept the transmission of a public key or even to insert a known key structure in place of the requested public key. Thus, attackers attempt to place themselves between the sender and receiver, and once they've intercepted the request for key exchanges, they send each participant a valid public key, which is known only to them. From the perspective of the victims of such attacks, their encrypted communication appears to be occurring normally, but in fact the attacker is receiving each

encrypted message and decoding it (with the key given to the sending party), and then encrypting and sending it to the originally intended recipient. Establishment of public keys with digital signatures can prevent the traditional man-in-the-middle attack, as the attacker cannot duplicate the signatures.

Correlation Attacks

As the complexities of encryption methods have increased, so too have the tools and methods of cryptanalysts in their attempts to attack cryptosystems. Correlation attacks are a collection of brute-force methods that attempt to deduce statistical relationships between the structure of the unknown key and the ciphertext that is the output of the cryptosystem. Differential and linear cryptanalysis, both of which are advanced methods of breaking codes that are beyond the scope of this discussion, have been used to mount successful attacks on block cipher encryptions such as DES. If these advanced approaches can calculate the value of the public key, and if this can be achieved in a reasonable time, all messages written with that key can be decrypted. The only defense against this kind of attack is the selection of strong cryptosystems that have stood the test of time, thorough key management, and strict adherence to the best practices of cryptography in the frequency of changing keys.

Dictionary Attacks

In a **dictionary attack**, the attacker encrypts every word in a dictionary using the same cryptosystem as used by the target. The attacker does this in an attempt to locate a match between the target ciphertext and the list of encrypted words from the same cryptosystem. Dictionary attacks can be successful when the ciphertext consists of relatively few characters, as for example files which contain encrypted usernames and passwords. If an attacker acquires a system password file, the individual can run hundreds of thousands of potential passwords from the dictionary he or she has prepared against the stolen list. Most computer systems use a well-known one-way hash function to store passwords in such files, but this can almost always allow the attacker to find at least a few matches in any stolen password file. After a match is located, the attacker has essentially identified a potential valid password for the system under attack.

Timing Attacks

In a **timing attack**, the attacker eavesdrops during the victim's session and uses statisticalanalysis of the user's typing patterns and inter-keystroke timings to discern sensitive session information. While timing analysis may not directly result in the decryption of sensitive data, it can be used to gain information about the encryption key and perhaps the cryptosystem in use. It may also eliminate' some algorithms as possible candidates, thus narrowing the attacker's search. In this narrower field of options, the attacker can increase the odds of eventual success. Once the attacker has successfully broken an encryption, he or she may launch a **replay attack**, which is an attempt to resubmit a recording of the deciphered authentication to gain entry into a secure source.

Defending From Attacks

Encryption is a very useful tool in protecting the confidentiality of information that is in storage and/or transmission. However, it is just that-another tool in the information security administrator's arsenal of weapons against threats to information security. Frequently, unenlightened individuals describe information security exclusively in terms of encryption (and possibly firewalls and antivirus software). But encryption is simply the process of hiding the true meaning of information. Over the millennia, mankind has developed dramatically more

sophisticated means of hiding information from those who should not see it. No matter how sophisticated encryption and cryptosystems have become, however, they have retained the same flaw that the first systems contained thousands of years ago: If you discover the key, that is, the method used to perform the encryption, you can determine the message. Thus, key management is not so much the management of technology but rather the management of people.

Encryption can, however, protect information when it is most vulnerable-that is, when it is outside the organization's systems. Information in transit through public or leased networks is an example of information that is outside the organization's control. With loss of control can come loss of security. Encryption helps organizations secure information that must travel through public and leased networks by guarding the information against the efforts of those who sniff, spoof, and otherwise skulk around. As such, encryption is a vital piece of the security puzzle.

**********END**********