

Chapter 1

Introduction

1.1 A HISTORICAL PERSPECTIVE OF INFORMATION AND NETWORK SECURITY

1.1.1 Hidden Messages

Delivering messages in secrecy has been a serious concern since antiquity. Messages that conveyed personal, business, or state affairs were very critical for the well-being of a person, business or country, and as history has shown even in more recent times, the outcome of a war depended on the prompt and safe delivery of a critical message. The players involved in the transport of a secret message are the author and rightful sender, the courier or the transporting medium, the authorized receiver and the interceptor. Because the sender of the secret message was aware that there are those other than the authorized recipient who would attempt to gain knowledge of the content of the secret message, the sender used a coding method to encrypt the message and assure secrecy. The courier was initially a trusted person who at risk of life had to deliver the message to the authorized recipient. The interceptor, depending on sophistication and opportunistic factors, had several choices: attack and capture the message; attack and destroy the message; acquire knowledge of the message content but do not alter it; get hold of the message, alter it and send it to the recipient. The authorized recipient of the message should be able to decode the message, verify the authenticity of the received message and also detect if the message was intercepted and altered.

The lessons learned over time forced senders to use more and more complex cryptographic methods to outsmart sophisticated and knowledgeable attackers.

The content of this book is intended to have illustrative and educational value and it should not be regarded as a complete specification of cryptography, network security or any protocol described herein. The information presented in this book is adapted from standards, published knowledge, and from the author's research activities; however, although a serious effort has been made, the author does not warrant that changes have not been made and typographical errors do not exist. The reader is encouraged to consult the most current standards recommendations for information and for network security.

Appreciating the art of cryptography, its sophistication, and how it evolved to its current status requires a review of some characteristic examples.

Ancient Mesopotamians wrote a private message in cuneiform script on a fresh clay tablet, which was exposed to the sun to dry. This tablet was then enclosed in a clay envelope on which the addressee's name was written (Figure 1.1). When the envelope was dry, it was dispatched with a trusted messenger. If this was intercepted, the clay envelope had to be broken, revealing to the recipient that the message was compromised. In other cultures, clay was substituted by papyrus, tree bark (such as white birch), carta-pergamena or parchment (processed skin of baby lamb or goat), or by fabric or paper on which text was written with a stylus and ink, and then rolled or folded and sealed with Spanish wax on which a symbol was impressed using either a signet ring or a stamping tool.

In addition to fabric and paper, the ancient Chinese had a method of hiding written messages hidden in cakes (known as *moon cake*). This cake was given to a trusted agent who passed it through the unsuspected guards of the gate. Similarly, the Egyptians used their own secret methods as is discerned in hieroglyphs, and so did the ancient Greeks as is discerned in ancient texts. It's worth to outline the following three examples because they have some similarities with modern encryption methods.

In all cultures, the trusted courier has been the most popular method of sending a secret short message, which was memorized so that there was no physical evidence of the message. A notable example is that of the soldier Philippides who ran from Marathon to Athens, a distance of about 40 km, to deliver to the Athenians the message that the battle at Marathon was *won*, a decisive victory for spreading civi-



Figure 1.1. Clay tablets enclosed in clay envelopes assured secrecy and authenticity of the message.

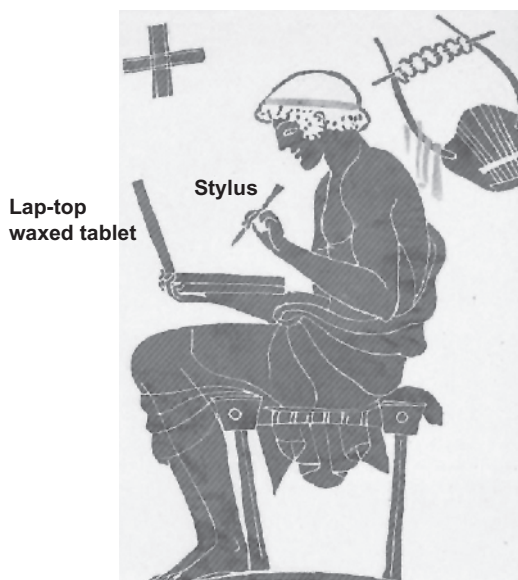


Figure 1.2. Hinged wooden tablets on which wax was spread was the scratch pad of antiquity. Scraping the wax, writing a secret message and spreading the wax on top of it was a form of cryptography of yester day.

lization and democracy; the modern *Marathon race* commemorates this victory and Philippides' accomplishment.

The Athenian Demaratus used a different method to transport a written message through the enemy ranks, which gave birth to the modern terms “*cryptography*” (from *crypto* and *graphe* or “hidden message”) and also “*steganography*” (from *steganos* and *graphe* or “sealed message”). He used waxed tablets (a popular writing medium like a paper scratch pad is today) that were hinged like a laptop (Figure 1.2). He scraped off the wax, wrote the message on the wooden surface with a carbon stick (a pencil) and then spread the wax over it, on which he wrote with a stylus an unclassified message. According to Herodotus, Demaratus' method worked very effectively and efficiently [1].

1.1.2 Encoded Messages

For long distances and for rapidly transmitting messages, the Greeks understood this can be done only with light. To accomplish this, small towers were built on top of hills and mountains forming a network. From these towers, optical messages were transmitted quickly, reaching far destinations. This method is discerned in Aeschylus' play *Agamemnon*, the Greek campaign general in the Trojan War (ca. 12th c. BCE) [2], and currently the method has been dubbed the *Agamemnon's Link*. According to Aeschylus, it took only few hours for a message to arrive at Argos from Troy, a

distance more than 600 kilometers; 3000 years ago, this was a remarkably short time [3]. Similarly, Herodotus wrote of communicating with light over long distances at the battle of Thermopylae (captured by Hollywood in the movie *300*).

What is not known is if and how optical messages were encoded then. However, historians do know that by about 350 BCE, optical messages were encoded according to a method developed by the military scientist Aeneas Tacitos of Stymphalos. According to the Greek historian, cryptographer and navigator Polybius (203–120 BCE), *communicating with encrypted light messages became the greatest service in war* [2].

It is believed that the encryption key changed from hour to hour (a method now known as the *cryptoperiod*) using a clepsydra, a water clock made with a leaky jug which had markings on the interior calibrated so that the water level in it determined the cipher key to be used.

Polybius, too, wrote about encoding light messages. Although his method was initially invented by *Cleoxenos The Engineer* and *Demokleitus The Inventor*, Polybius perfected it, and it is currently known as the *Polybius square*. This method is of current interest to cryptography, and is thus described below.

The alphabet letters were arranged in a 5×5 matrix; this example uses, the Latin alphabet and therefore since there are 26 letters, two least used letters (Y and Z) are placed in the same matrix element. Each row of the matrix is written in a tablet, letters on the tablet are numbered left to right from one to five, and each tablet is numbered from one to five (Table 1.1).

Both the transmitting and the receiving site had five lit torches. When the transmitting site wanted to send a message, the guards raised two torches, and the receiving site in response raised two torches as well, which were subsequently lowered; this established a “*request to send*” and the “*acknowledgment*” steps of modern communications protocols.

Now, to transmit the message VICTORY, the message was written on a tablet and each letter was encoded by writing the tablet number where the letter is, followed by the letter number on the tablet; the word VICTORY was then transformed to a sequence of two-digit numbers: 52, 24, 13, 45, 35, 41 and 55. Now, to send the first letter, five torches were raised and lowered followed by two torches; to make up the number 52 (for letter V). Then, two torches were raised and lowered, followed by four torches to make up the number 24 (for letter I), and so on.

Table 1.1. Arranging the alphabet in a matrix and on numbered tablets

	1	2	3	4	5
Tablet #1	A	B	C	D	E
Tablet #2	F	G	H	I	J
Tablet #3	K	L	M	N	O
Tablet #4	P	Q	R	S	T
Tablet #5	U	V	W	X	Y/Z

As trivial as this may seem, because of the distance between two towers and in order to remove errors based on human factors, Polybius provides training guidelines, optimum distance between torches, dimensions of relay towers, “viewing tubes” and screens, and more. What he did not publicize, for obvious reasons, is whether the letters of the alphabet were arranged on the tablets in an linear order as in the preceding example or if they were arranged in a random order, in which case all stations on the transmitting path should have copies of the same tablets, which also could change periodically to assure secrecy of the code (Table 1.2).

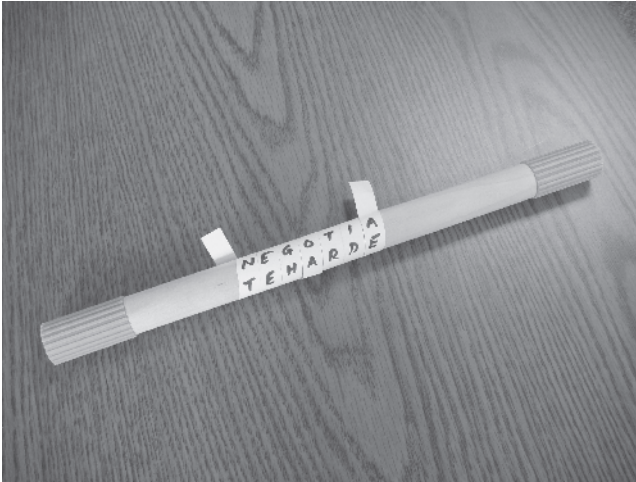
In addition to light as means of transmitting messages, Spartan ambassadors used another sophisticated method. Wrapping a ribbon helicoidally around a baton or staff of specific diameter that the Spartans called “skytale” (this was carried by all ambassadors), and then writing a message alongside would produce an unintelligible message on the unraveled ribbon; the method was recently used in a Hollywood movie (Figure 1.3). The message could only be read if the recipient had a baton of the same diameter with the original and if the ribbon would be wrapped around the baton in the same helicoidal sense and direction. Figure 1.4 illustrates the same phrase but wrapped in the opposite sense yielding an unintelligible message.

During the Roman era, Julius Caesar (100–44 BCE) proposed an encryption method, hence known as *Caesar’s Cipher*. This method shifted the letter in the message to another letter of the alphabet in a linear manner. For example, if a letter in a message is replaced by the next letter in the alphabet, the word *CAESAR* becomes *DBFTBS*, and if by the one after the next, then it becomes *ECGUCT*. Notice however that with this algorithm the frequency of occurrence of letters (such as A in Caesar) remains, although encoded to another letter; the frequency of occurrence of letters in words in a text reveals a vulnerability that modern crypto-analysts use to break the cryptographic algorithm and find the cipher key. The Caesar’s algorithm is considered trivial by modern cryptographers, who extended it to a more general encryption algorithm, by which each letter of a text is replaced by another letter of the alphabet according to a specified random shifting algorithm. This cryptographic method is currently known as *Shift Cipher* and it uses randomized arithmetic based on modulom operations.

