

**Rajendra Mane College of Engineering and Technology,  
AMBAV (Devrukh) - 415804**

University of Mumbai

Computer Engineering Sem - VII (Rev) Winter 2010 (10-WIN-COMP-VII-REV-SS)

**SYSTEM SECURITY**

Q. No. 1. a) **Explain different kinds of threats to information security.**

5

Ans: **Threat:** circumstances that have a potential to cause harm

Kinds of threats:

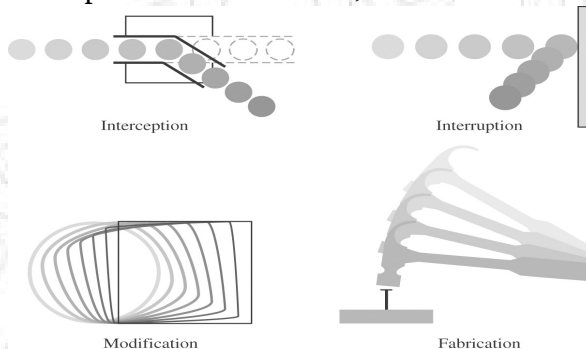
Interception: an unauthorized party (human or not) gains access to an asset

Interruption: an asset becomes lost, unavailable, or unusable

Modification: an unauthorized party changes the state of an asset

Fabrication: an unauthorized party counterfeits an asset

Examples for each for data, software and hardware.



Pfleeger/Pfleeger Fig. 01-02

b) **Does a PKI use symmetric or asymmetric for encryption?**

5

**Explain your answer.**

Ans: **Public Key Infrastructure (PKI)** is a set of hardware, software, people, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates.

In cryptography, a PKI is an arrangement that binds **public keys** with respective user identities by means of a **certificate authority (CA)**. The user identity must be unique within each CA domain. The binding is established through the registration and issuance process, which, depending on the level of assurance the binding has, may be carried out by software at a CA, or under human supervision. The PKI role that assures this binding is called the Registration Authority (RA). For each user, the user identity, the public key, their binding, validity conditions and other attributes are made unforgeable in public key certificates issued by the CA.

Sign message – to sign a message, S uses decryption algorithm D with S's private key. R authenticates signature using encryption algorithm E with S's public key

Encrypt certificate – after signing a (pre-)certificate, its issuer encrypts (E) the whole (pre-) certificate with his own private key. Anybody who receives certificate can verify it by using decryption algorithm. D with certificate issuers' public keys. But only certificate issuer can update a certificate she issued.

Thus PKI uses asymmetric encryption as it involves both public and private keys.

**Rajendra Mane College of Engineering and Technology,  
AMBAV (Devrukh) - 415804**

University of Mumbai

Computer Engineering Sem - VII (Rev) Winter 2010 (10-WIN-COMP-VII-REV-SS)

**SYSTEM SECURITY**

c) *What are the information security goals? Explain why the balance among different goals is needed.* 5

Ans: Confidentiality: Who is authorized?  
Integrity: Is the data "good?"  
Availability: Can access data whenever need it?

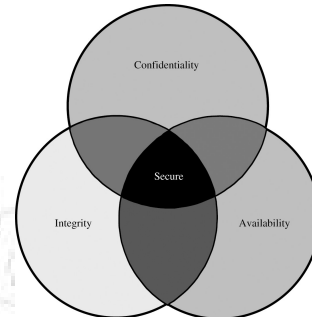


Fig: Relationship Between Confidentiality, Integrity, and Availability.

**Need to balance CIA**

Ex 1: Disconnect computer from Internet to increase confidentiality (availability suffers, integrity suffers due to lost updates)

Ex 2: Have extensive data checks by different people/systems to increase integrity (confidentiality suffers as more people see data, availability suffers due to locks on data under verification)

d) *What are the different types of malicious codes?* 5

Ans: **Kinds of Malicious Code**

1. **Bacterium** - A specialized form of virus which does not attach to a specific file. Usage obscure.
  2. **Logic bomb** - Malicious [program] logic that activates when specified conditions are met. Usually intended to cause denial of service or otherwise damage system resources.
  3. **Time bomb** - activates when specified time occurs
  4. **Rabbit** - A virus or worm that replicates itself without limit to exhaust resource
  5. **Trapdoor / backdoor** - A hidden computer flaw known to an intruder, or a hidden computer mechanism (usually software) installed by an intruder, who can activate the trap door to gain access to the computer without being blocked by security services or mechanisms.
- Combinations of the above kinds even more confusing  
E.g., virus can be a time bomb - spreads like virus, "explodes" when time occurs.

Q. No. 2. a) *Explain the Advanced Encryption Standard Algorithm in detail.* 10

Ans: **Advanced Encryption Standard :**

The AES Contest: 2001- ... winner - Rijndael (RINE-dahl)

Authors: Vincent Rijmen + Joan Daemen (Dutchmen)

Adopted by US gov't as Federal Info Processing Standard 197 (FIPS 197).

Overview of AES:

Similar to DES - cyclic type of approach. 128-bit blocks of P. # of iterations based on key length. 128-bit key => 9 "rounds" (called rounds, not cycles)

192-bit key => 11 rounds

256-bit key => 13 rounds

Each round uses 4 functions (in 3 "layers")

# Rajendra Mane College of Engineering and Technology, AMBAV (Devrukh) - 415804

University of Mumbai

Computer Engineering Sem - VII (Rev) Winter 2010 (10-WIN-COMP-VII-REV-SS)

## SYSTEM SECURITY

**ByteSub** (nonlinear layer) - byte level (confusion)

**ShiftRow** (linear mixing layer) - (transposition) – depends on key length (diff.)

**MixColumn** (nonlinear layer) - LSH and XOR (confusion +diffusion)

**AddRoundKey** (key addition layer) - XOR used (confusion)

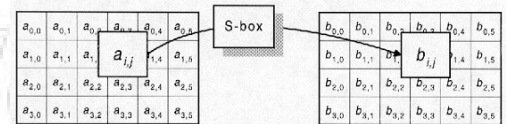
**ByteSub** (nonlinear layer) -

Assume 192 bit block, 4x6 bytes.

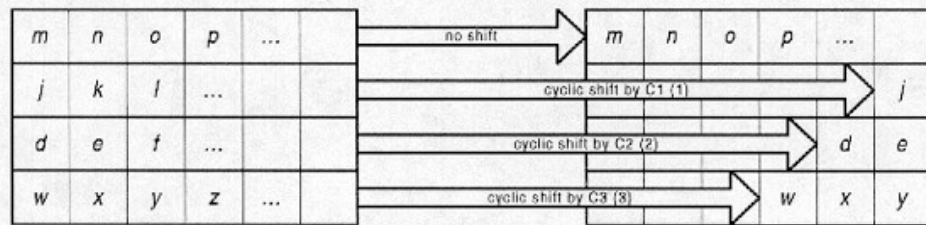
ByteSub is AES's "S-box".

Can be viewed as nonlinear (but invertible)

composition of two math operations

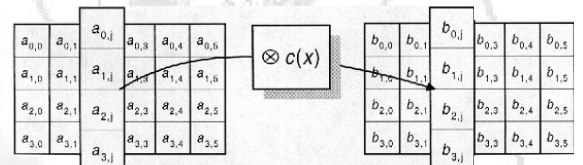


**ShiftRow** (linear mixing layer) - Cyclic shift rows.



**MixColumn** (nonlinear layer) -

Nonlinear, invertible operation applied to each column. Implemented as a (big) lookup table.



**AddRoundKey** (key addition layer) -

XOR subkey with block. RoundKey (subkey) determined by key schedule algorithm.

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} & a_{04} & a_{05} \\ a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{20} & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{30} & a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \end{bmatrix} \oplus \begin{bmatrix} k_{00} & k_{01} & k_{02} & k_{03} & k_{04} & k_{05} \\ k_{10} & k_{11} & k_{12} & k_{13} & k_{14} & k_{15} \\ k_{20} & k_{21} & k_{22} & k_{23} & k_{24} & k_{25} \\ k_{30} & k_{31} & k_{32} & k_{33} & k_{34} & k_{35} \end{bmatrix} = \begin{bmatrix} b_{00} & b_{01} & b_{02} & b_{03} & b_{04} & b_{05} \\ b_{10} & b_{11} & b_{12} & b_{13} & b_{14} & b_{15} \\ b_{20} & b_{21} & b_{22} & b_{23} & b_{24} & b_{25} \\ b_{30} & b_{31} & b_{32} & b_{33} & b_{34} & b_{35} \end{bmatrix}$$

### AES Decryption

To decrypt, process must be invertible. Inverse of MixAddRoundKey is easy, since  $\square$  is its own inverse. MixColumn is invertible (inverse is also implemented as a lookup table)

Inverse of ShiftRow is easy (cyclic shift the other direction)

ByteSub is invertible (inverse is also implemented as a lookup table)

### Strengths of AES

Not much experience so far (since 2001) but

Extensive cryptanalysis by US govt and independent experts

# Rajendra Mane College of Engineering and Technology, AMBAV (Devrukh) - 415804

University of Mumbai

Computer Engineering Sem - VII (Rev) Winter 2010 (10-WIN-COMP-VII-REV-SS)

## SYSTEM SECURITY

Dutch inventors have no ties to NSA or other US govt. bodies (less suspicion of trapdoor).

