CHAPTER 2

Authentication and Basic Cryptography

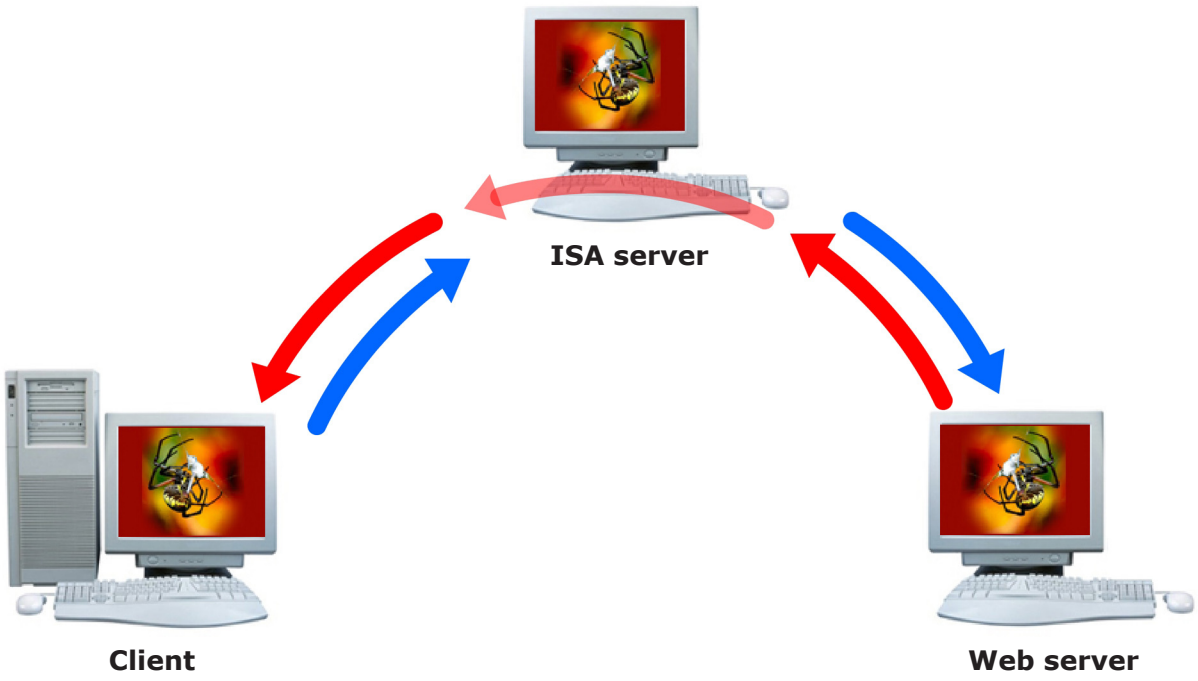
LEARNING OUTCOMES

By the end of this chapter, you should be able to:

1. Understand the mechanisms of authentication;
2. Determine and choose a good password to authenticate authorised user;
3. Understand the concept of cryptography and the algorithms used in cryptography; and
4. Identify the cryptography attacks.

INTRODUCTION

Observe the diagram below.



From the diagram, identify the authentication process and its purpose.

Authentication is a process used to verify transmitted data especially a message. Authentication services provide assurance of identity such as a particular username; an authentication service will provide a means of confirming that this claim is correct. Authentication is the most important of the security services because all other security services depend upon it to some extent.

Authentication relates to the scenario where some party (claimant) has presented a principal's identity and claims to be that principal. Authentication enables some other party (verifier) to gain confidence that the claim is legitimate. Authentication methods are based upon any of the following principles:

- The claimant demonstrates knowledge of something, e.g. password.
- The claimant demonstrates possession of something, e.g. a physical key or card.
- The claimant exhibits some required immutable characteristics, e.g. a finger print.
- Evidence is presented that the claimant is at some particular place or time.
- The verifier accepts that some other party, who is trusted, has already established authentication.

Authentication applies in a particular context. Two distinct contexts encountered in a computer network are (refer to Table 2.1):

Table 2.1: Gustafon's Three Purposes of a Teaching Outline Model

Context	Definition
Entity Authentication (EA)	An identity is presented by a remote party participating in a communication connection or session.
Data Origin Authentication (OA)	An identity is presented along with a data item. It is claimed that the data item originated from the principal identified.

Authenticity is an important factor of trust in the Internet. The trust relationship is under threat from spoofing, eavesdropping, modification and masquerading an insecure transmission. The key to the trust relationship among entities is to be able to verify the true identity of each other. According to that, passwords are a well-known way of providing authentication.

2.1 PASSWORD

The password is used to provide the authentication to the system.

Password is a character string used to authenticate an identity.

Knowledge of the password and its associated user ID is considered proof of authorisation to use the capabilities associated with that user ID. The use of password is straightforward, as shown in Figure 2.1.

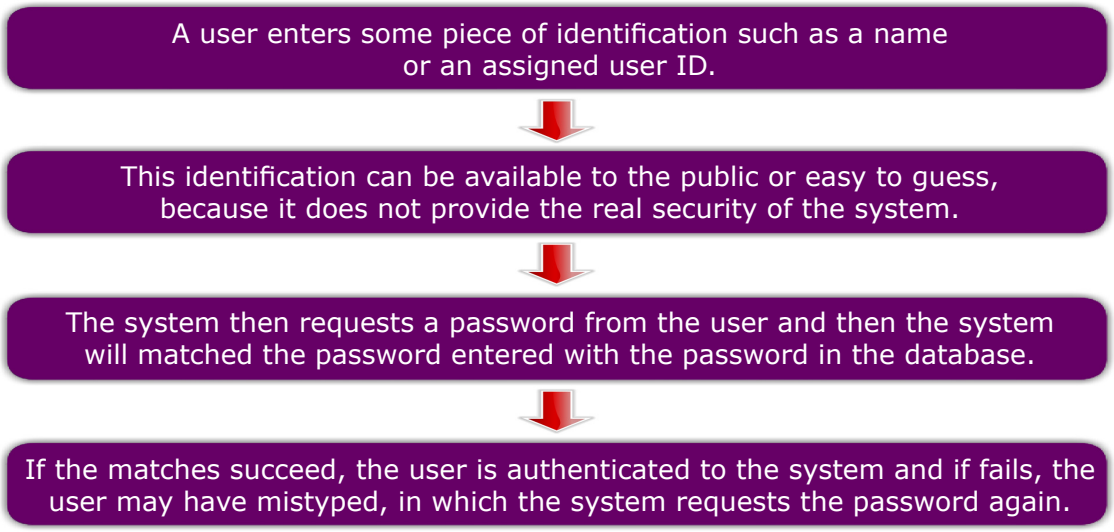


Figure 2.1: How password works

The most significant way to maintain information and system security is to select a good password and to keep it confidential. There are two major criteria; it should be hard to guess and it should be easy to remember. According to that we must know how to choose a good password and the simple common-sense precautions to protect our password.

2.1.1 Simple Common-Sense Precautions To Protect Passwords

There are a few steps and precautions in order for you to protect your password. These steps are shown in Figure 2.2 below:

Figure 2.2: Simple common sense to protect your password

2.1.2 How To Choose A Good Password?

Now that you have understood how to protect your password, let us look at what are the actions to consider in choosing a good password. Refer to Figure 2.3 for more details.

