COS 318: Operating Systems

Security and Privacy

Prof. Margaret Martonosi
Computer Science Department
Princeton University

http://www.cs.princeton.edu/courses/archive/fall11/cos318/

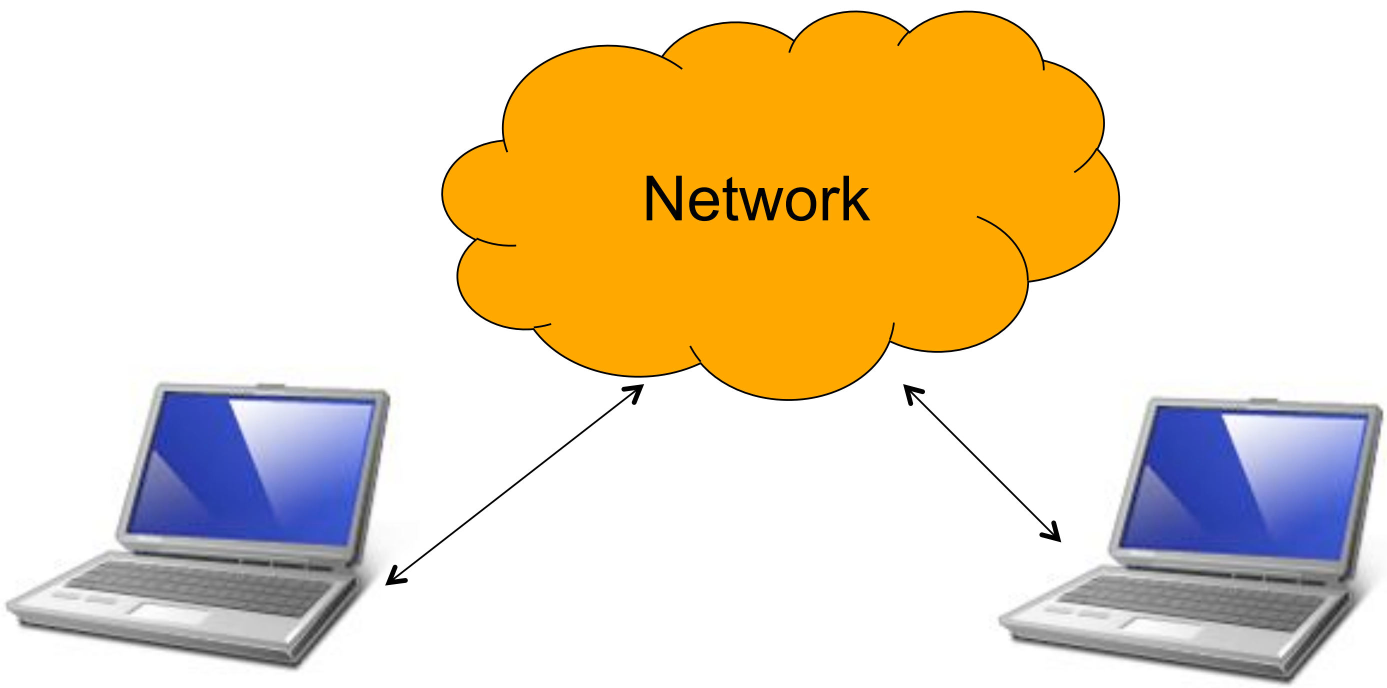
Announcements

- Precept tonight: Covers Project 3
  - Due to fall break, no design review for Project 3.
  - Due Weds Nov 9
- Midterm Thursday, Oct. 27 during normal class time
  - Covers material through Thur Oct 20 (last week)
  - Closed book.
  - 1-page cheatsheet allowed. (But since the class has few formulas etc, its utility is unclear…)
    - 8.5"x11"
    - One sided