Solid math basis despite seemingly simple steps within rounds.

b) Write a note on Kerberos system that supports authentication in distributed system. 10

Ans:

Kerberos – system for authentication between intelligent processes in distributed systems. Developed at MIT (1988+)

Design goal:

Enable systems to withstand attacks in distributed systems

Basic idea of Kerberos: Central server provides tickets to requesting app. Ticket is authenticated, non-forgable, non-replayable token => Implemented as encrypted data structure naming user U and service for which U has access permission (also contains time value, control info)

User's Step 1: Establishing session w/ Kerberos. U's workstation sends U's identity to Kerberos server (KS). KS verifies that U is authorized.

KS sends 2 msgs:

1) Msg to U, which contains:  $E(ST-GS + TT-GS, \text{pwd})$

ST-GS — U's session key for session with T-GS

TT-GS — U's ticket for T-GS

Enables U to request service from T-GS

pwd — user's pwd (Note: used as encryption key by KS)

2) Msg to T-GS, which contains: ST-GS and U's identity (encrypted under key shared by KS and T-GS)

ST-GS — T-GS's session key for session with U (same as U's session key for session with T-GS)

If U's workstation can decrypt  $E(ST-GS + TT-GS, \text{pwd})$  using its pwd, then U's authentication succeeds

Note: KS stores users' pwds => no need to pass pwds over network between U's workstation and KS. Security advantage.

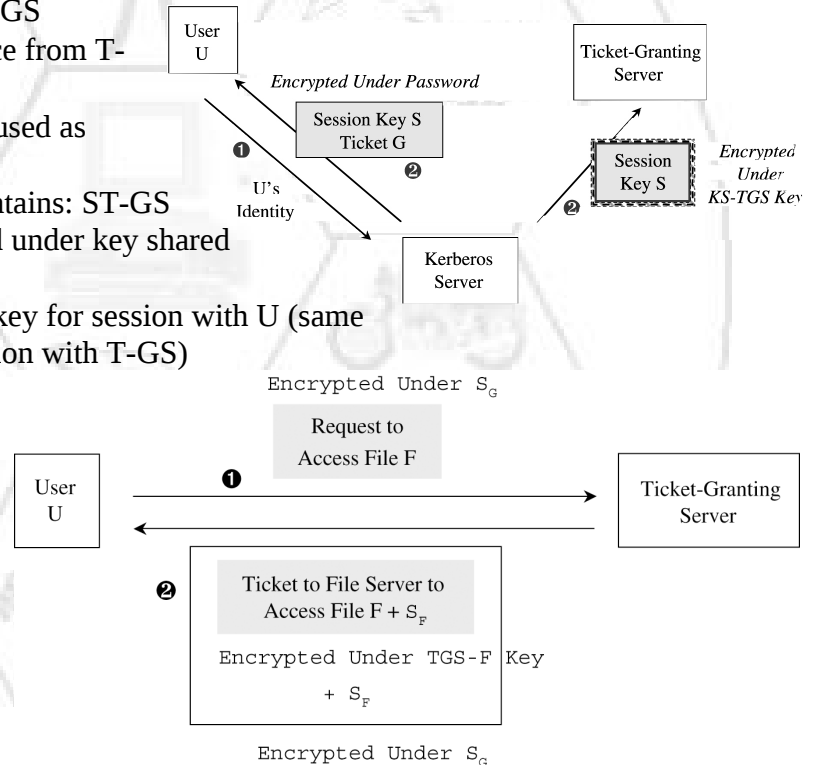


Figure: Obtaining a Ticket to Access a File.

User's Step 2: Access services of distributed system — e.g., access file F

Using U's ticket for T-GS (received fr. KS in Step 1), U sends to T-GS:

$E(\text{"request R for U's ticket for accessing F"} + ST-GS)$

Note: ST-GS (U's session key for session with T-GS obtained fr. KS in Step 1) is used to encrypt R. T-GS verifies U's access permission

# Rajendra Mane College of Engineering and Technology, AMBAV (Devrukh) - 415804

University of Mumbai

Computer Engineering Sem - VII (Rev) Winter 2010 (10-WIN-COMP-VII-REV-SS)

## SYSTEM SECURITY

### Strengths of Kerberos

1. No pwds communicated over network
2. Provides crypto protection against spoofing (e.g., masquerading, session hijacking, MITM)
3. Limits period of ticket validity (this disables some long-term attacks—e.g., brute force cryptanalysis)
4. Prevents replay attacks
5. Provides mutual authentication
6. Service user can be assured of any server's authenticity by requesting an authenticating response from S
7. Uses public key technology for key exchange

### Weaknesses of Kerberos system

1. Requires continuous availability of trusted ticket-granting server (T-GS)
2. Server S' authenticity requires trust between T-GS & S
3. Requires timely transactions (too quick ticket expiration will result in rejecting legitimate requests)
4. Subverted workstation can replay user pwds
5. Pwd guessing works (attacker can send initial —Step 1— authentication request to Kerberos server, receive response, try to decrypt response by guessing at pwd)
6. Kerberos does not scale well (due to system size might need > 1 KS and/or T-GS server; coordination and security problems if more than one KS and/or more than one T-GS is needed.)
7. Use of Kerberos requires compatibility of all apps in a given computing environment (to date few apps are compatible with Kerberos; modifying apps to make them compatible is not feasible)

Q. No. 3. a) **Explain control of access to general objects on operating system.**

10

Ans:

General objects in OS that need protection (examples)

Memory / File or data set on auxiliary storage device/ Pgm executing in memory / Directory of files / Hardware device/ Data structure / OS tables / Instructions, esp. privileged instructions/ Passwords and authentication mechanism / Protection mechanism

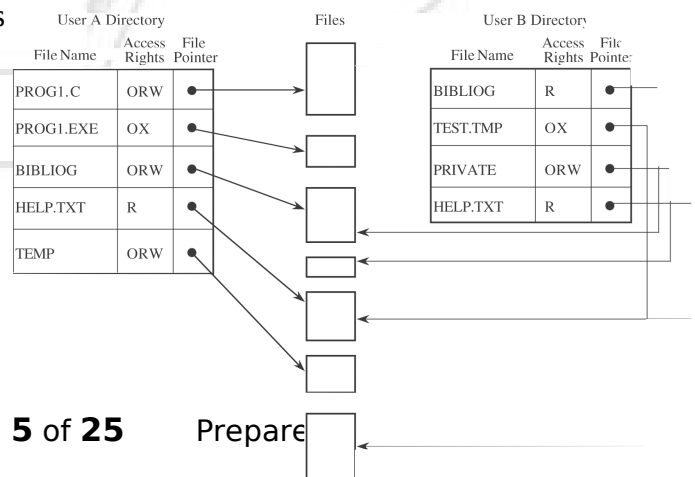
Subjects: User / Administrator / Programmer / Pgm/ Another object / Anything that seeks to use object

**Directory-like mechanism for access control.** Unique object

owner. Owner controls access rights: assigns/revokes them.

Access rights (ARs): Read, write, execute (possible others). Each user has access rights directory.

Example: (User A owns O1 and O3. User B owns O2, O4, O5).



# Rajendra Mane College of Engineering and Technology, AMBAV (Devrukh) - 415804

University of Mumbai

Computer Engineering Sem - VII (Rev) Winter 2010 (10-WIN-COMP-VII-REV-SS)

## SYSTEM SECURITY

Directory-like mechanism to control access to general objects. Analogous to file directory mechanism.

Advantage: Easy to implement. Just one list (directory) per user.

Difficulties: All user directories get too big for large # of shared objects — bec. each shared object in dir. of each user sharing it.