Figure 2.3: How to choose a good password

2.1.3 Calculations On Password

Numerous studies have shown that it is much easier for an adversary to penetrate a system on which there are user-generated passwords than on one using randomly selected passwords. Constraining controls on the user selections (minimum number of characters, maximum number of identical characters, etc.) are helpful, but this type of protection is at best a "patchwork" solution. It is more appropriate to address the real problem, which is the population from which the passwords are selected.

The population problem is that a good password routine can make a random selection

from a large (millions to hundreds of millions) population of potential passwords. A good dictionary may only contain tens of thousands of usable words. The words that come to mind during a user selection process may be in the range of tens or hundreds and the probability of selecting each one is not equally distributed. Thus, an adversary enjoys a significant advantage when attempting to breach a security system on which user-generated passwords are allowed.

Let us look at the formulas used to choose a good password and to protect our password from attacked by the hacker or any unauthorised user.

Password Population, N

Formula 1

N*, password population = *rs

Where:
r = the number of possibilities for each character;
s = the number of characters

Formula 2

1/N

Where it means that the probability of guessing a password on one attempt

Formula 3

P = nt/N

Where:
P = probability of success;
n = the number of guessing per unit time;
t = time;
N = password population

From the formulas given above, let's look at several examples given.

Example 1

Problem:

Let say you can choose character from A - Z and the number of character requires are 6. What is the population of password?

Solution:

r = 26 (A- Z); s = 6

Therefore,

$N = r^s = 26^6 = 308,915,776$

The probability of guessing on one attempt is;

= 1/N = 1/308,915,776 = 3.24 x 10-9

Example 2

Problem:

Let say you are given the number of possibilities for each character is 36, the number of characters is 6, 7 and 8 and the population of password is 2.9 x 1012. You are given 3 months to change the old password to the new password and your computer can process at 200MIPS (Million Instruction per Second) and also no limited attempt. Determine whether there is any possibility to penetrate to the system.

Solution:

P = nt/N; t = PN/n; n = 200 x 106 /sec

t = 3 month x (30 day/1 month) x (24 hour/1day) x (3600 second/1 hour)
= 7.776 x 106 seconds

Therefore:

$P = nt/N = [(7.776 \times 10^6 \text{ sec})(200 \times 10^6 \text{ /sec})] / (2.9 \times 10^{12}) \text{ /sec}$
$= 446.9 \times 10^{-1} = 44.7$

Conclusion:

With the ability of the computer is 200 MIPS, you can penetrate the system where P > 1.

Example 3

Problem:

You are given the population of password is 2.9 x 1012, the probability of success is 1 and the number guesses per unit time is 200 MIPS. How much time do you take to guess the right password?

Solution:

N = 2.9 x 10^{12}; P = 1; n = 200 MIPS

Therefore:

t = PN/n = [(1)(2.9 x 10^{12})]/(200 x 10^{6} /sec) = 14,500 sec
= 14,500 sec x (1 hour/3600 sec) = 4.03 hour

Conclusion:

You need at least 4 hours to guess the right password.

2.2 INTRODUCTION TO BASIC CRYPTOGRAPHY AN CRYPTOGRAPHY ALGORITHM

Source: http://www.youtube.com/watch?v=8zANm-GDWtQ

From the video, define your understanding on Cryptography.

2.2.1 What is Cryptography?

Cryptography is the science of secret writing. As we thrive into an information society, the technological means for global surveillance of millions of individual people are becoming available to major governments. Cryptography has become one of the main tools for privacy, trust, access control, electronic payments, corporate security, and countless other fields. We should be able to make full use of cryptography in our lives in modern society.

Encryption is a process of encoding a message so that its meaning is not obvious. **Decryption** is the reverse process: transforming an encrypted message back to its normal form. A system for encryption and decryption is called a **cryptosystem**. The terms **encode** and **decode** or **encipher** and **decipher** are used instead of the verbs **encrypt** and **decrypt**. More specifically, encoding is the process of translating entire words or phrases to other words or phrases, while enciphering is translating letters or symbols individually. Encryption covers both encoding and enciphering.

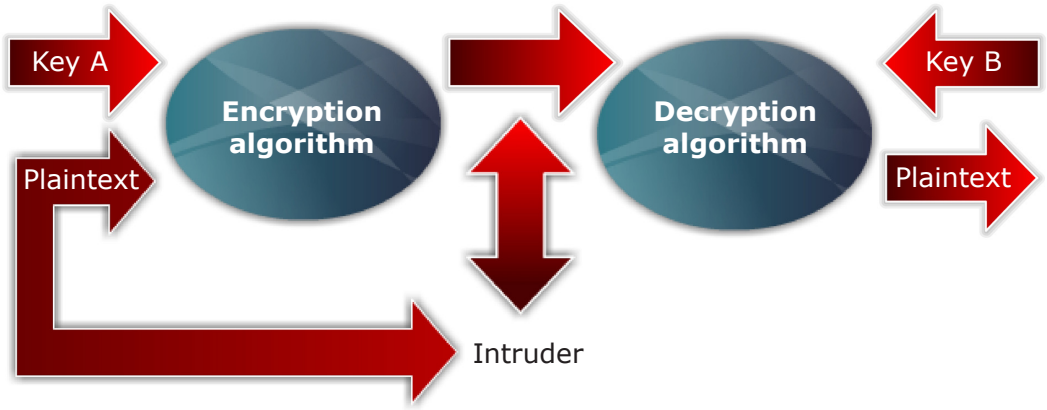


Figure 2.4: Simplified model of encryption

Referring to Figure 2.4, let say someone wants to send a message to a receiver, and he does not want anyone to read the message. The message can be encoded in such way to hide its contents from outsiders. This is called message encryption. The encryption process consists of an encryption algorithm and a key. Changing the key changes the output of the algorithm. The original intelligible message is called plaintext. The encrypted message is called ciphertext, in a form that is nonsense and unintelligible. The ciphertext may be transmitted, and once received by the receiver, a decryption algorithm is needed to transform the message back to original plaintext.

For most algorithms, the key used to transform the message back to plaintext is the same as the key which is used to transform the message to ciphertext.

While cryptography is on how to keep the messages secret, cryptanalysis is on breaking ciphers, i.e. retrieving the plaintext without knowing the proper key. People who do **cryptography** are cryptographers, and practitioners of **cryptanalysis** are **cryptanalysts**.

A cryptanalyst can do any or all of three different things that are attempt to break a single message, attempt to recognise patterns in encrypted messages, in order to be able to break subsequent ones by applying a straightforward decryption algorithm and attempt to find general weaknesses in an encryption algorithm, without necessarily having intercepted any messages.

The security of encryption depends on several factors that may include the algorithm must be powerful enough so that it is impractical to decrypt the algorithm on the basis of the ciphertext alone and the secrecy of the key, not the secrecy of the algorithm.

SELF-CHECK

What is the definition of:

- Cryptography
- Cryptanalysis
- Encryption
- Cryptosystem
- Ciphertext

2.2.2 Basic Cryptographic Algorithms

To be of practical value, a cryptosystem must be simple to be deciphered, yet unintelligible to an interceptor. Some encryption algorithms use a key K, so that the ciphertext message, C depends on both the original plaintext message, P and the key value, K, denoted $C = E(K,P)$ with E the encryption algorithm.