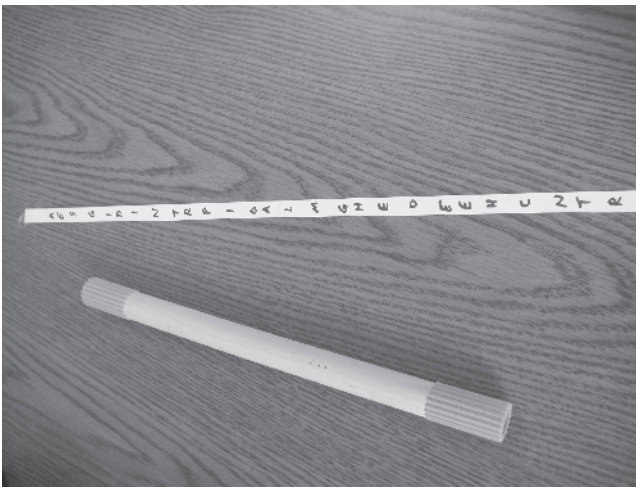
In recent times and until WWII, German intelligence used an encryption method known as the *enigma*, based on a modified typewriter, the *enigma typewriter*, which

Table 1.2. Arranging the alphabet in random order provided additional message security

	1	2	3	4	5
Tablet #1	A	E/Z	I	G	M
Tablet #2	J	B	Q	U	S
Tablet #3	C	V	D	O	X
Tablet #4	T	Y	N	W	P
Tablet #5	H	L	F	R	K



(a)



(b)

Figure 1.3. (a) A staff and a ribbon wrapped around it provided means of encoding a secret message such as “NEGOTIATE HARDER HELP IS ON THE WAY” (b) The unwrapped ribbon was unintelligible: AESGIDINTRIOALMGHEOEEHCNTR.

encrypted messages as they were typewritten. The enigma typewriter consisted of three alphabets that rotated separately after an alphabet key was depressed, thus yielding a combination of $26 \times 26 \times 26 = 17,576$ alphabets [4, 5]; these were reflected by a “mirror” causing three more transpositions. Thus, as the operator typed each letter of the message on the keyboard, the enigma typewriter scrambled it to another, producing an unintelligible message. The exact relationship of the three alphabets as they rotated on the drums established the “cryptographic key.” Conversely, by

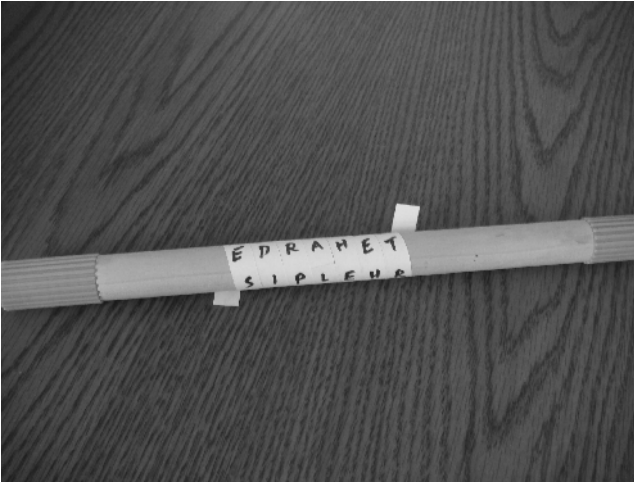


Figure 1.4. The ribbon of Figure 1.3 wrapped in the opposite sense also yields an unintelligible phrase.

typing the scrambled message on the enigma typewriter with the same key, the original message was recovered. Clearly, this method required two exact replicas of the enigma typewriters to cipher and decipher a message and therefore transporting the machines and the keys was done in utmost secrecy because the other side was very eager to know how the machine worked and what the key was. A WW II German Enigma machine is kept in the United States National Cryptologic Museum (in the collection of artifacts related to cryptology), located adjacent to the National Security Agency (NSA) Headquarters, Fort George G. Meade, Maryland.

With the evolution of radio transmission and particularly during World War II, message encoding gained particular importance because electromagnetic waves would reach both friendly and foe antennas. As a result, cryptography entered the realm of science (modulation methods) and mathematics (statistics, probability, and number theory) and several encryption algorithms were developed to provide authentication, no repudiation, data integrity, and confidentiality. Such algorithms required two fundamental elements: a unique *cipher key* that ciphers and deciphers a message, and a unique *key distribution method* so that no one can successfully intercept it.

In parallel to this, another effort of equal importance has been on the way: intercept the transport layer, copy the secret message (or cipher text) and try from this to figure out the secret key (that is, perform an automatic analysis of the cryptographic algorithm), or interrupt the key distribution process to disable secure communication. This effort can be used to test the hardness of the cryptographic method, the robustness of the distribution mechanism and also identify vulnerabilities; this is known as *cryptanalysis*. On the negative side, cryptanalysis can also be used by malicious attackers. It is typical that new encryption methods are internationally challenged at a prize for unbreakability or breakability of the cipher text and tolerance of the key distribution method.

1.2 MODERN CRYPTOGRAPHY, WATERMARKING, STEGANOGRAPHY, ESCROW AND CRYPTANALYSIS

1.2.1 Cryptography

Cryptography is a method that, based on an algorithm or some other method, transforms plain text (data) into unintelligible or undetectable *cipher text* to all but the authorized recipient who possesses special knowledge to convert the cipher text back to the original plain text. Based on this, if the cipher text is viewed by an unauthorized party, an eavesdropper or malicious attacker, the original text cannot be read from it; thus, unintelligible means that the plain text has been transformed to an alpha-numeric or binary string that makes no sense, or that the cipher text may seem intelligible but it is not the original plain text. For example, consider the plaintext “TRANSFER_ACCOUNT_TOMORROW_AM”. Using a cryptographic algorithm, this is transformed to the unintelligible string “SIGPZFKV_BCIRQQT_QNPRQSX_GP,” or using double meaning linguistic to the undetectable cipher text “BEES GO FOR HONEY AT SUNRISE,” or some other intelligible phrase that triggers some specific action without explicitly revealing to a third party what the action is. The latter is widely used in many cases particularly in the classified section of newspapers, over radio waves and TV; only the knowledgeable receiver understands the exact meaning of the message.

In digital communication systems, information is transmitted in binary form; alphanumeric, math and other symbols in a text are each converted in eight-bit bytes and transmitted as a string of ones and zeros. However, prior to transmitting this binary string, the string is scrambled with a cipher key that preferably is as long as the string. As an example, if the message in binary notation is 10001101000011011010011011 and the established cipher key is 010101001111101101000110111, where the vertical bar | delineates each byte, then the cipher text is obtained by bit modulo-2 or Exclusive OR (XOR) logic operation yielding the cipher text 1101100111111011010101111. If C_k is the cipher-key and T_x is the plain text then the cipher text C_T is $C_T = C_k \otimes T_x$. The XOR function is very convenient because at the receiver, the original plain text is recovered from $T_x = C_T \otimes C_k$ as a result of the XOR identity (if $A = B \otimes C$ then $B = A \otimes C$ and $C = A \otimes B$). This is known as *decrypting* or *deciphering*.

It is evident that deciphering requires that the cipher text has not errored bits as a result of noise, attenuation and pulse shape deformations. Similarly, it is important that the cipher key is bit synchronized with the cipher text.

In the example above, when the message is represented by a block of bytes, the cipher key is known as *block cipher*. Certain encryption algorithms use a long key to create a cryptographically strong *keystream*. It is this keystream that is exclusive-ORed with the plain text; this algorithm is known as *stream cipher*. Cipher keys may be *permanent*, they may change periodically (in which case a *cipherperiod* is defined), or they may change for each message (known as *one-time pad keys* or *ephemeral keys*).

1.2.1.1 *Symmetric and Asymmetric Keys*

There are two fundamental cryptographic methods in ciphering/deciphering. One uses the same key at both ends of the channel and the other different keys at each end.

- When the cipher key is the same at both ends of the cryptographic channel, then cryptography is referred to as *symmetric cryptography* and the key as *symmetric key*. To generate the symmetric key another symmetric key may be needed first, the *seed key*, which is used to calculate the final cipher key, hence known as the *key encrypting key*; this process is known as *key wrapping*. In applications where the text is binary serial (in data communications), the cipher key operates on the serial bit-stream a bit or a byte at a time using XOR. Such cryptographic methods are also known as *stream ciphers*. The stream cipher text is decoded serially by applying the same XOR logic operation with the same key (as in the aforementioned example). When the key is generated by a pseudo-random generator, it is called a *key-stream generator*. One of the issues with symmetric cryptography is the *key distribution*. That is, the transportation or communication of the key from the sender of the cipher text to the rightful recipient of it. If the cipher key is intercepted and copied by an intruder during its transportation, the value of the cipher text becomes meaningless. However, symmetric cryptography may become stronger if the symmetric key changes often and randomly according to a secretly timed algorithm.
- When the cipher key is not the same at either end of the cryptographic channel, then cryptography is referred to as *asymmetric cryptography*. Based on this, the encoding end uses one key that keeps it secret, and the decoding end another key that also keeps it secret; the two keys are mathematically interrelated. Thus, even after intercepting the cipher text, it is very difficult to decipher it or identify the decoding key. In some asymmetric cryptographic systems, a publicly transported key is needed to calculate the decoding key, known as a *public key* and hence *public key cryptography*; in this case, the public key is transported over a *public channel* (such as wireless). Intercepting the public key does not help in decoding because the mathematical algorithm that produces the deciphering key is either secret or too difficult. Thus, the asymmetric key method has a lot of appeal and it is the method that has inspired some more complex cryptographic methods applicable to the Internet and others. To summarize, public key cryptography uses a pair of two key ciphers, a public and a private. Plain-text encrypted with the public key can only be decrypted with the associated private key. Public key cryptography is also used in conjunction with one-way hash functions to produce digital signatures. According to this, messages signed with the private key can be verified with the public key [13, 14].

1.2.1.2 Hash Functions

Several algorithms map a bit string (or plain text) of arbitrary length to a bit string of fixed length, known as *message digest*, according to an approved function known as *one-way encryption* or *hash function*. The result of applying the hash function on the bit string is the *hash value* or a *digital fingerprint* of a file's contents. Thus, hash functions provide a measure of the integrity of a message. Hash functions satisfy the property that it is computationally infeasible to map any input string length to a pre-specified output bit string, and also that it is computationally infeasible for two distinct input strings to map onto the same output pre-specified string. Thus, the output is always the same for the same input.

There are two primary hash functions in use, the message digest 5 (MD5) and the secure hash algorithm-1 (SHA-1).

- MD5 was developed by RSA labs and SHA-1 by the National Security Agency (NSA).
- SHA-1 supports messages up to 2^{64} bits at the input and it produces a 160-bit digest [6–8].

The *Secure Hash Algorithm* (SHA) is a public key protocol included in Suite B, consistent with the National Institute of Standards and Technology (NIST) publications. Yet, vulnerabilities or insecurities are not absent from these algorithms [6] and recently it was announced that the SHA-1 algorithm was broken by a Chinese research team (March 1, 2005, *The Australian*). In cryptosystem language, the public and private keys, symmetric keys, block ciphers, and hash functions are known as *cryptographic primitives*.

1.2.1.3 Security Services

The information security services that are performed in cryptography are *authentication* (of source and destination), *authorization*, *data integrity*, *data privacy*, *message confidentiality*, and *non-repudiation*.

- **Authentication** is a process that verifies the identity of source and/or destination, verifies that the received message was sent by the rightful sending entity, and that the received message has not been altered during transmission [9–11]. Authentication is accomplished with cryptographic checksums known as the *authentication code* calculated according to an approved cryptographic algorithm; the authentication code is also known as *message authentication code*. A message authentication code is a one-way hash function computed from a message and a secret key. It is difficult to forge without knowing the secret key.
- **Authorization** is a process that grants access privileges to an entity such as the intended destination or to a third party. Authorization to the destination is granted after the destination authentication. Access authorization to a third party is granted only after an official and justified request to access a message

and perhaps modify the message. Access is under the control of an *access authority* (or function), which is responsible for monitoring and granting privileges to other authorities that request access.

