UNIT 4

SECURITY TECHNOLOGY: INTRUSION DETECTION, ACCESS CONTROL & OTHER SECURITY TOLLS

4.1 SECURITY TECHNOLOGY

Introduction

The protection of an organization's information assets relies at least as much on people as on technical controls, but technical solutions, guided by policy and properly implemented, are an essential component of an information security program.

This concept builds on that discussion by describing additional and more advanced technologies—intrusion detection and prevention systems, honey pots, honey nets, padded cell systems, scanning and analysis tools, and access controls—that organizations can use to enhance the security of their information assets.

4.1.1 Intrusion Detection and Prevention Systems

An intrusion detection and prevention system (IDPS) is defined as a system that monitors a network and scans it for possible threats to alert the administrator and prevent potential attacks.

IDPS Terminology

In order to understand IDPS operational behaviour, you must first become familiar with some IDPS terminology. The following list of IDPS industry standard terms and definitions is taken from a well-known information security company, TruSecure:

1. **Alert:** Raising an alert in the form of audible signals, email messages, page notifications or pop-up windows.
2. **Evasion:** Changing format by an attacker to avoid from detecting by IDPS.
3. **False negative:** Failure of detecting a real attack by IDPS. Whereas, the main function of IDPS is detect and respond to attacks.
4. **False attack stimulus:** Triggering of alert by an event in the absence of an actual attack.
5. **False positive:** Raising alert by IDPS in the absence of an actual attack. They tend to pervasive users to be insensitive to alerts so, they reduce their activity to real intrusion events.

6. **Noise:** Alarm events that are accurate but do not pose significant threats to information security. Unsuccessful attacks are the most popular source of IDPS noise.
7. **Site policy:** Configuration and policy prepared by organization for implementation of IDPS
8. Site policy awareness: So-called smart IDPS means the ability of IDPS to dynamical modify its configuration in response to environment activity.
9. True attack stimulus: Alarm triggering by an event and causes an IDPS to react as if a real attack is in progress.
10. **Tuning:** The process of adjusting an IDPS to maximize its efficiency in order to detecting true positives while minimizing both false positive and false negatives.
11. **Confidence value:** Measurement of correct detections by IDPS and identify certain types of attacks.
12. **Alarm filtering:** Classification of IDPS alerts so that they are manageable more efficiently. They are similar to packet filters whereas they can filter items by source and destination while they can filter by operating systems, confidence values and alert types.
13. **Alarm clustering and compaction:** Categorization of similar behavioral alerts on time into a single higher-level alarm. This clustering maybe based on combination of frequency, attack signature or target similarity that lead to reducing the number of alert generated.

Types of Intrusion Detection Systems

For the purpose of dealing with IT, there are three main types of IDS:

1. **Network intrusion detection system (NIDS) :** It is an independent platform that identifies intrusions by examining network traffic and monitors multiple hosts. Network intrusion detection systems gain access to network traffic by connecting to a network hub, network switch configured for port mirroring, or network tap. In a NIDS, sensors are located at choke points in the network to be monitored, often in the demilitarized zone (DMZ) or at network borders. Sensors captures all network traffic and analyzes the content of individual packets for malicious traffic. An example of a NIDS is Snort.
2. **Host-based intrusion detection system (HIDS) :** It consists of an agent on a host that identifies intrusions by analyzing system calls, application logs, file-system modifications (binaries, password files, capability databases, Access control lists, etc.) and other host activities and state. In a HIDS, sensors usually consist of a software agent. Some application-based IDS are also part of this category. An example of a HIDS is OSSEC.

Intrusion detection systems can also be system-specific using custom tools and honeypots. In the case of physical building security, IDS is defined as an alarm system designed to detect unauthorized entry.

For the purpose of protecting perimeters of critical infrastructures and high risk assets, there is a primary type of IDS:

3. **Perimeter Intrusion Detection System (PIDS) :** Detects and pinpoints the location of intrusion attempts on perimeter fences of critical infrastructures. Using either electronics or more advanced fibre optic cable technology fitted to the perimeter fence, the PIDS detects disturbances on the fence, and this signal is monitored and if an intrusion is detected and deemed by the system as an intrusion attempt, an alarm is triggered.

IDPS Detection Methods

The majority of intrusion prevention systems utilize one of three detection methods: signature-based, statistical anomaly-based, and stateful protocol analysis.

1. **Signature-based Detection**: This method of detection utilizes signatures, which are attack patterns that are preconfigured and predetermined. A signature-based intrusion prevention system monitors the network traffic for matches to these signatures. Once a match is found the intrusion prevention system takes the appropriate action. Signatures can be exploit-based or vulnerability-based. Exploit-based signatures analyze patterns appearing in exploits being protected against, while vulnerability-based signatures analyze vulnerabilities in a program, its execution, and conditions needed to exploit said vulnerability.

2. **Statistical Anomaly-based Detection**: This method of detection baselines performance of average network traffic conditions. After a baseline is created, the system intermittently samples network traffic, using statistical analysis to compare the sample to the set baseline. If the activity is outside the baseline parameters, the intrusion prevention system takes the appropriate action.

3. **Stateful Protocol Analysis Detection**: This method identifies deviations of protocol states by comparing observed events with "predetermined profiles of generally accepted definitions of benign activity.

IDPS Response Behaviour

Each IDPS responds to external stimulation in a different way, depending on its configuration and function. Some respond in active ways, collecting additional information about the intrusion, modifying the network environment, or even taking action against the intrusion. Others respond in passive ways, for example by setting off alarms or notifications or collecting passive data through SNMP traps.