Maintenance difficulties: Deletion of shared objects - Requires deleting entry from each directory referencing it

Revocation of access : If owner A revokes access rights for X from every subject, OS must search dir's of all subjects to remove entries for X

### Access control lists

A list attached to an object.

Specifying ARs for each subject (who accesses this object)

For some subjects specified individually, for others — via being member of a group.

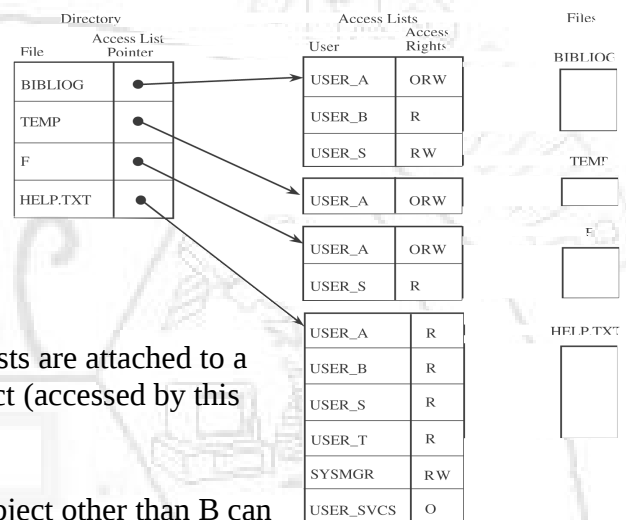


Fig: ACL →

Reverses directory approach where: lists are attached to a subject; specifying ARs for each object (accessed by this subject).

Example: Subjects: A, B, C, D, E

Use of wild card (\*) for 'any' (any subject other than B can R/W Object 4)

Significant advantages over directory approach: Can have default ARs for subjects w/o specific ARs

### Access control matrices

A sparse matrix (a table)

Rows — subjects / columns — objects

Cell (i, j) — subjects i's ARs for access to object j

	Object 1	Object 2	Object 3	Object 4	Object 5
Subject A	OWR	R	OWR	WR	-
Subject B	R	OWR	-	OWR	OWR
Subject C	R	R	-	WR	-
Subject D	R	-	-	WR	-
Subject E	-	R	-	WR	R

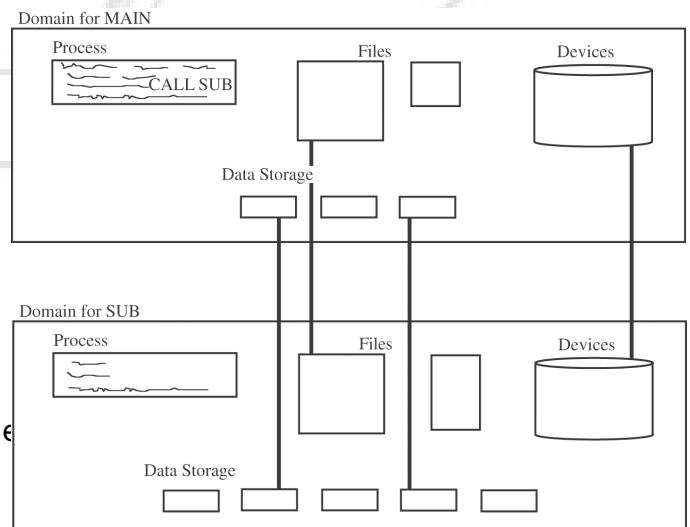
Fig: ACM

### Capabilities for access control

Capability — a kind of an unforgeable token/ticket/pass giving to subject certain ARs for an object.

Subjects access objects only via capabilities. To see (kind of access) a movie (object), a moviegoers (subject) must have a ticket (capability)

Capability to transfer ARs — allows subject to pass copies of its capabilities to other subjects. S1



**Rajendra Mane College of Engineering and Technology,  
AMBAV (Devrukh) - 415804**

University of Mumbai

Computer Engineering Sem - VII (Rev) Winter 2010 (10-WIN-COMP-VII-REV-SS)

**SYSTEM SECURITY**

can copy its capability to access O1 and transfer it to S2. If S1 omits 'transfer' rights for O1 in capability passed to S2, S2 can't transfer these rights to any other subject.

Fig: Process domain

Capability is limited by its domain (= local name space). Not all cap's passed from caller domain to subroutine domain. Subr. can have cap's that its calling pgm doesn't .

Capabilities help OS keep track of ARs during execution. Backed up by more detailed table (e.g. acc. ctrl matrix). Capabilities for objects accessed by current process are kept readily available (for speed).

Protecting capabilities: Capabilities in memory are accessible to OS only. E.g., stored in protected memory.

Capability are unforgeable - two basic ways:

1) Only OS holds and writes capabilities. OS issues to subjects only pointers to capabilities

2) Capability is encrypted. Key known only to OS's access control mechanism

Problem: Capability revocation can be complicated. When capability revoked by its issuing subject, OS must find & stop corresponding accesses.

Flexibility, Complexity, Overhead increase in the following order.

Directory - Access control lists - Access control matrices – Capabilities

b) *Explain non-malicious program errors with examples.*

10

Ans:

Non-malicious Program Errors

- Buffer overflows
- Incomplete mediation
- Time-of-check to time-of-use errors
- Combinations of non-malicious program flaws

**Buffer Overflows**

Buffer overflow flaw — often inadvertent (=>non-malicious) but with serious security consequences. Many languages require buffer size declaration

C language array overflow example.

Similar problem caused by pointers. No reasonable way to define limits for pointers.

Affects user's data - overwrites user's data

Affects users code - changes user's instruction

Affects OS data - overwrites OS data

Affects OS code - changes OS instruction

Implications of buffer overflow:

Attacker can insert malicious data values/instruction codes into "overflow space"

Suppose buffer overflow affects OS code area. Attacker code executed as if it were OS code. Attacker might need to experiment to see what happens when he inserts A into OS code area. Can raise attacker's privileges (to OS privilege level). Attacker can gain full control of OS.

Example: Supp. buffer overflow affects a call stack area

Stack: [data][data][...]

# Rajendra Mane College of Engineering and Technology, AMBAV (Devrukh) - 415804

University of Mumbai

Computer Engineering Sem - VII (Rev) Winter 2010 (10-WIN-COMP-VII-REV-SS)

## SYSTEM SECURITY

Pgm executes a subroutine => return address pushed onto stack (so subroutine knows where to return control to when finished)

Stack: [ret\_addr][data][data][...]

Subroutine allocates dynamic buffer char sample[10] => buffer (10 empty spaces) pushed onto stack

Stack: [.....][ret\_addr][data][data][...]

Subroutine executes: sample[i] = 'A' for i = 10

Stack: [.....][A][data][data][...]

Note: ret\_address overwritten by A (Assumed: size of ret\_address is 1 char)

Stack: [.....][A][data][data][...]

Subroutine finishes

Buffer for char sample[10] is deallocated

Stack: [A][data][data][...]

RET operation pops A from stack (considers it ret. addr.)

Stack: [data][data][...]

Pgm (which called the subroutine) jumps to A => shifts program control to where attacker wanted.

### Incomplete mediation

Sensitive data are in exposed, uncontrolled condition

Example: URL to be generated by client's browser to access server, e.g.:

http://www.things.com/order/final&custID=101&part=555A&qy=20&price=10&ship=boat&shipcost=5&total=205

Instead, user edits URL directly, changing price and total cost as follows:

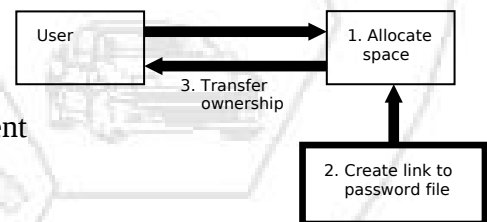
http://www.things.com/order/final&custID=101&part=555A&qy=20&price=1&ship=boat&shipcost=5&total=25

User uses forged URL to access server. The server takes 25 as the total cost.

Possible solution: Anticipate problems. Don't let client return a sensitive result (like total) that can be easily recomputed by server.

Use drop-down boxes / choice lists for data input. Prevent user from editing input directly

Check validity of data values received from client



### Time-of-check to Time-of-use Errors

Time-of-check to time-of-use flaw — often inadvertent (=>

nonmalicious) but with serious security consequences. A.k.a. synchronization flaw / serialization flaw

Example: Security processes should be atomic. Occur “all at once”. Race conditions can arise when security-critical process occurs in stages. Attacker makes change between stages. Often, between stage that gives authorization, but before stage that transfers ownership. Example: Unix mkdir

Prevention of TOCTTOU errors: make security-critical processes atomic.