- **Data integrity** pertains to identifying possible unauthorized alterations in the transported information. In communications, one of the mechanisms to secure document or data integrity is digital signatures and steganography; that is a non-removable signature that authenticates the original text or invisible text superimposed to the original text which only the rightful recipient knows how to remove. Thus, when part or all of the text is altered by an unauthorized entity, the signature or the steganogram has been altered as well.
- **Data privacy** secures the authorship of data, such as for example the ownership and copyright of a text, picture, or movie. This is accomplished by superimposing a visible text or a symbol (A) over the original data (B) such that the two ($A + B$) become inseparable using a cryptographic key; this is known as *watermarking*. Altering watermarked data also alters the original data. Thus, watermarking does not make a text unintelligible and undetectable, but it secures the ownership of it. A common application of watermarking is the word *draft* or *classified* across the page of text, or the name of the photographer on a picture.
- **Message confidentiality** is the service that warranties that information during its transport from source to destination will not be disclosed to one or more unauthorized parties. Cryptography with strong encoding is the mechanism that ensures message confidentiality. The network also provides an *accountability* function that monitors and ensures that actions of any entity on the network security can be traced back to it.
- **Non-repudiation** provides proof of the integrity and of the origin of a message to a third party. Non-repudiation prevents an entity from denying sourcing or receiving a message. For example, a signed message provides undeniable proof that the message was sourced by the transmitting entity and a coded time stamp that the message was received by the rightful owner. The digital signature is calculated with a secret key. In similar applications, the equipment signature ID of the sourcing computer is embedded on data prior to transmitting, and the signature ID of the receiving computer is embedded as soon as it is received. The equipment signature ID is a unique code that identifies the particular end device; it is encoded and it cannot be duplicated.

1.2.2 Watermarking

Digital watermarking is a cryptographic method that safeguards the copyright and the ownership of data and its privacy. Watermarking is not designed to embed a hidden message like steganography does, but to fuse a notice (A) with a target file (b) that contains text, image (photo, video, etc.) or audio, so that an unauthorized user cannot extract from the fused file ($A + B$) the target file. Based on this, stock

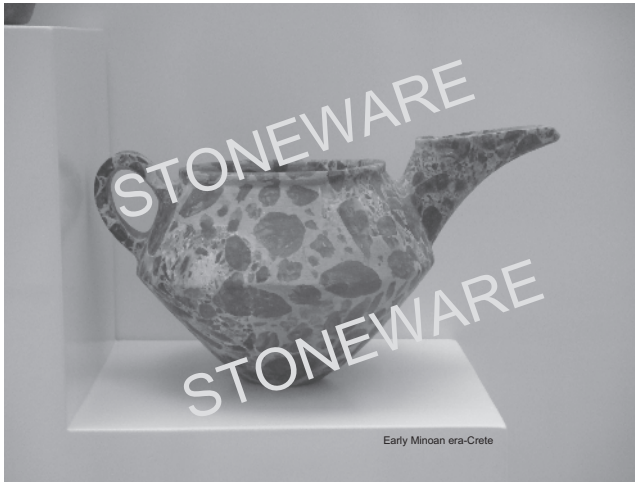


Figure 1.5. An image fused with a label produces a watermarked image, which can be separated with a key that only the owner of the image holds.

photography agencies that display their images on the web, watermark those images with their logo or agency name; when an end user purchases the rights to use, then the agency removes the watermark (Figure 1.5). The watermark, at the pixel level, has typically lower amplitude than the target file and it appears semitransparent when superimposed with text or image. In a different application, watermarking has been used on paper money to warranty authenticity. Commercial software watermarks text or image.

1.2.3 Steganography

A form of cryptography known as *steganography* is a technology that embeds a secret text (A) within a non-secret text (B) the *carrier* to form a composite text ($A + B$) known as *steganogram*. In this case, the steganogram looks like the intelligible and detectable text B , but embedded text A , is only detectable or intelligible to the rightful and authorized recipient who holds the proper *steganographic key*; to the naked eye, the hidden text A is invisible.

Steganography takes advantage of quantization noise after digitizing an analog signal (picture, voice or text). Thus, if the steganogram ($A + B$) has the same statistical characteristics of the carrier signal B and the *steganographic key* is not known, then it is very difficult to extract signal A from the steganogram. We can parallel steganography with traditional telephony that encodes analog signals using μ -law coders; because some small bandwidth is needed for signaling, the least significant bit of certain time slots is superimposed by signaling bits, a method known as *bit robbing*; this is a well-tested and well-used method and it does not affect the quality of voice signal when it is heard by the callee. Thus, bit robbing may be considered

the first form of steganography in telecommunications. One of the applications in steganography uses a digitized color picture in which some bits at random (according to a secret key) have been altered to host bits of a digitized secret text. To retrieve the secret message in it, one must know the algorithm and find the embedded bits in it.

1.2.4 Escrow

In cryptography, the term *key escrow system* or *escrow* means that the two components that comprise a cryptographic key are entrusted to two key holders called *escrow agents*. The escrow agents provide the components of the key to a *grantee* entity only upon fulfillment of pre-specified conditions. The grantee entity reconstructs a unique key from the two components and generates the session key, which is then used to decrypt the cipher text [12].

An overall end-to-end cryptographic process consists of the transmitter or source with an algorithm that generates a secret key A, the receiver or destination with another algorithm that generates a secret key B, and the link or medium between the two ends which may be attacked; the attacker has means for capturing or discovering secret keys (Figure 1.6). The major functions at the two ends are encapsulated as:

- An algorithm at the transmitter that generates a key (or cipher key); this key may consist of more than one component.
- A secure mechanism for transporting the key to the receiver; the transported key may or may not be the same with that at the source. The method by which the cipher key is exchanged and agreed upon by both sender and receiver is known as *key establishment*.

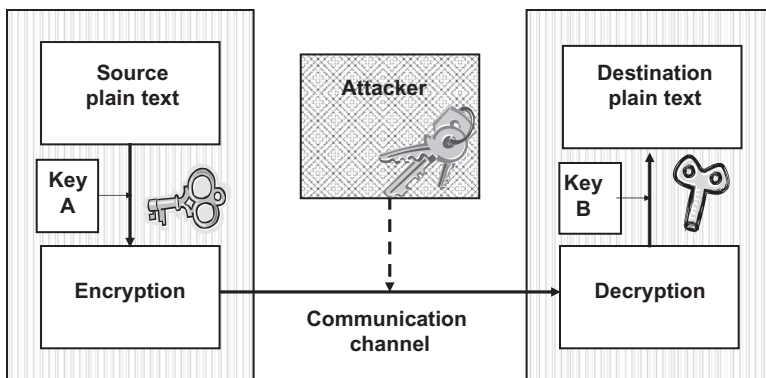


Figure 1.6. At both ends of a communications channel the plain text is ciphered using a secret key and is deciphered to plain text using the same or another cipher key. Bad actors evaluate vulnerabilities of the channel and cryptographic systems to attack.

- A secure process and a key *registration authority* that registers the agreed upon key in a *key depository* or *key management archive*.
- A secure mechanism at the transmitter that transforms the original text (or *plain text*) to an encrypted text (or *cipher text*). This process is called *encoding* or *ciphering*.
- A secure medium for transporting the cipher text to the rightful recipient.
- A protocol at the receiver that authenticates the source and one that authenticates the destination. In other words, a *public key certificate* that uniquely identifies an entity, contains the entity's public key, uniquely binds the public key with the entity, and a trusted *certificate authority* that digitally signs it, as part of the X.509 protocol (or ISO Authentication framework).
- An algorithm that transforms the received cipher text back to its original form, the plain text, using the decipher key. This process is called *decoding* or *deciphering*.

1.2.5 Cryptanalysis

If cryptography tries to cipher a text and make it unintelligible to an unauthorized agent, cryptanalysis tries to break cryptography and discover the cipher key or read the unintelligible cipher text using various methods that include mathematics, social engineering, deception and other useful clues that help to break the code.

In its pure form, cryptanalysis does not necessarily know the cipher key or the cryptographic algorithm and therefore it is much more difficult than cryptography. Examples include the decipherment of ancient scripts that were unreadable for millennia (such as, the Mesopotamian cuneiform, and the Mycenaean Linear B scripts, the Mayan glyphs and the Egyptian hieroglyphs), as well as some that are still unbreakable (such as the Harappan and the Minoan Linear A scripts). The term cryptanalysis is a compound word from Greek *kryptós* (hidden) and *analysis* (to untie or to *analyze*).

Cryptanalysis is also used to test the hardness of cryptography. The faster a cipher text is deciphered with cryptanalysis, the softer the cryptographic algorithm is. Therefore, new cryptographic algorithms challenge cryptanalysts and offer a reward or a prize if and when they break the secret key. On the negative side, bad actors also try to break the secret key for their own profit in any way they can, using cryptanalysis and also other illegal and unethical means (including social engineering tactics). Hollywood has captured this in many spy movies.

Malevolent cryptanalysts also use methods for facilitating and expediting the code breaking. Among them are capturing a plain text and its corresponding cipher text from which they determine the cryptographic system and algorithm; this is extremely valuable information because they can decipher future cryptograms much more easily, even if the cipher key has changed. In certain cases, bad actors impersonate a third party and transmit a short plain text to the source, which encrypts it; capturing the encrypted message will provide the desirable result to the bad actor.

In a different scenario, the attacker may capture the same plain text that has been ciphered with two different keys. Although the keys are not known, their interrelationship is found as well as the workings of the cryptosystem.

Currently, cryptanalysis is classified as either linear or differential.

- **Linear cryptanalysis** is based on probabilities and it takes advantage of the statistical occurrences in the cipher text. The premise is that the cipher breaker already has some statistical information of known plain texts and their corresponding cipher texts. Thus, if the cipher process is considered to be a black box, its input the plain text and its output the cipher text, the cipher breaker tries to discern the general workings of the black box.
- **Differential cryptanalysis** exploits the statistical behavior of plain text differences and the differences in corresponding cipher texts. That is, this class of cryptanalysis examines the difference ΔX of plain texts X_i , and the difference ΔY of their corresponding cipher texts Y_i at the output of the black box. Differential cryptanalysis assumes that the attacker has knowledge of or is able to select plain texts and also to observe their corresponding cipher texts from which ΔX and ΔY are calculated.

In conclusion, with modern cryptography there is also modern cryptanalysis, which is based in problem or puzzle solving techniques using computers. Because a cryptosystem may be based on symmetric or asymmetric keys and on different mathematical approaches, cryptanalysis may use many computers to collectively and in parallel try to solve the cipher problem. With fast execution time, computers have been successful and have broken several codes.

