Table 1.2. Security Services (X.800)

AUTHENTICATION
The confidentiality of selected fields within the user data on a connection or in a single data block.
Traffic Flow Confidentiality
The protection of the information that might be derived from observation of traffic flows.
Connection Integrity with Recovery
Provides for the integrity of all user data on a connection and detects any modification, insertion, deletion, or replay of any data within an entire data sequence, with recovery attempted.
Connection Integrity without Recovery
As above, but provides only detection without recovery.
Selective-Field Connection Integrity
Provides for the integrity of selected fields within the user data of a data block transferred over a connection and takes the form of determination of whether the selected fields have been modified, inserted, deleted, or replayed.
Connectionless Integrity
Provides for the integrity of a single connectionless data block and may take the form of detection of data modification. Additionally, a limited form of replay detection may be provided.
Selective-Field Connectionless Integrity
Provides for the integrity of selected fields within a single connectionless data block; takes the form of determination of whether the selected fields have been modified.
NONREPUDIATION
Provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication.
Nonrepudiation, Origin
Proof that the message was sent by the specified party.
Nonrepudiation, Destination

(iii)Polyalphabetic ciphers

Another way to improve on the simple monoalphabetic technique is to use different monoalphabetic substitutions as one proceeds through the plaintext message. The general name for this approach is polyalphabetic cipher. All the techniques have the following features in common.

- A set of related monoalphabetic substitution rules are used
- A key determines which particular rule is chosen for a given transformation.

(iv)Vigenere cipher

In this scheme, the set of related monoalphabetic substitution rules consisting of 26 caesar ciphers with shifts of 0 through 25. Each cipher is denoted by a key letter. e.g., Caesar cipher with a shift of 3 is denoted by the key value 'd' (since a=0, b=1, c=2 and so on). To aid in understanding the scheme, a matrix known as vigenere tableau is constructed.

	PLAIN TEXT															
KEY LETTERS		a	b	c	d	e	f	g	h	i	j	k	…	x	y	z
	a	A	B	C	D	E	F	G	H	I	J	K	…	X	Y	Z
	b	B	C	D	E	F	G	H	I	J	K	L	…	Y	Z	A
	c	C	D	E	F	G	H	I	J	K	L	M	…	Z	A	B
	d	D	E	F	G	H	I	J	K	L	M	N	…	A	B	C
	e	E	F	G	H	I	J	K	L	M	N	O	…	B	C	D
	f	F	G	H	I	J	K	L	M	N	O	P	…	C	D	E
	g	G	H	I	J	K	L	M	N	O	P	Q	…	D	E	F
	:	:	:	:	:	:	:	:	:	:	:	:	…	:	:	:
	:	:	:	:	:	:	:	:	:	:	:	:		:	:	:
	x	X	Y	Z	A	B	C	D	E	F	G	H	…			W
	y	Y	Z	A	B	C	D	E	F	G	H	I	…			X
	z	Z	A	B	C	D	E	F	G	H	I	J	…			Y

Each of the 26 ciphers is laid out horizontally, with the key letter for each cipher to its left. A normal alphabet for the plaintext runs across the top. The process of

encryption is simple: Given a key letter X and a plaintext letter y, the cipher text is at the intersection of the row labeled x and the column labeled y; in this case, the ciphertext is V.

To encrypt a message, a key is needed that is as long as the message. Usually, the key is a repeating keyword.

e.g., key = d e c e p t i v e d e c e p t i v e d e c e p t i v e

PT = w e a r e d i s c o v e r e d s a v e y o u r s e l f

CT = ZICVTWQNGRZGVTWAVZHCQYGLMGJ

Decryption is equally simple. The key letter again identifies the row. The position of the cipher text letter in that row determines the column, and the plaintext letter is at the top of that column.

Strength of Vigenere cipher

- There are multiple ciphertext letters for each plaintext letter.
- Letter frequency inforamiton is obscured.