There are two categories of basic algorithms in cryptography as shown in Figure 2.5:

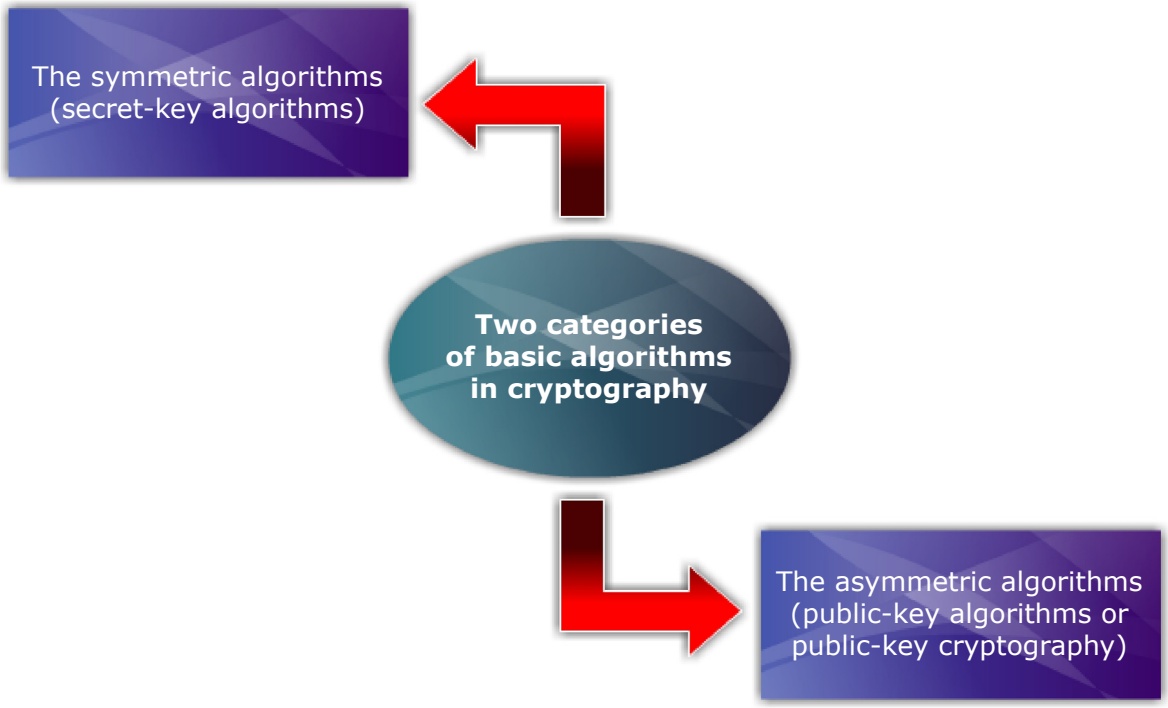


Figure 2.5: The 2 categories of algorithms in cryptography

From Figure 2.5, we will look into each of the category in details.

1. The symmetric algorithms (secret-key algorithms)

These algorithms uses the same key for encryption and decryption, or the decryption key can easily be derived from the encryption key. Since the keys are the same;

$$P = D(K, E(K, P))$$

The algorithms can be divided into stream ciphers and block ciphers (see Figure 2.6).

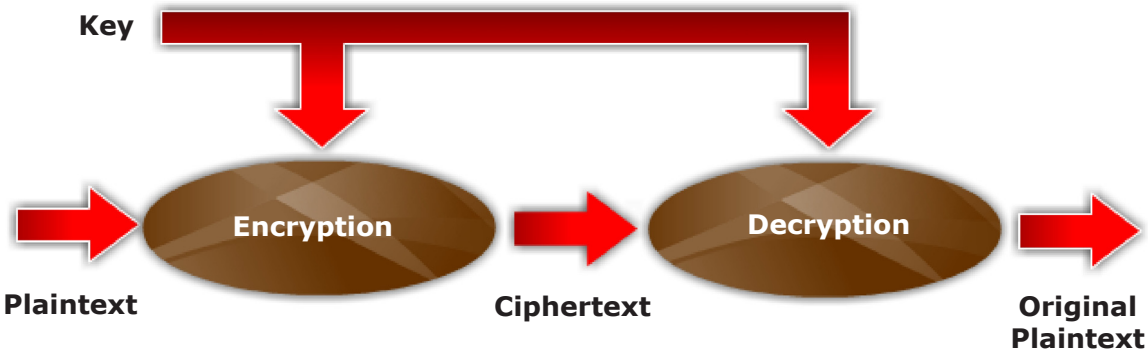


Figure 2.6: Symmetric cryptosystem

2. The asymmetric algorithms (public-key algorithms or public-key cryptography)

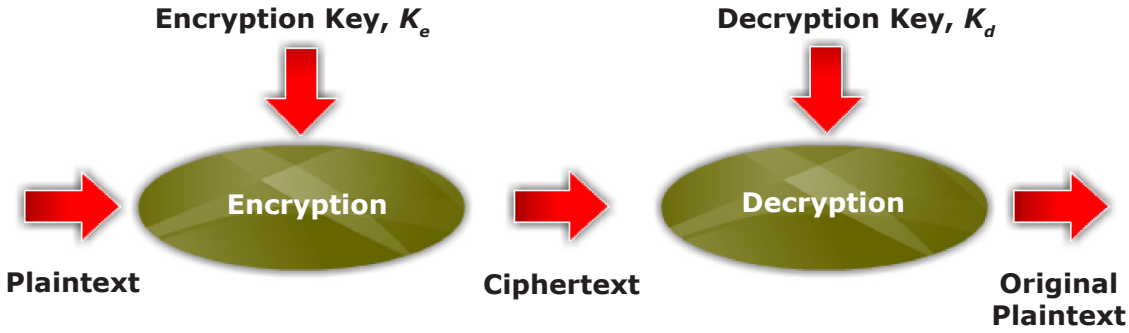


Figure 2.7: Asymmetric cryptosystem

In Figure 2.7, these algorithms allow the key to be publicised, hence anyone can encrypt using the key, and the proper recipient who knows the decryption key can decrypt the message. The encryption key is called public key and the decryption key is called secret/private key. The decription key, K_d, inverts the encryption of key, K_e, so that;

$$P = D(K_d, E(K_e, P))$$

Most encryption algorithms are mathematical in nature, i.e. can be explained in mathematical form. All ciphers are based on two basic methods as shown in Figure 2.8.

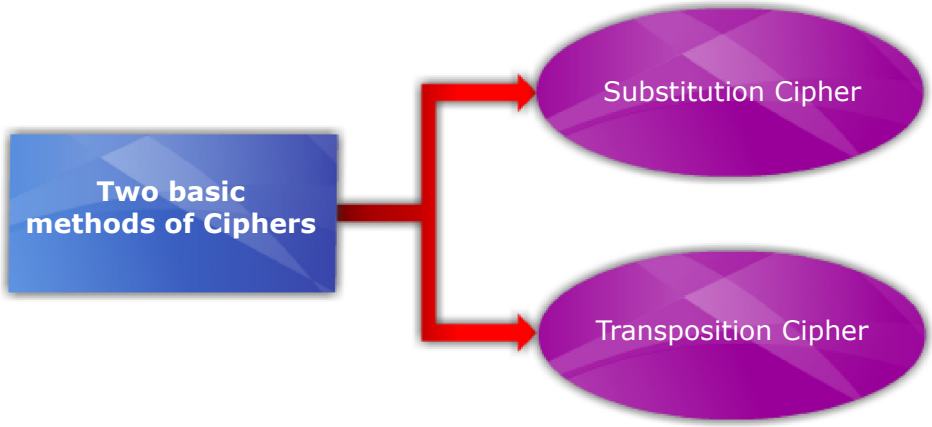


Figure 2.8: The two basic methods of ciphers

(a) **Substitution Cipher**

In a substitution technique, a letter in the plaintext is replaced by other letters or by numbers or symbols. If the plaintext is a sequence of bits, then the substitution involves replacing the bit patterns with the ciphertext bit patterns.