1.3 NETWORK SECURITY

Up to this point, we defined vulnerabilities associated with computer-based nodes and end-terminals, which have attracted a number of different attack types. In traditional synchronous communications, accessing the loop of a circuit-switched network required moderate networking know-how to tap a two-wire pair to eavesdrop on a conversation, and more know-how to mimic signaling codes with the so called “blue box,” and establish end-to-end connectivity without being billed. Despite these attacks, the synchronous network was not the subject of attacks as the data networks are. The reason is that data did flow continuously without being buffered or stored, as a characteristic of the circuit switched method. In addition, demultiplexing time-slots at the core network required specialized equipment and substantial know how of complex protocols and time-slot assignments. Thus, the security of the synchronous network was not challenged with virus attacks by outside bad actors other than eavesdropping on individual conversations at risk of been caught. Virus and other malicious soft attacks are associated with Internet and computer communication networks for which many “cyber-security” reports have been drafted for governments and enterprise [13–16] in an attempt to develop counter cyber-attack strategies and thwart the humongous destructive attempts and permit

to trace back and discover the offending terminal and its user. Unfortunately, such attempts can be made remotely (from another country) by high school and college students who don't have specialized equipment, just personal computer, and who are not seeking profit but to demonstrate intelligence!

In general, information assurance and security aims to ensure a level of trust to the client and by the client commensurate with client expectations. Such expectations include information or data protection during its creation, use, transformation, storage, and transport. In addition, the expectation is that data are not retained at layer boundaries of the reference model (such as the ISO, ATM, TCP/IP), and at the transport layer. In a computation environment, information security also aims to ensure that *access to information and the network* is under authorized control and the network is capable of self-defense and countermeasures; the latter is accomplished by specific mechanisms that monitor, detect, react and respond to attacks, vulnerabilities and deficiencies.

Among the most interesting networks in terms of assurance and security are the wireless and the fiber optic. Copper twisted wires are currently used in telephony and for high speed digital at the subscriber loop (DSL).

- *Wireless network* is not a complete end-to-end network; wireless is the access part of the network, which is connected to a traditional network (Internet for data, synchronous for voice and video) and with a traditional medium. With this clarification, there are several wireless networks, each with different security classifications and idiosyncrasies because the transmitted electromagnetic waves reach both friendly and foe antennas. Thus, eavesdropping is a bonus to an intruder as reception is undetectable and security depends on secret keys, terminal ID authentication, and protocols.
 - *The wireless local area network (WLAN)* connects many hundreds of end terminals to a control module, which is connected to a data network server and from there to the data network (Figure 1.7). In the general application of this network, wireless access is point-to-point in a star topology, the wireless link has limited distance (up to 300m) and the geographic area is limited as well (such as offices on the floor of a building or open space in a campus environment). Wireless LANs are described in the IEEE 802.11 standards [17–26].
 - *The wireless ad-hoc LAN* is a peer-to-peer network. That is, there is no centralized control module, the network topology is meshed, and all end-users maintain connectivity maps continuously, if end-users are mobile, or periodically, if movement is slow or if the network is scalable; wireless sensor networks are among these, as well as mobile cluster end-terminals (for data and voice) that are on a continuous move (Figure 1.8) [18].
 - *The mobile cellular network* is a wireless technology that also provides wireless access to mobile end users. Because the geographic area is partitioned in hexagonal cells, each having its own antenna, the base station (BS), and each having different frequency allocation (known as frequency

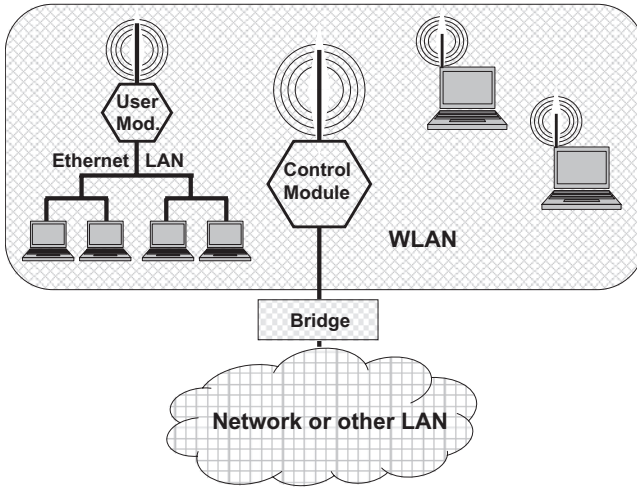


Figure 1.7. End terminals with a wireless interface access a WLAN via a control module. Similarly, other LANs (Ethernet) with wireless interface connect with the WLAN. The control module is connected with the network or another LAN via a bridge that support a high data rate interface (1GbE or OC-48).

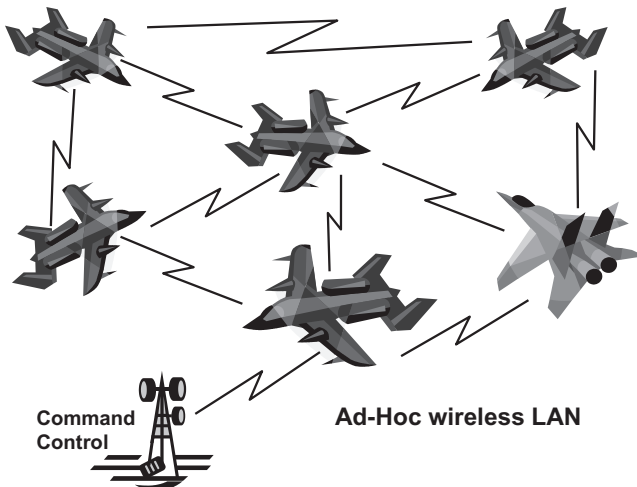


Figure 1.8. A fully connected (mesh) fast moving and fast reconfiguring ad-hoc wireless network. Nodes may be added and deleted while the mesh topology is reconfigured. Such applications require fast and complex protocols with robust security.

reuse) to reduce cross-talk between adjacent cells. Users may move within a cell and through cells, in which case connectivity changes from base station to base station according to a protocol known as *handover*. A cluster of cells communicates with a centralized cell, known as the mobile telephone switching office (MTSO). The latter communicates with other

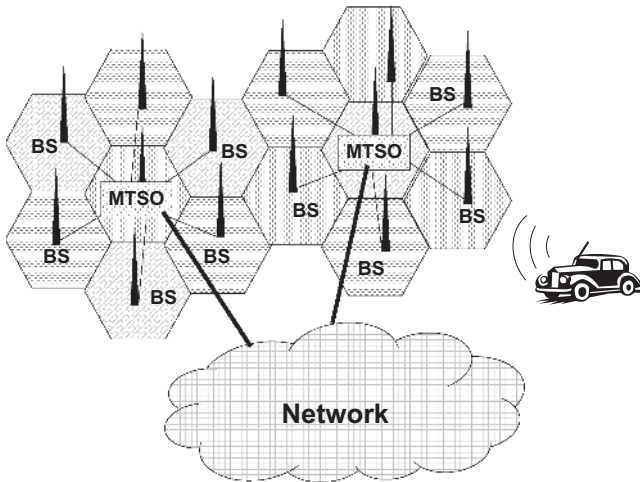


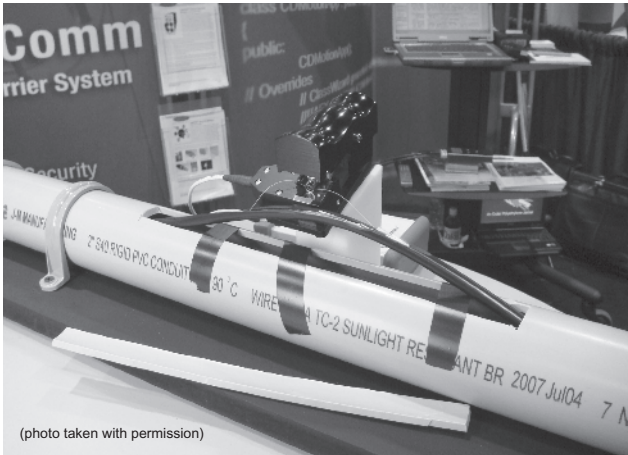
Figure 1.9. The cellular mobile wireless network consists of hexagonal cells in a symmetric cluster; the central cell (MTSO) provides connectivity with the network and with the other surrounding cells (BS) in a star topology.

MTSOs and with the public switched digital network (PSDN) (Figure 1.9) [19].

- *The fiber optical networks* currently transport up to 40Gbps per optical channel, or Tbps per fiber using dense wavelength division multiplexing (DWDM) technology [27]. Although optical technology is more complex than its predecessors, it attracts bad actors because of the huge amount of information that transports in a single fiber. Bad actors with the proper know-how and sophisticated tools may attack the medium (Figure 1.10) and harvest huge amounts of information, mimic the source, alter information or disable the proper operation of the network. Thus, in order to eliminate this risk and assure data security and privacy, highly complex, difficult and sophisticated algorithms are necessary for the generation of the cipher keys and for the key distribution. Moreover, the network itself should be intelligent [28], sophisticated to identify and authenticate channel ID [29], able to detect malicious attackers and outsmart them by adopting sophisticated countermeasure strategies [30] and also vulnerability-free or vulnerability-hardened [31].

1.3.1 ISO/OSI Reference Model and Security

There are many functions in a computer-based node in a data network. These functions have been grouped into abstractions and a hierarchical network reference model of abstraction layers has been defined known as Open System Interconnection (ISO). That is, as information starts from the highest layer and progresses toward the lower layer, specific overhead is added by each layer. Based on this model, a



(photo taken with permission)

Figure 1.10. Photo showing a fiber under tap. The protective PVC tube is cut open, the protective plastic wrap of the fiber cable has been carefully sliced open and a particular fiber has been pulled and tapped. From the tap, another fiber is connected to copying and cryptanalysis equipment. (Photo taken with permission).

Application	7	•Overall management of a transaction, including segmentation and reassembly
Presentation	6	•Data transformation, formatting and syntax
Session	5	•Negotiated connection parameter •Dialogue control and interrupts •Synchronization mechanisms
Transport	4	•Connection management •Data transfer •Flow control
Network	3	•Move data through the network •Switching and routing functions •Error recovery on network layer
Link	2	•Reliable interchange of data •Error recovery on the link layer •Recovery from abnormal conditions
Physical	1	•Compliance with physical interface specifications •Electrical/optical/wireless, mechanical, functional, procedural

Figure 1.11. The ISO/OSI seven layers model.

node communicates with another peer node of the network and standard protocol (known as peer-to-peer communication) only via its corresponding layer.

All data protocols do not define the same ISO reference model. The International Standards Organization Open Systems Interconnection (ISO/OSI) reference model defines seven layers (Figure 1.11) [32, 33]. The lowest layer (layer 1) is the

physical (PHY) and the highest layer (layer 7) is the information. More specifically, the seven layers are defined and contain functions (also known as abstractions). According to the ISO/IEC 7498-1 [5] standard, each protocol layer is composed of three functional planes, the user or bearer, the signalling and control, and the management plane.