One Time Pad Cipher

It is an unbreakable cryptosystem. It represents the message as a sequence of 0s and 1s. this can be accomplished by writing all numbers in binary, for example, or by using ASCII. The key is a random sequence of 0's and 1's of same length as the message. Once a key is used, it is discarded and never used again. The system can be expressed as follows:

$$C_i = P_i \oplus K_i$$

C_i - i^{th} binary digit of cipher text

P_i - i^{th} binary digit of plaintext

K_i - i^{th} binary digit of key

- $\oplus$ – exclusive OR opearaiton

Thus the cipher text is generated by performing the bitwise XOR of the plaintext and the key. Decryption uses the same key. Because of the properties of XOR, decryption simply involves the same bitwise operation:

$$P_i = C_i \oplus K_i$$

BLOCK CIPHER PRINCIPLES

Virtually, all symmetric block encryption algorithms in current use are based on a structure referred to as Fiestel block cipher. For that reason, it is important to examine the design principles of the Fiestel cipher. We begin with a **comparison of stream cipher with block cipher.**

- **A stream cipher** is one that encrypts a digital data stream one bit or one byte at a time. E.g, vigenere cipher. **A block cipher** is one in which a block of plaintext is treated as a whole and used to produce a cipher text block of equal length. Typically a block size of 64 or 128 bits is used.

Block cipher principles

- most symmetric block ciphers are based on a **Feistel Cipher Structure**
- needed since must be able to **decrypt** ciphertext to recover messages efficiently
- block ciphers look like an extremely large substitution
- would need table of 264 entries for a 64 bit block
- instead create from smaller building blocks
- using idea of a product cipher

in 1949 Claude Shannon introduced idea of substitution-permutation (S-P) networks called modern substitution-transposition product cipher

these form the basis of modern block ciphers

- S-P networks are based on the two primitive cryptographic operations we have seen before:
  - *substitution* (S-box)
  - *permutation* (P-box)
- provide *confusion* and *diffusion* of message
- **diffusion** – dissipates statistical structure of plaintext over bulk of ciphertext
- **confusion** – makes relationship between ciphertext and key as complex as possible

Feistel cipher structure

- If A wishes to send a confidential message to B, A encrypts the message using B's public key.
- When B receives the message, it decrypts using its private key. No other recipient can decrypt the message because only B knows B's private key.

With this approach, all participants have access to public keys and private keys are generated locally by each participant and therefore, need not be distributed. As long as a system controls its private key, its incoming communication is secure.

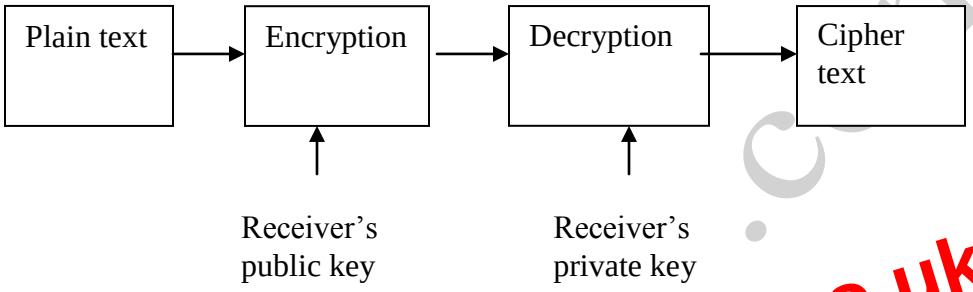


Fig: encryption

Let the plaintext be X=[X1, X2, X3, …,Xm] where m is the number of letters in some finite alphabets. Suppose A wishes to send a message to B. B generates a pair of keys: a public key KU_b and a private key KR_b. KR_b is known only to B, whereas KU_b is publicly available and therefore accessible by A.

