SFWR 4C03: Computer Networks and Computer Security	Mar 1-4 2004
Lecturer: Kartik Krishnan	Lectures 19-21

Information, Computer and Network Security:

Information security is the security of information. With the introduction of the computer, the need for automated tools for the protection of files, and other information stored on the computer has become evident. This is especially true for a shared system, such as a time sharing system, and the need is even more acute for systems that can be accessed over a public telephone network, data network or the internet. The generic name for the collection of tools designed to protect data and thwart hackers is *computer security*. Another nuisance computer security tools have to guard against is the computer virus, which can be introduced into the system when it arrives on a diskette, and is subsequently loaded onto the computer.

In the course, we will be more interested in a second kind of security called *internet/network security*. This deals with the security of information during its transmission from user on one computer network to another. Of course, computer security is important too, since if someone can access your computer's resources, he/she will have access to the network, and other computers attached to this network. Thus, computer and network security measures go hand in hand. However, we will discuss internet security first and then return to computer security.

Network security problems can be divided roughly into four intertwined areas: secrecy, authentication, nonrepudiation and integrity control.

1. **Secrecy:** This is also called confidentiality, and has to do with keeping information out of the hands of unauthorized users. This is what usually comes to mind when people think about network security.

2. **Authentication:** This deals with determining whom you are talking to before revealing sensitive information or entering into a business deal.

3. **Nonrepudiation:** This deals with signatures: How does amazon.ca prove that Kartik indeed placed an order for a book, which Kartik claims he never placed?.

4. **Integrity of service:** How does one ensure that the message received was really the one sent, and not something that a malicious adversary modified in transit or concocted?.

We will also classify the attacks that compromise network security as *passive attacks* and *active attacks*.

1. **Passive Attacks:** These attacks are in the nature of eavesdropping on, or monitoring of, transmissions. The goal of the opponent is to obtain information that is being transmitted. Two types of passive attacks are *release of message contents* where an eavesdropper tries to learn the contents of what is being transmitted. This can be prevented by encryption (see model for cryptography below). A second type of passive attack is called *traffic analysis*, where the opponent tries to observe the pattern, frequency and length of messages being exchanged which could be used in guessing the nature of the communication that is taking place. Passive attacks are very difficult to detect since they do not involve the alteration of the data. The emphasis, therefore, is on prevention via a good encryption algorithm.

2. **Active Attacks:** Active attacks involve some modification of the data stream or the creation of a false stream. These attacks present the opposite characteristics of passive attacks. It is difficult to prevent active attacks absolutely because to do so would require physical protection of all communications facilities and paths at all times. Instead, the goal is to detect them and to recover from any disruption or delays caused by them.

A model for Network Security

The general model is shown in Figure 1. A message (plaintext) is to be transferred from one party (Alice) to another (Bob) across some sort of internet. The two parties, who are the *principals* in this transaction, must cooperate for the exchange to take place. A communication channel is established by defining a route through the internet from source to destination and by the cooperative use of communication protocols, e.g. TCP/IP by the two principals. As shown in the figure, the communication channel is not secure since there is an eavesdropper (opponent, Oscar/Trudy) who presents a threat to confidentiality, authenticity, and so on. All techniques for providing security have two components.

1. A security related transformation on the information to be sent. Examples include the encryption of the message, which scrambles the message (called the ciphertext) so that it is unreadable by the opponent.

2. Some secret information shared by the two principals, and it is hoped unknown to the opponent. An example is an encryption key used in conjunction with the transformation to scramble (encrypt) the message before transmission and unscramble (decrypt) it on reception.

A trusted third party (big brother) may be needed to achieve secure transmission. For example, big brother may be responsible for distributing the secret information to the two principals while keeping it from any opponent. Else, he/she may be needed to arbitrate disputes between the two principals concerning the authenticity of a message transmission.

There are four tasks in designing a particular security service:

1. Design an algorithm for performing the security related transformation. This algorithm is assumed to be known to the opponent (Kerckhoff's principle), but it should be such that an opponent cannot defeat its purpose.