Substitution Techniques

- **Monoalphabetic Substitution Cipher (Caesar Cipher)**

  The cipher alphabet is formed by shifting the letters of the original alphabet. For example, by replacing each letter of the alphabet with the letter three places down the alphabet. It is monoalphabetic as only one letter in plaintext is exchanged for one letter of ciphertext.

Plaintext alphabet	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Ciphertext key	D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

A brute-force cryptanalysis is easily performed: by simply trying all the keys available. The characteristics, which enable the use of brute-force cryptanalysis, are:

1. The encryption and decryption algorithms are known.
2. There are only 26 keys to try.
3. The language of the plaintext is known and easily recognisable. The pattern is obvious.

Caesar cipher is far from secure. It is seldom used by professional cryptographers.

- **Polyalphabetic Substitution Cipher**

  This cipher ensures greater secrecy than Caesar cipher. Two or more cipher alphabets, which usually are interrelated, are used to encipher a message. An extension of the monoalphabetic system through 26 alphabets can be formed.

  For example, in the Figure 2.5, **A** is represented as **B** in the second alphabet, C in the third alphabet and so on. The square is called Vigenere tableau.

  Example:

Plaintext	**descr iptio ncann otsui titse lfinw ordsx** (the 'x' added is called null)
Key	**lifel essli feles slife lessl ifele sslif** (letters of **lifeless** act as repeating key for each letter of the plaintext)

For example, by using the Vigenere tableau, the letter **S** enciphered by key **E** gives cipher letter **G**. Then for the plain text above yields:

omxgc mhktw sglrf geazm emlkp tkmya gjoac

Polyalphabetic substitutions are not immune to breaking. The method to break such an encryption is to determine the number of alphabets used, break the ciphertext into pieces that were enciphered with the same alphabet; and solve each piece as a monoalphabetic substitution (see Figure 2.9).

	Plaintext Letter																								
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Key Letter A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Figure 2.9: Polyalphabetic substitution cipher

There are two tools that can decrypt messages written even with a large number of alphabets; the Kasiski method, to determine when a pattern of encrypting permutations has repeated and the index of coincidence, to predict the number of alphabets used for substitutions.

(b) **Transposition Cipher**

This mapping is achieved by performing some sort of permutation on the plaintext letters. In this method, the letters are not replaced, but rearranged. The letters are retained but moved from its position.

The goals of a substitution are:

i. Confusion, an attempt to make it difficult to determine how a message and key were transformed into ciphertext.
ii. Diffusion, to spread the information from the message or they key out widely across the ciphertext.

- **Unkeyed Single Transposition**

It is one of the simplest methods in encipherment. For example, the plaintext is '**time is of essence**'. It is inscribed in matrix with predetermined vertical and horizontal components, e.g. a 3 (horizontal) X 6 (vertical) by matrix.

Table 2.2: Unkeyed Single Transposition

d	e	s	c	r	i	p	t	i
o	n	c	a	n	n	o	t	s
u	i	t	i	t	s	e	s	f
i	n	w	o	r	d	d	x	x

The **ciphertext** is taken from horizontal rows. For example, in arbitrary units of three letters each:

doui enin sctw caio rntr insd poes ttlx isfx

Decipherment: The decipherer must make a matrix identical to the original and inscribe the ciphertext, which is read from the vertical columns.

- **Keyed Single Transposition - Columnar Transposition**

In this scheme, the message is written in a rectangle, row by row, and the message is read column by column. It makes use of a key which determines the horizontal length of the matrix. For example: **the plaintext is 'attack postponed until tomorrow**' with the key '2584316'. The ciphertext is '**aodo tsum ttno aptr coir knlo petw**'.

Any other random letter can be used for better security.

Table 2.3: Columnar Transposition

2	5	8	4	3	1	6
a	t	t	a	c	k	p
o	s	t	p	o	n	e
d	u	n	t	i	l	t
o	m	m	r	r	o	w

A pure transposition cipher is easily recognised because it has the same letter frequencies as the original plaintext. The transposition cipher can be made more secure by performing more than one stage of transposition.

(c) Stream Cipher

A stream cipher is one that encrypts a digital data stream one bit or one byte at a time. The transformation depends on the symbol, the key and control information of the encipherment algorithm.

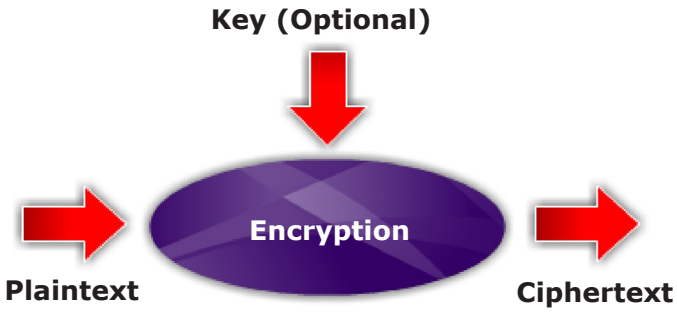


Figure 2.10: Model of Stream Enciphering

Advantages and Disadvantages of Stream Ciphers

Table 2.4 below shows the advantages and disadvantages of stream ciphers.

Table 2.4: Advantages and Disadvantages of Stream Ciphers

Advantages	Disadvantages
1. Speed of transformation. • Each symbol in encrypted without regard for any other plaintext symbols, so each symbol can be encrypted as soon as it is read. • The time it takes to encrypt each symbol depends on the encryption algorithm itself.	1. Each symbol is separately enciphered. • All the information is contained in one symbol of the ciphertext. • A symbol is then considered as a separate entity and attempt can be made to break it using the characteristics of all individual symbols in the ciphertext using tools.
2. Low error propagation. • Each symbol is separately encoded; therefore each error in the encryption process affects only that character. • The conversion of later characters in the stream is not affected.	2. Susceptibility to malicious insertions and modifications. • If the code is broken, the interceptor can insert pieces of previous messages and transmit a new message.

(d) Block Cipher

A block cipher, as shown in Figure 2.11 is one in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length. Normally a block size of 64 bits is used. Examples of block ciphers are columnar transposition and other transpositions. In the columnar transposition, the entire message is translated as one block.

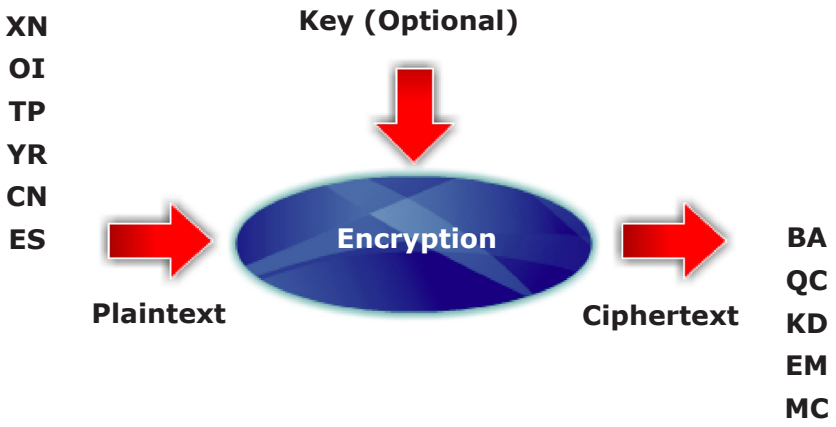


Figure 2.11: Model of block enciphering

Advantages and Disadvantages of Block Ciphers

There are several advantages and disadvantages of block ciphers. These can be found in Table 2.5 below.