1. **IDPS Response Options :** When an IDPS detects a possible intrusion, it has a number of response options, depending on the implementing organization's policy, objectives, and system capabilities. When configuring an IDPS's responses, the system administrator must exercise care to ensure that a response to an attack (or potential attack) does not inadvertently exacerbate the situation.
2. **Reporting and Archiving Capabilities** : Many, if not all, commercial IDPSs can generate routine reports and other detailed information documents, such as reports of system events and intrusions detected over a particular reporting period (for example, a week or a month). Some provide statistics or logs in formats suitable for inclusion in database systems or for use in report generating packages.
3. **Failsafe Considerations for IDPS Responses** : Failsafe features protect an IDPS from being circumvented or defeated by an attacker. There are several functions that require failsafe measures. For instance, IDPSs need to provide silent, reliable monitoring of attackers. Should the response function of an IDPS break this silence by broadcasting alarms and alerts in plaintext over the monitored network, attackers can detect the IDPS and might then directly target it in the attack. Encrypted tunnels or other cryptographic measures that hide and authenticate communications are excellent ways to secure and ensure the reliability of the IDPS.

Selecting IDPS Approaches and Products

The wide array of available intrusion detection products addresses a broad range of organizational security goals and considerations; the process of selecting products that represent the best fit for any particular organization is challenging.

1. **Technical and Policy Considerations :** In order to determine which IDPS best meets an organization's needs, first consider the organizational environment in technical, physical, and political terms.
2. **Organizational Requirements and Constraints :** Your organization's operational goals, constraints, and culture will affect the selection of the IDPS and other security tools and technologies to protect your systems. Consider the organizational requirements and limitations selecting the IDPS approach
3. **IDPSs Product Features and Quality :** It's important to carefully evaluate any IDPS product by considering the features and quality security tools for selecting the IDPS approach.

Strengths and Limitations of IDPSs

Although intrusion detection systems are a valuable addition to an organization's security infrastructure, there are things they do well and things they do not do well. As you plan the security strategy for your organization's systems, it is important for you to

understand what IDPSs should be trusted to do and what goals might be better served by other security mechanisms.

Strengths of Intrusion Detection and Prevention Systems

Intrusion detection and prevention systems perform the following functions well:

1. Monitoring and analysis of system events and user behaviours
2. Testing the security states of system configurations
3. Baselining the security state of a system, then tracking any changes to that baseline
4. Recognizing patterns of system events that correspond to known attacks
5. Recognizing patterns of activity that statistically vary from normal activity
6. Managing operating system audit and logging mechanisms and the data they generate
7. Alerting appropriate staff by appropriate means when attacks are detected
8. Measuring enforcement of security policies encoded in the analysis engine
9. Providing default information security policies
10. Allowing non-security experts to perform important security monitoring functions

Limitations of Intrusion Detection and Prevention Systems

Intrusion detection systems cannot perform the following functions:

1. Compensating for weak or missing security mechanisms in the protection infrastructure, such as firewalls, identification and authentication systems, link encryption systems, access control mechanisms, and virus detection and eradication software.
2. Instantaneously detecting, reporting, and responding to an attack when there is a heavy network or processing load.
3. Detecting newly published attacks or variants of existing attacks
4. Effectively responding to attacks launched by sophisticated attackers.
5. Automatically investigating attacks without human intervention
6. Resisting all attacks that are intended to defeat or circumvent them
7. Compensating for problems with the fidelity of information sources
8. Dealing effectively with switched networks.

Deployment and Implementation of an IDPS

Deploying and implementing an IDPS is not always a straightforward task. The strategy for deploying an IDPS should take into account a number of factors, the foremost being how the IDPS will be managed and where it should be placed. These factors determine the

number of administrators needed to install, configure, and monitor the IDPS, as well as the number of management workstations, the size of the storage needed for retention of the data generated by the systems, and the ability of the organization to detect and respond to remote threats.

1. **IDPS Control Strategies :** An IDPS can be implemented via one of three basic control strategies. A control strategy determines how an organization supervises and maintains the configuration of an IDPS. It also determines how the input and output of the IDPS is managed. The three commonly utilized control strategies are centralized, partially distributed, and fully distributed.

2. **IDPS Deployment :** Given the highly technical skills required to implement and configure IDPSs and the imperfection of the technology, great care must be taken when deciding where to locate the components, both in their physical connection to the network and host devices and in how they are logically connected to each other and the IDPS administration team. Since IDPSs are designed to detect, report, and even react to anomalous stimuli, placing IDPSs in an area where such traffic is common can result in excessive reporting. Moreover, the administrators monitoring systems located in such areas can become desensitized to the information flow and may fail to detect actual attacks in progress.

Measuring the Effectiveness of IDPSs

When selecting an IDPS one typically looks at the following four measures of comparative effectiveness:

1. **Thresholds :** A threshold is a value that sets the limit between normal and abnormal behavior. Thresholds usually specify a maximum acceptable level, such as x failed connection attempts in 60 seconds, or x characters for a filename length. Thresholds are most often used for anomaly-based detection and stateful protocol analysis.

2. **Blacklists and whitelists :** A blacklist is a list of discrete entities, such as hosts, TCP or UDP port numbers, ICMP types and codes, applications, usernames, URLs, filenames, or file extensions, that have been associated with malicious activity. Blacklists, also known as hot lists, typically allow IDPSs to block activity that is highly likely to be malicious, and may also be used to assign a higher priority to alerts that match blacklist entries. Some IDPSs generate dynamic blacklists that are used to temporarily block recently detected threats (e.g., activity from an attacker's IP address).

   A whitelist is a list of discrete entities that are known to be benign. Whitelists are typically used on a granular basis, such as protocol-by-protocol, to reduce or ignore false positives involving known benign activity from trusted hosts. Whitelists and blacklists are most commonly used in signature-based detection and stateful protocol analysis.

3. **Alert settings :** Most IDPS technologies allow administrators to customize each alert type. Examples of actions that can be performed on an alert type include:
   a. Toggling it on or off
   b. Setting a default priority or severity level
   c. Specifying what information should be recorded and what notification methods (e.g., e-mail, pager) should be used
   d. Specifying which prevention capabilities should be used

   Some products also suppress alerts if an attacker generates many alerts in a short period of time and may also temporarily ignore all future traffic from the attacker. This is to prevent the IDPS from being overwhelmed by alerts.
4. **Code viewing and editing :** Some IDPS technologies permit administrators to see some or all of the detection-related code. This is usually limited to signatures, but some technologies allow administrators to see additional code, such as programs used to perform stateful protocol analysis.