Use digital signatures and certificates to “lock” data values after checking them. So nobody can modify them after check & before use.

**Rajendra Mane College of Engineering and Technology,  
AMBAV (Devrukh) - 415804**

University of Mumbai

Computer Engineering Sem - VII (Rev) Winter 2010 (10-WIN-COMP-VII-REV-SS)  
**SYSTEM SECURITY**

---

**Combinations of Non-malicious Pgm Flaws**

The above flaws can be exploited in multiple steps by a concerted attack  
Non-malicious flaws can be exploited to plant malicious flaws.

Q. No. 4. a) **If generator  $g = 2$  and  $n$  or  $p = 11$ , using Diffie – Hellman algorithm solve the following.**

i. **Show that 2 is a primitive root of 11.**

4

Ans:

$$2 \bmod 11 = 2$$

$$2^2 \bmod 11 = 4$$

$$2^3 \bmod 11 = 8$$

$$2^4 \bmod 11 = 5$$

$$2^5 \bmod 11 = 10$$

$$2^6 \bmod 11 = 9$$

$$2^7 \bmod 11 = 7$$

$$2^8 \bmod 11 = 3$$

$$2^9 \bmod 11 = 6$$

$$2^{10} \bmod 11 = 1.$$

Since  $2^i \bmod 11$  for  $0 < i < 11$  contains all numbers from 1 to 11-1, the size of this set is equal to  $\phi(11)$ , the order of  $q$ . Hence 2 is a primitive root of 11.

ii. **If A has a public key '9', what is A's private key?**

2

Ans:

From the above values,  $2^6 \bmod 11 = 9$ , therefore A's private key  $X_A = 6$ .

iii. **If B has a public key '3', what is B's private key?**

2

Ans:

From the above values,  $2^8 \bmod 11 = 3$ , therefore B's private key  $X_B = 8$ .

iv. **Calculate the shared secret key.**

2

Ans:

$$K = (Y_B)^{X_A} \bmod q = (3)^6 \bmod 11 = 3$$

OR

$$K = (Y_A)^{X_B} \bmod q = (9)^3 \bmod 11 = 3.$$

b) **Explain different denial of service attacks.**

10

Ans:

**Denial-of-service (DoS) attacks** = attacks on availability.

DoS attacks include:

- Physical DoS attacks
- Electronic DoS attacks

**Physical DoS attacks**

examples: Line cut deliberately, Noise injected on a line, Bringing down a node/device via h/w manipulation.

# Rajendra Mane College of Engineering and Technology, AMBAV (Devrukh) - 415804

University of Mumbai

Computer Engineering Sem - VII (Rev) Winter 2010 (10-WIN-COMP-VII-REV-SS)

## SYSTEM SECURITY

---

### Electronic DoS attacks

examples: (2a) Crashing nodes/devices via s/w manipulation

(2b) Saturating devices (due to malicious injection of excessive workload/ or traffic) Includes: (i) Connection flooding- Syn flood

(2c) Redirecting traffic: Includes: (i) Packet-dropping attacks - DNS attacks

**Connection flooding** = flooding a connection with useless packets so it has no capacity to handle (more) useful packets

ICMP (Internet Control Msg Protocol) - designed for Internet system diagnostic (3rd class of Internet protocols next to TCP/IP & UDP)

ICMP msgs can be used for attacks.

Some ICMP msgs:

- echo request – source S requests destination D to return data sent to it (shows that link from S to D is good)
- echo reply – response to echo request sent from D to S
- destination unreachable – msg to S indicating that packet can't be delivered to D
- source quench – S told to slow down sending msgs to D (indicates that D is becoming saturated)

Example attacks using ICMP msgs

#### (i1) Echo-charge attack

- charge protocol – generates stream of packets; used for testing network

- Echo-charge attack example 1:

(1) attacker uses charge on server X to send stream of echo request packets to Y

(2) Y sends echo reply packets back to X. This creates endless “busy loop” betw. X & Y

- Echo-charge attack example 2:

(1) attacker uses charge on X to send echo request packet stream to X

(2) X sends echo reply packets back to itself

#### (i2) Ping of death attack, incl. smurf attack

- Ping of death example :

(1) attacker uses ping after ping on X to flood Y with pings (ping uses ICMP echo req./reply)

(2) X responds to pings (to Y). This creates endless “busy loop” betw. X & Y

Note: In cases (i1-Ex.1) & (i2):

- if X is on 10 MB connection and path to victim Y is 100 MB, X can't flood Y

- if X is on 100 MB connection and path to victim Y is 10 MB, X can easily flood Y

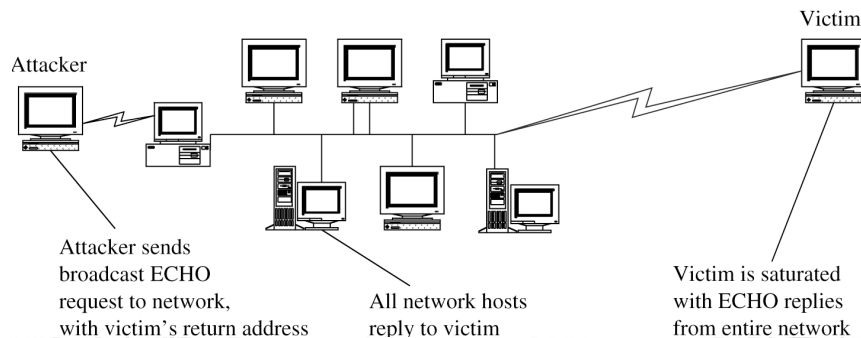
#### Smurf attack:

(1) attacker spoofs source address of ping packet sent fr. X – appears to be sent by Z

(2) att. broadcasts spoofed pkt to N hosts

**SYSTEM SECURITY**

(3) all N hosts echo to Z – flood it.



**Syn flood attack**

Attack is based on properties/implementation of a session in TCP protocol suite

Session = virtual connection between protocol peers

Session established with three-way handshake (S = source, D = destination) as follows:

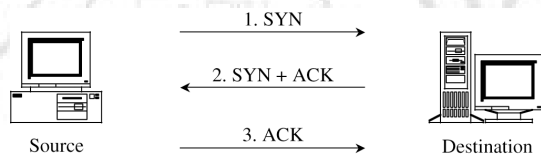


Figure: Three-Way Connection Handshake.

Now session between S and D is established.

D keeps SYN\_RECV queue which tracks connections being established for which it has received no ACK. Normally, entry is in SYN\_RECV for a short time. If no ACK received within time T (usu. k minutes), entry discarded (connection establ. times out).

Normally, size of SYN\_RECV (10-20) is sufficient to accommodate all connections under establishment.

Syn flood attack scenario: Attacker sends many SYN requests to D. Attacker never replies to D's SYN+ACK packets. D puts entry for each unanswered SYN+ACK packet into SYN\_RECV queue

With many unanswered SYN+ACK packets, SYN\_RECV queue fills up. When SYN\_RECV is full, no entries for legitimate unanswered SYN+ACK packets can be put into SYN\_RECV queue on D => nobody can establish legitim. connection with D.

**Redirecting traffic**

(i) Redirecting traffic by advertising a false best path

Routers find best path for passing packets from S to D. advertise their connections to their neighbors.

Example of traffic redirection attack:

Router R taken over by attacker. R advertises (falsely) to all neighbors that it has the best (e.g., shortest) path to hosts H1, H2, ..., Hn

Hosts around R forward to R all packets addressed to H1, H2, ..., Hn

R drops some or all these packets

**Rajendra Mane College of Engineering and Technology,  
AMBAV (Devrukh) - 415804**

University of Mumbai

Computer Engineering Sem - VII (Rev) Winter 2010 (10-WIN-COMP-VII-REV-SS)

**SYSTEM SECURITY**

drops some => packet-dropping attack

drops all => black hole attack

(ii) Redirecting traffic by DNS attacks

Domain name server (DNS): resolving domain name = converting domain names into IP addresses. DNS queries other DNSs (on other hosts) for info on unknown IP addresses. DNS caches query replies (addresses) for efficiency. Most common DNS implementation: BIND s/w (BIND = Berkeley Internet Name Domain) a.k.a. named i.e. name daemon). Numerous flaws in BIND including buffer overflow. Attacks on DNS (e.g., on BIND): Overtaking DNS / fabricating cached DNS entries using fabricated entry to redirect traffic.

Q. No. 5. a) *List, explain and compare different kinds of firewalls used in network security.* 10

Ans: All traffic (incoming / outgoing) must pass thru firewall and only authorized traffic allowed to pass.