The definition of each OSI layer, from the highest to lowest layer, is:

- **Layer 7:** The *Application Layer* deals with communication issues of applications and provides the interface with the user. Examples include the Hyper-Text Transfer Protocol (HTTP), the File Transfer Protocol (FTP), the Session Initiation Protocol (SIP), the Simple Mail Transfer Protocol (SMTP) and Telnet.
- **Layer 6:** The *Presentation Layer* receives data from the Application Layer, translates data, and performs data compression/decompression and data encryption/decryption according to standards. Examples include ASCII, ZIP, JPEG, TIFF, RTP, and the MIDI format.
- **Layer 5:** The *Session Layer* initiates the contact between two computers and sets up the communication lines. It formats the data for transfer and it maintains the end-to-end connection. Examples are the Remote Procedure Call (RPC) and the Secure Sockets Layer (SSL) protocols.
- **Layer 4:** The *Transport Layer* defines how to address the physical locations of the network and to establish connections between host computers, and it handles network messaging. It maintains the session end-to-end integrity and it provides mechanisms to support session establishment for the upper layers. Examples of protocols at this layer are the Transport Control Protocol (TCP), the User Datagram Protocol (UDP), and the Stream Control Transmission Protocol (SCTP).
- **Layer 3:** The *Network Layer* is responsible for routing and relaying data through the network host computers with integrity. It sends *packets* from a source to a destination host computer and it manages bit errors (detection and correction), message routing and traffic control. The Internet Protocol (IP) is performed at this layer.
- **Layer 2:** The *Data Link Layer* defines conditions so that a host can access the network. It ensures message delivery to the proper device over the physical link and it translates, encodes and scrambles the packet bits for the lowest physical layer. Examples at this layer are Ethernet and Token Ring protocol requirements such as 4B/5B or 8B/10B encoding, data scrambling, RZ, NRZI, Manchester, etc.
- **Layer 1:** The *Physical Layer* defines the physical connection between a host computer interface and the network. Its main function is to convert the logical bits received from the data link layer into a physical signal that meets standard transmission specifications, such as min–max voltages or optical impulse power, modulation method (FSK, PSK, etc.) frequency spectrum, bit rate, medium impedance, and so on and for various medium interfaces (wireless,

wired, infrared or optical fiber). Hardware drivers at this layer responsible for communication interfaces are network interface cards (NIC).

In addition to grouping functions in layers and defining responsibilities of each layer, ISO [34] and International Telecommunications Union (ITU) standards [35] describe security objectives, services and related mechanisms for each plane and layer. These objectives are accomplished according to a set of criteria, known as *security policies*, which define the provision of security services provided by a layer. Security services are implemented by *security mechanisms*. According to standards, the basic security services are (see also section 1.2.1.3) Authorization, Access Control, Data Confidentiality, Data Integrity, and Non-Repudiation. Similarly, the defined security mechanisms are Encipherment/Decipherment, Digital signatures, Access control, Data integrity, Authentication, Traffic padding, Routing control, Notarization. Security extensions to the security architecture recommended in [35] are provided in [36–53].

1.4 SECURITY THREATENING ATTACKS AND ACTIONS

A communication system that transmits sensitive and proprietary information should be architected and designed to provide the expected security services with secure mechanisms, as described in the previous section. Security services and mechanisms, however, should also vary according to type of network, whether synchronous (circuit switched) or asynchronous (connectionless packet switched), wireless, wired or fiber optic, because the amount of information transported per second as well as node and medium accessibility varies.

In connectionless networks such as the Internet, the first defense is to protect the boundary between the network and the LAN using a firewall device [54, 55]; a firewall runs a special software program that controls, as a single pipe, and examines all the data flow between two hosts or between the data network and the corporate LAN (Figure 1.12). Thus, firewalls can provide a protective mechanism if all configuration parameters have been set correctly and if the running software is the most recent; this occasionally becomes a weak point to new smart attacks that are not recognized by the firewall and thus pass through.

Because of this, security threats may include attacks to the physical medium (tapping wired or fiber optic synchronous networks), or to the node layers (information and/or control of computer-based host) as already described.

Attacking the medium or the information/control layer may be done to enable eavesdropping and copying sensitive data that may be of value to a bad actor, even if data is ciphered (see following section). In such case, the bad actor, by trying different possible keys, may be able to *break* the cipher text. This is known as *brute force attack*. Similarly, a bad actor may capture the key and the cipher text and figure out how to decipher it or how the cryptographic system works. Because attacks of any form have been on the increase, modern networks that are security-minded should include certain additional functions:

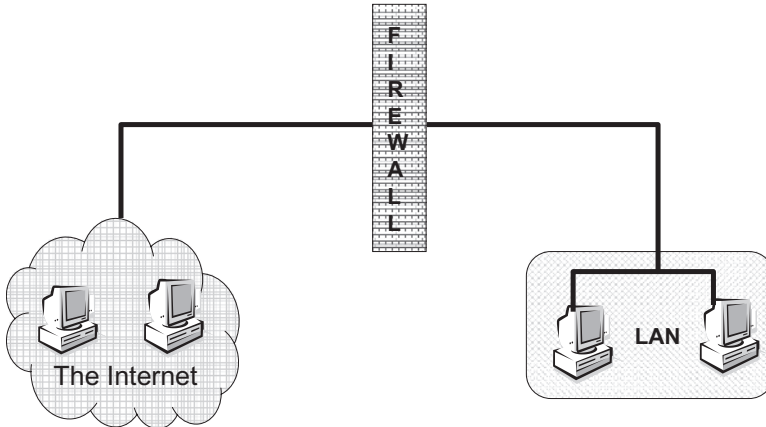


Figure 1.12. A firewall controls the flow of all data traffic from the network to a LAN; if properly configured and running the most recent security software, it is expected to act as a barrier to malevolent attacks and it protects the LAN.

- Detect that an attack by a bad actor took place.
- Be able to differentiate between attacks and malfunctions or faults.
- Locate where the attack took place.
- Detect and verify that the cipher text is not altered or the key is not compromised. If it is suspected that the key has been *compromised*, the key should be *revoked*. In this case, the key establishment may restart and the key may be *updated*.
- Activate a self-defensive countermeasure strategy.
- Activate a counterattack to the bad actor.

Does this sound like a war scenario? It is a war scenario; it is called *cyber-war*.

As a consequence, end-to-end overall security needs to be designed as multi-layer framework that delivers reliability, resiliency, remote access, management, availability and bandwidth and mitigates multi-vendor protocols and interfaces. At the same time, it lowers risk, encompasses services, anticipates and resolves security issues of privacy, intrusion, authentication, policy, and proactive prevention.

1.4.1 Information Security Attacks

Information attacks are alarmingly on an exponential increase. Attacks come in different forms. Among the known viruses are email viruses, Trojan horses, worms, phishing, web pharming, and so on. All have a malicious intention: destroy files, alter files, and harvest and steal sensitive personal data. Many of them may not seem harmful, although they flood the network causing congestion, slow down execution

and traffic, cause denial of service, and perhaps can cause routers and the network to shut down. This section examines some types of attacks.

1.4.1.1 *Virus*

A virus is short length software that is attached (piggyback) to a real program, a picture, etc. Each time the program is executed, the virus runs in the background. Depending on the type of virus, when it runs it reproduces copies of itself or clones itself without permission and be attached to other programs while it destroys files, alters the contents of files, or harvests data among specific files. The original virus may modify the reproduced copies, or the copies may modify themselves, known as a metamorphic virus. In such cases, although the initial virus may be detected and destroyed, the modified copies may not be detected.

Such viruses were initially spread from computer to computer via floppy disks. Floppy disks are obsolete and read-only CD-ROMs are used to load programs; CD-ROMs are safer than magnetic media, which can be rewritten and thus modified.

Currently, the most popular method of spreading a virus is through the Internet via email. An email-virus is propagated as an attachment to a message. When activated by clicking, it replicates itself, harvests email addresses from the victim's address book, and automatically sends itself to them, causing damage to all who activate the virus. Thus, a virus may be propagated at an alarming rate and cause havoc. Some email viruses enter a hibernation state and they are activated at a specific time or date, or by a particular triggering event. Other destructive viruses target the boot sector of a rewriteable hard disk. The boot sector contains the first part of the operating system software that controls the computer loading the remaining of the operating system. Modern computers contain the boot program in read-only memory, which cannot be altered by viruses, or the more advanced operating system does not permit other programs to alter the boot sector.

Some viruses are benign; they do not mean to cause damage but to replicate and spontaneously present some text, video, image, or sound. These viruses waste computer resources and execution power, and they slow down the execution of important programs.

Many viruses were written in the scripting languages for Microsoft programs such as Word (.doc), Excel (.xls) or as an executable program (.exe), thus infecting documents and spreadsheets in Microsoft Office or Mac OS based computers. Cross-site scripting viruses use vulnerabilities to propagate, infecting notable sites.

Viruses are also classified as resident and non-resident.

- **Resident viruses** load themselves into memory on execution and transfer control to the host program. The virus stays active in the background and infects new hosts when the infected files are transferred.
- **Non-resident viruses** search for a new target host to infect and they transfer control to the application program they infected.

Anti-virus programs perform a self-integrity check and if they are infected they detect the virus. However, viruses have been programmed to recognize and avoid

anti-virus programs. In response to this, anti-virus programs create small bait files that entice viruses to infect them, thus detecting the virus, isolating it, analyzing it, and then destroying it. Thus, bait files are also used as virus forensics tools that examine virus behavior, particularly of polymorphic viruses (or mutating viruses), and to develop improved anti-virus programs.

As anti-virus programs become more sophisticated, so, too, do the viruses. Although they may have infected a file, when the anti-virus program examines the file, the virus sends to the anti-virus program an uninfected portion of the file to trick it. These are known as *stealth viruses*. To counter stealth viruses, anti-viruses employ other methods, and thus the race who tricks whom is on.

Anti-viruses typically scan files to detect specific byte patterns known as virus signatures. When such pattern is detected, the file is placed in quarantine and the user may delete or clean the infected file. If the virus is encrypted, detecting the virus signature becomes more difficult. In this case, the virus is encrypted with a different key for each infected file and only the decrypting portion remains unencrypted and the same; thus, the anti-virus looks for decrypting modules within files.

1.4.1.2 Trojan Horses

A *Trojan horse* is a computer program that pretends to be one thing (game, movie or sound clip) and in effect it is an executable virus or worm that does harm. The term is borrowed from the historical wooden Trojan horse that was presented by the Greeks to the Trojans at the famous Homeric battle of Troy. According to Homer, many Greek soldiers were hidden in the belly of the wooden horse, after the Trojans brought the horse inside the wall of the citadel, during the night the Greeks opened a trap door, descended from the horse and opened the wall gates. The fall of the city and the disaster that followed it brings to mind what the computer Trojan horse can accomplish. Trojan horses are also known as *security loopholes*.

Typically, Trojan horses exploit social engineering and propagate via email hiding behind legitimate animations, screensavers, or some other fun files (a typical Trojan horse program has the `.src` extension). When the user clicks on the fun file to watch it, the malicious program is installed and it runs its destructive course. A Trojan horse may affect the registry by adding entries so that the virus runs each time the computer boots, or by adding services to the computer, opening ports and allowing hackers to remotely access the user's computer (called a RAT, remote administration tool), disable security software, destroy files, cause denial of service, affect the downloader, send email, upload or download files, harvest email addresses, stealing passwords, spreading viruses, phishing for bank accounts, turn off the computer, and more.

Trojan horses are not known to replicate themselves. Instead, they reside on the host computer and propagate from computer to computer when an email with the virus file is transmitted to another host computer.

There is no single method to delete Trojan horses. The simplest method is to clear all temporary Internet files on the computer, or edit the registry. It may even

be necessary to reset the computer back to its factory defaults. The latest version of reputable anti-virus software is strongly recommended for all virus types including Trojan horses.

1.4.1.3 Worms

A worm is a very short length software (few hundreds of lines) that exploits software security vulnerabilities (known as *security holes*) in a computer; if it finds a hole, it replicates itself. Copies of the worm propagate across the computer network and when a worm finds a computer with similar holes, starts replicating there as well, and so on. Effectively, this results in an avalanche effect of worm propagation across the computer network that causes considerable harm. Thus, worms waste computer execution time and considerable network bandwidth that could clog the network, effectively causing the network to slow or shut down.