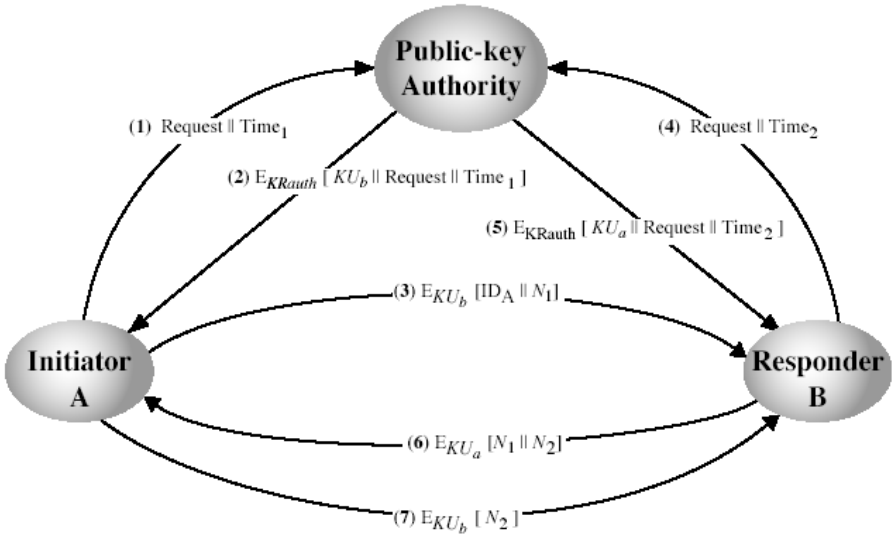
With the message X and encryption key KU_b as input, A forms the cipher text Y=[Y1, Y2, Y3, … Yn].

$$\text{i.e., } Y = E\ KU_b(X)$$

The receiver can decrypt it using the private key KR_b.

$$\text{i.e., } X = D\ KR_b()$$

The other approach (using sender's private key for encryption and sender's public key for decryption) will provide authentication which is illustrated in the following diagram.



Public-Key Certificates

- Certificates allow key exchange without real-time access to public-key authority
- A certificate binds **identity** to **public key**
  - Usually with other info such as period of validity, rights of use etc
- With all contents **signed** by a trusted Public-Key or Certificate Authority (CA)
- Can be verified by anyone who knows the public-key authorities public-key

E-mail facilities often are restricted to a maximum length. E.g., many of the facilities accessible through the internet impose a maximum length of 50,000 octets. Any message longer than that must be broken up into smaller segments, each of which is mailed separately.

To accommodate this restriction, PGP automatically subdivides a message that is too large into segments that are small enough to send via e-mail. The segmentation is done after all the other processing, including the radix-64 conversion. At the receiving end, PGP must strip off all e-mail headers and reassemble the entire original block before performing the other steps.

Cryptographic keys and key rings

Three separate requirements can be identified with respect to these keys:

- A means of generating unpredictable session keys is needed.
- It must allow a user to have multiple public key/private key pairs.
- Each PGP entity must maintain a file of its own public/private key pairs as well as a file of public keys of correspondents.

We now examine each of the requirements in turn.

1. Session key generation

Each session key is associated with a single message and is used only for the purpose of encryption and decryption of that message. Random 128-bit numbers are generated using CAST-128 itself. The input to the random number generator consists of a 128-bit key and two 64-bit blocks that are treated as plaintext to be encrypted. Using cipher feedback mode, the CAST-128 produces two 64-bit cipher text blocks, which are concatenated to form the 128-bit session key. The plaintext input to CAST-128 is itself derived from a stream of 128-bit randomized numbers. These numbers are based on the keystroke input from the user.

2. Key identifiers

If multiple public/private key pair are used, then how does the recipient know which of the public keys was used to encrypt the session key? One simple

Private Key Ring

Timestamp	Key ID*	Public Key	Encrypted Private Key	User ID*
• • •	• • •	• • •	• • •	• • •
T_i	PU_i mod 2^{64}	PU_i	E(H(P_i), PR_i)	User i
• • •	• • •	• • •	• • •	• • •

Public Key Ring

Timestamp	Key ID*	Public Key	Owner Trust	User ID*	Key Legitimacy	Signature(s)	Signature Trust(s)
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
T_i	PU_i mod 2^{64}	PU_i	trust_flag$_i$	User i	trust_flag$_i$		
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •

* = field used to index table

Figure 15.4 General Structure of Private and Public Key Rings

Timestamp – the date/time when this entry was made.