4.1.2 Scanning and Analysis Tools

Scanner and analysis tools can find vulnerabilities in systems, holes in security components, and unsecured aspects of the network. The some information security experts may not perceive them as defensive tools, scanners, sniffers, and other such vulnerability analysis tools can be invaluable because they enable administrators to see what the attacker sees. Some of these tools are extremely complex and others are rather simple. The tools also range from expensive commercial products to free.

Many of the best scanning and analysis tools are those developed by the hacker community and are available free on the Web.

Types of Scanning and Analysis Tools

The types of Scanning and Analysis Tools are as follows :

1. **Port Scanners :** Port scanning utilities, or **port scanners**, are tools used by both attackers and defenders to identify (or fingerprint) the computers that are active on a network, as well as the ports and services active on those computers, the functions and roles the machines are fulfilling, and other useful information. These tools can scan for specific types of computers, protocols, or resources, or their scans can be generic. It is helpful to understand the network environment so that you can use the tool most suited to the data collection task at hand.

   For instance, if you are trying to identify a Windows computer in a typical network, a built-in feature of the operating system, nbtstat, may be able to get the answer you

need very quickly without the use of a scanner. This tool will not work on other types of networks, however, so you must know your tools in order to make the best use of the features of each.

2. **Firewall Analysis Tools :** Understanding exactly where an organization's firewall is located and what the existing rule sets on the firewall do are very important steps for any security administrator. There are several tools that automate the remote discovery of firewall rules and assist the administrator (or attacker) in analyzing the rules to determine exactly what they allow and what they reject.

   The Nmap tool mentioned earlier has some advanced options that are useful for firewall analysis. The Nmap option called idle scanning (which is run with the -I switch) will allow the Nmap user to bounce your scan across a firewall by using one or more idle DMZ hosts as the initiator of the scan. More specifically, since most operating systems do not use truly random IP packet identification numbers (IP IDs), if there is more than one host in the DMZ and one host uses nonrandom IP IDs, then the attacker can query the server (server X) and obtain the currently used IP ID as well as the known algorithm for incrementing the IP IDs.

   The attacker can then spoof a packet that is allegedly from server X and destined for an internal IP address behind the firewall. If the port is open on the internal machine, the internal machine re plies to server X with a SYN-ACK packet, which forces server X to respond with a TCP RESET packet.

3. **Operating System Detection Tools :** Detecting a target computer's operating system is very valuable to an attacker, because once the OS is known, all of the vulnerabilities to which it is susceptible can easily be determined.

   There are many tools that use networking protocols to determine a remote computer's OS. One specific tool worth mentioning is XProbe, which uses ICMP to determine the remote OS. This tool can be found at www.sourceforge.net/projects/xprobe. When run, XProbe sends many different ICMP queries to the target host. As reply packets are received,

   XProbe matches these responses from the target's TCP/IP stack with its own internal database of known responses. Because most OSs have a unique way of responding to ICMP requests, Xprobe is very reliable in finding matches and thus detecting the operating systems of remote computers. System and network administrators should take note of this and restrict the use of ICMP through their organization's firewalls and, when possible, within its internal networks.

4. **Vulnerability Scanners : Active vulnerability scanners** scan networks for highly detailed information. An active scanner is one that initiates traffic on the network in

order to determine security holes. As a class, this type of scanner identifies exposed usernames and groups, shows open network shares, and exposes configuration problems and other vulnerabilities in servers.

5. **Packet Sniffers :** Another tool worth mentioning is the packet sniffer. A **packet sniffer** (sometimes called a network protocol analyzer) is a network tool that collects copies of packets from the network and analyzes them. It can provide a network administrator with valuable information for diagnosing and resolving networking issues. In the wrong hands, however, a sniffer can be used to eavesdrop on network traffic. There are both commercial and open-source sniffers—more specifically, Sniffer is a commercial product, and Snort is open-source software.

An excellent free, client-based network protocol analyzer is Wireshark (www.wireshark.org), formerly known as Ethereal. Wireshark allows the administrator to examine data from both live network traffic and captured traffic. Wireshark has several features, including a language filter and TCP session reconstruction utility.

6. **Wireless Security Tools :** 802.11 wireless networks have sprung up as subnets on nearly all large networks. A wireless connection, while convenient, has many potential security holes. An organization that spends all of its time securing the wired network and leaves wireless networks to operate in any manner is opening itself up for a security breach. As a security professional, you must assess the risk of wireless networks. A wireless security toolkit should include the ability to sniff wireless traffic, scan wireless hosts, and assess the level of privacy or confidentiality afforded on the wireless network. In 2006, Insecure.org conducted a survey to identify the top five wireless tools.

4.1.3 Biometric Access Controls Devices

Biometric access control is based on the use of some measurable human characteristic or trait to authenticate the identity of a proposed systems user (a supplicant). It relies upon recognition— the same thing you rely upon to identify friends, family, and other people you know. The use of biometric-based authentication is expected to have a significant impact in the future as technical and ethical issues with the technology are resolved. Biometric authentication technologies include the following:

1. Fingerprint comparison of the supplicant's actual fingerprint to a stored fingerprint.
2. Palm print comparison of the supplicant's actual palm print to a stored palm print.
3. Hand geometry comparison of the supplicant's actual hand to a stored measurement.

4. Facial recognition using a photographic ID card, in which a human security guard compares the supplicant's face to a photo
5. Facial recognition using a digital camera, in which a supplicant's face is compared to a stored image
6. Retinal print comparison of the supplicant's actual retina to a stored image
7. Iris pattern comparison of the supplicant's actual iris to a stored image

Among all possible biometrics, only three human characteristics are usually considered truly unique. They are as follows:

1. Fingerprints
2. Retina of the eye (blood vessel pattern)
3. Iris of the eye (random pattern of features found in the iris, including freckles, pits, striations, vasculature, coronas, and crypts)

Figure depicts some of these human recognition characteristics.