For example, the worm dubbed *Code Red* (July 2001), on July 19, 2001 replicated itself more than 250,000 times in approximately nine hours. This worm infected computers causing web pages to be replaced by a page displaying the message “Hacked by Chinese.” It was designed to replicate itself 20 days of each month, and it attacked the White House; it made 100 simultaneous attempts to port 80 of the address www.whitehouse.gov (198.137.240.91). This address has changed since then and a security patch was issued.

In 2003, the *Slammer worm* exploited a buffer overflow vulnerability. It generated random Internet addresses (it infected hosts over the UDP) and it sent out copies of the worm to those addresses. Because the worm was small (376 bytes), it fit in a single small packet and it was routed faster than longer packets (some routing algorithms push short packets through faster than longer ones). The worm clog that was generated caused some routers to shut or to slow down, creating severe congestion. Neighboring routers, which were trying to reroute traffic and avoid the congested routers, started sending messages to other computers in the network to update their routing tables, causing more traffic over the network. The avalanche of worms and table updates flooded the network, eventually causing data traffic to significantly slow down or to be dropped.

In 2007 the *Storm* worm was launched. Storm used social engineering and tried to entice users to load free music from well-known artists. The worm hid in the music files. The worm in the infected computer (called *zombie* or *bot*) opened a back door, adding the Microsoft Window-based OS computer to a peer-to-peer group of computer networks called a *botnet* without the consent of the computer user; the term botnet derives from *robot-network*. Each group of computers in a botnet may consist of few hundreds to several thousands, and therefore it’s estimated that many millions of computers may have been infected with the Storm worm. The originator of a botnet, known as a *botnet herder*, may remotely control the botnet.

Although the initial motivation of a botnet was not malicious, and therefore botnets comply with RFC 1459 [56], botnets were exploited and evolved to accommodate malicious intentions and compromise computers. Thus, *bots* on the botnet deliver spam or adware (estimated in the millions or billions), attack websites, and

harvest computer data and resources, thus wasting bandwidth and causing denial-of-service [57]. Harvesting computer resources is also known as “scurumping.”

1.4.1.4 Phishing and Pharming

Phishing is an illegal attempt via email or instant messaging to entrap someone to divulge personally sensitive information using social engineering methods. Such an attempt may be impersonating a trustworthy bank and asking for bank accounts; masquerading as a credit card service provider and asking to verify the credit card number and pin; or masquerading as a trustworthy entity (police, technical, education, training, legislative, web service, and so on) and asking for specific information (username and password, health status, personal data, social security number, etc). The term *phishing* derives from fishing, whereby the impersonator lures the user to fall victim.

Here is a real example of phishing that the author received on November 7, 2007 (at 10:47 EDT):

Security Alerts

RBC Royal Bank [ibanking@app.rbc.com]

To: Kartalopoulos, Stamatios V.

Cc: _____

Dear Customer,

Our Technical Service department has recently updated our online banking services, and due to this upgrade we sincerely call your attention to follow below link and reconfirm your online account details. Failure to confirm the online banking details will suspend you from accessing your account online.

<https://www1.royalbank.com/cgi-bin/rbaccess/rbunxcgi>

We use the latest security measures to ensure that your online banking experience is safe and secure. The administration asks you to accept our apologies for the inconvenience caused and expresses gratitude for cooperation.

Thanks for banking with us.

Royal Bank of Canada Security Advisor

RBC Financial Group

This is an automatic message. Please do not reply.

Phishing is a very serious offense and in some ways it can be more harmful than a worm or virus. If the victim believes that the phishing email comes from a bank, and thus gives away a bank account and other sensitive personal data the phishing culprit may steal the user’s identity and wealth. Thus, one anti-phishing defense is for users to be alert and to not give personal data over email. In general, legitimate and trustworthy entities do not solicit personal information via email or

instant messaging. Banks and government agencies do not, as a rule, ask via email for accounts and social security numbers.

The Anti-phishing Act of 2005 [58] is a bill that was referred to the U.S. Senate and also to the Subcommittee on Crime, Terrorism, and Homeland Security of the U.S. House of Representatives to combat phishing and pharming, and to define penalties for individuals who commit such crimes and identity theft by falsifying corporate and other entity websites or emails. Similarly, in the United Kingdom, a general offense of fraud was introduced with the Fraud Act 2006 [59]. Despite these two bills and many others, the computer user should also be prudent and make a serious effort to combat phishing and pharming by using the latest versions of spam filters.

Phishing emails use social engineering tactics that aim to upset and panic the recipient of the email, particularly the elderly. For example, this bogus email has specific key-words that can upset and disturb the recipient:

From: 1st United Services Credit Union [mailto:services@1stuscu.org]
Sent: Mon 11/12/2007 10:12 AM
Subject: Important Member Service Information

Dear Member,

This is your official notification from 1st United Services Credit Union that the service(s) listed below will be deactivated and deleted if not renewed immediately.

Previous notifications have been sent to the Billing Contact assigned to this account.

As the Primary Contact, you must renew the service(s) listed below or it will be deactivated and deleted.

Renew Now your 1st United Services Credit Union and ON-LINE BANKING.

SERVICE: 1st United Services Credit Union ON-LINE BANKING.

EXPIRATION: November, 16 2007

Thank you for using 1st United Services Credit Union.

We appreciate your business and the opportunity to serve you.

1st United Services Credit Union ON-LINE BANKING

Copyright © 2007 1st United Services Credit Union. All rights are reserved.

Examination of this message reveals the following tricks of appearance and intimidation designed to induce panic:

- First, it attempts to look like a legitimate notice with a fake copyright at the end of the email. However, there is no “to:” and it is addressed to an impersonal “member” and not to a specific name.
- It is addressed to the “Primary Contact” and states that the recipient “must renew the service(s) listed below,” but there is no list of services. Instead, it entices the recipient you to click on the “Renew now” to look for services, etc., and get trapped.
- It uses the strong words and it gives only 5 days margin to renew or “the account will be deactivated and deleted immediately”

Pharming (a derivative of farming) also attempts to illegally collect personal sensitive information by *domain spoofing*. That is, instead of using bogus email requests, pharming “poisons” a domain node server (DNS) by infusing false information in it and redirecting a website’s traffic to another bogus website which harvests personal data from the user’s computer; as far as the browser is concerned, the user is connected with the correct website.

DNS servers are responsible for resolving Internet names into their real addresses. For example, the user types a name such as Amazon.com, AOL.com, Google.com, etc, and the DNS server translates it to a numeric address.

A key difference between phishing and pharming is that the first targets one user at a time via email whereas the second targets large groups of users through domain spoofing; spoofing uses someone else’s IP address in TCP/IP packets (source or destination address). With spoofing, the attacker pretends that is the source or the destination and may cause redirection of routing to the hacker’s host, or may flood the receive queues causing a large group of users to respond to the victim. Pharming has become a major concern to e-commerce.

1.4.1.5 Protecting the Computer from Viruses

The best defense against viruses, Trojan horses, worms and other known attacks is to install the latest version of reputable anti-virus and anti-spam security software. In addition, the user may also take few simple protection steps, such as:

- Saving files periodically on an external hard disk. If the computer is infected, the user can always use back-ups on a clean computer.
- Not opening emails that do not seem legitimate, either because the subject does not read right, or the sender does is not familiar, or it promises some free fun adventure. Remember, no one gives anything for nothing. Remember that bad actors are unscrupulous and very sophisticated.
- If an email has an attachment of an executable file (.exe, .com, .vbs, .src), never open it. In general, do not open attachments from sources that seem suspicious. Even a picture (.gif) or video clip may contain a virus, a worm or a Trojan horse. Ask, what is more valuable? The attachment or the files in the computer?
- Install software programs from CD-ROMs only. Reputable software companies have enough security safeguards and the software they sell comes on a one time writeable disk, the CD-ROM, which subsequently cannot be infected by viruses. Thus, avoid programs from online, unknown sources.
- The Macro Virus Protection of the computer should be enabled and in general, users should avoid running macros in a document.

1.4.2 VPN Networks

1.4.2.1 Tunneling

Tunneling is a term that encompasses encapsulation, routing, and decapsulation. Encapsulation entails forming a new packet by adding header overhead to the original packet. The overhead of the new packet may have a new address and routing information that is used to route the new packet through a logical data path. To the original source, the tunnel appears as a point-to-point transparent connection across the network and is unaware of routers, switches, security gateways and proxy servers on the tunnel path. During the encapsulation process, the original packet can be encrypted for confidentiality purposes. When tunneling is combined with confidentiality, it can be used to provide a virtual private network (VPN). Encapsulated packets travel through the network tunnel. When the encapsulated packet reaches its destination at the end of the tunnel, the encapsulation overhead is removed and the destination information on the original packet is used to route the packet to its final destination. That is, routing becomes a two-step process.

1.4.2.2 VPN Security

A Virtual Private Network (VPN) is a private network, which by configuring a public data network like the Internet emulates a point-to-point private secure link between two sites (site-to-site) or between a remote client computer and a corporate server; this is also known as tunneling. Thus, two corporations at different sites and clients who travel or who work from home are able to obtain remote access connectivity between their VPN computer (VPN client) and an organization server (corporate VPN server) using provisioned links of the public data network infrastructure (such as the Internet). Based on this, there are two types of VPN connections:

- *Site-to-site* by which a router establishes VPN connections through the public data network between two private networks.
- *Remote access* by which a remote access client makes a remote access VPN connection with a private network.

VPN configuration for the initial connection is common to both types and this includes tunneling protocols, authentication methods, access network and address assignment.

Typically, servers provide several VPN protocols for remote client access connections and for site-to-site connections. Among them are the Point-to-Point Tunneling Protocol (PPTP), the Layer Two Tunneling Protocol over Internet security (L2TP/IPsec), and the IPsec tunnel mode.

- *The Point-to-Point Tunneling Protocol (PPTP)* supports on-demand, multi-protocol, and it enables the secure transfer of data from a remote client to a private corporate server by creating a VPN link across TCP/IP-based data

networks. PPTP connections require only user-level authentication through a point-to-point authentication protocol.

- **The Layer Two Tunneling Protocol (L2TP)** provides encapsulation for Point-to-Point Protocol (PPP) frames across packet-oriented (IP) networks, and it allows IP traffic to be encrypted. L2TP may use Internet Protocol security (IPsec) encryption to protect the data stream over the link between the VPN client and the VPN server. In this case, it is termed L2TP over IPsec (L2TP/IPsec). L2TP/IPsec connections require the same user-level authentication with PPTP and, in addition, computer-level authentication using computer certificates.
- **The IPsec tunnel mode** allows IP datagrams to be encrypted, encapsulated by adding IP header, and sent across a corporate IP network or a public IP network. In this mode, IPsec provides encapsulation for IP traffic only. IPsec tunnel mode provides interoperability with routers, gateways and end systems that do not support L2TO/IPsec or PPTP VPN tunneling (Figure 1.13).

In some applications, the VPN server is integrated into the firewall functionality, from which site-to-site VPN connections as well as VPN client access to the corporate network can be managed.

In VPN point-to-point links, data is encapsulated with a header that includes routing information through the public network to its destination address. For confidentiality, data is also encrypted.

A call from a remote client (or gateway) to establish VPN connectivity entails several steps:

- A remote access client (or remote site gateway) dials the server.
- The server sends a challenge to the client (according to an authentication protocol, such as CHAP or MS-CHAP).

