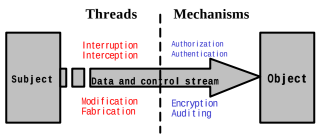
Chapter 10: Security



- Objects: passive entities whose security attributes must be protected
- Subjects: active entities that access objects
- Threads: potential dangers which harm security
- Security Policy: a precise specification to describe appropriate levels of security
- Security Mechanism: an implementation of a given security policy

Security Strategy

- Security Strategy consists of two steps: Security Policy and Security Mechanism.
- A Security Policy describes precisely which actions the entities in a system are allowed to take and which ones are prohibited.
- A Security Mechanism is the implementation of a given security policy such that a policy can be enforced.

Types of Threats

- Interception: an unauthorized subject has gained access to an object, such as stealing data, overhearing others communication, etc.
- Interruption: services or data become unavailable, unusable, destroyed, and so on, such as lost of file, denial of service, etc.
- Modification: unauthorized changing of data or tempering with services, such as alteration of data, modification of messages, etc.
- Fabrication: additional data or activities are generated that would normally no exist, such as adding a password to a system, replaying previously send messages, etc.

Methods of Attack

- Eavesdropping: obtaining copies of messages without authority
- Masquerading: sending/receiving messages using other's identifier
- Tempering: stealing messages and altering their contents
- Replaying: storing messages and sending them at later date
- Infiltrating: accessing system in order to run programs that implement the attack (virus, worm, Trojan horse)
- Unknown yet: new attacking methods may appear later

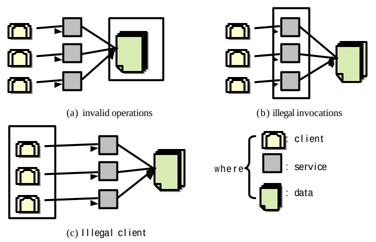
Indirect Infiltration

Trojan Horse:	A piece of code that misuses its environment. The program seems innocent enough, however when executed, unexpected behavior occurs.
Worms:	Use spawning mechanism; standalone programs. Such facilities may exist accidentally as well as intentionally.
Viruses:	Fragment of code embedded in a legitimate program. Mainly effects personal PC systems. These are often downloaded via e-mail or as active components in web pages.

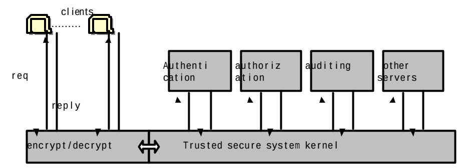
Security Mechanisms

- Encryption: transforming data into something an attacker cannot understand, i.e., providing a means to implement confidentiality, as well as allowing user to check whether data have been modified.
- Authentication: verifying the claimed identity of a subject, such as user name, password, etc.
- Authorization: checking whether the subject has the right to perform the action requested.
- Auditing: tracing which subjects accessed what, when, and which way. In general, auditing does not provide protection, but can be a tool for analysis of problems.

Focus of Control

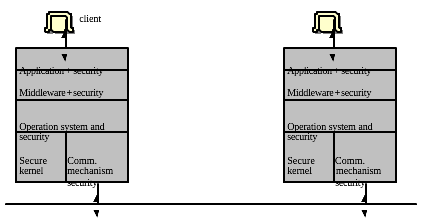


Dedicated Security Mechanism

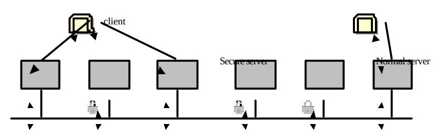


- Special servers dedicated to different security issues

Layered Security Mechanism



RISSC Security Mechanism



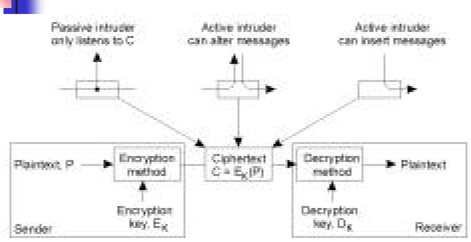
RISSC (Reduced Interface for Secure System Components)

Any security-critical server is placed on a separate machine isolated from end-user systems using low-level secure network interface. Clients run on different machines and can access the secured server only through these network interface.