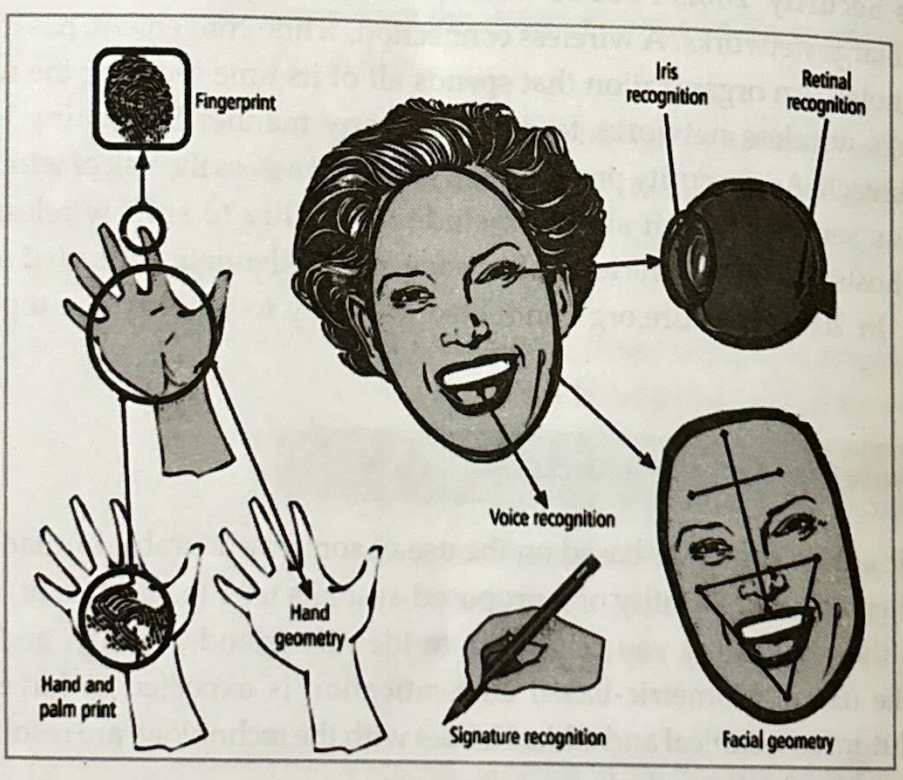

Figure : Biometric Recognition Characteristics

Most of the technologies that scan human characteristics convert these images to some form of minutiae. Minutiae are unique points of reference that are digitized and stored in an encrypted format when the user's system access credentials are created. Each subsequent access attempt results in a measurement that is compared with the encoded value to determine if the user is who he or she claims to be. A problem with this method is that

some human characteristics can change over time, due to normal development, injury, or illness, which means that system designers must create fallback or failsafe authentication mechanisms.

Signature and Voice Recognition Technologies

Signature and voice recognition technologies are also considered to be biometric access controls measures. Signature recognition has become commonplace. Retail stores use signature recognition, or at least signature capture, for authentication during a purchase. The customer signs a digital pad with a special stylus that captures the signature. The signature is digitized and either saved for future reference, or compared with a signature on a database for validation.

Currently, the technology for signature capturing is much more widely accepted than that for signature comparison, because signatures change due to a number of factors, including age, fatigue, and the speed with which the signature is written.Voice recognition works in a similar fashion in that an initial voiceprint of the user reciting a phrase is captured and stored. Later, when the user attempts to access the system, the authentication process requires the user to speak this same phrase so that the technology can compare the current voiceprint against the stored value.

Effectiveness of Biometrics

Biometric technologies are evaluated on three basic criteria: first, the false reject rate, which is the percentage of supplicants who are in fact authorized users but are denied access; second, the false accept rate, which is the percentage of supplicants who are unauthorized users but are granted access; and third, the crossover error rate, which is the level at which the number of false rejections equals the false acceptances.

1. False Reject Rate

The false reject rate is the percentage of identification instances in which authorized users are denied access as a result of a failure in the biometric device. This failure is known as a Type I error. While a nuisance to supplicants who are authorized users, this error rate is probably of least concern to security professionals since rejection of an authorized user represents no threat to security. The false reject rate is often ignored unless it reaches a level high enough to generate complaints from irritated supplicants.

Most people have experienced the frustration of having a credit card or ATM card fail to perform because of problems with the magnetic strip. In the field of biometrics, similar problems can occur when a system fails to pick up the various information points it uses to authenticate a prospective user properly.

2. False Accept Rate

The false accept rate is the percentage of identification instances in which unauthorized users are allowed access to systems or areas as a result of a failure in the biometric device. This failure is known as a Type II error, and is unacceptable to security professionals.

3. Crossover Error Rate (CER)

The crossover error rate (CER) is the level at which the number of false rejections equals the false acceptances, and is also known as the equal error rate. This is possibly the most common and important overall measure of the accuracy of a biometric system. Most biometric systems can be adjusted to compensate for both false positive and false negative errors. Adjustment to one extreme creates a system that requires perfect matches and results in high false rejects, but almost no false accepts. Adjustment to the other extreme produces low false rejects, but high false accepts. The trick is to find the balance between providing the requisite level of security and minimizing the frustration level of authentic users.

A biometric device that provides a CER of 1 percent is a device for which the failure rate for false rejection and the failure rate for false acceptance are both 1 percent. A device with a CER of 1 percent is considered superior to a device with a CER of 5 percent.

Acceptability of Biometrics

As you've learned, a balance must be struck between how acceptable a security system is to its users and how effective it is in maintaining security. Many biometric systems that are highly reliable and effective are considered somewhat intrusive to users. As a result, many information security professionals, in an effort to avoid confrontation and possible user boycott of the biometric controls, don't implement them. Table shows how certain biometrics rank in terms of effectiveness and acceptance. Interestingly, the order of effectiveness is nearly exactly opposite the order of acceptance.