Key ID – the least significant bits of the public key.

Public key – public key portion of the pair.

Private key – private key portion of the pair.

User ID – the owner of the key.

Key legitimacy field – indicates the extent to which PGP will trust that this is a valid public key for this user.

Signature trust field – indicates the degree to which this PGP user trusts the signer to certify public key.

Owner trust field – indicates the degree to which this public key is trusted to sign other public key certificates.

PGP message generation

First consider message transmission and assume that the message is to be both signed and encrypted. The sending PGP entity performs the following steps:

PEM(PRIVACY ENHANCED MAIL)

Introduction

On the Internet, the notions of privacy and security are practically non-existent. Although email is one of the most popular uses of the Internet, security experts have estimated that only about one in every 100 messages is secured against interception and alteration.

Many people may think that sending an email in plain text is privacy-protected and enhancement of privacy is not necessary. This is simply not the fact. Whether you realize it or not, those messages you've been sending to business partners or friends over the Internet have been sent in the clear; information you thought was enclosed in a sealed envelope was instead sent just like a postcard.

When an email message is sent between two distant sites, it will generally transit dozens of machines on the way. Any of these machines can read the message and/or record it for future work.

Email Security

Let's look at some of the assumptions many people have about the security and integrity of email [1].

Authenticity

Many people assume that the name given as the sender of an email message identifies who actually sent it. In fact, this depends on the honesty of the sender and the flexibility of their mail package. For example, the Netscape Navigator mail function allows people to enter their own description of who they are, and what their email address is. While this will not allow them to receive mail that is not properly addressed to them, they can still send mail.

Integrity

When you send a message via e-mail, there is no guarantee that it will be received, or that what is received is exactly what you sent. You have no way of knowing that your message was not read or forwarded by third parties. This is due to the passing of messages from machine to machine, between your email server and that of the intended recipient.

At any point along the way, the mail server could lose the message, or the staff supporting the server could read and/or alter it. This is obvious if you consider that a mail message is only a file that gets passed from person to person along a delivery chain. Any person in the chain can drop the whole file in the garbage, or copy, add, delete, or replace documents in it. The next person in the chain doesn't know it's coming, what's in it, or how big it should be. These people don't work for the same company, and quite possibly aren't even on the same continent.

If you mis-spell the recipient's address, the mail server at their end may send the note back to you as undeliverable. However, it may also send it to somebody else, who happens to have the address you typed, or it may send it to the "Postmaster", who administers the system. Normally the postmaster will re-send it to the appropriate

person, but this is a manual process, which may take some time, or it may not be done at all.

To add to the confusion, incoming and outgoing mail is stored in plain text files on a hard disk in your mail server. These files can be altered by authorized administrators or by anybody capable of assuming authority. While University employees do not do this on a whim, the capability exists.

Reliability

As a sender, you have no way of knowing when a message was delivered. It could have been delayed due to system problems at an intermediate link in the delivery chain. Also, there is no standard way of requesting a receipt when the message is read. If you request a return receipt, and the receiver's mail system does not recognize that function, it will not send you an email note confirming delivery.

Because of the wide-spread nature of these problems, a number of competing solutions are being developed that address the authentication and integrity issues. The general consensus is to use some form of public-key cryptography, so that messages can be decrypted only by the intended recipient, are unalterable, and can be verified as coming from the sender.

Pretty Good Privacy, PGP, and Privacy-Enhanced Mail, PEM, are both "systems" that provide secrecy and non-repudiation of data that is sent over the Internet, mostly by email (figure 1).

Figure 1: PGP, PEM are external packages for message encryption, signing, etc.