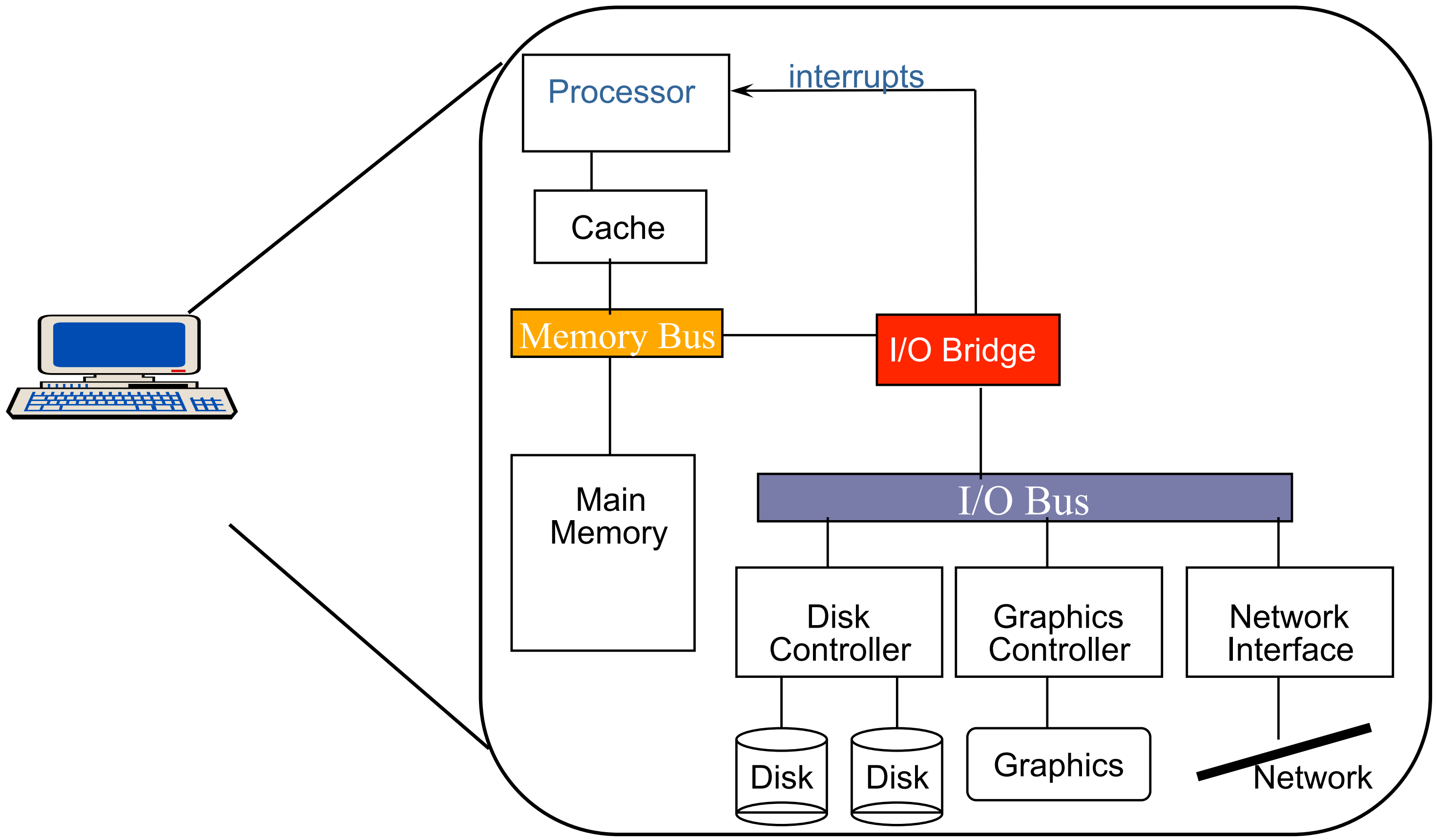
Today's Topics

- Security, Privacy, and OS role…

What are we trying to secure? And from whom?



What are we trying to secure? And from whom?



What sorts of security/privacy issues are we protecting against?

The Security Environment

- Security goals and threats

Goal	Threat
Data confidentiality	Exposure of data
Data integrity	Tampering with data
System availability	Denial of service
Exclusion of outsiders	System takeover by viruses

A couple categories…

Intruders

- Casual prying by nontechnical users
- Snooping by insiders
- Determined attempt to make trouble (or personal gain)
- Commercial or military espionage

Accidental Data Loss

- Acts of God
  - fires, floods, wars
- Hardware or software errors
  - CPU malfunction, bad disk, program bugs
- Human errors
  - data entry, wrong tape mounted, rm *

How to protect?

- Hardware?
  - Parity and error-correcting codes: Memory, Caches, Disk, …
  - Blurring techniques for covert channels: even out power consumption etc etc.
  - Physical access: it shouldn't be so easy to unscrew the back of the voting machine…
  - Zeroing out memory
  - Hardware help with memory isolation & protection
  - Timers…
- OS?
  - Process isolation: scheduling, memory spaces, encryption, process privileges, passwords!, driver security…
- Languages?
- Constraints on memory access, …

- Communication protocols?

The Security Environment

- Security goals and threats

Goal	Threat
Data confidentiality	Exposure of data
Data integrity	Tampering with data
System availability	Denial of service
Exclusion of outsiders	System takeover by viruses

Data Integrity: Step 0

Redundancy and ECC

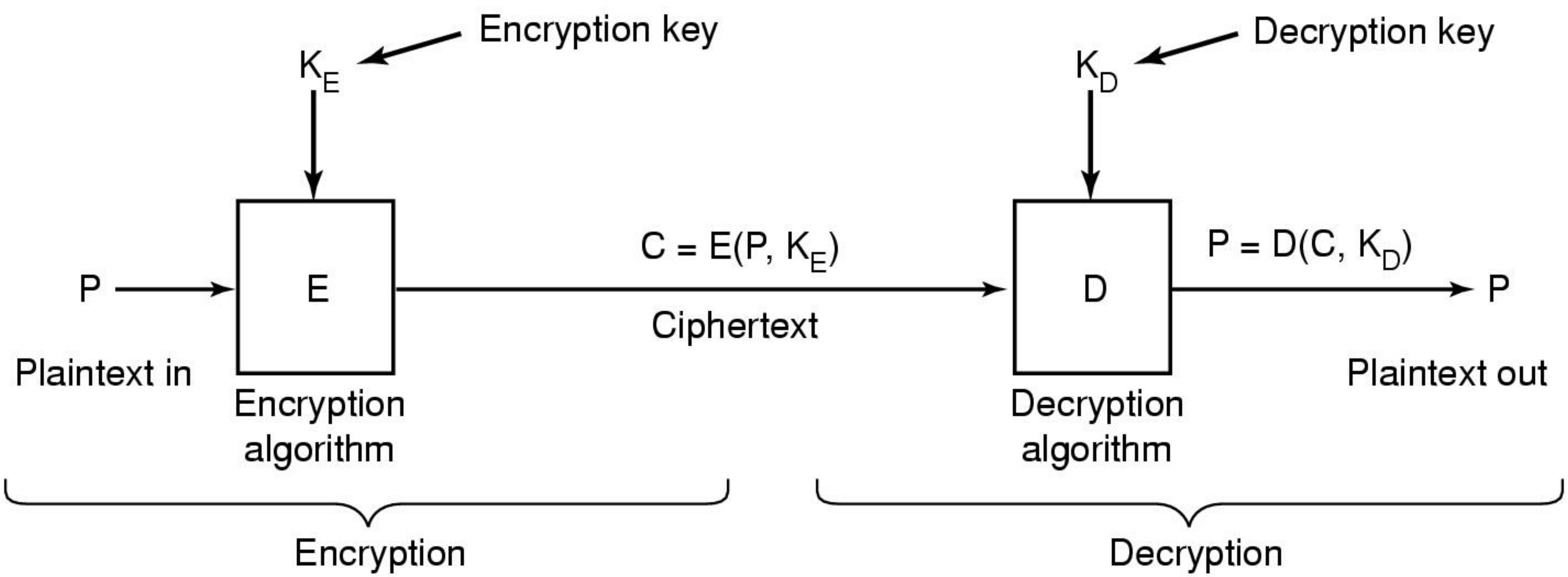
- Replication of data, geographically distributed
  - As simple as backups
  - First-class replication (Coda)
  - Voting schemes
- Error detection-correction
  - Erasure codes (encode n blocks into >n blocks, requiring r blocks to recover original content of original n)
  - Parity bits, checksums

Data Confidentiality: Step 0

Encryption

- symmetric key cryptography
- public key cryptography
- digital signatures
- one-way functions
- hashes

Basics of Cryptography



Challenges?