Biometrics	Universality	Uniqueness	Permanence	Collectability	Performance	Acceptability	Circumvention
Face	H	L	M	H	L	H	L
Fingerprint	M	H	H	M	H	M	H
Hand Geometry	M	M	M	H	M	M	M
Keystroke Dynamics	L	L	L	M	L	M	M
Hand Vein	M	M	M	M	M	M	H
Iris	H	H	H	M	H	L	H
Retina	H	H	M	L	H	L	H
Signature	L	L	L	H	L	H	L
Voice	M	L	L	M	L	H	L
Facial Thermogram	H	H	L	H	M	H	H
DNA	H	H	H	L	H	L	L

Table : Ranking of Biometric Effectiveness and Acceptance

H=High, M=Medium, L=Low.

4.2 CRYPTOGRAPHY

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 **cipher text**) 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 cipher text message back into plaintext so that it can be readily understood.

The field of cryptology is so complex it can fill many volumes. This textbook provides only a general overview of cryptology and some specific information about cryptographic tools. In the early sections of this chapter you learn the background of cryptology as well as key concepts in cryptography and common cryptographic tools. In later sections you will learn about common cryptographic protocols and some of the attack methods used against cryptosystems.

Terminology

To understand the fundamentals of cryptography, you must know the meanings of the following terms:

1. **Algorithm:** The programmatic steps used to convert an unencrypted message into an encrypted sequence of bits that represent the message; sometimes refers to the programs that enable the cryptographic processes.
2. **Cipher** or **cryptosystem:** An encryption method or process encompassing the algorithm, key(s) or crypto variable(s), and procedures used to perform encryption and decryption.
3. **Ciphertext** or **cryptogram:** The encoded message resulting from an encryption **Code:** The process of converting components (words or phrases) of an unencrypted message into encrypted components.
4. **Decipher:** To decrypt, decode, or convert, ciphertext into the equivalent plaintext
5. **Encipher:** To encrypt, encode, or convert, plaintext into the equivalent ciphertext
6. **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 used by a computer program

7. **Keyspace:** The entire range of values that can be used to construct an individual key.
8. **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
9. **Plaintext** or **cleartext:** The original unencrypted message, or a message that has been successfully decrypted.
10. **Steganography:** The hiding of messages—for example, within the digital encoding of a picture or graphic
11. **Work factor:** The amount of effort (usually in hours) required to perform cryptanalysis to decode an encrypted message when the key or algorithm (or both) are unknown.

4.2.1 Foundations of Cryptology

Cryptology has a long and multicultural history. Table provides an overview of the history of cryptosystems.

Date	Event
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 by a tablet that was discovered to contain an encrypted formula for pottery glazes; the tablet used symbols that have different meanings than when used in other contexts.
500 B.C.	Hebrew scribes writing the book of Jeremiah used a reversed alphabet substitution cipher known as 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.
Fourth to sixth centuries	The *Kama Sutra of Vatsayana* listed cryptography as the 44th and 45th of the 64 arts (yogas) that men and women should practice:*(44) The art of understanding writing in cipher, and the writing of words in a peculiar way; (45) The art of speaking by changing the forms of the word.*
725	Abu 'Abd al-Rahman al-Khalil ibn Ahman ibn 'Amr ibn Tammam al Farahidi al-Zadi al Yahmadi wrote a book (now lost) on cryptography; he also solved a Greek cryptogram by guessing the plaintext introduction.

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, as well as ciphers with multiple substitutions, a technique that had never been used before.
1466	Leon Battista Alberti, the Father of Western cryptography, worked with polyalphabetic substitution and also designed a cipher disk.
1518	Johannes Trithemius wrote the first printed book on cryptography and invented a steganographic cipher, in which each letter 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 introducing 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 is called "The Vigenère 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 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.
1790s	Thomas 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 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	During World War I, the Germans, British, and French used a series of transposition and substitution ciphers in radio communications throughout the war. All sides expended considerable effort to try to intercept and decode communications, and thereby created 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 nonrepeating 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 protect the privacy of messages used in criminal activities.

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 to enable secure e-commerce, such as online banking and Web shopping.

4.2.2 Cipher Methods

There are two methods of encrypting plaintext: the bit stream method or the block cipher method. In the bit stream method, each bit in the plaintext is transformed into a cipher bit one bit at a time. In 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 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. Note that most computer-based encryption methods operate on data at the level of its binary digits (bits), but some operate at the byte or character level.

1. **Substitution Cipher :** To use a substitution cipher, you substitute one value for another, for example a letter in the alphabet with the letter three values to the right. Or you can substitute one bit for another bit that is four places to its left. A three-character substitution to the right results in the following transformation of the standard English alphabet:

   Initial alphabet yields ABCDEFGHIJKLMNOPQRSTUVWXYZ

   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. This type of substitution is based on a monoalphabetic substitution, because it only uses one alphabet. More advanced substitution ciphers use two or more alphabets, and are referred to as polyalphabetic substitutions.

2. **Transposition Cipher :** Like the substitution operation, the transposition cipher is simple to understand, but it can, if properly used, produce ciphertext that is difficult 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.

3. **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 0. 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.

4. **Vernam Cipher :** Also known as the one-time pad, the Vernam cipher, which was developed by 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—that is, a pad of paper. To perform the Vernam cipher encryption operation, the pad values are added to numeric values that represent the plaintext that needs to be encrypted.

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.

5. **Book or Running Key Cipher :** One encryption method made popular by spy movies involves using the text in a book as the key to decrypt a message. The ciphertext consists of a list of codes representing the page number, line number, and word number of the plaintext word. The algorithm is the mechanical process of looking up the references from the ciphertext and converting each reference to a word by using the ciphertext's value and the key.

6. **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 specific message and to confirm that there have not been any changes to the content. While they do not create a ciphertext, hash functions confirm message identity and integrity, both of which are critical functions in e-commerce.

Hash algorithms are public 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 compared with the recipient'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 same message always provides the same hash value, but the hash value itself cannot be used to determine the contents of the message.