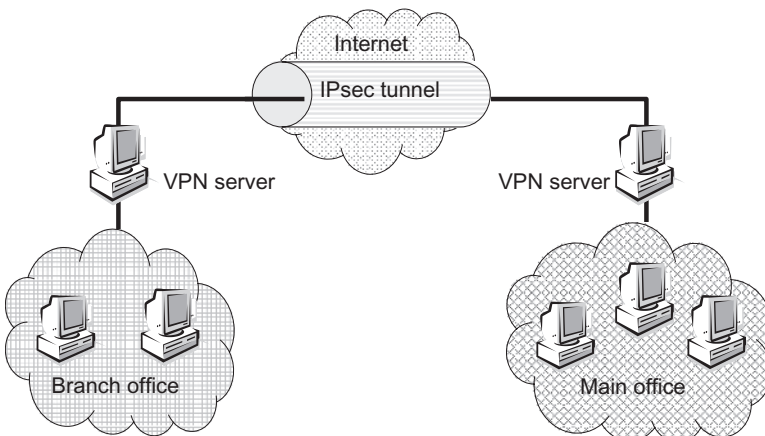


Figure 1.13. The path between a branch and main offices is connected with VPN servers and IPsec tunnel mode through the Internet network.

Table 1.3. Security comparison of certain data protocols

Protocol	Usage	Security
PPTP	Connecting to VPN server	Moderate
L2TP/IPsec	Connecting to VPN server	High
IPsec Tunnel mode	Third-party VPN server	High

- The client (or gateway) responds to the challenge and sends an encrypted response to the server that consists of a user name, a domain name, and a password; these credentials may differ according to the authentication protocol in use.
- The server checks the response against a valid user account database.
- If (according to the database) the client response is authenticated, the server uses the dial-in properties of the user account and the remote access policies to authorize the connection.
- If the client does not respond with acceptable credentials, the access attempt fails.

These steps repeat each time the client attempts to access a network resource.

While a remote VPN client has requested connectivity and until the credentials have been verified, the client may have been placed under quarantine control. This determines that the client's computer configuration is in accordance with the organization's specific quarantine restrictions (type of anti-virus software) and VPN policy. If the client fails to meet configuration requirements and if they cannot be corrected, the client fails to connect after some predetermined interval. Quarantine control may be optional.

A comparison of three VPN protocols is provided in Table 1.3.

1.4.2.3 IPsec

The Internet Protocol Security or IPsec is a suite of security protocols designed to secure IP traffic between computers. As already discussed, a secure channel starts with two peer computers, one at each end of the channel, which define a secret session key known as authenticated key exchange. Then, the peers encrypt the information over the channel via MAC protocols (such as HMAC).

IPsec is designed by the Internet Engineering Task Force (IETF) as the security architecture for the Internet Protocol (IP) to provide interoperable cryptographically-based security for IPv4 and IPv6.

IPsec defines (at the IP layer) IP packet formats to provide end-to-end (client-to-client, client-to-server, server-to-server) security services such as access control, connectionless integrity, data origin or data source authentication (filtering incoming

IP addresses to eliminate spoofing), integrity, anti-replay (detecting and rejecting partial sequence integrity), confidentiality via encryption, and limited traffic flow confidentiality.

Based on this, the MACs of the two (end-to-end) peers authenticate each other and with IPsec the receiving MAC is assured that the source IP address was initiated at the sourcing MAC, although there is no warranty that the message is that of the sourcing terminal and that a third party has not mimicked the source. That is, IPsec protects communication between computers but it is not involved in the user authentication or authorization between VPN and user-terminal.

The IPsec suite of protocols consists of several protocols, two of which are for channel authentication, and they are cryptographic algorithm-independent. That is, they are modular, permitting selection of different cryptographic algorithms and they can run either independently or one on top of the other to provide IPv4 and IPv6 security:

- ***The Authentication Header (AH)*** protocol provides data origin authentication and integrity protection but it does not encrypt the channel.
- ***The Encapsulating Security Payload (ESP)*** protocol provides the same services AH does, and it provides confidentiality and limited traffic flow confidentiality for concealing packet length and facilitating generation and discard of dummy packets; ESP requires more bandwidth.

Both AH and ESP protocols support two distinct modes of operation: transport and tunnel. To get the channel ready for communication, a key exchange authentication is performed as described by two protocols, the Internet Security Association Key Management Protocol (ISAKMP) and the Internet Key Exchange (IKE). IKE also helps to establish a simplex (unidirectional) Security Association (SA) between two computers, which is defined by a peer's IP address, AH or ESP, and an index to a set of parameters such as the encryption and hash algorithms.

IPsec creates a boundary between unprotected and protected interfaces and traffic traversing the boundary is subject to access controls. As such, IPsec provides secure gateway-to-gateway connections across private wide area networks (WAN), internet-based connections using Layer Two Tunneling Protocol over IPsec tunnels (L2TP/IPsec), or pure IPsec tunnel mode; the latter is not designed for VPN remote access. Additionally, IETF defines on-demand security negotiation and automatic key management service.

The Point-to-Point Tunneling Protocol (PPTP) and L2TP/IPsec are among the tunneling protocols that establish a tunnel between two VPN servers (VPN site-to-site). Instructions how to set up IPsec remote networks are provided by the manufacturer. Typical steps are:

- Select IPsec as the tunneling protocol of the remote site network.
- Configure the VPN end-servers in IPsec tunnel mode and the advanced IPsec settings based on instruction and access policy. Caution: some VPN servers

will erase the IPsec configuration when a restart is invoked and all traffic from clients may be forwarded to the Internet unencrypted.

In VPN site-to-site with IPsec if L2TP over IPsec is used, the IPsec tunnel (shown in Figure 1.12) is replaced by L2TP/IPsec and the VPN servers are configured accordingly, following manufacturer's instructions.

Although IPsec consists of a large suite of protocols [60–78] to provide the required level of detail and protocol inter-relationship, its understanding and implementation is not as easy. For example: different RFCs describe the AH and ESP protocols, cryptographic algorithms for integrity and encryption for use with AH, ESP and different IKE versions.

1.4.3 Network Security Attacks

1.4.3.1 Network and Service Availability

Network availability is a primary requirement of a communication system. Availability means that the system should be ready, within a specified and tight margin, to provide the expected services requested by customers under any state of the network, whether normal, congested, or under attack condition. That is, the network must be available to provide the agreed-upon services whenever the user needs it.

However, availability of network and of services depend on network technology, architecture, design, medium, protocols, topology and survivability tactics. Because networks are dissimilar, security of each network is addressed with different mechanisms. For example, wireless ad hoc networks and asynchronous connectionless networks do not have centralized control in contrast to mobile cellular and to synchronous circuit switched networks, and thus availability of each network is not expected to be the same. Similarly, a residential access network (wired or fiber) is not expected to have the same availability with a backbone mesh fiber optic network. Thus, the vulnerabilities of each network are not the same.

Network, protocol and service vulnerability is important to security of both information and network. Bad actors exploit vulnerabilities to harvest information, attack the network to cause denial of service, or bring down the network. Thus, a unified security approach is sought. In fact, because bad actors become more aggressive in their attacks, the modern network should be more intelligent, self-defensive, self-protective and in many cases counter-attacking in order to protect itself and be survivable and available for user service.

1.4.3.2 Network Attacks

Network attack encompasses any malicious actions that aim to intentionally interrupt, disturb, slow down, stop, or cause the network to malfunction or operate at degraded performance. There are two classifications of network attacks: passive and active.

- **Passive attacks** are considered those that do not directly affect the medium or the network. For example, in wireless a suitably-equipped eavesdropper is “listening” to conversations harvesting information, authentication, ID codes, pin numbers, and encryption keys, as well as networking information such as OA&M (operation, administration and management) that can be used at a later time to launch an assault.
- **Active attacks** are considered those that directly impact the transmission characteristics of the network. For example, in wireless access networks the attacker may broadcast a multi-frequency signal to jam or corrupt communication channels. In wired or optical networks, that attacker may intercept traffic by tapping the medium or the switch and eavesdrop, mimic the source of data, alter data, reverse engineer the cryptographic system (that is cryptanalyze cipher-texts), derail routing, affect the control, operations and management of nodes, affect billing data, and also cause nodes to not cooperate with the protocols and expected behavior in the network.

Both types of attacks may be attempted by a disgruntled employee or an unauthorized person inside communications buildings (central office, communications vault or closet), or inside an outdoor cabinet or hut that houses communications equipment. Attacks may also be outside the premises; such as attacking cable links (tapping aerial cables or cables in underground pipes and pathways and in manholes).

Among such attacks, the most typical are:

- *Eavesdropping* attempts to “listen” to data transported over a medium (wireless, wired, or optical) and to copy data, keys, protocols, and any sensitive information that can be useful to a bad actor.
- *Interception* implies capturing a message and either destroying it or altering and retransmitting it by mimicking the source.
- *Impersonation* is also source-mimicking, although it does not imply that a message must have been intercepted. Impersonation may source false original data to a destination in an attempt to mislead the receiver, or to engage the receiver in a cryptographic protocol process which can facilitate the bad actor’s cryptanalysis work.
- *Routing* are attacks that aim the routing tables and protocols of nodes in the network, including provisioned tunnels. Such attacks may aim to derail specific channels to other destinations, to cause (undesirable) broadcasting of data, and to cause severe congestion and denial of service over specific paths or node-clusters of the networks.

In conclusion, network attacks are on several layers, particularly on the information, control and physical layer, and a network cannot be considered secure if only one or two of the three are considered secure and not all three.

The best cryptography does NOT warranty secrecy if nodes are not immune to attacks; and the most immune node does NOT warranty privacy if the network is NOT secure.

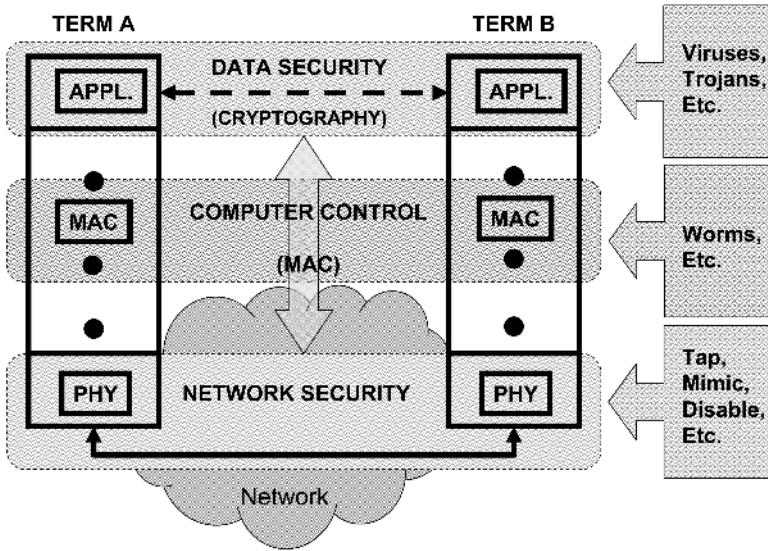


Figure 1.14. The three most vulnerable layers according to the ISO model.