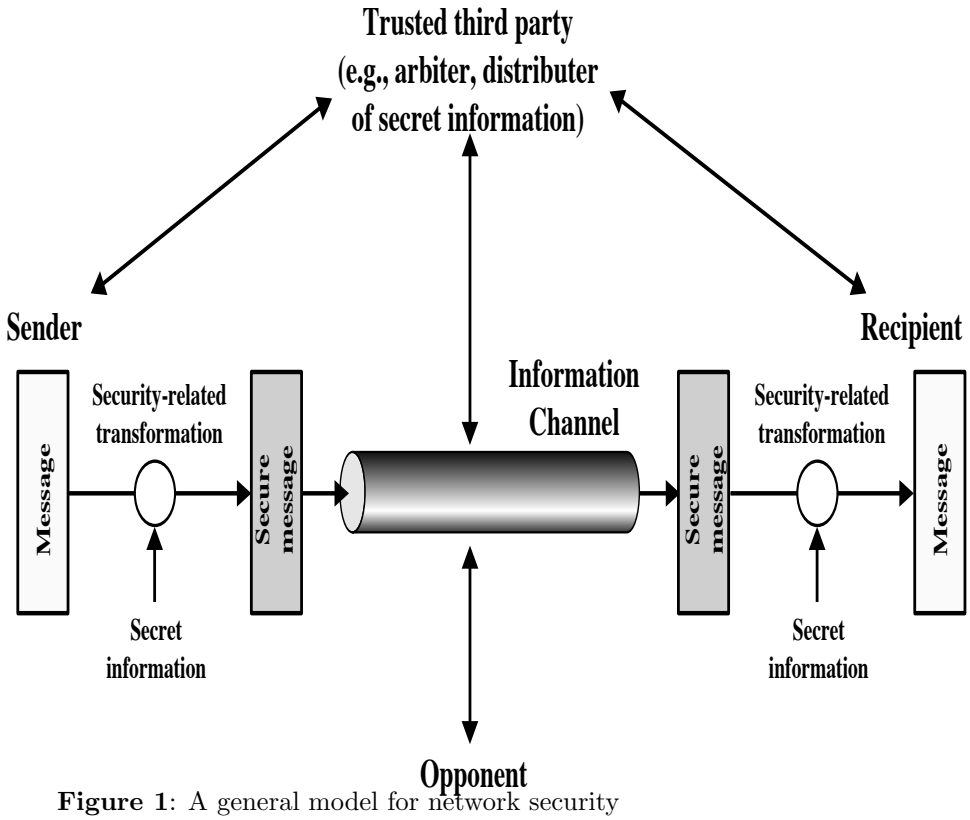

Figure 1: A general model for network security

Figure 1.1 Model for Network Security

2. Generate the secret information (key) to be used with the algorithm.

3. Develop methods for the distributing and sharing of the secret information.

4. Specify a protocol to be used by the two principals that makes use of the security algorithm and the secret information to achieve a particular security service.

Before we discuss these technicalities, we need to introduce some notation:

Definition 1 *A cryptosystem is a five tuple* $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$*, where the following conditions are satisfied:*

1. $\mathcal{P}$ *is the finite set of possible plaintexts.*

2. $\mathcal{C}$ *is a finite set of possible ciphertexts.*

3. $\mathcal{K}$*, the keyspace, is a finite set of possible keys.*

4. *For each* $K \in \mathcal{K}$*, there is an encryption rule* $e_K \in \mathcal{E}$ *and a corresponding decryption rule* $d_K \in \mathcal{D}$*. Each* $e_K : \mathcal{P} \rightarrow \mathcal{C}$ *and* $d_K : \mathcal{C} \rightarrow \mathcal{P}$ *are functions such that* $d_K(e_K(x)) = x$ *for every plaintext element* $x \in \mathcal{P}$*.*

Property 4 says that if a plaintext x is encrypted using e_K, and the resulting ciphertext is subsequently decrypted using d_K, then the original plaintext x results.

Alice and Bob employ the following protocol to use in a specific cryptosystem. First, they choose a random key $K \in \mathcal{K}$. This can be done when they are in the same place and not being observed by Oscar, or, alternatively when they do have access to a prior secure channel. At a later time, Alice wants to communicate a message to Bob over an insecure channel (as shown in the figure). We suppose that this message is a string $x = x_1 x_2 \dots x_n$, for some integer $n \geq 1$, where each plaintext symbol $x_i \in \mathcal{P}$, $1 \leq i \leq n$. Each x_i is encrypted using the encryption rule e_K specified by the predetermined key K (one can think as the encryption as being a function of the algorithm e_K and the key K). Hence, Alice computes $y_i = e_K(x_i)$, $1 \leq i \leq n$, and the resulting ciphertext $y = y_1 y_2 \dots y_n$ is sent over the channel. When Bob receives $y_1 y_2 \dots y_n$, he decrypts it using the decryption function d_K (once again decryption is a function of the algorithm d_K and the key K), obtaining the original text string $x_1 x_2 \dots x_n$.