**Basic kinds of firewalls:**

Hardware firewalls: More common. Implemented on router level. More expensive / more difficult to configure

Software firewalls: Used in single workstations. Less expensive / Easier to configure

**Types of firewalls**

i. Packet filters / packet filtering firewalls

(i-1) Simple packet filters / (simple) packet filtering gateways / screening routers

(i-2) Stateful packet filters / stateful inspection firewalls

ii. Application proxies / proxy firewalls / application-level gateways

(ii-1) Guards (a special case of app proxies)

iii. Personal firewalls

**Packet filters** — a.k.a. packet filtering firewalls

(i-1) Simple packet filters (“memoryless”)

(i-2) Stateful packet filters (with “memory”)

Basis for packet filtering

1) Packet IP addresses: Filtering based on both source/destination addresses

2) Port number determines TCP transport protocol type: Filtering based on port nr

Packet filtering firewalls do not “see” other packet fields. See only IP address' transport protocol type. E.g., can not allow only some Telnet commands OR exclude only some other Telnet commands.

# Rajendra Mane College of Engineering and Technology, AMBAV (Devrukh) - 415804

University of Mumbai

Computer Engineering Sem - VII (Rev) Winter 2010 (10-WIN-COMP-VII-REV-SS)

## SYSTEM SECURITY

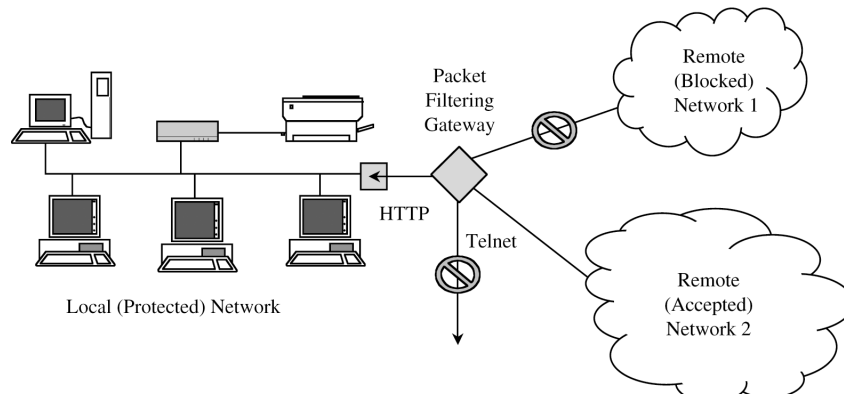


Figure: Packet Filter Blocking Addresses and Protocols.

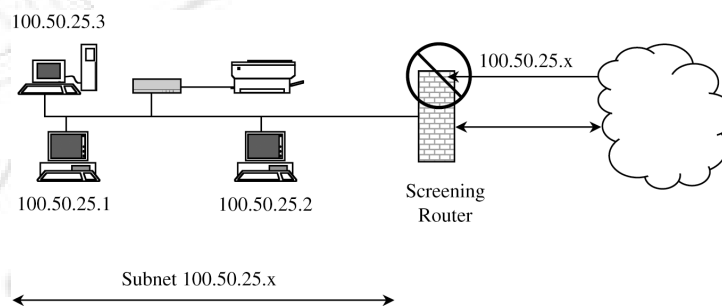


Figure: Filter Screening Outside Addresses.

### (i-1) Simple packet filters

Simple packet filters / (simple) packet filtering gateways / screening routers — simplest firewall type. are memoryless => can not perform attack detections that require remembering state (history/context) of  $\geq N$  last pkts. E.g., can not see that prev. & curr. pkt indicate attack. "Attack signature" (i.e., attack pattern) would be clearly visible if both pkts were put together.

Example: Certain attack script known to use Telnet (port 23) and then SNMP (port 161). The same source address in previous pkt, using port 23, and in current packet, using port 161, constitutes attack signature.

Cheating simple (memoryless) PF:

Attacker divides pkt (including attack signature) into many v. short pkts. Attack signature (pattern) would be visible in original long pkt. Even memoryless simple PF would detect it. Pattern of attack is completely invisible in any single short pkt => memoryless simple PF is unable to detect attack. Additionally, TCP pkts arrive in order different than their sending order => remembering just last packet would not be enough – must remember  $N$  last packets.

One very important task for simple packet filtering gateways: Validating inside IP addresses. Blocks any packet coming from outside network that claims to be sent by internal host.

# Rajendra Mane College of Engineering and Technology, AMBAV (Devrukh) - 415804

University of Mumbai

Computer Engineering Sem - VII (Rev) Winter 2010 (10-WIN-COMP-VII-REV-SS)

## SYSTEM SECURITY

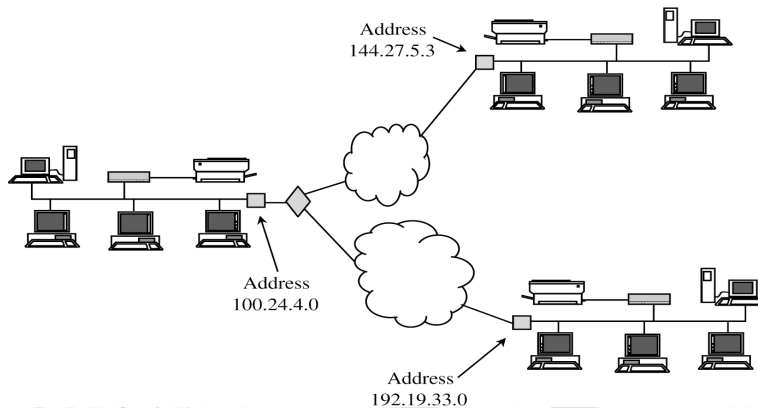


Figure: Three Connected LANs.

Disadvantage: high granularity.

Advantage: reduces complexity.

### (ii) Application proxies

Application proxies / proxy firewalls / application-level gateways / application proxy gateways.

Application proxies include — as a special case — Guards.

App proxy firewalls fix basic problem with packet filtering firewalls because they:

See all pkt data (not just IP addresses and port #s). (In addition, they are stateful => can analyze multiple pkts)  
=> can detect/derail more sophisticated attacks. Can filter out harmful commands in pkt stream.

For example, app proxies can prevent:

1. Use of back door open to pkts inbound to SMTP port (Port 25)
2. Flawed application run by user U (e.g., an e-mail agent) has all U's privileges => can cause damage.

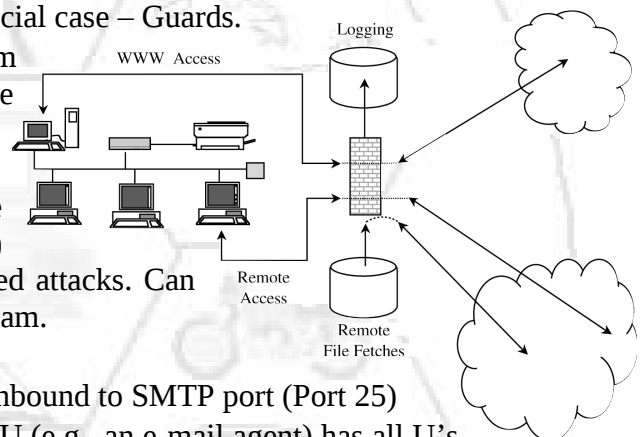
(ii-1) Guards = most sophisticated category of app proxies ("top model"). Limited only by what is computable (& by human creativity).

No sharp boundary between app proxies and guards. At some point of upgrading app proxy, it becomes a guard.

### (iii) Personal firewalls

Regular firewalls protect subnetworks. Personal firewalls protect single hosts. For small business / home office / home. Can be used to complement conventional firewall. Next line of defense. Customized to user(s) of particular host. Firewall capabilities at a lower price. Personal firewall is application program

Products include: Norton Personal Firewall (Symantec), McAfee Personal Firewall, Zone Alarm (Zone Labs).



b) List and explain the contents of a security plan for administrative security.

10

# Rajendra Mane College of Engineering and Technology, AMBAV (Devrukh) - 415804

University of Mumbai

Computer Engineering Sem - VII (Rev) Winter 2010 (10-WIN-COMP-VII-REV-SS)

## SYSTEM SECURITY

Ans: **Security plan for administrative security:** An official record of current security practices, plus a blueprint for orderly change to improve those practices. By following the plan, developers and users can measure the effect of proposed changes, leading eventually to further improvements.

A carefully written plan, supported by management, notifies employees that security is important to management (and therefore to everyone). Thus, the security plan has to have the appropriate content and produce the desired effects.