Cryptography

- In ancient Greece, the Spartan generals used a form of cryptography so that the generals could exchange secret messages: the messages were written on narrow ribbons of parchment that were wound spirally around a cylindrical staff called a *scytale*. After the ribbon was unwound, the writing on it could only be read by a person who had a matching cylinder of exactly the same size. This primitive system did a reasonably good job of protecting messages from interception and from the prying eyes of the message courier as well.

Cryptography Basics



- Intruders and eavesdroppers in communication

Cryptography System

DEFINITIONS:

Encryption:

C = E(P, Ke)

- **E = Encrypting Algorithm**
- **P = Plain text**
- **Ke = Encryption key**
- **C = Cipher text**

Decryption:

P = D(C, Kd)

- **D = Decrypting Algorithm**
- **Kd = Decryption key**

Symmetric cryptosystem:

Ke = Kd = K

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

Asymmetric cryptosystem:

Ke ¹ Kd

P = D(E(P, Ke), Kd)

Example: **Symmetric cryptosystem**

Ceasar Cipher:

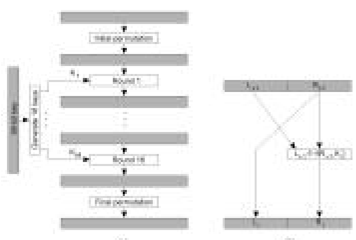
K = 1

Encryption: C[i] = P[i] + K

Decryption: P[i] = C[i] - K

P = Attack at dawn
C = Buubdl!bu!ebxo

DES: Data Encryption Standard



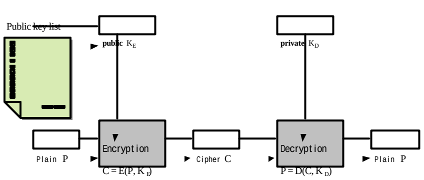
A symmetric cryptosystem: operate on 64-bit blocks:

- **The principle of DES**
- **Outline of one encryption round**

Discussion of DES

- The principle of DES is quite simple: initial permutation, 16 rounds of transformation, and final permutation.
- Even through the DES algorithm is well known, but the key or cipher is difficult to break using analytical methods.
- Using a brute-force attack by simply searching for a key is possible. However, for 56-bit key, there are 2^{56} possible key combinations, if we could search one key in 1 µs, then we need 2283 years to try all keys. (Distributed.net broke a DES-56 within 22 hours and 15 minutes, by using 100,000 PCs).
- Use 3DES (K1, K2, K3), or DES-128 for high security.

Public-Key Cryptosystems: RSA



An asymmetric cryptosystem (Rivest, Shamir, and Adleman, 1978):

- **Based on the fact that no methods are known to efficiently find the prime factors of larger numbers.**

Generating RSA Keys

(1) Pick up 3 large prime numbers, let S be the maximum, and X, Y be the rest;

(2) Let N = X * Y;

(3) Assume a unknown number Q, such that

(S * Q) mod (X – 1)(Y – 1) = 1

From (1), we know that S is an prime, and (X -1)(Y-1) is an even number, so there GCD is 1, that is GCD(S, (X-1)(Y-1) ≡ 1. We can use Euclid Algorithm to calculate: S*Q + (X-1)(Y-1) * R0 = 1

(4) Now, we got a triple (S, Q, N), and have

P^{SQ} mod N ≡ P, that is

(P^Q mod N $)^S$ mod N = P

encryption

decryption

Example: RSA Cryptosystem (1)

(1) Pick up 97, 47, 79. Let S = 97, X = 47, and Y = 79.