- Agreeing on a key
- Selecting a useful encryption/decryption function

Secret-Key Cryptography

- Secret-key crypto called symmetric-key crypto
  - If keys are long enough there are OK algorithms
  - Secret key must be shared by both parties

  - How to distribute?

Public-Key Cryptography

- All users pick a public key/private key pair
  - publish the public key
  - private key not published
- Public key is (usually*) the encryption key
- Private key is (usually*) the decryption key
- RSA

The Security Environment

- Security goals and threats

Goal	Threat
Data confidentiality	Exposure of data
Data integrity	Tampering with data
System availability	Denial of service
Exclusion of outsiders	System takeover by viruses

Exclusion of Outsiders: User Authentication

Basic Principles. Authentication must identify:

1. Something the user knows
2. Something the user has
3. Something the user is

This is done before user can use the system for access control

Authentication Using Passwords

LOGIN: ken
PASSWORD: FooBar
SUCCESSFUL LOGIN

(a)

LOGIN: carol
INVALID LOGIN NAME
LOGIN:

(b)

LOGIN: carol
PASSWORD: Idunno
INVALID LOGIN
LOGIN:

(c)

(a) A successful login

(b) Login rejected after name entered

(c) Login rejected after name and password typed

Authentication Using Passwords

LBL> telnet elxsi
ELXSI AT LBL
LOGIN: root
PASSWORD: root
INCORRECT PASSWORD, TRY AGAIN
LOGIN: guest
PASSWORD: guest
INCORRECT PASSWORD, TRY AGAIN
LOGIN: uucp
PASSWORD: uucp
WELCOME TO THE ELXSI COMPUTER AT LBL

- How a cracker broke into LBL
  - a U.S. Dept. of Energy research lab

Login Spoofing

(a) (b)

(a) Correct login screen
(b) Phony login screen

Authentication Using Passwords

Bobbie, 4238, e(Dog,4238)
Tony, 2918, e(6%%TaeFF,2918)
Laura, 6902, e(Shakespeare,6902)
Mark, 1694, e(XaB@Bwcz,1694)
Deborah, 1092, e(LordByron,1092)

Salt Password

- Salt = random bits used in function with provided password
- Helps defeat precomputation of encrypted passwords

One-Time Passwords

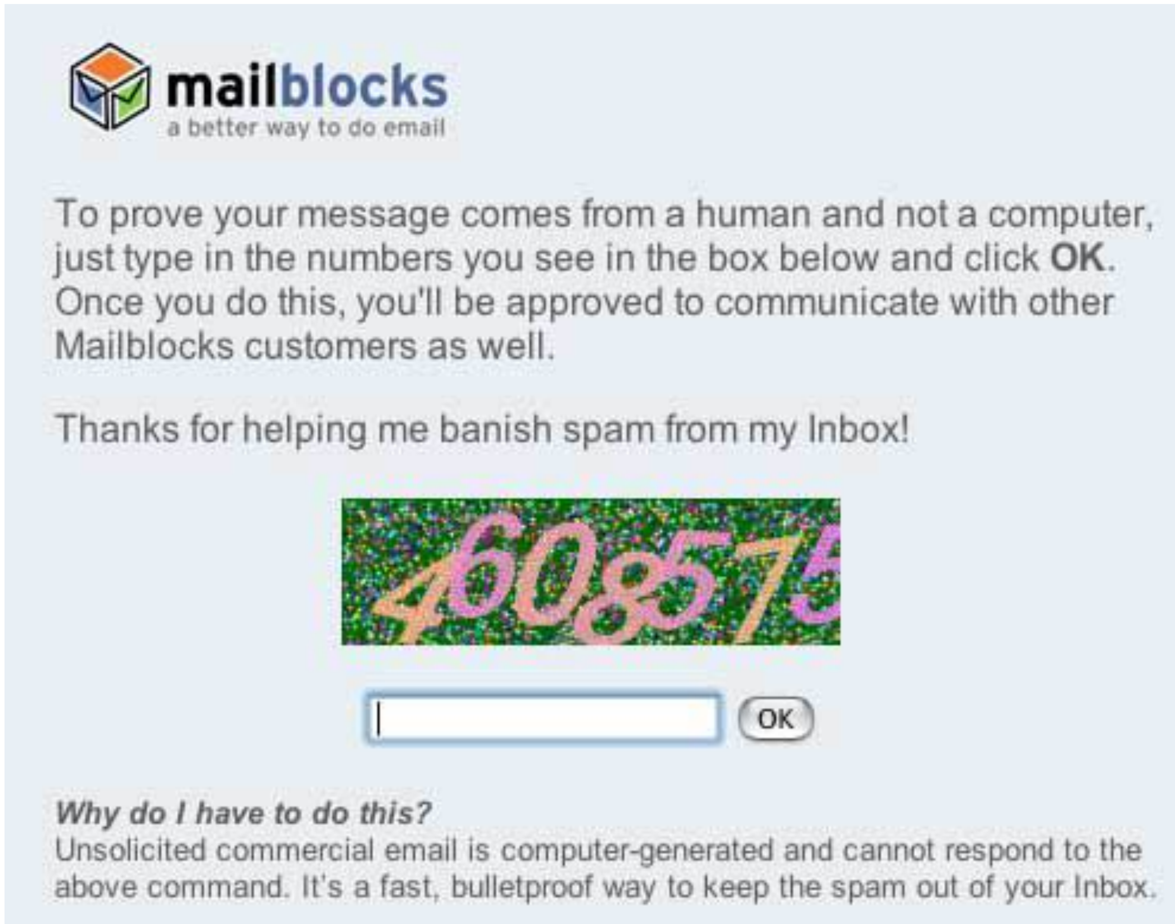
Using 1-way function:

- Function such that given formula for f(x)
  - easy to evaluate y = f(x)
- But given y
  - computationally infeasible to find x
- One-time passwords
  - Choose password s and integer n
  - 1st time P_1=f(f(f(f(s)))), 2nd time P_2= f(f(f(s))), etc
  - Login name supplies current integer value
  - Server stores old password, f(newpassword)==old?