### Contents of a Security Plan

A security plan identifies and organizes the security activities for a computing system. The plan is both a description of the current situation and a plan for improvement. Every security plan must address seven issues.

- 1) Policy:- indicating the goals of a computer security effort and the willingness of the people involved to work to achieve those goals
- 2) Current state:- Describing the status of security at the time of the plan.
- 3) Requirements:- recommending ways to meet the security goals.
- 4) Recommended controls:- mapping controls to the vulnerabilities identified in the policy and requirements.
- 5) Accountability:- describing who is responsible for each security activity.
- 6) Timetable:- identifying when different security functions are to be done.
- 7) Continuing attention:- specifying a structure for periodically updating the security plan

### 1. Policy

A security plan must state the organization's policy on security. A security policy is a high-level statement of purpose and intent.

Initially, you might think that all policies would be the same: to prevent security breaches. But in fact the policy is one of the most difficult sections to write well

The policy statement must answer three essential questions:

- 1) Who should be allowed access?
- 2) To what system and organizational resources should access be allowed?
- 3) What types of access should each user be allowed for each resource?

The policy statement should specify the following: The organization's goals on security. For example, Should the system protect data from leakage to outsiders, protect against loss of data due to physical disaster, protect the data's integrity, or protect against loss of business when computing resources fail? What is the higher priority: serving customers or securing data?

Where the responsibility for security lies. For example, should the responsibility rest with a small computer security group, with each employee, or with relevant managers?

The organization's commitment to security. For example, who provides security support for staff, and where does security fit into the organization's structure?

### 2. Current Security Status

# **Rajendra Mane College of Engineering and Technology, AMBAV (Devrukh) - 415804**

University of Mumbai

Computer Engineering Sem - VII (Rev) Winter 2010 (10-WIN-COMP-VII-REV-SS)

## **SYSTEM SECURITY**

---

To be able to plan for security, an organization must understand the vulnerabilities to which it may be exposed. The organization can determine the vulnerabilities by performing a risk analysis.

Risk analysis:- a careful investigation of the system, its environment, and the things that might go wrong.

The risk analysis forms the basis for describing the current status of security. The status can be expressed as a listing of organizational assets, the security threats to the assets, and the controls in place to protect the assets.

The status portion of the plan also defines the limits of responsibility for security. It describes not only which assets are to be protected but also who is responsible for protecting them.

### **3. Requirements**

The heart of the security plan is its set of security requirements: functional or performance demands placed on a system to ensure a desired level of security. The requirements are usually derived from organizational needs. Sometimes these needs include the need to conform to specific security requirements imposed from outside, such as by a government agency or a commercial standard.

We must distinguish the requirements from constraints and controls.

Constraint is an aspect of the security policy that constrains, circumscribes, or directs the implementation of the requirements.

Control is an action, device, procedure, or technique that removes or reduces vulnerability.

The requirements should always leave the implementation details to the designers, whenever possible.

We should make sure that the requirements have these characteristics:

Correctness: Are the requirements understandable? Are they stated without error?

Consistency: Are there any conflicting or ambiguous requirements?

Completeness: Are all possible situations addressed by the requirements?

Realism: Is it possible to implement what the requirements mandate?

Need: Are the requirements unnecessarily restrictive?

Verifiability: Can tests be written to demonstrate conclusively and objectively that the requirements have been met? Can the system or its functionality be measured in some way that will assess the degree to which the requirements are met?

Traceability: Can each requirement be traced to the functions and data related to it so that changes in a requirement can lead to easy reevaluation?

### **4. Recommended Controls**

All control that we studied so far come under this topic.....

### **5. Responsibility for Implementation/Accountability**

The plan notes who is responsible for implementing controls when a new vulnerability is discovered or a new kind of asset is introduced.

# **Rajendra Mane College of Engineering and Technology, AMBAV (Devrukh) - 415804**

University of Mumbai

Computer Engineering Sem - VII (Rev) Winter 2010 (10-WIN-COMP-VII-REV-SS)

## **SYSTEM SECURITY**

---

People building, using, and maintaining the system play many roles. Each role can take some responsibility for one or more aspects of security.

Personal computer users may be responsible for the security of their own machines. Alternatively, the security plan may designate one person or group to be coordinator of personal computer security.

Project leaders may be responsible for the security of data and computations.

Managers may be responsible for seeing that the people they supervise implement security measures.

Database administrators may be responsible for the access to and integrity of data in their databases.

Information officers may be responsible for overseeing the creation and use of data; these officers may also be responsible for retention and proper disposal of data.

Personnel staff members may be responsible for security involving employees, for example, screening potential employees for trustworthiness and arranging security training programs.

### **6. Timetable**

The security plan includes a timetable that shows how and when the elements of the plan will be performed. These dates also give milestones so that management can track the progress of implementation.

The plan should specify the order in which the controls are to be implemented so that the most serious exposures are covered as soon as possible. A timetable also gives milestones by which to judge the progress of the security program.

### **7. Continuing Attention**

The inventory of objects and the list of controls should periodically updated, and risk analysis performed a new. The security plan should set times for these periodic reviews, based either on calendar time or on the nature of system changes.

### **Security Planning Team Members**

Who performs the security analysis, recommends a security program, and writes the security plan? a committee that represents all the interests involved. The size of the committee depends on the size and complexity of the computing organization and the degree of its commitment to security.

Security planning team should represent each of the following groups.

1. computer hardware group
2. system administrators
3. systems programmers
4. applications programmers
5. data entry personnel
6. physical security personnel
7. representative users

**Rajendra Mane College of Engineering and Technology,  
AMBAV (Devrukh) - 415804**

University of Mumbai

Computer Engineering Sem - VII (Rev) Winter 2010 (10-WIN-COMP-VII-REV-SS)

**SYSTEM SECURITY**

Q. No. 6. Write a detail note on (any two):

20

**a) E-mail security**

Ans: Although we now take the email for granted, it is important to realise that in its most basic form, at least - it is not necessarily a very secure or private means of communication. In fact, email has often been likened to the use of the postcard in conventional postal systems: it is open to being read or tampered with during transmission, and it might not even actually come from the person who apparently sent it.

There are two main strategies for making your email secure and private:

Use email encryption software to encode your messages, which are then decoded by the recipient after delivery. Even if a message is viewed in transit by someone else, they will not be able to decipher it. Email encryption software can also be used to digitally sign a message to guarantee it really did originate from the apparent sender. Two of the most widely used email encryption systems is called **PGP** (Pretty Good Privacy) and **S/MIME** (Secure/Multipurpose Internet Mail Extensions).

Use a secure connection system between your own machine and the server - this is rather like the use of a 'scrambler' on a conventional phone system. This protects all forms of information - for example passwords - passing between a workstation and a remote server, not just email messages. It prevents against anyone 'snooping' on your connection's network traffic, although offers no protection on information once it passes beyond the secure connection. One of the most widely used forms of secure connection system is known as SSL.

Most popular security systems, including those mentioned above, are based on the concept of Public Key Encryption. This involves the use of a linked pair of digital "keys":

A public key, freely publicized by its owner, which is used to encrypt information being sent to that person or system.

A private key, known only to its owner, used to decode incoming information encrypted with the corresponding public key.

**b) RSA algorithm (public key algorithm).**

Ans: Invented by Rivest, Shamir and Adleman (MIT).

Let  $p$  and  $q$  be two large prime numbers. Let  $N = pq$  be the modulus. Choose  $e$  relatively prime to  $(p-1)(q-1)$ . Find  $d$  s.t.  $ed = 1 \pmod{(p-1)(q-1)}$ .

Public key is  $(N, e)$ . Private key is  $d$ .

To encrypt message  $M$ , compute

$$C = M^e \pmod N$$

To decrypt  $C$ , compute

$$M = C^d \pmod N$$

If attacker can factor  $N$ , he can use  $e$  to easily find  $d$  since

# Rajendra Mane College of Engineering and Technology, AMBAV (Devrukh) - 415804

University of Mumbai

Computer Engineering Sem - VII (Rev) Winter 2010 (10-WIN-COMP-VII-REV-SS)

## SYSTEM SECURITY

$$ed = 1 \pmod{(p-1)(q-1)}$$

Factoring the modulus breaks RSA. It is not known whether factoring is the only way to break RSA.