4.2.3 Cryptographic Algorithms

In general, cryptographic algorithms are often grouped into two broad categories—symmetric and a symmetric—but in practice, today's popular cryptosystems use a hybrid combination of symmetric and asymmetric algorithms. Symmetric and asymmetric algorithms are distinguished by the types of keys they use for encryption and decryption operations.

1. Symmetric Encryption

Encryption methodologies that require the same secret key to encipher and decipher the message are using what is called 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 executed quickly by even small computers. 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 ciphertext) to avoid interception.

There are a number of popular symmetric encryption cryptosystems. One of the most widely known is the Data Encryption Standard (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, after which it became widely employed in commercial applications.

DES enjoyed increasing popularity for almost twenty years, until 1997, when users realized that a 56-bit key size did not provide acceptable levels of security. In 1998, a group called the 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 (not necessarily a specialized computer) can break a DES key in less than four hours.

2. Asymmetric Encryption

While symmetric encryption systems use 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 encryption 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 in 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.

Asymmetric algorithms are one-way functions. A one-way function is simple to compute in one direction, but complex to compute in the opposite direction. 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 essentially a summary of the original input values. It is virtually impossible to derive the original values without knowing how those 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." 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 the true key, and the private key is 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 and Netscape 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 learn how this algorithm works, see the Technical Details box entitled "RSA Algorithm."

Encryption Key Size

When deploying ciphers, users have to decide on the size of the cryptovariable or key. This is very important, because the strength of many encryption applications and cryptosystems is measured by key size. How exactly does key size affect the strength of an algorithm? Typically, the length of the key increases the number of random guesses that have to be made in order to break the code. Creating a larger universe of possibilities increases the time required to make guesses, and thus a longer key directly influences 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, to enable research to uncover their weaknesses.

In fact, 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.

4.2.4 Cryptographic Tools

The ability to conceal the contents of sensitive messages and to verify the contents of messages and the identities of their senders have the potential to be useful in all areas of business. To be actually useful, these cryptographic capabilities must be embodied in tools that allow IT and information security practitioners to apply the elements of cryptography in the everyday world of computing.

Types of Cryptographic Tools

This section covers a number of the more widely used tools that bring the functions of cryptography to the world of information systems

1. Public-Key Infrastructure (PKI)

Public-key Infrastructure (PKI) is an integrated system of software, encryption methodologies, protocols, legal agreements, and third-party services that enables users to communicate securely. 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:

1. Authentication: Individuals, organizations, and Web servers can validate the identity of each of the parties in an Internet transaction.
2. Integrity: Content signed by the certificate is known to not have been altered while in transit from host to host or server to client.
3. Privacy: Information is protected from being intercepted during transmission.
4. Authorization: The validated identity of users and programs can enable authorization rules that remain in place for the duration of a transaction; this reduces some of the overhead and allows for more control of access privileges for specific transactions.
5. Non-repudiation: Customers or partners can be held accountable for transactions, such as online purchases, which they cannot later dispute.

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

1. A certificate authority (CA), which issues, manages, authenticates, signs, and revokes users' digital certificates, which typically contain the user name, public key, and other identifying information.
2. A registration authority (RA), which operates under the trusted collaboration of the certificate authority and can handle day-to-day certification functions, such as verifying registration information, generating end-user keys, revoking certificates, and validating user certificates.
3. Certificate directories, which are central locations for certificate storage that provide a single access point for administration and distribution.
4. Management protocols, which organize and manage the communications among 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.
5. Policies and procedures, which assist an organization in the application and management of certificates, in the formalization of legal liabilities and limitations, and in actual business use.

Common implementations of PKI include systems that issue digital certificates to users and servers; directory enrollment; key issuing systems; tools for managing the key issuance; 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 engage in authenticated and secure communications and transactions.

2. Digital Signatures

Digital signatures were created in response to the rising need to verify information transferred via electronic systems. 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 is successful, the process verifies that the message was sent by the sender and thus cannot be refuted. This process is known as non-repudiation and is the principle of cryptography that underpins the authentication mechanism collectively known as a digital signature. Digital signatures are, therefore, encrypted messages that can be mathematically proven authentic.

The management of digital signatures is built into most Web browsers. The Internet Explorer digital signature management screen is shown in Figure. In general, digital

signatures should be created using processes and products that are based on the Digital Signature Standard (DSS). When processes and products are certified as DSS compliant, they have been approved and endorsed by U.S. federal and state governments, as well as by many foreign governments, as a means of authenticating the author of an electronic document. NIST has approved a number of algorithms that can be used to generate and verify digital signatures.