(2) N = X * Y = 3713;

(3) (X-1)(Y-1) = 3588, thus we should solve:

97 * Q + 3588 * R0 = 1, (calculation process omitted)

we have Q = 37, and R0 = -1 (we do not need R0)

(4) Now, we got a triple (S = 97, Q = 37, N = 3713)

char	blank	A	B	C	…	Y	Z
code	00	01	02	03	…	25	26

- From the above char/code table, we have:
  ATTACK AT DAWN ➔ 0120200103110001200401231 4

Example: RSA Cryptosystem (2)

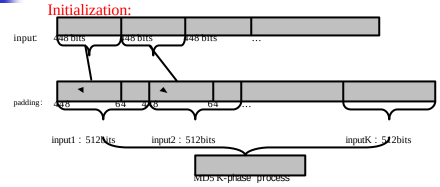
- ATTACK AT DAWN ➔ 01202001031100012004012314
- Message is first divided into fixed-length blocks, such as
  (0120)(2001)(0311) …
- To encrypt message, calculate each block by using Q = 37, N = 3713:
  $(0120)^{37} \bmod 3713 = 1404$
  $(2001)^{37} \bmod 3713 = 2932$
  $(0311)^{37} \bmod 3713 = 3536$
  …
- Integrate block coding together, we have:
  140429323536…
- Decryption at the receiver side uses S = 97, N = 3713:
  $(1404)^{97} \bmod 3713 = 0120$
  $(2932)^{97} \bmod 3713 = 2001$
  $(3536)^{97} \bmod 3713 = 0311$
  …
  } 01202001031100012004012314
  ATTACK AT DAWN

Hashing Function Cryptosystem

- A hash function h = H(m) takes a message m of arbitrary length as input and produces a fixed-length bit string h as output.
- A hash function is a one-way function, i.e., it is computationally infeasible to find the input m that corresponds to a known output h.
- The weak collision resistance property, i.e., given m and h = H(m), it is computationally infeasible to find another m' (m' ≠ m), such that H(m) = H(m').
- The strong collision resistance property, i.e., when only given H, it is computationally infeasible to find two different m and m', such that H(m) = H(m').

MD5: Message-Digest algorithm 5

Initialization:



- MD5 is a hash function for computing a 128-bit, fixed-length message digest from an arbitrary length binary input.
- Initialization: dividing input into 448-bit blocks and then padding these blocks into 512-bit blocks.

MD5: K-phase hashing

- K is the number of padded blocks
- Each phase consists four rounds of computations by using four different functions.
- Typical application of MD5 is Digital Signature.

Authentication

- How to make the communication between clients and servers (or senders and receivers) secure? We need to authentication of communication parties.
- Authentication and message integrity are closely related, cannot go without each other.
- Commonly use authentication models:
  (1) based on a shared secret key
  (2) based on a key from KDC (Key Distribution Center)
  (3) based on public key

Protocol Terminologies

Symbol	Meaning
A	principal A on one machine
B	principal B on another machine
S	authentication server or key distribution center
K_{AS}	secret key shared only by A and S
K_{BS}	secret key shared only by B and S
K_{AB}	session key shared only by A and B after authentication
K_A^+	public key of A
K_A^-	private key of A
A → B : M	sender A transmits message M to receiver B
K(M)	a cipher of M encrypted by a key K
N_A	a nonce generated by A
N_B	a nonce generated by B
T_S	timestamp of machine (server) S

Needham-Schroeder Protocol

steps	transmit	message
(1)	A → S	A, B, N_A
(2)	S → A	$K_{AS}(N_A, B, K_{AB}, K_{BS}(A, K_{AB}))$
(3)	A → B	$K_{BS}(A, K_{AB})$
(4)	B → A	$K_{AB}(N_B)$
(5)	A → B	$K_{AB}(N_B - 1)$

- S is the key distribution server, and A, B are two principals for establishing an interactive connection.
- Nonce,such as N_A,is a random number and used-only-once.
- A drawback of this protocol is that if the session key between A and B is compromised, and the certificate to B containing it is recorded, an intruder can impersonate A by carrying out the last three steps of the protocol to trick B to use the compromised session key and to think it was communicating with A.