Table 2.5: Advantages and Disadvantages of Block Ciphers

Advantages	Disadvantages
1. Information from the plaintext is diffused into several ciphertext symbols.	1. Slowness of encryption. • In block ciphering, the encryption process can only start only after the entire block of plaintext is received. • For columnar transposition, which is the extreme case, the delay must be for the entire message.
2. Immunity to insertions. • Since enciphering is done by blocks, insertion of a single symbol into one block is impossible. The length of the block would be incorrect. The insertion is then revealed during decipherment. • One character in plaintext is encrypted into not just one ciphertext. Therefore, an interceptor cannot just exchange a character in a message with a new one.	2. Error propagation • An error in encryption will affect the transformation of all other characters in the same block.

2.3 PKI (PUBLIC-KEY INFRASTRUCTURE) AND KEY MANAGEMENT

PKI is a method for authenticating a message sender or encrypting a message. It enables users of an insecure public network, such as the Internet, to securely and privately exchange data through the use of a public and a private cryptographic key pair that is obtained and shared through a trusted authority. It provides for a digital certificate that can identify an individual or an organisation and directory services that can store and, when necessary, revoke the certificates.

PKI is consists from a set of (see Figure 2.12):

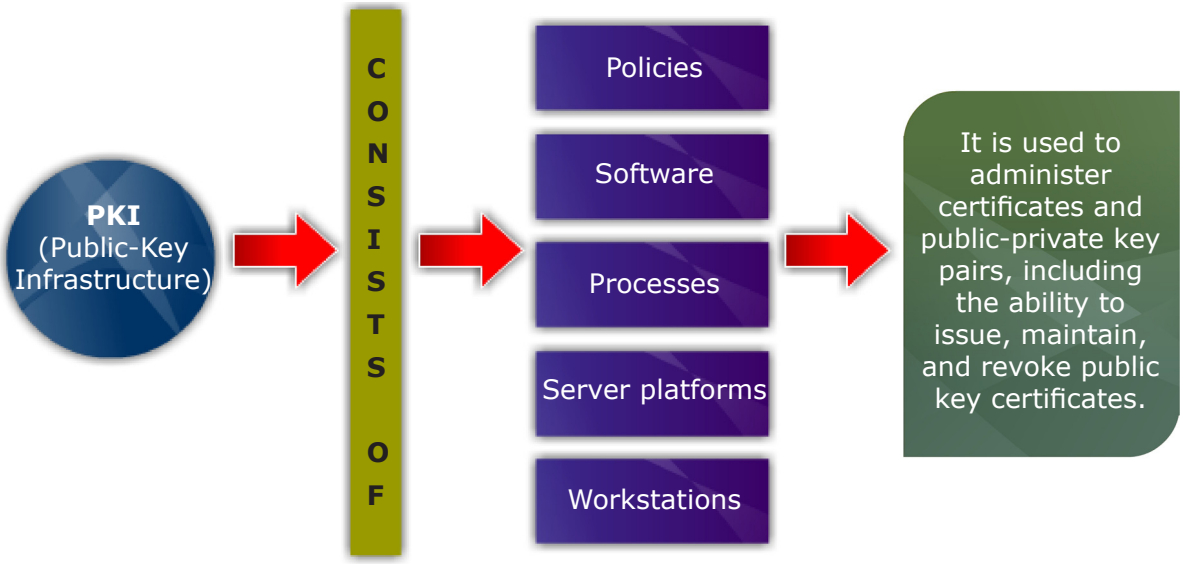


Figure 2.12: The content in PKI (Public Key Infrastructure)

The infrastructure used to create a secure chain of trust for Internet-based communications. A PKI solution consists of:

- A security policy, a Certificate Authority (CA);
- A Registration Authority (RA);
- Certificate distribution system, and
- PKI-enabled applications.

A PKI is based on asymmetric encryption and digital signatures technologies. It enables two parties to exchange confidential electronic messages and to enter into legally binding agreements over the Internet.

In cryptography, a public key infrastructure (PKI) is an arrangement which provides for third-party vetting of, and vouching for, user identities. It also allows binding of public keys to users. This is usually carried by software at a central location together with other coordinated software at distributed locations. The public keys are typically in certificates.

The term is used to mean both the certificate authority and related arrangements as well as, more broadly and somewhat confusingly, to mean use of public key algorithms in electronic communications. The later sense is erroneous since PKI methods are not required to use public key algorithms.

2.3.1 How the Public Key Cryptography Concept Works

Public-key cryptography uses a pair of mathematically related cryptographic key. If one key is used to encrypt information, then only the related key can decrypt that information. If you know one of the keys, you cannot easily calculate what the other one is.

As a result, in a 'public key system' you have a public key that is something that you make public - it is freely distributed and can be seen by all users, and a corresponding (and unique) private key that is something that you keep secret which it is not shared amongst users. Your private key enables you to prove, unequivocally, that you are who you claim to be.

The next sections describe how these keys are used in practice. They consider using separate public and private keys for encryption than those used for signing.

2.3.2 The Public Key Used for Encryption

Another person uses your public encryption key when they want to send you confidential information. The information to be sent is encrypted using your public key. You can provide your public key to the sender, or it can be retrieved from the directory in which it is published.

In normal practice, the actual information being sent is encrypted using a secret key algorithm (symmetric cryptography). Symmetric algorithms are much faster than public/private key algorithms (asymmetric cryptography). A random key (the session key) is generated, and it is used with the symmetric algorithm to encrypt the information. The public key is then used to encrypt that key and both are sent to the recipient.

2.3.3 The Private Key Used for Decryption

A private key is used to decrypt information that has been encrypted using its corresponding public key. The person using the private key can be certain that the information it is able to decrypt must have been intended for them, but they cannot be certain who the information is from.

In normal practice the private key is used to decrypt the session key, and that key is used to decrypt the actual information rather than the private key decrypting all the information.

2.3.4 The Private Key for Signature

If the sender wishes to prove to a recipient that they are the source of the information (perhaps they accept legal responsibility for it) they use a private key to digitally sign a message (a digital signature). Unlike the handwritten signature, this digital signature is different every time it is made.

A unique mathematical value, determined by the content of the message, is calculated using a 'hashing' or 'message authentication' algorithm. Then this value is encrypted with the private key which is creating the digital signature for this specific message. The encrypted value is either attached to the end of the message or is sent as a separate file together with the message. The Public Key corresponding to this private key may also be sent with the message, either on its own or as part of a certificate.

Anyone receiving information protected simply by a digital signature can check the signature and can read and process the information. Adding a digital signature to information does not provide confidentiality.

2.3.5 The Public Key for Signature

The receiver of a digitally signed message uses the correct public key to verify the signature by performing the following steps. A non-technical example is given after these steps:

- The correct public key is used to decrypt the hash value that the sender calculated for the information.
- Using the hashing algorithm (where certificates are in use it will be stated in the public key certificate sent with the message), the hash of the information received is calculated.
- The newly calculated hash value is compared to the hash value that the sender originally calculated. This was found in step 1 above. If the values match, the receiver knows that the person controlling the private key corresponding to the public key sent the information. They also know that the information has not been altered since it was signed.
- If a public key certificate was sent with the information it is then validated with the CA that issued the certificate to ensure that the certificate has not been falsified and therefore the identity of the controller of the private key is genuine.
- Finally, if one is available, the revocation list for the CA is checked to ensure that the certificate has not been revoked, or if it has been revoked, what the date and time of revocation.