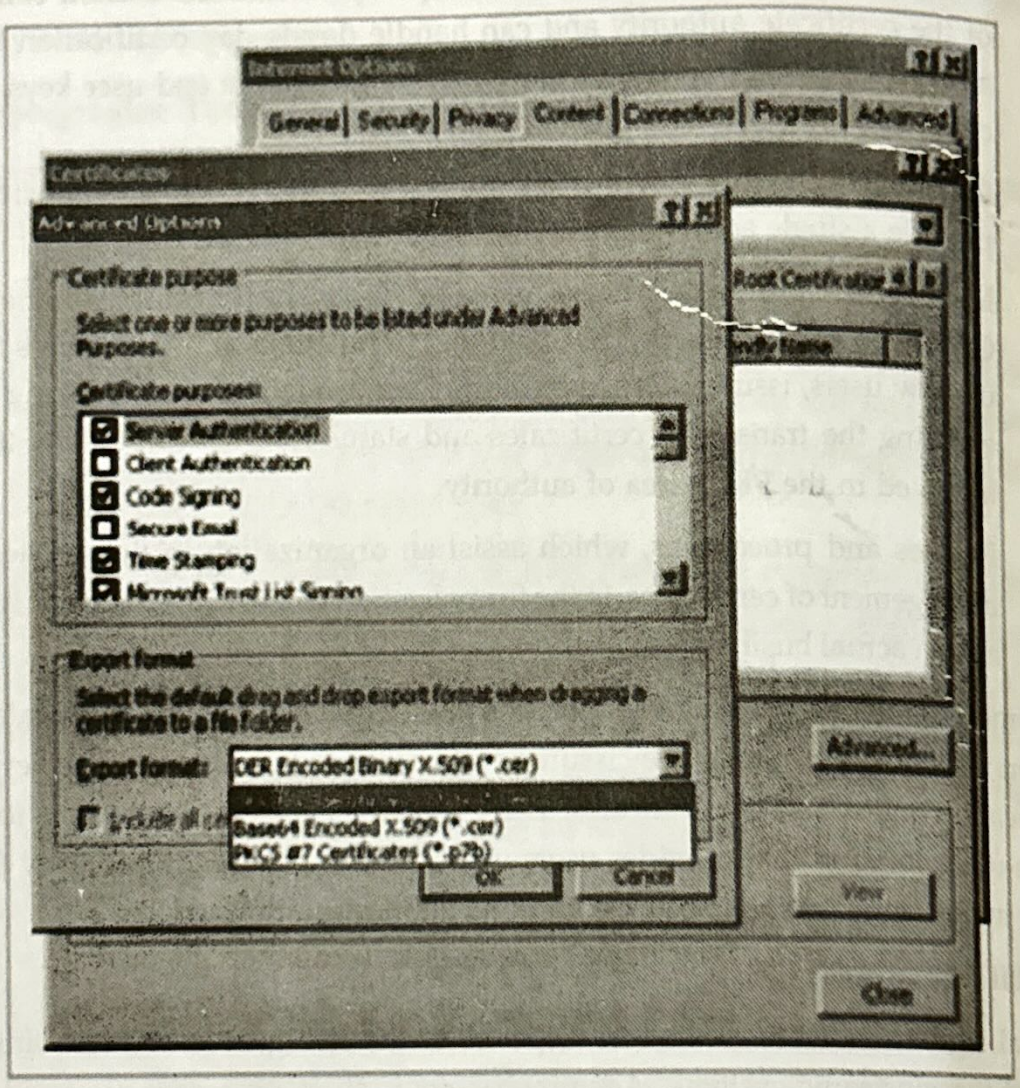

Figure : Managing Digital Signature

These algorithms can be used in conjunction with the sender's public and private keys, the receiver's public key, and the Secure Hash Standard (described earlier in this chapter) to quickly create messages that are both encrypted and nonrepudiable. This process first creates a message digest using the hash algorithm, which is then input into the digital signature algorithm along with a random number to generate the digital signature.

The digital signature function also depends upon the sender's private key and other information provided by the CA. The resulting encrypted message contains the digital signature, which can be verified by the recipient using the sender's public key.

3. Digital Certificates

As you learned earlier in this chapter, a digital certificate is an electronic document or container file that contains a key value and identifying information about the entity that controls the key. The certificate is often issued and certified by a third party, usually a certificate authority. A digital signature attached to the certificate's container file certifies the file's origin and integrity. This verification process often occurs when you download or update software via the Internet.

Client-server applications use different types of digital certificates

The Different client-server applications use different types of digital certificates to accomplish their assigned functions, as follows:

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

4. Hybrid Cryptography Systems

Except in digital certificates, asymmetric key encryption in its pure form is not widely used, but it is 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, which is a method for exchanging private keys using public key encryption. Diffie-Hellman key exchange uses asymmetric encryption to exchange session keys. These are limited-use symmetric keys for temporary communications; they allow two entities to conduct quick, efficient, secure communications based on symmetric encryption, which is more efficient than asymmetric encryption for sending messages.

Diffie-Hellman provides the foundation for subsequent developments in public key encryption. It protects data from exposure to third parties, which is sometimes a problem when keys are exchanged out-of-band.

5. Steganography

The word steganography the art of secret writing—is derived from the Greek words steganos, meaning "covered" and graphein, meaning "to write." The Greek historian Herodotus described one of the first steganographers, a fellow Greek who sent a message to warn of an imminent invasion by writing it on the wood beneath a wax writing tablet. While steganography is technically not a form of cryptography, it is another way of protecting the confidentiality of information in transit. The most popular modern version of steganography involves hiding information within files that contain digital pictures or other images.

4.2.5 Protocols for Secure Communications

Much of the software currently used to protect the confidentiality of information are not true cryptosystems. Instead, they are applications to which cryptographic protocols have been added. This is perhaps particularly true of Internet protocols; some experts claim that the Internet and its corresponding protocols were designed without any consideration for security, which was added later as an afterthought. Whether or not this is true, the lack of threats in the environment in which it was launched allowed the Internet to grow rapidly. But as the number of threats grew, so did the need for additional security measures.

1. Securing Internet Communication with S-HTTP and SSL

S-HTTP (Secure Hypertext Transfer Protocol) and SSL (Secure Sockets Layer) are two protocols designed to enable secure network communications across the Internet. S-HTTP and SSL ensure Internet security via different mechanisms and can be used independently or together. Netscape developed the Secure Sockets Layer (SSL) protocol to use public key encryption to secure a channel over the Internet, thus enabling secure communications. Most popular browsers, including Internet Explorer, use SSL. In addition to providing data encryption, integrity, and server authentication, SSL can, when properly configured, provide client authentication.

Secure HTTP (S-HTTP) is an extended version of Hypertext Transfer Protocol that provides for the encryption of individual messages transmitted via the Internet between a client and server. S-HTTP is the application of SSL over HTTP, which allows the encryption of all information passing between two computers through a protected and secure virtual connection. Unlike SSL, in which a secure channel is established for the duration of a session, S-HTTP is designed for sending individual messages over

the Internet and therefore a session for each individual exchange of data must be established. To establish a session, the client and server must have compatible cryptosystems and agree on the configuration. S-HTTP can provide confidentiality, authentication, and data integrity through a variety of trust models and cryptographic algorithms. In addition, this protocol is designed for easy integration with existing HTTP applications and for implementation in conjunction with HTTP.