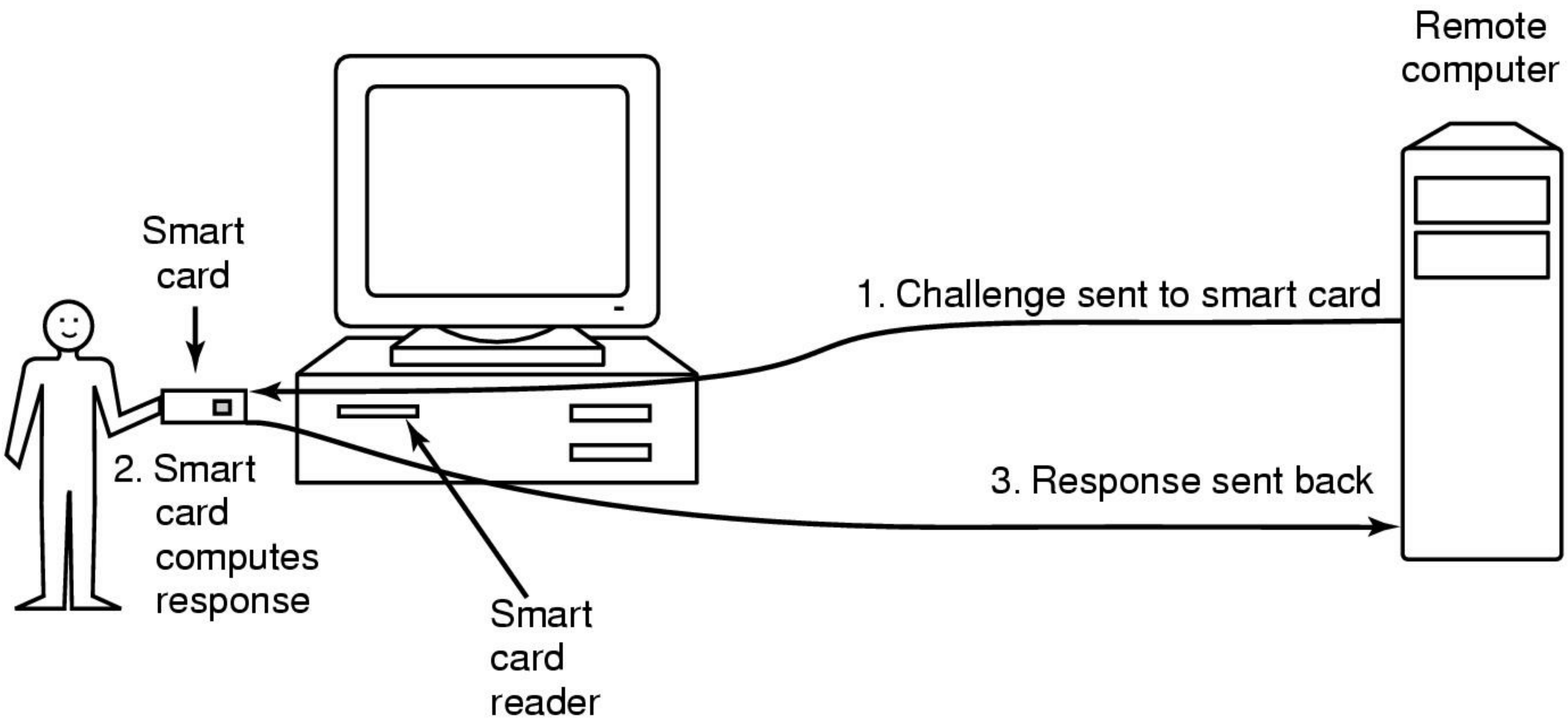
Challenge - Response

- Sets of question – answer pairs
  - Server picks one and asks
  - User knows answer
- User picks function f(x)
  - Server sends a value for x
  - User sends back f(x) as password
- Using symmetric encryption
  - Server sends random value r
  - User encrypts with secret key – e(r,k)
- Server compares