As an example, suppose you are sent a Word document by e-mail. The sender has signed it by calculating a hash value for that Word document, and then encrypted that value with their private key. You receive the Word document, and calculate the hash value for it. You decrypt the hash vale that the sender encrypted and compare the two. If they are equal, the document hasn't changed and you are certain who sent the document. If they don't match you know that the document has changed or the sender is not who they claimed. If no errors have been found, the receiver can now be certain of the authenticity and accuracy of the information that has been received.

The following table summarises who uses public and private keys and when:

Table 2.6: Summarisation of Public and Private Keys

Key Function	Key Type	Whose Key Used
Encrypt data for a recipient	Public key	Receiver
Sign data	Private key	Sender
Decrypt data received	Private key	Receiver
Verify a signature	Public key	Sender

To encrypt information that will be stored for your own use (that is, you will be the only person able to read it), you must use your own Public Key as the recipients key (you are the recipient) in order to be able to decrypt and read the information later. If you use someone else's Public Key, then only they will be able to decrypt and read the information. To avoid the difficulty associated with not being able to read encrypted messages if you are not one of the recipients, some systems do not delete the original message after encryption whilst others store a copy of the key used for encryption either under the sender's Public Key or under a System Recovery Key.

Public Key Cryptography is therefore used for the encryption/decryption and signing/verification of information. Encrypting information ensures privacy by preventing unintended disclosure, and signing messages authenticates the sender of the message and ensures the message has not been modified since it was sent. It has to be remembered that only the information signed/encrypted has been protected. Commonly in e-mail systems headers, addresses and body messages may have no protection at all and should not be considered secure or part of the protected information.

2.3.6 The Certificate

In the section on public and private keys, references were made to certificates. A certificate is information referring to a public key that has been digitally signed by a Certification Authority (CA). The information normally found in a certificate conforms to the ITU (IETF) standard X.509 v3. Certificates conforming to that standard includes:

- Information about the published identity of the owner of the corresponding private key;
- The key length;
- The algorithm used and associated hashing algorithm;
- Dates of validity of the certificate; and
- The actions the key can be used for.

A certificate is not essential to operating a Public Key Infrastructure (PKI), however, some scheme is necessary to locate information about the controller of a private key and X.509 certificate is the most commonly implemented the scheme.

2.3.7 Controlling Key Usage

One of the fields in a public key certificate (certificate) is the key usage field. It is used by the CA to state the uses the CA has approved. It does not mean that the corresponding private key cannot be used in any other ways. There is no certificate with a private key. People receiving information protected using a public key system should check, where a certificate is provided, that the key usage stated in the certificate corresponds to the actual use.

2.3.8 Storing Methods for Public and Private Keys Certificates

Public keys are stored within digital certificates along with other relevant information (user information, expiration date, usage, who issued the certificate etc.). The CA enters the information contained within the certificate when it is issued and this information cannot be changed. Since the certificate is digitally signed and all the information in it is intended to be publicly available there is no need to prevent access to reading it, although you should prevent other users from corrupting, deleting or replacing it.

2.3.9 Protection

If someone gains access to your computer they could easily gain access to your private key(s). For this reason, access to a private key is generally protected with a password of your choice. Private Key passwords should never be given to anyone else and should be long enough so that they are not easily guessed. This is the same as looking after your ATM CARD and it's PIN. If someone manages to get hold of your card then the

only thing that prevents him or her using it is the PIN (password) protecting it. If someone has your PIN then they can take your money and you can't stop them.

Different vendors often use different and sometimes proprietary storage formats for storing keys. For example, Entrust uses the proprietary .epf format, while Verisign, GlobalSign, and Baltimore, to name a few, use the standard *.p12* format.

2.3.10 The Components of PKI

A public key infrastructure is created by combining a number of services and technologies as shown in Figure 2.13:

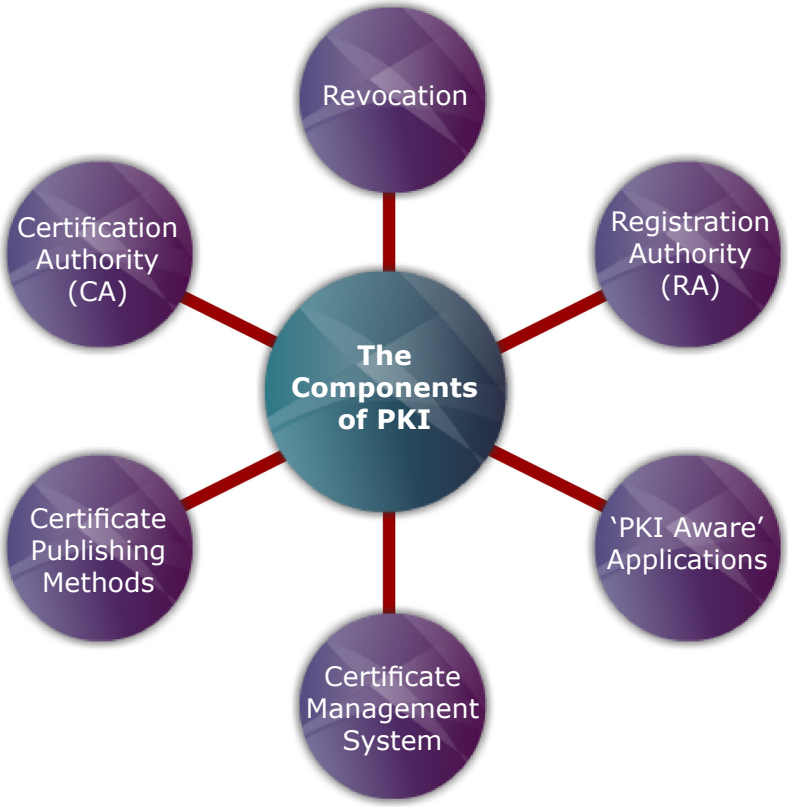


Figure 2.13: The components of PKI

1. Certification Authority (CA)

A ***Certification Authority (CA)*** issues and verifies certificates. The Certification Authority (CA) takes responsibility for identifying (to a stated extent) the correctness of the identity of the person asking for a certificate to be issued, and ensures that the information contained within the certificate is correct and digitally signs it.

(a) **Generating Key Pairs**

The Certification Authority (CA) may generate a public key and a private key (a key pair) or the person applying for a certificate may have to generate their own key pair

and send a signed request containing their public key to the CA for validation. The person applying for a certificate may prefer to generate their own key pair so as to ensure that the private key never leaves their control and as a result is less likely to be available to anyone else.

(b) **Issuing Certificates**

Unless you generate your own certificate (some applications software will enable you do this) you will generally have to purchase one from a CA. Before a CA issues you with a certificate they will make various checks to prove that you are who you say you are. The Certification Authority (CA) could be thought of as the PKI equivalent of a passport agency where the Certification Authority (CA) issues you a certificate after you provide the credentials they require to confirm your identity, and then the CA signs (stamps) the certificate to prevent modification of the details contained in the certificate.

A CA may also state the quality of the checks that were carried out before the certificate was issued. Different classes of certificate can be purchased that correspond to the level of checks made. There are **three or four** general classes of certificate:

- Class 1 certificates can be easily acquired by supplying an email address;
- Class 2 certificates require additional personal information to be supplied; and
- Class 3 certificates can only be purchased after checks have been made as to the requestors' identity.

A 4th class may be used by governments and organisations needing very high levels of checking.