2. **Securing E-mail with S/MIME, PEM, and PGP**

A number of cryptosystems have been adapted to work with the dominant e-mail protocols in an attempt to incorporate some degree of security into this notoriously insecure communication medium. Some of the more popular adaptations included Secure Multipurpose Internet Mail Extensions, Privacy Enhanced Mail (PEM), and Pretty Good Privacy (PGP).

3. **Securing Web Transactions with SET, SSL, and S-HTTP**

Just as PGP, PEM, and S/MIME work to secure e-mail operations, a number of related protocols work to secure Web browsers, especially at electronic commerce sites. Among these are Secure Electronic Transactions (SET), Secure Sockets Layer (SSL), Secure Hypertext Transfer Protocol (S-HTTP), Secure Shell (SSH-2), and IP Security (IPSec).

SSL uses a number of algorithms, but mainly relies on RSA for key transfer and IDEA, DES, or 3DES for encrypted symmetric key-based data transfer. Figure 8-8, shown earlier, illustrates the kind of certificate and SSL information that is displayed when you are checking out of an e-commerce site. If your Web connection does not automatically display such certificates, you can right-click in your browser's window and select Properties to view the connection encryption and certificate properties.

4. **Securing Wireless Networks with WEP and WPA**

Wireless local area networks (also known by the brand name Wi-Fi, or wireless fidelity networks) are thought by many in the IT industry to be inherently insecure. The communication channel between the wireless network interface of any computing device and the access point that provides its services uses radio transmissions. Without some form of protection, these signals can be intercepted by anyone with a wireless packet sniffer. In order to prevent interception of these communications, these networks must use some form of cryptographic security control.

Two sets of protocols are currently widely used to help secure wireless transmissions: Wired Equivalent Privacy and Wi-Fi Protected Access. Both are designed for use with the IEEE 802.11 wireless networks.

5. Securing TCP/IP with IPSec and PGP

Internet Protocol Security (IPSec) is an open-source protocol framework for security development within the TCP/IP family of protocol standards. It is used to secure communications across IP-based networks such as LANs, WANs, and the Internet. The protocol is designed to protect data integrity, user confidentiality, and authenticity at the IP packet level.

IPSec is the cryptographic authentication and encryption product of the IETF's IP Protocol Security Working Group. It is often described as the security system from IP version 6 (the future version of the TCP/IP protocol), retrofitted for use with IP version 4 (the current version). IPSec is defined in Request for Comments (RFC) 1825, 1826, and 1827 and is widely used to create virtual private networks (VPNs).

IPSec includes the IP Security protocol itself, which specifies the information to be added to an IP packet as well as how to encrypt packet data; and the Internet Key Exchange, which uses an asymmetric-based key exchange and negotiates the security associations. IPSec operates in two modes: transport and tunnel. In transport mode only the IP data are encrypted, not the IP headers. This allows intermediate nodes to read the source and destination addresses.

In tunnel mode the entire IP packet is encrypted and is then placed into the content portion of another IP packet. This requires other systems at the beginning and end of the tunnel to act as proxies and to send and receive the encrypted packets. These systems then transmit the decrypted packets to their true destinations.

4.2.6 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 ciphertext attacks 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, is used along with published frequency of occurrence patterns of various languages and can allow an experienced attacker to crack almost any code quickly with 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 ciphertext.

Occasionally, an attacker may obtain duplicate texts, one in ciphertext and one in plaintext, and thus 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 which cryptosystem is being employed.

Most publicly available encryption methods are generally released to the information and computer security communities to test 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 does prevent attackers from wasting their time, 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:

1. Man-in-the-middle,
2. Correlation,
3. Dictionary
4. Timing.

1. Man-in-the-Middle Attack

A man-in-the-middle attack, as you learned in Chapter 2, attempts to intercept 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.

To the victims of such attacks, 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 intended recipient. Establishing public keys with digital signatures can prevent the traditional manin-the-middle attack, as the attacker cannot duplicate the signatures.

2. Correlation Attacks

As the complexities of encryption methods have increased, so too have the tools and methods of cryptanalysts. 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 generated by the cryptosystem. Differential and linear cryptanalysis, which are advanced methods of code breaking that are beyond the scope of this text, 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 in a reasonable time, all messages written with that key can be decrypted. The only defense against this 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 key changes.

3. **Dictionary Attacks**

In a dictionary attack, the attacker encrypts every word in a dictionary using the same cryptosystem as used by the target in an attempt to locate a match between the target ciphertext and the list of encrypted words. Dictionary attacks can be successful when the ciphertext consists of relatively few characters, as for example files which contain encrypted usernames and passwords. An attacker who acquires a system password file 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 an attacker can almost always find at least a few matches in any stolen password file. After a match is found, the attacker has essentially identified a potential valid password for the system.

4. **Timing Attacks**

In a timing attack, the attacker eavesdrops on the victim's session and uses statistical analysis of 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. It may also eliminate some algorithms, thus narrowing the attacker's search and increasing the odds of eventual success. Having broken an encryption, the attacker 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 Against Attacks

Encryption is a very useful tool in protecting the confidentiality of information that is in storage or transmission. However, it is just that—another tool in the information security administrator's arsenal against threats to information security. Frequently, the uninformed

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 millennia, mankind has developed dramatically more sophisticated means of hiding information from those who should not see it, but no matter how sophisticated encryption and cryptosystems have become, they retain the flaw that was present in the very first such system: If you discover the key, that is, the method used to perform the encryption, you can read 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 outside the organization's control, and 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.

IMPORTANT QUESTIONS

1. Explain about Intrusion detection, access control and other security tolls.
2. Discuss briefly about Intrusion detection and prevention systems.
3. Explain about Scanning and analysis tools and Access control devices.
4. What is Cryptography? Explain about Foundations of cryptology.
5. Discuss briefly about Cipher methods, Cryptographic Algorithms and Cryptographic tools.
6. Explain about Protocols for secure communications and Attacks on cryptosystems.