Given  $C = M^e \pmod N$  we must show

$$M = C^d \pmod N = M^{ed} \pmod N$$

Use **Euler's Theorem**: If  $x$  is relatively prime to  $n$  then  $x^{\phi(n)} = 1 \pmod n$

$$\text{Facts: } ed = 1 \pmod{(p-1)(q-1)}$$

By definition of "mod",  $ed = k(p-1)(q-1) + 1$

$$\phi(N) = (p-1)(q-1)$$

$$\text{Then } ed - 1 = k(p-1)(q-1) = k\phi(N)$$

$$M^{ed} = M^{(ed-1)+1} = M \cdot M^{ed-1} = M \cdot M^{k\phi(N)} = M \cdot (M^{\phi(N)})^k \pmod N$$

$$= M \cdot 1^k \pmod N = M \pmod N$$

### c) Data Encryption Standard (symmetric key algorithm).

Ans:

Overview of DES

DES - a block cipher. A product cipher. 16 rounds (iterations) on the input bits (of P). substitutions (for confusion) and permutations (for diffusion). Each round with a round key. Generated from the user-supplied key.

Easy to implement in S/W or H/W

Input: 64 bits (a block)

$L_i/R_i$  - left/right half of the input block for iteration  $i$  (32 bits)

- subject to substitution  $S$  and permutation  $P$

$K$  - user-supplied key

$K_i$  - round key: 56 bits used +8 unused (unused for E but often used for error checking)

Output: 64 bits (a block)

Note:  $R_i$  becomes  $L_{i+1}$ .

All basic op's are simple logical ops Left shift / XOR

#### Generation of Round Keys

key - user-supplied key (input)

PC-1, PC-2 - permutation tables. PC-2 also extracts 48 of 56 bits

$K_1 - K_{16}$  - round keys (outputs)

Length( $K_i$ ) = 48

$C_i / D_i$  - confusion / diffusion

LSH - left shift (rotation) tables

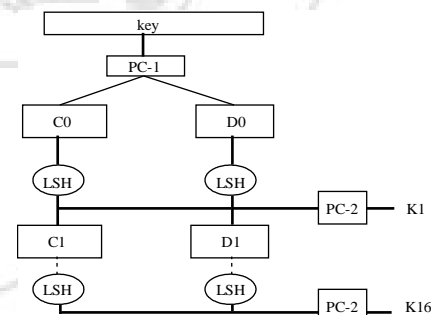
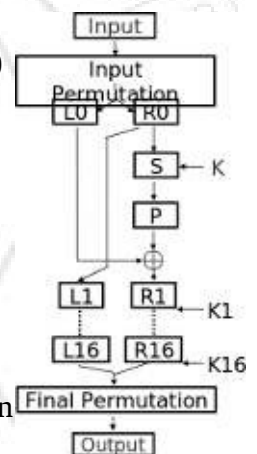
#### Substitution & Permutation

S-box, P-box

#### Problems with DES

Key length is fixed (= 56) - too short for faster computers. Design decisions not public

#### Double DES:



**Rajendra Mane College of Engineering and Technology,  
AMBAV (Devrukh) - 415804**

University of Mumbai

Computer Engineering Sem - VII (Rev) Winter 2010 (10-WIN-COMP-VII-REV-SS)

**SYSTEM SECURITY**

Use double DES encryption.  $C = E(k_2, E(k_1, P))$ . Expected to multiply difficulty of breaking the encryption. Not true.

**Triple DES:**

Tricks used: D not E in the 2nd step,  $k_1$  used twice (in steps 1 & 3) It is:

$C = E(k_1, D(k_2, E(k_1, P)))$  and  $P = D(k_1, E(k_2, D(k_1, C)))$

Doubles the effective key length. 112-bit key is quite strong. Even for today's computers. For all feasible known attacks.

**d) covert channel.**

Ans:

**Covert Channels:** pgms that disclose confidential/secret info. They violate confidentiality, secrecy, or privacy of info.

Examples: An old military radio communication network. The busiest node is most probably the command center.

Creating covert channels: Providing pgm with built-in Trojan horse. Uses covert channel to communicate extracted data.

Example: leaked data hidden in output reports (or displays). Different 'marks' in the report. Varying report format. Changing line length / changing nr of lines per page. Printing or not certain values, characters, or headings. Each 'mark' can convey one bit of info.

Trojan signals value of X as follows:

Bit-1 = 1 if >1 space follows 'ACCOUNT CODE:'; 0 otherwise

Bit-2 = 1 if last digit in 'seconds' field is >5; 0 otherwise

Bit-3 = 1 if heading uses 'TOTALS'; 0 otherwise (uses 'TOTAL')

Bit-4 = 1 if no space follows subtotals line; 0 otherwise => Trojan signaled and spy got: X = '1101'

Types of Covert Channels

**Storage Covert Channels**

Convey info by presence or absence of an object in storage

Protected variable X has n bits:  $X_1, \dots, X_n$ . Trojan within Service Pgm leaks value of X. Trojan and Spy Pgm synchronized, so can "slice" time into n intervals. File FX (not used by anybody else). To signal that  $X_k=1$ , Trojan locks file FX for interval k ( $1 \leq k \leq n$ ). To signal that  $X_k=0$ , Trojan unlocks file FX for interval k. Spy Pgm tries to lock FX during each interval. If it succeeds during k-th interval,  $X_k = 0$  (FX was unlocked). Otherwise,  $X_k = 1$  (FX was locked).

**SYSTEM SECURITY**

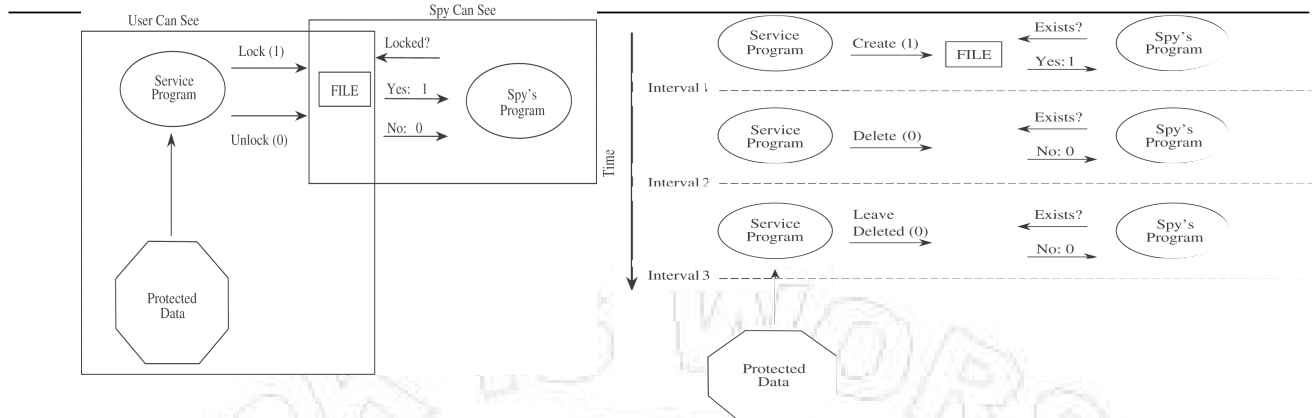
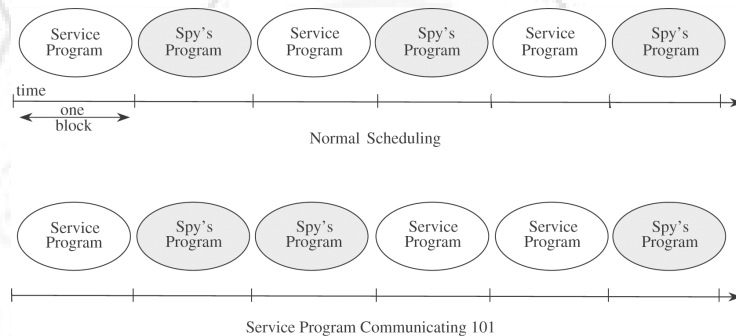


Figure: File Lock Covert Channel. Figure: File Existence Channel Used to Signal 100.

**Timing Covert Channels**

Convey info by varying the speed at which things happen. Example: Multiprogramming system “slices” processor time for programs running on the processor. 2 processes only: Trojan (Pgm w/ Trojan) and Spy Pgm. Trojan receives all odd slices (unless abstains). Spy Pgm receives all even slices (unless abstains). Trojan signals  $X_k=1$  by using its time slice, signals  $X_k=0$  by abstaining from using its slice.

Details: Trojan takes Slice 1 (its 1st slice) signaling  $X_1=1$ . Trojan abstains from taking Slice 3 (its 2nd slice) signaling  $X_2=0$ . Trojan takes Slice 5 (its 3rd slice) signaling  $X_3=1$ .