(c) **Using Certificates**

An individual may have any number of certificates issued by any number of Certification Authorities. Different Web applications may insist that you use certificates issued only by certain Certification Authorities. For example, a bank may insist that you use a certificate issued by them in order to use their services, whereas a public Web site may accept any certificate you offer (just as some allow free choice of ID and password). The CA can be a unit within your organisation, a company (i.e. a bank or a post office), or an independent entity (VeriSign).

(d) **Verifying Certificates**

The public key certificate is signed by the CA to prevent its modification or falsification. This signature is also used when checing that the public key is still valid. The signature is validated against a list of 'Root CAs' contained within various 'PKI aware' applications (e.g. your browser). Some CA certificates are called 'Root Certificates' as they form the root of all certificate validation. Certificate validation occurs automatically using the appropriate public certificate contained within the root CA list.

PGP (Pretty Good Privacy) users normally act as their own issuing authority, so you accept their certificate on the basis that they are who they say they are without further verification. This method is called the 'Web of trust' because it is based upon people you trust rather than liability by contract.

2. Revocation

Where a system relies upon publishing certificates so that people are able to communicate with each other, there has to be a system for letting people knows when certificates are no longer valid. It can be done in one of two ways. Certificates can be deleted from the Directory or database in which they should be found. As a result, any attempt to find them to check that they still exist will fail and anyone looking for them would know that they have been revoked.

There are two problems with this approach:

i. The first is that a denial of service attack on the Directory or database might create the appearance of a failed certificate.
ii. The second is that the Directory was designed to optimise the time to read information, so deleting information is normally avoided, as is updating. Also, deleting the record does not tell the person asking for the information why it is not there, and they may need to know why and when it was removed.

As a result, a system of revocation lists has been developed that exists outside the Directory or database. This is a list of certificates that are no longer valid (for whatever reason), equivalent to a lost or stolen ATM card list.

There are currently two different methods for checking for certificate revocation which are **'CRL'** or **'OCSP'**. Revocation lists may be publicly available even when the matching Directory or database is not. This is because certificates may have been distributed for use beyond the private network of the organisation involved.

3. Registration Authority (RA)

A CA may use a third-party called a Registration Authority (RA) to perform the necessary checks on the person or company requesting the certificate to ensure that they are who they say they are. That RA may appear to the certificate requestor as a CA, but they do not actually sign the certificate that is issued.

4. Certificate Publishing Methods

One of the fundamentals of PKI systems is the need to publish certificates so that users can find them. (You must be able to get hold of the public encryption key for the recipient of encrypted information.) There are two ways of achieving this. One is to publish certificates in the equivalent of an electronic telephone directory. The other is

to send your certificate out to those people you think might need it by one means or another.

The most common approaches are listed in Table 2.7:

Certificate Publishing Methods	Descriptions
Directories	Directories are databases that are X.500/LDAP-compliant. This means that they contain certificates in the X.509 format, and that they provide specific search facilities as specified in the LDAP standards published by the IETF. Directories may be made publicly available or they may be private to a specific organisation – i.e. a company may have its own directory where it holds the certificates for its users and only its users can access this directory. A Directory is kept private when it contains information that the owner does not wish to be publicly available. Public directories on the other hand can be read by anyone with access to them.
Databases	A database can be configured to accept X.509 format certificates. This may be done for private systems where the search methods for locating certificates do not follow the LDAP structure. Because it is essentially proprietary, this method is not used for public systems.
Email, floppy discs etc.	Certificates may be sent within an e-mail so that the recipient can add them to their own collection on their server or desktop, depending upon the way their security systems have been configured. They may also be put onto floppy discs, or any other medium.

5. Certificate Management System

This term refers to the management system through which certificates are published, temporarily or permanently suspended, renewed or revoked. Certificate management systems do not normally delete certificates because it may be necessary to prove their status at a point in time, perhaps for legal reasons. A CA (and perhaps an RA) will run certificate management systems to be able to keep track of their responsibilities and liabilities.

6. 'PKI aware' Applications

This term usually refers to applications that have had a particular CA software supplier's toolkit added to them so that they are able to use the supplier's CA and certificates

to implement PKI functions. The term does not mean that the applications have any 'knowledge' built into them about what the security requirements really are, or which PKI services are relevant to delivering them. These issues are quite separate from having PKI services available.

Strength of Cryptographic Algorithms

Good cryptographic systems should always be designed so that they are as difficult to break as possible. It is possible to build systems that cannot be broken in practice (though this cannot usually be proved). This does not significantly increase system implementation effort; however, some care and expertise is required. There is no excuse for a system designer to leave the system breakable. Any mechanisms that can be used to circumvent security must be made explicit, documented, and brought into the attention of the end users.

In theory, any cryptographic method with a key can be broken by trying all possible keys in sequence. If using brute force to try all keys is the only option, the required computing power increases exponentially with the length of the key.

- A 32 bit key takes 232 (about 109) steps. This is something anyone can do on his/her home computer.
- A system with 40 bit keys takes 240 steps - this kind of computation requires something like a week (depending on the efficiency of the algorithm) on a modern home computer.
- A system with 56 bit keys (such as DES) takes a substantial effort (with a large number of home computers using distributed effort, it has been shown to take just a few months), but is easily breakable with special hardware. The cost of the special hardware is substantial but easily within reach of organised criminals, major companies, and governments.
- Keys with 64 bits are probably breakable now by major governments, and within reach of organised criminals, major companies, and lesser governments in few years.
- Keys with 80 bits appear good for a few years, and keys with 128 bits will probably remain unbreakable by brute force for the foreseeable future. Even larger keys are sometimes used.

However, key length is not the only relevant issue. Many ciphers can be broken without trying all possible keys. In general, it is very difficult to design ciphers that could not be broken more effectively using other methods. Designing your own ciphers may be fun, but it is not recommended for real applications unless you are a true expert and know exactly what you are doing.

One should generally be very wary of unpublished or secret algorithms. Quite often the designer is then not sure of the security of the algorithm, or its security depends on the secrecy of the algorithm. Generally, no algorithm that depends on the secrecy of the algorithm is secure. Particularly in software, anyone can hire someone to disassemble

and reverse-engineer the algorithm. Experience has shown that the vast majority of secret algorithms that have become public knowledge later have been pitifully weak in reality.

The key lengths used in public-key cryptography are usually much longer than those used in symmetric ciphers. This is caused by the extra structure that is available to the cryptanalyst. There the problem is not that of guessing the right key, but deriving the matching secret key from the public key.

In the case of RSA, this could be done by factoring a large integer that has two large prime factors. In the case of some other cryptosystems it is equivalent to computing the discrete logarithm modulo a large integer (which is believed to be roughly comparable to factoring when the moduli are a large prime number). There are public key cryptosystems based on yet other problems.

To give some idea of the complexity for the RSA cryptosystem;

- A 256 bit modulus is easily factored at home, and 512 bit keys can be broken by university research groups within a few months;
- Keys with 768 bits are probably not secure in the long term;
- Keys with 1024 bits and more should be safe for now unless major cryptographical advances are made against RSA; and
- Keys of 2048 bits are considered by many to be secure for decades.

It should be emphasised that the strength of a cryptographic system is usually equal to its weakest link. No aspect of the system design should be overlooked, from the choice algorithms to the key distribution and usage policies.

2.4 METHODS OF CRYPTOGRAPHY ATTACKS

Cryptanalysis is the art of deciphering encrypted communications without knowing the proper keys. There are many cryptanalytic techniques. Some of the more important ones for a system implementer are described in Table 2.8 below.