One of the requirements of the encryption function is that it has been one-to-one (injective), i.e. if $x_1 \neq x_2$, then it must be the case that $y_1 \neq y_2$, where $y_1, y_2 = e_K(x_1), e_K(x_2)$. Else, the message cannot be decrypted in an unambiguous manner.

Here is an example of the cryptosystem called the shift cipher. Let $\mathcal{Z}_{26} = \{A, \dots, Z\}$. We will use the following numbering scheme for the 26 alphabets, where $A = 0$, $B = 1$, …, $Z = 25$ etc. The shift cipher can be formally defined as follows:

Definition 2 *Let* $\mathcal{P} = \mathcal{C} = \mathcal{K} = \mathcal{Z}_{26}$*. For* $0 \leq K \leq 25$*, define*

$$\begin{aligned} e_K(x) &= (x + K) \bmod 26 \\ d_K(y) &= (y - K) \bmod 26 \end{aligned}$$

Here $(x + K)$ mod 26 is the remainder obtained after dividing $(x + K)$ by 26; $(y-K)$ mod 26 is defined in a similar fashion. Now suppose $(y-K)$ is a negative number, the mod operation works as follows: (-7) mod $31 = (-1) \times 31 + 24 = 24$.

We consider an example for the shift cipher below. In the example we will use upper case letters for ciphertext and lower case letters for plaintext, in order to improve readability. We will do this elsewhere as well. Suppose the key for the shift cipher is $K = 11$, and the plaintext is

$$wewillmeetatmidnight$$

Using the correspondence between alphabets and numbers, we obtain the following sequence of integers

22	4	22	8	11	11	12	4	4	19
0	19	12	8	3	13	8	6	7	19

Next, we add 11 to each value, reducing each sum modulo 26:

7	15	7	19	22	22	23	15	15	4
11	4	23	19	14	24	19	17	18	4

Finally, we convert the sequence of integers to alphabetic characters, obtaining the ciphertext

$$HPHTWWXPPELEXTOYTRSE$$

To decrypt the text, Bob will first convert the ciphertext to a sequence of integers, then subtract 11 from each value (reducing modulo 26), and finally convert the sequence of integers to alphabetic characters. I'd expect everyone to try decrypting the ciphertext using the key to see if you can recover the original text.

We will consider three other cryptosystems in this lecture.

Definition 3 *Let $\mathcal{P} = \mathcal{C} = \mathcal{Z}_{26}$. Let $\mathcal{K}$ consist of all possible permutations of the 26 symbols $0, 1, \ldots, 25$. For each permutation $\pi \in \mathcal{K}$, define*

$$\begin{aligned} e_\pi(x) &= \pi(x) \\ d_\pi(y) &= \pi^{-1}(y) \end{aligned}$$

where π^{-1} is the inverse permutation to π.

Here is an example of a permutation π, which could comprise an encryption function. (As before, plaintext characters are written in lower case and ciphertext characters are written in upper case). Thus, $e_\pi(a) = X$ etc. The decryption

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

function is the inverse permutation. This is formed by writing the second lines first, and then sorting in alphabetical order. The following is obtained. Hence, $d_\pi(A) = d$ etc. As an exercise, try encrypting and decrypting the previous message using the substitution cipher.

A key for the substitution cipher consists of a permutation of all the 26 alphabetic characters. The number of possible permutations is 26!, which is

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

more than 4×10^{26}, a very large number. Thus, an exhaustive key search is infeasible, even for a computer. This is better than the shift cipher where $K = 11$ where there are only 25 possibilities (excluding $K = 0$).

The substitution cipher is a *monoalphabetic* cipher, since each alphabetic character is mapped to a unique alphabetic character. Thus, although, the keyspace is large the substitution cipher can be broken using the statistical properties of the English language. This attack uses the relative frequencies of the 26 letters of the English language; it is also useful to consider sequences of two or three consecutive letters called *digrams* and *trigrams* and the relative frequencies of their occurrence. I have a link on the course webpage which illustrates the cryptanalysis of the substitution cipher; you are all encouraged to see how this is done.