Denning-Sacco Protocol

steps	transmit	message
(1)	A → S	A, B
(2)	S → A	$K_{AS}(B, T_S, K_{AB}, K_{BS}(A, T_S, K_{AB}))$
(3)	A → B	$K_{BS}(A, T_S, K_{AB})$

- Message freshness is guaranteed by including a timestamp instead of using a nonce handshake.
- A and B can verify that their messages are not replays by checking that:

  $$C - T_S < \Delta t1 + \Delta t2$$

  where C is the local time, T_S is the timestamp of S, $\Delta t1$ is the interval representing the normal discrepancy between S'clock and the local clock, and $\Delta t2$ is the interval representing the expected network delay. As long as $\Delta t1 + \Delta t2$ is less than the interval between two contiguous authentication, this protocol can protect against replay attack.

Otway-Rees Protocol

steps	transmit	message
(1)	A → B	M, A, B, $K_{AS}(M, A, B, N_A)$
(2)	B → S	M, A, B, $K_{AS}(M, A, B, N_A)$, $K_{BS}(M, A, B, N_B)$
(3)	S → B	M, $K_{AS}(N_A, K_{AB})$, $K_{BS}(N_B, K_{AB})$
(4)	B → A	M, $K_{AS}(N_A, K_{AB})$

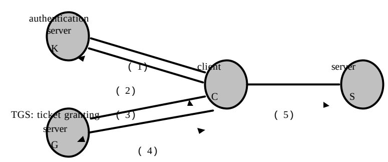
- This protocol attempts to provide timely authentication in a small number of messages without synchronized clocks.
- A and B issue their own nonces, N_A and N_B,and a common nonce M (a special message for challenge/response purpose) is issued by A and must be included in both encrypted messages.
- The most attractive property is that it can be implemented as two nested RPC's, such as A calls B, B calls S. But a major drawback is that B has no way to check that A's request is genuine and fresh.

Kerberos Protocol

steps	transmit	message
(1)	A → S	A, B
(2)	S → A	$K_{AS}(K_{AB}, ticket_{AB})$, where $ticket_{AB} = K_{BS}(B, A, IP_A, T_S, L, K_{AB})$
(3)	A → B	$authenticator_{AB}$, $ticket_{AB}$, $authenticator_{AB} = K_{AB}(A, IP_A, T_A)$
(4)	B → A	$K_{AB}(T_A + 1)$

- Kerberos is a part of project at MIT, one of the most promising implementation of the authentication service. It was designed for the client/server model.
- It places the authentication service on two kinds of servers: (1) Kerberos server authenticates the user at login time and issues a ticket for a TGS; (2) TGS: Ticket Granting Server issues tickets for individual servers to a client.

Kerberos v.5 Process



- A ticket and authenticator pair is called a credential. When making a service request, the client presents the request along with the credential which authenticates the client and its right to access the server.

Public Key Protocol

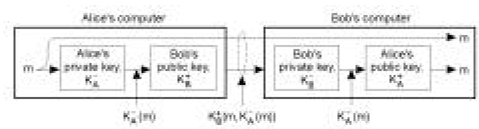
steps	transmit	message
(1)	A → B	$K_B^+(A, R_A)$
(2)	B → A	$K_A^+(R_A, R_B, K_{AB})$
(3)	A → B	$K_{AB}(R_B)$

- Suppose that we have a trusted public key distribution centre.
- It is important that B must trust that it got the right public key (as well as the most updated key) to A, and not the public key of someone impersonating A.
- How such guarantee can be given involving another protocol: Key management protocol.

Digital Signatures

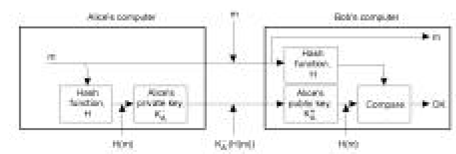
- A digit signature has the same authentication and legally binding functions as a handwritten signature.
- An electronic document or message M can be signed by an entity A by encrypting a copy of M in a key K_A and attaching it to a plain-text copy of M and A's identifier, such as <M, A, E(M, K_A)>.
- Once a signature is attached to a electronic document, it should be possible (1) any party that receives a copy of message to verify that the document was originally signed by the signatory, and (2) the signature can not be altered either in transmit or the receivers.