Graphical Challenge-Response



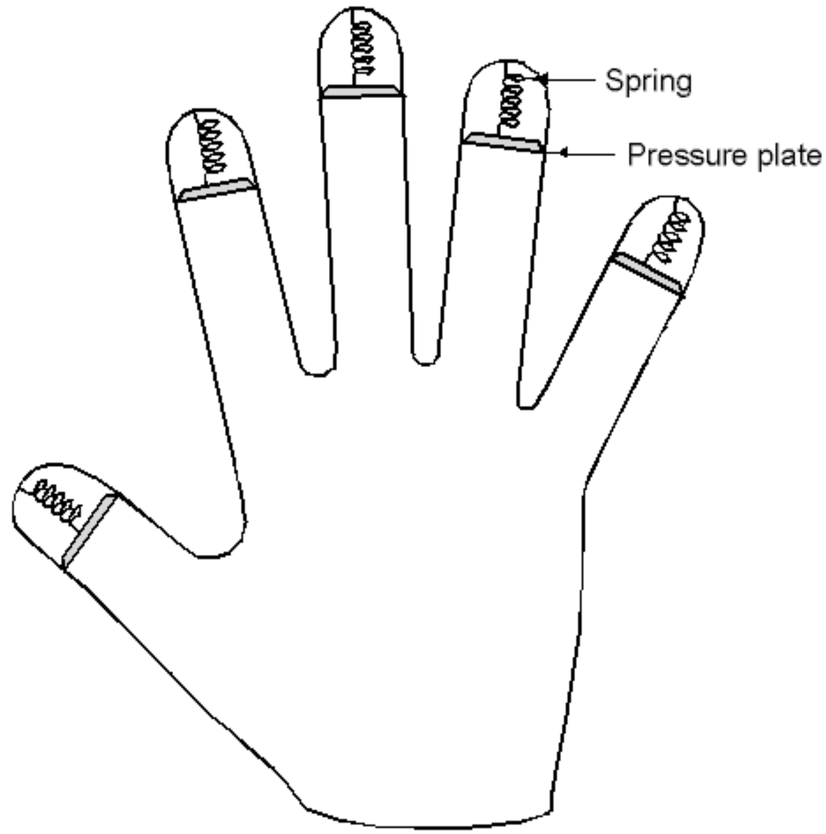
Authentication Using a Physical Object



- Magnetic cards
  - magnetic stripe cards
  - chip cards: stored value cards, smart cards
- RFIDs

Authentication Using Biometrics

- A device for measuring finger length.
- Retinal scans
- Voice recognition
- Surveillance tech
  - Image analysis
  - Gait analysis



Countermeasures

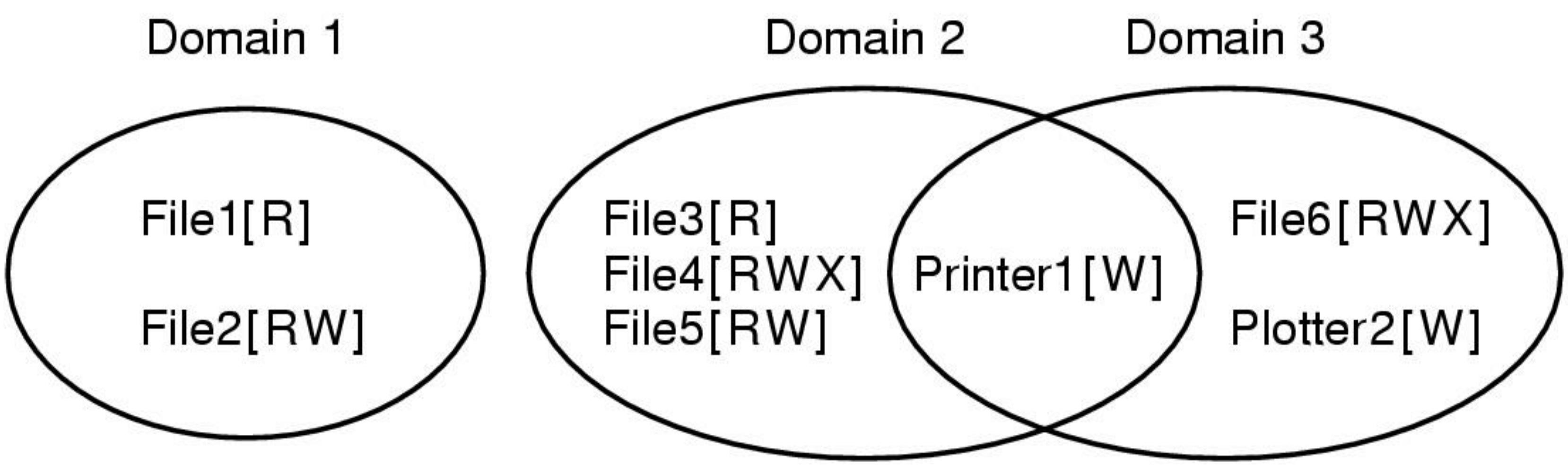
- Limiting times when someone can log in
- Automatic callback at number prespecified
- Limited number of login tries
- A database of all logins
- Simple login name/password as a trap
  - security personnel notified when attacker bites

The Security Environment

- Security goals and threats

Goal	Threat
Data confidentiality	Exposure of data
Data integrity	Tampering with data
System availability	Denial of service
Exclusion of outsiders	System takeover by viruses

Access Control Mechanisms: Protecting software and data from other programs



Examples of three protection domains

The Access Model

- Authorization problems can be represented abstractly by of an *access model*.
  - each row represents a subject/principal/domain
  - each column represents an object
  - each cell: accesses permitted for the *{subject, object}* pair
    - read, write, delete, execute, search, control, or any other method
- In real systems, the access matrix is sparse and dynamic.
    - need a flexible, efficient representation

Access Control Matrix

- Processes execute in a protection domain, initially inherited from subject

	gradefile	solutions	proj1	luvltr	hotgossip
TA	rw	rw	rx	r	
grp		r	rwx		
Terry					rw
Lynn				rw	rw

Two Representations

- ACL - Access Control Lists
  - Columns of previous matrix
  - Permissions attached to Objects
  - ACL for file hotgossip: Terry, rw; Lynn, rw
- Capabilities
  - Rows of previous matrix
  - Permissions associated with Subject
  - Tickets, Namespace (what it is that one can name)
  - Capabilities held by Lynn: luvltr, rw; hotgossip,rw