**Identifying Potential Covert Channels:**

Two techniques for locating covert channels:

- 1) Shared Resource Matrix
- 2) Information Flow Method

Q. No. 7. a) **What is the term Risk Analysis? Explain in detail the steps in Risk Analysis.** 10

Ans: A risk is a potential problem that the system or its users may experience. We distinguish a risk from other project events by looking for three things,  
1. A loss associated with an event. The event must generate a negative effect: compromised security, lost time, diminished quality, lost money, lost control, lost understanding, and so on. This loss is called the risk impact.

# Rajendra Mane College of Engineering and Technology, AMBAV (Devrukh) - 415804

University of Mumbai

Computer Engineering Sem - VII (Rev) Winter 2010 (10-WIN-COMP-VII-REV-SS)

## SYSTEM SECURITY

2. The likelihood that the event will occur. The probability of occurrence associated with each risk is measured from 0 (impossible) to 1 (certain). When the risk probability is 1, we say we have a problem.

3. The degree to which we can change the outcome. For example, if the likelihood of virus attack is 0.3 and the cost to clean up the affected files is \$10,000, then the risk exposure is \$3,000.

So we can use a calculation like this one to decide that a virus checker is worth an investment of \$100, since it will prevent a much larger potential loss. Clearly, risk probabilities can change over time, so it is important to track them and plan for events accordingly.

RISK EXPOSURE :- ( risk impact X risk probability)

Risk is inevitable in life: Crossing the street is risky but that does not keep us from doing it. We can identify, limit, avoid, or transfer risk but we can seldom eliminate it.

We have three strategies for dealing with risk:

1. avoiding the risk, by changing requirements for security or other system characteristics
2. transferring the risk, by allocating the risk to other systems, people, organizations, or assets; or by buying insurance to cover any financial loss should the risk become a reality
3. assuming the risk, by accepting it, controlling it with available resources, and preparing to deal with the loss if it occurs

$$\frac{(\text{risk exposure before reduction}) - (\text{risk exposure after reduction})}{(\text{cost of risk reduction})}$$

If the leverage value of a proposed action is not high enough, then we look for alternative but less costly actions or more effective reduction techniques.

Risk analysis is the process of examining a system and its operational context to determine possible exposures and the potential harm they can cause.

Steps of a Risk Analysis

1. Identify assets.
2. Determine vulnerabilities.
3. Estimate likelihood of exploitation.
4. Compute expected annual loss.
5. Survey applicable controls and their costs.
6. Project annual savings of control.

### Step 1: Identify Assets

Before we can identify vulnerabilities, we must first decide what we need to protect. Thus, the first step of a risk analysis is to identify the assets of the computing system.

The assets can be considered in categories, as listed below.

# Rajendra Mane College of Engineering and Technology, AMBAV (Devrukh) - 415804

University of Mumbai

Computer Engineering Sem - VII (Rev) Winter 2010 (10-WIN-COMP-VII-REV-SS)

## SYSTEM SECURITY

- 1) Hardware: processors, boards, keyboards, monitors, terminals, microcomputers, workstations, tape drives, printers, disks, disk drives, cables, connections, communications controllers, and communications media
- 2) Software: source programs, object programs, purchased programs, in-house programs, utility programs, operating systems, systems programs (such as compilers), and maintenance diagnostic programs
- 3) Data: data used during execution, stored data on various media, printed data, archival data, update logs, and audit records
- 4) People: skills needed to run the computing system or specific programs
- 5) Documentation: on programs, hardware, systems, administrative procedures, and the entire system
- 6) Supplies: paper, forms, laser cartridges, magnetic media, and printer fluid

### Step 2: Determine Vulnerabilities

We want to predict what damage might occur to the assets and from what sources. We can enhance our imaginative skills by developing a clear idea of the nature of vulnerabilities. This nature derives from the need to ensure the three basic goals of computer security: confidentiality, integrity, and availability. Thus, vulnerability is any situation that could cause loss of confidentiality, integrity, and availability.

Assets and Security Properties			
Asset	Confidentiality	Integrity	Availability
Hardware			
Software			
Data			
People			

### Step 3: Estimate Likelihood of Exploitation

Risk analysis is determining how often each exposure is likely to be exploited. Likelihood of occurrence relates to the stringency of the existing controls and the likelihood that someone or something will evade the existing controls.

In security, it is often not possible to directly evaluate an event's probability by using classical techniques. However, we can try to apply frequency probability by using observed data for a specific system. Local failure rates are fairly easy to record, and we can identify which failures resulted in security breaches or created new vulnerabilities. In particular, operating systems can track data on hardware failures, failed login attempts, numbers of accesses, and changes in the sizes of data files.

Ratings of Likelihood	
Frequency	Rating
More than once a day	10
Once a day	9
Once every three days	8
Once a week	7
Once in two weeks	6
Once a month	5
Once every four months	4
Once a year	3
Once every three years	2
Less than once in three years	1

### Step 4: Compute Expected Loss

We have gained an understanding of the assets we value, their possible vulnerabilities, and the likelihood that the vulnerabilities will be exploited. We must determine the likely loss if the exploitation does indeed occur. As with likelihood of occurrence, this value is difficult to determine. Some costs, such as the cost to replace a hardware item, are easy to obtain. The cost to replace a piece of software can be approximated reasonably well from the initial cost to buy it (or specify, design, and write it).

# Rajendra Mane College of Engineering and Technology, AMBAV (Devrukh) - 415804

University of Mumbai

Computer Engineering Sem - VII (Rev) Winter 2010 (10-WIN-COMP-VII-REV-SS)

## SYSTEM SECURITY

If a computing system, a piece of software, or a key person is unavailable, causing a particular computing task to be delayed, there may be serious consequences. If a program that prints paychecks is delayed, employees' confidence in the company may be shaken, or some employees may face penalties from not being able to pay their own bills. If customers cannot make transactions because the computer is down, they may choose to take their business to a competitor. For some time-critical services involving human lives, such as a hospital's life-support systems or a space station's guidance systems, the costs of failure are infinitely high.

### Step 5: Survey and Select New Controls

We understand the system's vulnerabilities and the likelihood of exploitation. We turn next to an analysis of the controls to see which ones address the risks we have identified. We want to match each vulnerability with at least one appropriate security technique, as shown in Figure. Once we do that, we can use our expected loss estimates to help us decide which controls, alone or in concert, are the most cost effective for a given situation. Notice that vulnerabilities E and F are countered by primary techniques 2 and 4, respectively. The secondary control techniques 2 and 3 for vulnerability F are good defense in depth. The fact that there is no secondary control for vulnerability E is a minor concern. But vulnerability T is a serious caution, because it has no control whatsoever.

	Technique 1	Technique 2	Technique 3	Technique 4
Vulnerability A		↑	↑	↑
Vulnerability B				
Vulnerability C				
Vulnerability D				
Vulnerability E	→	Primary		Primary
Vulnerability F	→	Secondary	Secondary	Primary
Vulnerability G				
⋮				
Vulnerability T	←		Caution	

### Step 6: Project Savings

The next step is to determine whether the costs outweigh the benefits of preventing or mitigating the risks. Recall that we multiply the risk probability by the risk impact to determine the risk exposure. The risk impact is the loss that we might experience if the risk were to turn into a real problem. There are techniques to help us determine the risk exposure.

b) *How is the physical security provided for protection needed outside the computer system.*

10

Ans: Physical security is the term used to describe protection needed outside the computer system. Typical physical security controls include guards, locks, and fences to deter direct attacks. In addition, there are other kinds of protection

**Rajendra Mane College of Engineering and Technology,  
AMBAV (Devrukh) - 415804**

University of Mumbai

Computer Engineering Sem - VII (Rev) Winter 2010 (10-WIN-COMP-VII-REV-SS)

**SYSTEM SECURITY**

---

against less direct disasters, such as floods and power outages; these, too, are part of physical security.

But many threats to security involve human or natural disasters, events that should also be addressed in the security plan. For this reason, we consider how to cope with the non-technical things that can go wrong.

There are two pieces to the process of dealing with non-technical problems: preventing things that can be prevented and recovering from the things that cannot be prevented.

- 1) Definition
- 2) Natural Disasters: Flood, Fire, Other Natural Disasters
- 3) Power Loss: Uninterruptible Power Supply, Surge suppressor
- 4) Human Vandals: Unauthorized access and Use, Theft
- 5) Control
  - 1) Backup: Complete Backup, Revolving Backup, Selective backup
  - 2) Offsite Backup
  - 3) Network Storage: Cold site, Hot site.