One way to improve the security of the substitution cipher is to use a *polyalphabetic* cipher, where each alphabet is mapped into more than one alphabet in the ciphertext. One common example is the *Vignere* cipher.

Definition 4 *Let m be a positive integer. Define $\mathcal{P} = \mathcal{C} = \mathcal{K} = (\mathcal{Z}_{26})^m$. For a key $K = (k_1, \ldots, k_m)$, we define*

$$\begin{aligned} e_K(x_1, x_2, \ldots, x_m) &= ((x_1 + k_1) \bmod 26, (x_2 + k_2) \bmod 26, \ldots, (x_m + k_m) \bmod 26) \\ d_K(y_1, y_2, \ldots, y_m) &= ((y_1 - k_1) \bmod 26, (y_2 - k_2) \bmod 26, \ldots, (y_m - k_m) \bmod 26) \end{aligned}$$

The key is chosen randomly; using the correspondence $A \leftrightarrow 0, \ldots, Z \leftrightarrow 25$ described earlier, we can associate each key K with an alphabetic string of length m called a *keyword*. The Vignere cipher encrypts m alphabetic characters at a time: each plaintext element is equivalent to m alphabetic characters. As an instance, try decrypting the previous message with the keyword $CIPHER$. Here $m = 6$.

The number of possible keywords of length m in a Vignere cipher is 26^m, so even for relatively small values of m, an exhaustive key search would require a long time. This is large enough to preclude exhaustive key search by hand. Also, note that in a Vignere cipher having keyword length m, an alphabetic character can be mapped to one of m possible alphabetic characters (assuming the keyword contains m distinct characters). Hence this cipher is an example of a polyalphabetic cryptosystem. In general, cryptanalysis is more difficult for polyalphabetic than for monoalphabetic cryptosystems.

All the cryptosystems we have discussed so far involve substitution: plaintext characters are replaced by different ciphertext characters. The idea of a *permutation* or *transposition* cipher is to keep the plaintext characters unchanged, but to alter their positions by rearranging them using a permutation.

Definition 5 *Let m be a positive integer. Let $\mathcal{P} = \mathcal{C} = (\mathcal{Z}_{26})^m$, and let $\mathcal{K}$ consist of all permutations of $\{1, \ldots, m\}$. For a key (i.e., a permutation) π, we define*

$$\begin{aligned} e_\pi(x_1, \ldots, x_m) &= (x_{\pi(1)}, \ldots, x_{\pi(m)}) \\ d_\pi(y_1, \ldots, y_m) &= (y_{\pi^{-1}(1)}, \ldots, y_{\pi^{-1}(m)}) \end{aligned}$$

where π^{-1} is the inverse permutation to pi.

Here is an example to illustrate the permutation cipher. Suppose $m = 6$ and the key is the following permutation π: The inverse permutation π^{-1} is given

x	1	2	3	4	5	6
$\pi(x)$	3	5	1	6	4	2

by Now, suppose we are given the plaintext

x	1	2	3	4	5	6
$\pi^{-1}(x)$	3	6	1	5	2	4

$$shesellsseashellsbytheseashore$$

We first partition the plaintext into groups of six letters (since the keysize $m = 6$).

$$shesel|lsseas|hellsb|ythere|ashore$$

Now each group of six letters is rearranged according to the permutation π, yielding the following

$$EESLSH|SALSES|LSHBLE|HSYEET|HRAEOS|$$

So, the ciphertext is

$$EESLSHSALSESLSHBLEHSYEETHRAEOS$$

The ciphertext can be decrypted in a similar fashion, using the inverse permutation π^{-1}; I urge you all to try this out.

Recommended Reading

1. Chapter 1 of Stallings [1] for an introduction to network and computer security. The model for network security is also discussed here.

2. Chapter 1 of Stinson [2] provides a nice introduction to cryptography. The discussion on the shift, substitution, Vignere, and permutation ciphers in this lecture is taken from this reference. There is also a good discussion of these ciphers in Chapter 2 of Stallings. Section 1.2 of Stinson illustrates how the substitution and Vignere ciphers can be broken using a statistical analysis on the ciphertext; there is also some discussion in Section 2.2 of Stallings.

References

[1] W. Stallings, *Cryptography and Network Security: Principles and Practices*, 3rd edition, Prentice Hall, NJ, 2003.

[2] D.R. Stinson, *Cryptography: Theory and Practice*, 2nd edition, Chapman & Hall/CRC, 2002.