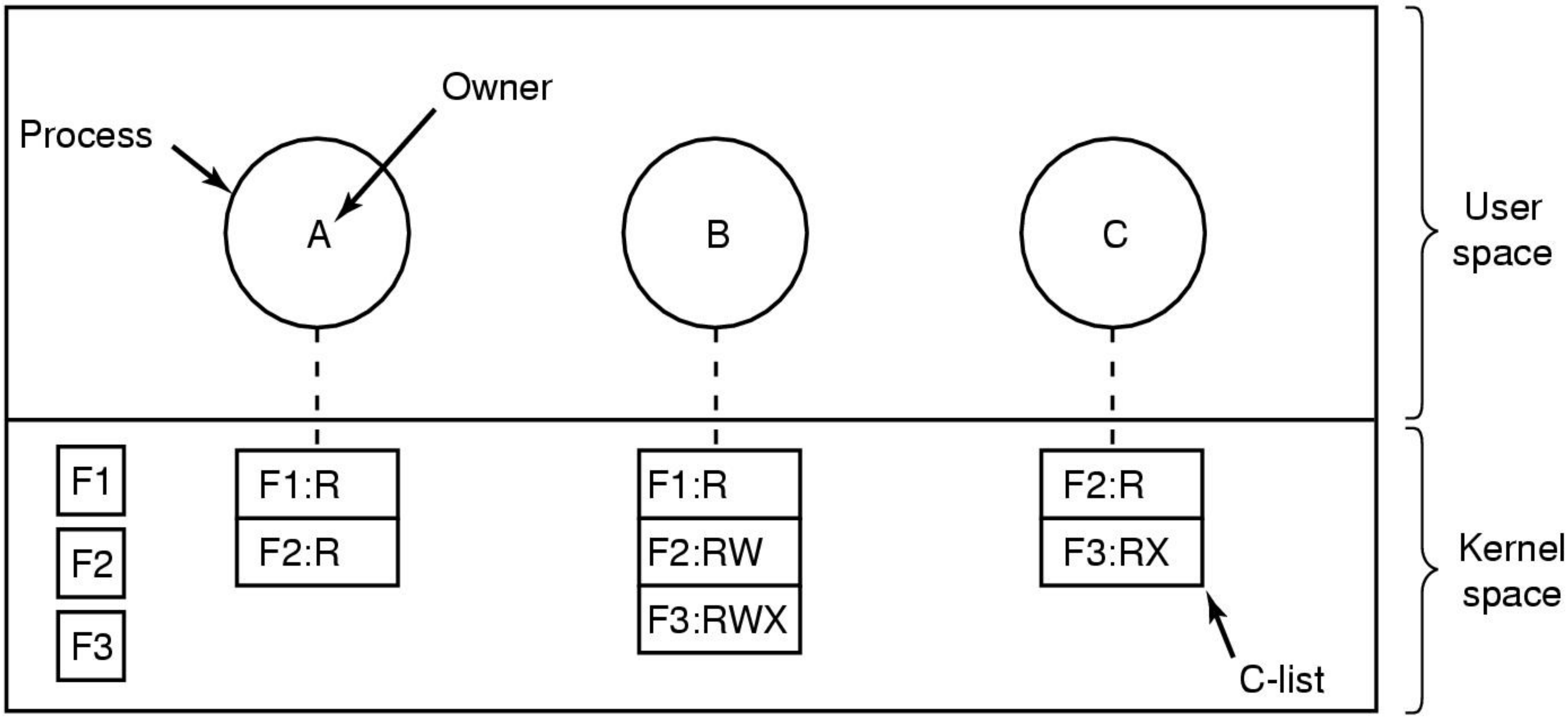
Access Control Lists

- *Approach*: represent the access matrix by storing its columns with the objects.
  - Tag each object with an *access control list* (ACL) of authorized subjects/principals.
- To authorize an access requested by *S* for *O*
  - search *O*'s ACL for an entry matching *S*
  - compare requested access with permitted access
  - access checks are often made only at bind time

Capabilities

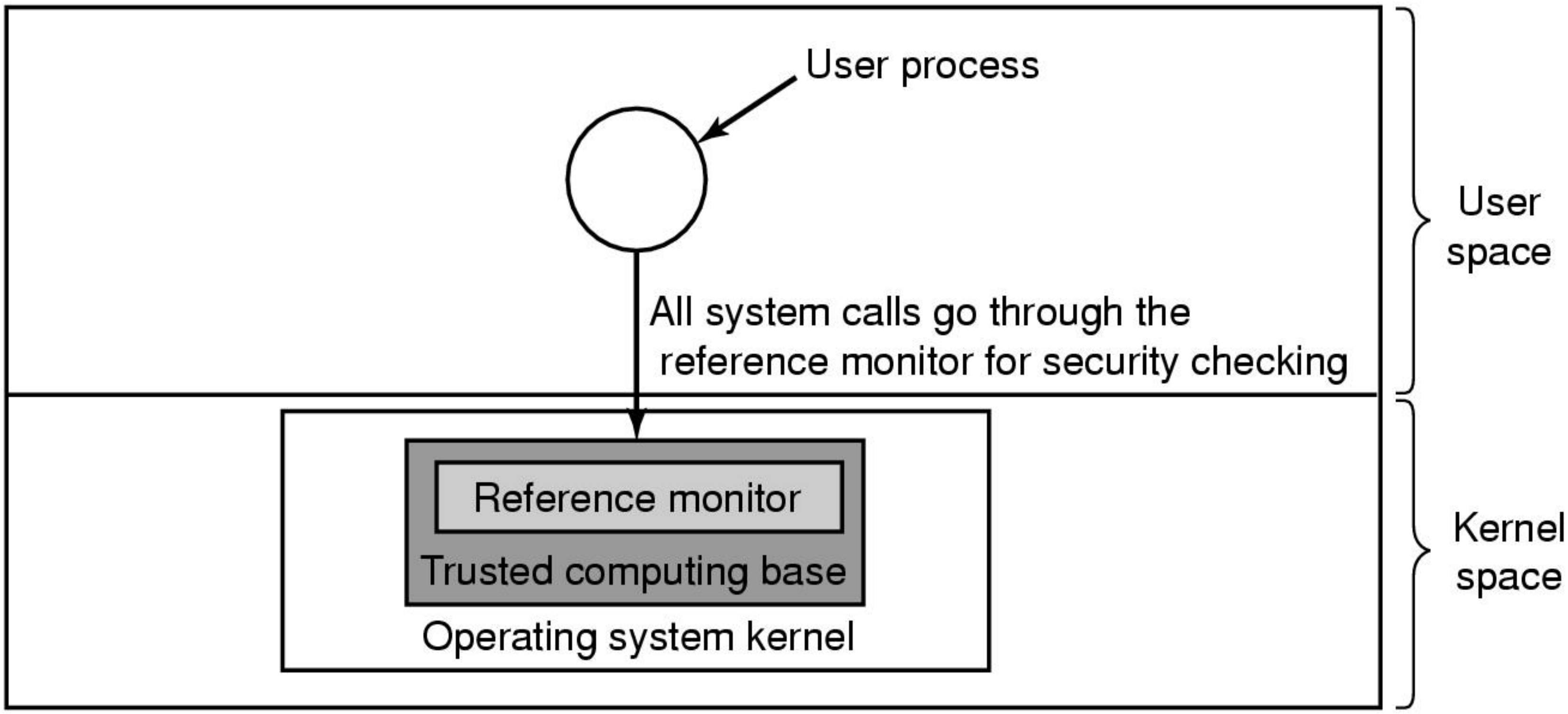
- Approach: represent the access matrix by storing its rows with the subjects.
    - Tag each subject with a list of capabilities for the objects it is permitted to access.
  - A capability is an unforgeable object reference, like a pointer.
  - It endows the holder with permission to operate on the object
    - e.g., permission to invoke specific methods
  - Typically, capabilities may be passed from one subject to another.
    - Rights propagation and confinement problems