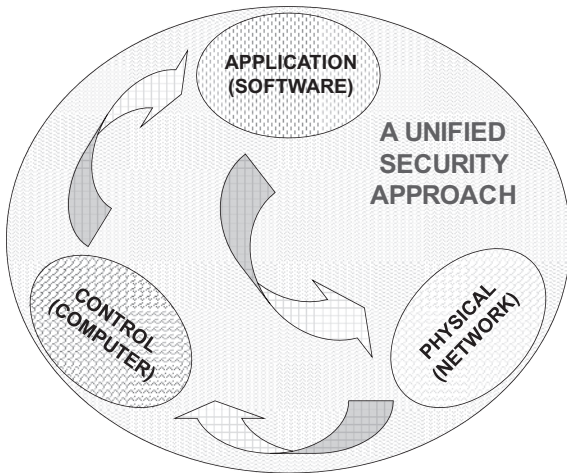


Figure 1.15. A synergistic approach definitely enforces network security. This synergy may also be used to architect self-defensive as well as counter-attacking networks.

Figure 1.14 addresses the three particular ISO layers that are most vulnerable to attacks—information, control and physical—and Figure 1.15 encapsulates a synergistic approach to security.

1.4.3.3 Counter-Attacking Intelligent Networks

Counter-attacking is best understood based on the scenario of a bad actor attacking the medium.

- First, this attack is detected as it occurs. Chapter 7 discusses methods that do exactly this; that is, they detect channel attacks in real-time and with no service interruption by the method.
- As soon as the attack is detected, the nodes at either side of the link initiate a counter-measure strategy, whereby sensitive data is moved to another secure channel and whereas decoy data are being transmitted over the attacked channel.
- In addition, since the attacker operates with computer-based equipment, the decoy messages contain hidden viruses and Trojan horses, which although not destructive to link receivers (by preconditioning the receivers), are destructive to the bad actor's computer.

REFERENCES

1. Herodotus, *Histories*, (many publishers).
2. Homer, *The Illiad* (many Publishers and translations).
3. Aeschylus, *Agamemnon*, Loeb Classical Library.
4. D. KAHN, "Seizing the Enigma: The Race to Break the German U-Boats Codes, 1939–1943" (1991).
5. A. STRIPP, "The Enigma Machine: Its Mechanism and Use", in Hinsley and Stripp (eds.) *Codebreakers: The Inside Story of Bletchley Park*, 1993, pp. 83–88.
6. FIPS 180-1, "Secure Hash Standard", National Institute of Standards and Technology, April 17, 1995.
7. FIPS 186-2, "Digital Signature Standard", National Institute of Standards and Technology, 15 February 2000.
8. FIPS Pub 198, The Keyed-Hash Message Authentication Code (HMAC), March 2002.
9. FIPS Pub 190, *Guideline for the use of advanced authentication technology alternatives*, September 28, 1994.
10. IEEE P1363, "Standard Specifications for Public Key Cryptography", Institute of Electrical and Electronics Engineers, 2000.
11. W. DIFFIE and M. HELLMAN, "New Directions in Cryptography", *IEEE Trans. Info. Theory*, IT-22, 1976, pp. 644–654.
12. FIPS Pub 185, Escrowed Encryption Standard, February 9, 1994.
13. CRS Report to Congress, Order Code RL33123, *Terrorist Capabilities for Cyberattack: Overview and Policy Issues*, January 22, 2007.
14. CRS Report to Congress, Order Code RL32114, *Computer Attack and Cyberterrorism: Vulnerabilities and Policy Issues for Congress*, April 1, 2005.
15. CRS Report to Congress, Order Code RL32331, *The Economic Impact of Cyber-Attacks*, April 1, 2004.
16. CRS Report to Congress, Order Code RL31542, *Homeland Security-Reducing the Vulnerability of Public and Private Information Infrastructures to Terrorism: An Overview*, December 12, 2002.
17. IEEE 802.11, 1999 Edition, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.
18. IEEE 802.11a-1999, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications—Amendment 1: High-speed Physical Layer in the 5GHz band.
19. IEEE 802.11b-1999, Supplement to 802.11-1999, Wireless LAN MAC and PHY specifications: Higher speed Physical Layer (PHY) extension in the 2.4GHz band.

20. 802.11b-1999/Cor1-2001, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications—Amendment 2: Higher-speed Physical Layer (PHY) extension in the 2.4GHz band—Corrigendum1.
21. IEEE 802.11d-2001, Amendment to IEEE 802.11-1999, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Specification for Operation in Additional Regulatory Domains.
22. IEEE 802.11e-2005, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements.
23. IEEE 802.11g-2003, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications—Amendment 4: Further Higher-Speed Physical Layer Extension in the 2.4GHz Band.
24. IEEE 802.11h-2003, IEEE Standard for Information technology—Telecommunications and Information Exchange Between Systems—LAN/MAN Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Spectrum and Transmit Power Management Extensions in the 5GHz band in Europe.
25. IEEE 802.11i-2004, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications—Amendment 6: Medium Access Control (MAC) Security Enhancements Interpretation.
26. IEEE 802.11j-2004, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications—Amendment 7: 4.9GHz–5GHz Operation in Japan.
27. S.V. KARTALOPOULOS, *DWDM: Networks, Devices and Technology*, Wiley/IEEE Press, 2003.
28. S.V. KARTALOPOULOS, *Next Generation Intelligent Optical Network: From Access to Backbone*, Springer, 2007.
29. S.V. KARTALOPOULOS, “Channel signature authentication for secure optical communications”, SPIE Defense & Security, 9/26–29/05, Bruges, Belgium, paper no: 5986-41, on CD-ROM: CDS191.
30. S.V. KARTALOPOULOS, “Optical Network Security: Countermeasures in view of channel attacks”, Unclassified Proceedings of Milcom 2006, October 23–25, 2006, Washington, D.C., on CD-ROM, ISBN 1-4244-0618-8, Library of Congress 2006931712, paper no. US-T-G-404.
31. S.V. KARTALOPOULOS, “Identifying vulnerabilities of quantum cryptography in secure optical data transport”, Unclassified Proceedings of Milcom 2005, 10/17–20/05, Atlantic City, session: Comm. Security I, invited paper # 678, on CD-ROM, ISBN # 0-7803-9394-5.
32. ISO/OSI 7498-1: 1984, *Open Systems Interconnection—Part 1: Basic Reference Model*; also ISO/IEC 7498-1, 1994.
33. ITU-T Recommendation X.200, *Open Systems Interconnection—Basic Reference Model: The basic model*, 1994.
34. ISO/IEC 7498-2, *Open Systems Interconnection—Part 2: Security Architecture*, 1989.
35. ITU-T Recommendation X.800, *Security Architecture for Open Systems Interconnection for CCIT Applications*, 1991.
36. ITU-T Recommendation X.802, *Open Systems Interconnection—Lower Layers Security Model*, 1995.
37. ITU-T Recommendation X.802, *Open Systems Interconnection—Upper Layers Security Model*, 1994.
38. ITU-T Recommendation X.810, *Open Systems Interconnection—Security Frameworks for Open Systems: Overview*, 1995.
39. ITU-T Recommendation X.811, *Open Systems Interconnection—Security Frameworks for Open Systems: Authentication Framework*, 1995.
40. ITU-T Recommendation X.812, *Open Systems Interconnection—Security Frameworks for Open Systems: Access Control Framework*, 1995.
41. ITU-T Recommendation X.813, *Open Systems Interconnection—Security Frameworks for Open Systems: Non-repudiation Framework*, 1996.

42. ITU-T Recommendation X.814, *Open Systems Interconnection—Security Frameworks for Open Systems: Confidentiality Framework*, 1995.
43. ITU-T Recommendation X.815, *Open Systems Interconnection—Security Frameworks for Open Systems: Integrity Framework*, 1995.
44. ITU-T Recommendation X.815, *Open Systems Interconnection—Security Frameworks for Open Systems: Security Audit and Alarms Framework*, 1995.
45. ISO/IEC 10181-1: 1996, *Open Systems Interconnection—Part 1: Security Frameworks for Open Systems: Overview*.
46. ISO/IEC 10181-2: 1996, *Open Systems Interconnection—Part 2: Security Frameworks for Open Systems: Authentication Framework*.
47. ISO/IEC 10181-3: 1996, *Open Systems Interconnection—Part 3: Security Frameworks for Open Systems: Access Control Framework*.
48. ISO/IEC 10181-4: 1996, *Open Systems Interconnection—Part 4: Security Frameworks for Open Systems: Non-repudiation Framework*.
49. ISO/IEC 10181-5: 1996, *Open Systems Interconnection—Part 5: Security Frameworks for Open Systems: Confidentiality Framework*.
50. ISO/IEC 10181-6: 1996, *Open Systems Interconnection—Part 6: Security Frameworks for Open Systems: Integrity Framework*.
51. ISO/IEC 10181-7: 1996, *Open Systems Interconnection—Part 7: Security Frameworks for Open Systems: Security Audit and Alarms Framework*.
52. ISO/IEC 11586-1: 1996, *Open Systems Interconnection—Generic upper layers security: Overview, models and notation*.
53. ISO/IEC 11586-2: 1996, *Open Systems Interconnection—Generic upper layers security: Exchange Service Element (SESE) service definition*.
54. NIST 800-41, *Guidelines for Firewalls and Firewall Policy*, January, 2002.
55. GAO-04-467, United States General Accounting Office, Report to Congressional Requesters, *INFORMATION SECURITY, Technologies to Secure Federal Systems*, March 2004.
56. RFC 1459, Internet Relay Chat Protocol, May 1993.
57. SHARON GAUDIN, “Storm Worm Botnet More Powerful Than Top Supercomputers”, Information Week, September 6, 2007. <http://www.informationweek.com/news/showArticle.jhtml?articleID=201804528>
58. Anti-Phishing Act of 2005, “A bill to criminalize Internet scams involving fraudulently obtaining personal information, commonly known as phishing”, March 1, 2005; <http://www.govtrack.us/congress/billtext.xpd?bill=h109-1099>
59. ELIZABETH II, UK, Fraud Act 2006, Chapter 35, 8th November, 2006.
60. RFC 2401 “Security Architecture for the Internet Protocol”.
61. RFC 2402, “IP Authentication Header”.
62. RFC 2451, “The ESP CBC-Mode Cipher Algorithms”.
63. RFC 2403, “The Use of HMAC-MD5-96 within ESP and AH”.
64. RFC 2404, “The Use of HMAC-SHA-1-96 within ESP and AH”.
65. RFC 2405, “The ESP DES-CBC Cipher Algorithm With Explicit IV”.
66. RFC 2406, “IP Encapsulating Security Payload (ESP)”.
67. RFC 2407, “The Internet IP Security Domain of Interpretation for ISAKMP”.
68. RFC 2408, “Internet Security Association and Key Management Protocol (ISAKMP)”.
69. RFC 2409, “The Internet Key Exchange (IKE)”.
70. RFC 2857, “The Use of HMAC-RIPEMD-160-96 within ESP and AH”.
71. RFC 3526, More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)”.
72. RFC 3554, “On the Use of Stream Control Transmission Protocol (SCTP) with IPsec”.
73. RFC 3566, “The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec”.
74. RFC 3602, “The AES-CBC Cipher Algorithm and Its Use with IPsec”.

75. RFC 3664, "The AES-XCBC-PRF-128 algorithm for IKE".
76. RFC 3686, "Using AES Counter Mode With IPsec ESP".
77. RFC 3706, "A Traffic-Based Method of Detecting Dead IKE Peers".
78. RFC 3715, "IPsec-NAT Compatibility Requirements".