Table 2.8: Methods of Cryptography Attacks

Cryptanalytic Techniques	Descriptions
Ciphertext-only attack	This is the situation where the attacker does not know anything about the contents of the message, and must work from ciphertext only. In practice it is quite often possible to make guesses about the plaintext, as many types of messages have fixed format headers. Even ordinary letters and documents begin in a very predictable way. For example, many classical attacks use frequency analysis of the

	ciphertext, however, this does not work well against modern ciphers. Modern cryptosystems are not weak against ciphertext-only attacks, although sometimes they are considered with the added assumption that the message contains some statistical bias.
Known-plaintext attack	The attacker knows or can guess the plaintext for some parts of the ciphertext. The task is to decrypt the rest of the ciphertext blocks using this information. This may be done by determining the key used to encrypt the data, or via some shortcut. One of the best known modern known-plaintext attacks is linear cryptanalysis against block ciphers.
Chosen-plaintext attack	The attacker is able to have any text he likes encrypted with the unknown key. The task is to determine the key used for encryption. A good example of this attack is the differential cryptanalysis which can be applied against block ciphers (and in some cases also against hash functions). Some cryptosystems, particularly RSA, are vulnerable to chosen-plaintext attacks. When such algorithms are used, care must be taken to design the application (or protocol) so that an attacker can never have chosen plaintext encrypted.
Man-in-the-middle attack	This attack is relevant for cryptographic communication and key exchange protocols. The idea is that when two parties, A and B, are exchanging keys for secure communication (e.g., using Diffie-Hellman), an adversary positions himself between A and B on the communication line. The adversary then intercepts the signals that A and B send to each other, and performs a key exchange with A and B separately. A and B will end up using a different key, each of which is known to the adversary. The adversary can then decrypt any communication from A with the key he shares with A, and then resends the communication to B by encrypting it again with the key he shares with B. Both A and B will think that they are communicating securely, but in fact the adversary is hearing everything. The usual way to prevent the man-in-the-middle attack is to use a public key cryptosystem capable of providing digital signatures. For set up, the parties must know each other's public keys in advance. After the shared secret has been generated, the

	parties send digital signatures of it to each other. The man-in-the-middle can attempt to forge these signatures, but fails because he cannot fake the signatures.
Correlation	Correlation between the secret key and the output of the cryptosystem is the main source of information to the cryptanalyst. In the easiest case, the information about the secret key is directly leaked by the cryptosystem. More complicated cases require studying the correlation (basically, any relation that would not be expected on the basis of chance alone) between the observed (or measured) information about the cryptosystem and the guessed key information. For example, in linear (resp. differential) attacks against block ciphers the cryptanalyst studies the known (resp. chosen) plaintext and the observed ciphertext. Guessing some of the key bits of the cryptosystem the analyst determines by correlation between the plaintext and the ciphertext whether she guessed correctly. This can be repeated, and has many variations. The differential cryptanalysis introduced by Eli Biham and Adi Shamir in late 1980's was the first attack that fully utilised this idea against block ciphers (especially against DES). Later Mitsuru Matsui came up with linear cryptanalysis which was even more effective against DES. More recently, new attacks using similar ideas have been developed.
Attack against or using the underlying hardware	In the last few years as more and smaller mobile crypto devices have come into widespread use, a new category of attacks has become relevant which aim directly at the hardware implementation of the cryptosystem. The attacks use the data from very fine measurements of the crypto device doing, say, encryption and compute key information from these measurements. The basic ideas are then closely related to those in other correlation attacks. For instance, the attacker guesses some key bits and attempts to verify the correctness of the guess by studying correlation against her measurements. Several attacks have been proposed such as using careful timings of the device, fine measurements of the power consumption, and radiation patterns. These measurements can be used to obtain the secret key or other kinds information stored on the

	device. This attack is generally independent of the used cryptographical algorithms and can be applied to any device that is not explicitly protected against it.
Faults in cryptosystems	Faults in cryptosystems can lead to cryptanalysis and even the discovery of the secret key. The interest in cryptographical devices leads to the discovery that some algorithms behaved very badly with the introduction of small faults in the internal computation. For example, the usual implementation of RSA private key operations is very susceptible to fault attacks. It has been shown that by causing one bit of error at a suitable point can reveal the factorisation of the modulus (i.e. it reveals the private key). Similar ideas have been applied to a wide range of algorithms and devices. It is thus necessary that cryptographical devices are designed to be highly resistant against faults (and against malicious introduction of faults by cryptanalysts).

SUMMARY

1. **Authentication** is a process used to verify transmitted data especially a message and it's considered as the most important of the security services because towards some extent all other security services depend upon it.
2. **Cryptography** is the science of secret writing and it has become one of the main tools for privacy authentication, trust, access control, electronic payments, corporate security, and countless other fields.
3. **Cryptanalysis** is the art of deciphering encrypted communications without knowing the proper keys.

GLOSSARY

Cryptography	The art of secret writing that enables an individual to hide the contents of a message or file from all but intended recipient.
Cryptanalysis	The process of attempting to break a cryptographic system.
Cryptosystem	A system for encryption and decryption.
Decryption	It is the reverse process: transforming an encrypted message back to its normal form.
Encipher	The process of translating letters or symbols individually.
Encode	The process of translating entire words or phrases to other words or phrases.
Encryption	The art of obscuring data by making it cryptic (as in scrambling data). It covers both encoding and enciphering.

DISCUSSION QUESTION

1. Let say you can choose character from A-Z and the number of character requires are 5. What is the population of password?
2. Let say you can choose character from A-Z, a-z and 0-9, and the number of character requires are 7. What is the population of password?
3. Let say you are given the number of possibilities for each character is 36, the number of characters is 6, 7 and 8 and the population of password is 2.9 x 1012. You are given 3 months to change the old password to the new password and your computer can processed at 400 MIPS (Million Instruction per Second) and

also no limited attempt. Determine whether there is any possibility to penetrate to the system.

4. You are given the population of password is 2.9 x 1012, the probability of success is 1 and the number guesses per unit time is 600 MIPS. How much time do you take to guess the right password?
5. Given that the password selected is 4 characters from 26 characters. Assume an intruder can try the password at the rate of 1 second. Let say there are no response back to the intruder until any attempt is done. Determine how much time will be needed to get the right password.
6. Encipher the message using Caesar Cipher as in the example: HARTFORD PUBLIC
7. Decipher this secret message using Caesar Cipher as in the example: ZIXPPFZXI JXDKBG
8. Encipher this message using columnar transposition as in the example: ASIA E UNIVERSITY.

REFERENCES

Charles P. Pfleeger (2003). *Security in Computing*. New Jersey, United States: Prentice Hall.

Ciphers and Codes: http://www.optonline.com/comptons/ceo/01004_A.html

Crazy Linux: http://www.crazylinux.net/mirrors/www.ssh.fi/intro.htm

Dieter Gollmann (2000). *Computer Security*. USA: John Wiley & Sons.

Donald E. Eastlake & Kitty Niles (2002). *Digital Cryptography: A Subtle Art, 1st Ed*. Addison Wesley Professional.

Glossary on Technical Cryptography: http://www.io.com/~ritter/GLOSSARY.htm

Matt Bishop (2002). *Computer Security: Art and Science*. Boston: Pearson.

Paul Campbell, Ben Calvert & Steven Boswell (2003). *Security + Guide to Network Security Fundamentals*. Canada: Thomson Learning.

William Stallings (2003). *Cryptography and Network Security: Principles and Practices, 3rd Ed*. USA: Pearson, USA.