Capabilities



Each process has a capability list

Trusted Systems
Trusted Computing Base



A reference monitor

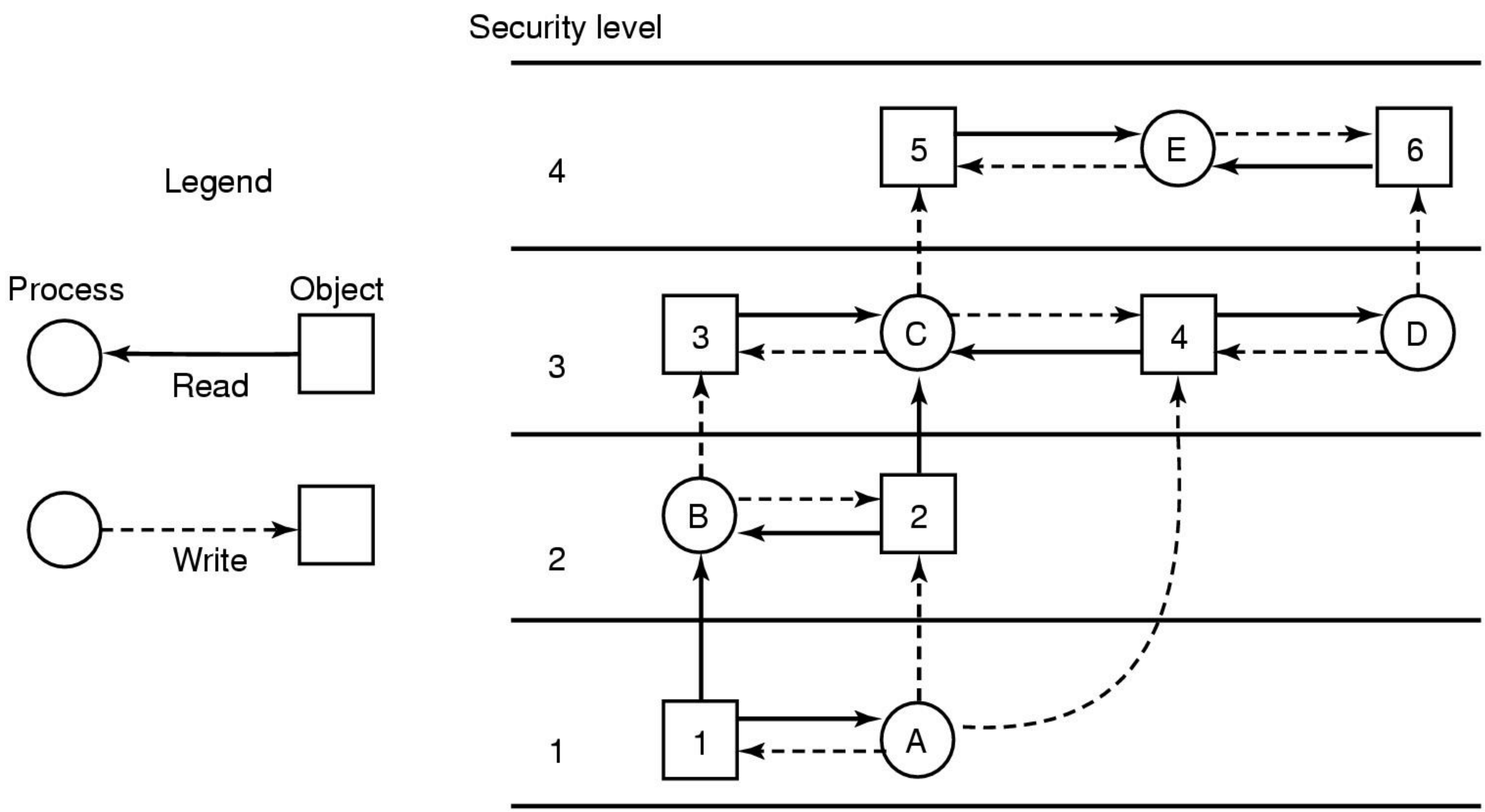
Multilevel Security: The Bell-La Padula Model

Rules for the Bell-La Padula model:

- **The simple security property**: A process running at security level k can read only objects at its level or lower.
- **The * property**: A process running at security level k can write only objects at its level or higher.

- Military inspired:
- A lieutenant can read less stuff than a general
- Generals should be careful where they write down what they know, lest a lieutenant read it.