Public Key Digital Signatures (1)



- Digital signing a message using public-key cryptography.
- Problem: the validity of Alice's signature holds only as long as Alice's private key remains a secret and unchanged.
- Problem: the signature is too big.

Public Key Digital Signatures (2)



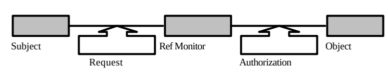
- Digitally signing a message using a message digest.
- Problem: hash function based signature is no longer safe, such as MD5.

Needham-Schroeder Digital Signatures

steps	transmit	message
(1)	A → S	$A, K_{AS}(M)$
(2)	S → A	$K_S(A, M, T)$
(3)	A → B	$A, M, K_S(A, M, T)$
(4)	B → S	$B, K_S(A, M, T)$
(5)	S → B	$K_{BS}(A, M, T)$

- S verifies A's signature (step 2). B trusts S.
- It would be difficult for A to claim that the signature was forged, for B has a copy that can be checked with S. On the other hand, A could not claim that B forged the signature, for B does not know the S's secret key.

Access Control



- A request from a client can be carried out only if the client has sufficient access rights for that requested operation.
- Verifying access rights is called access control, whereas authorization is about granting access rights.
- Many access control models:
  - Access Control Matrix
  - Access Control List (Capability List)
  - Firewalls

Access Control Matrix

Sub/Obj	file 1	file 2	file 3	file 4
user 1	owner	R/W	Exec	owner
user 2	--	R	owner	R/W
user 3	Copy/R	owner	--	--

(a) Resource ACM

Sub/Obj	process 1	process 2	process 3
process 1	--	send	Unblock send
process 2	receive	--	receive
process 3	Block receive	send	--

(b) Process communication ACM

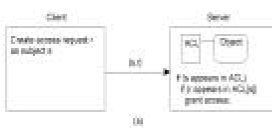
Sub/Obj	domain A	domain B	domain C
domain A	--	enter	--
domain B	--	--	enter
domain C	enter	--	--

(c) Domain communication ACM

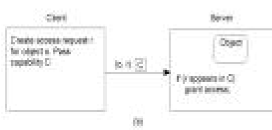
Access Control List

- ACM is simple and straightforward, but if a system supports thousands of users and millions of objects, the ACM will be a very sparse matrix.
- An ACL (Access Control List) is a column of ACM with empty entries removed, each object is assumed to have its own associated ACL.
- Another approach is to distribute the matrix row-wise by giving each subject a list of CL (Capability List).

Comparison between ACL and CL



ACL is associated with Object

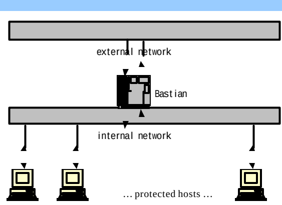


CL is associated with Subject

Firewalls

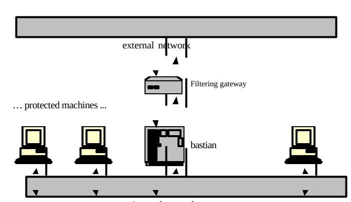
- A Firewall is a special kind reference monitor to control external access to any part of a distributed system.
- A Firewall disconnects any part of a distributed system from outside world, all outgoing and incoming packets must be routed through the firewall.
- A firewall itself should be heavily protected against any kind of security threads.
- Models of firewall:
  - Packet-filtering gateway
  - Proxy:
    - Application-level Proxy
    - Circuit-level Proxy

Firewalls: Bastian structure



- A Bastian is a special computer which provides secure services, including authentication and access control.
- Bastian can be a single machine or a dual-machine.

Firewalls: Bastian + Filtering gateway



- Gateway implements IP packet filtering functions.
- A Bastian provides secure services.