Multilevel Security



The Bell-La Padula multilevel security model

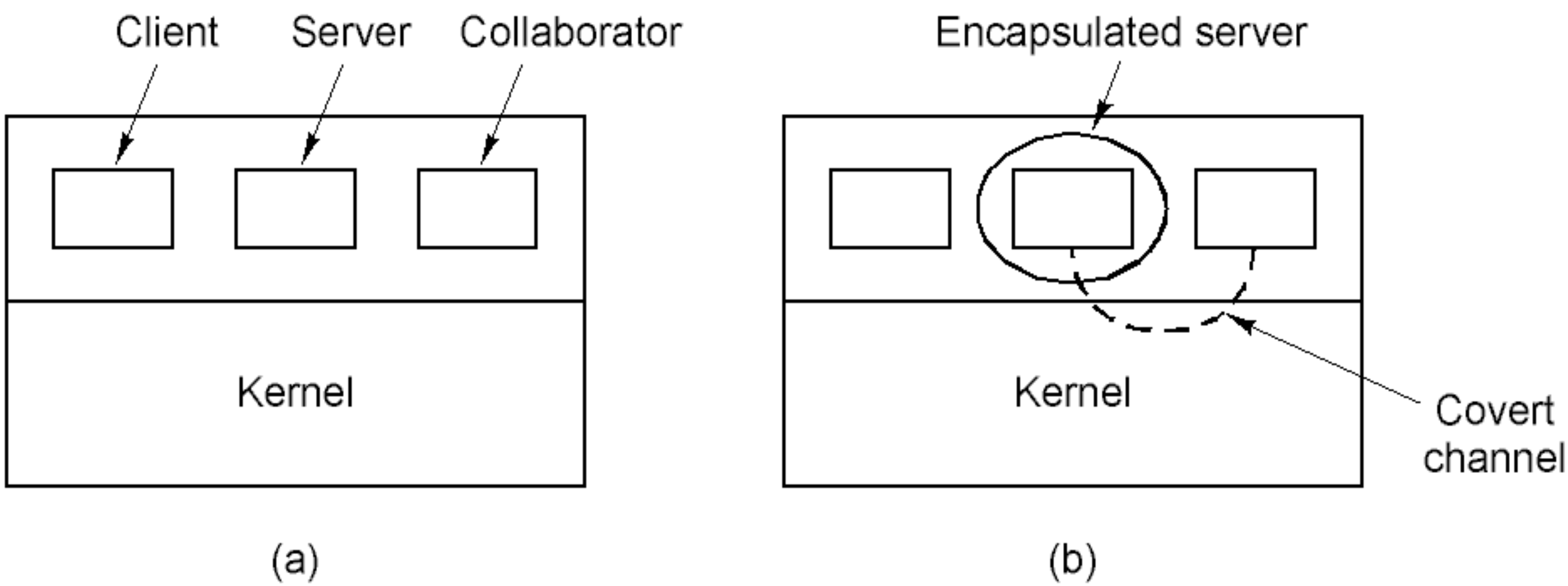
The Security Environment

- Security goals and threats

Goal	Threat
Data confidentiality	Exposure of data
Data integrity	Tampering with data
System availability	Denial of service
Exclusion of outsiders	System takeover by viruses

Covert Channels

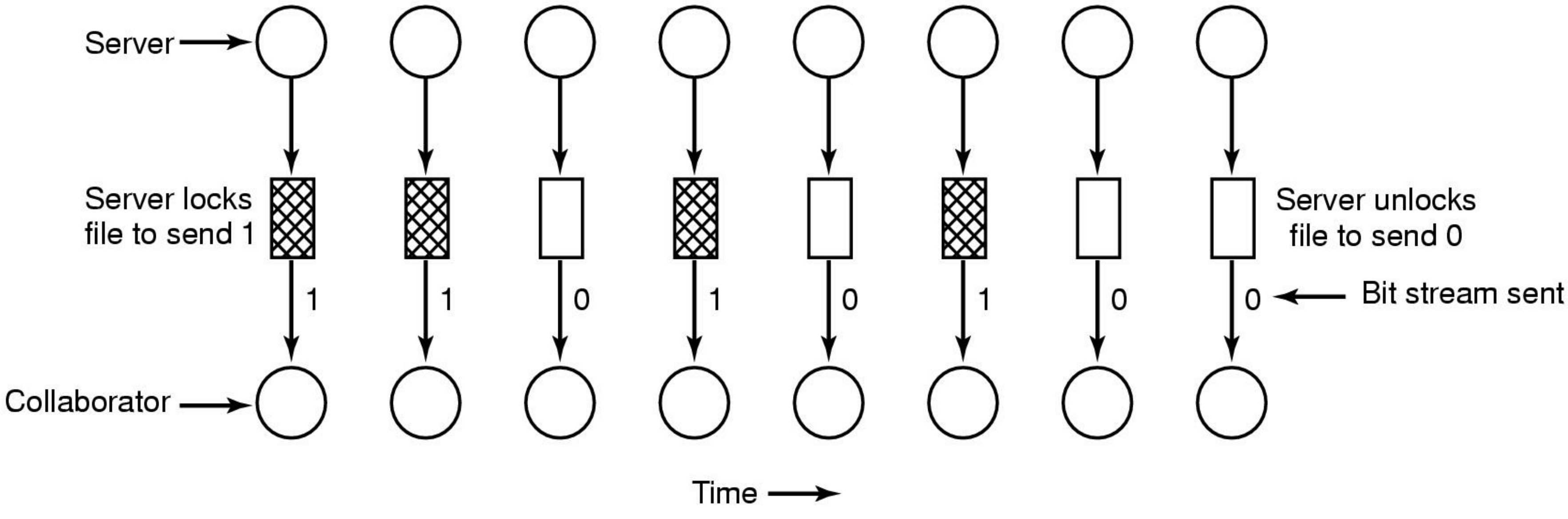
- Encode information someplace unexpected…



Client, server and collaborator processes

Encapsulated server can still leak to collaborator via covert channels

Covert Channels



A covert channel using file locking

Covert Channels

- Pictures appear the same
- Picture on right has text of 5 Shakespeare plays
  - encrypted, inserted into low order bits of color values

Zebras

Hamlet, Macbeth, Julius Caesar
Merchant of Venice, King Lear