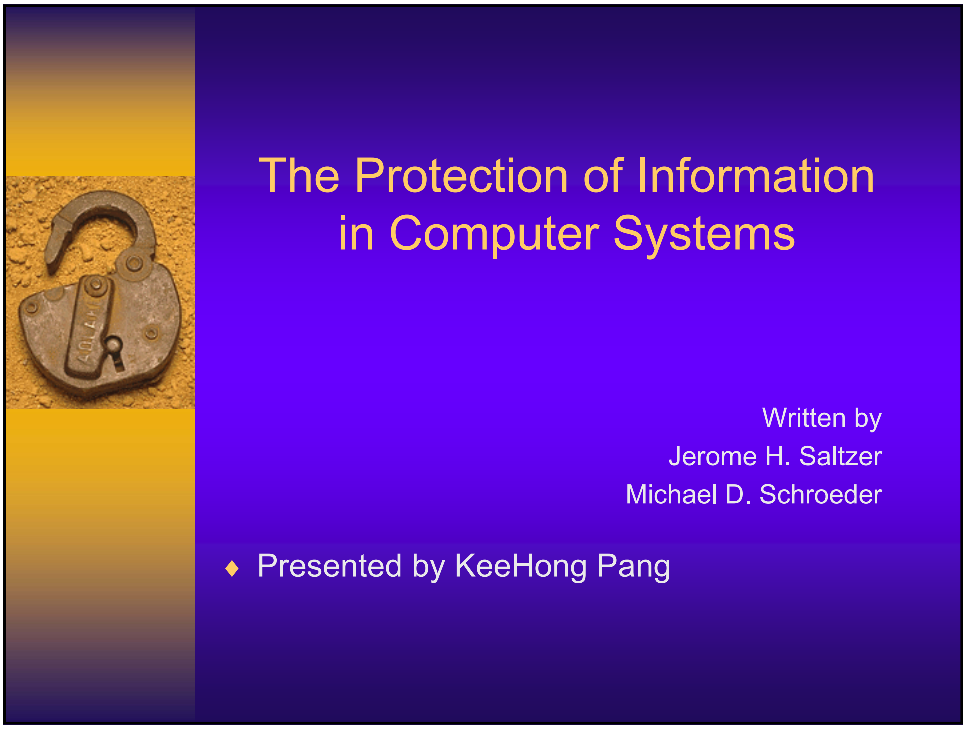


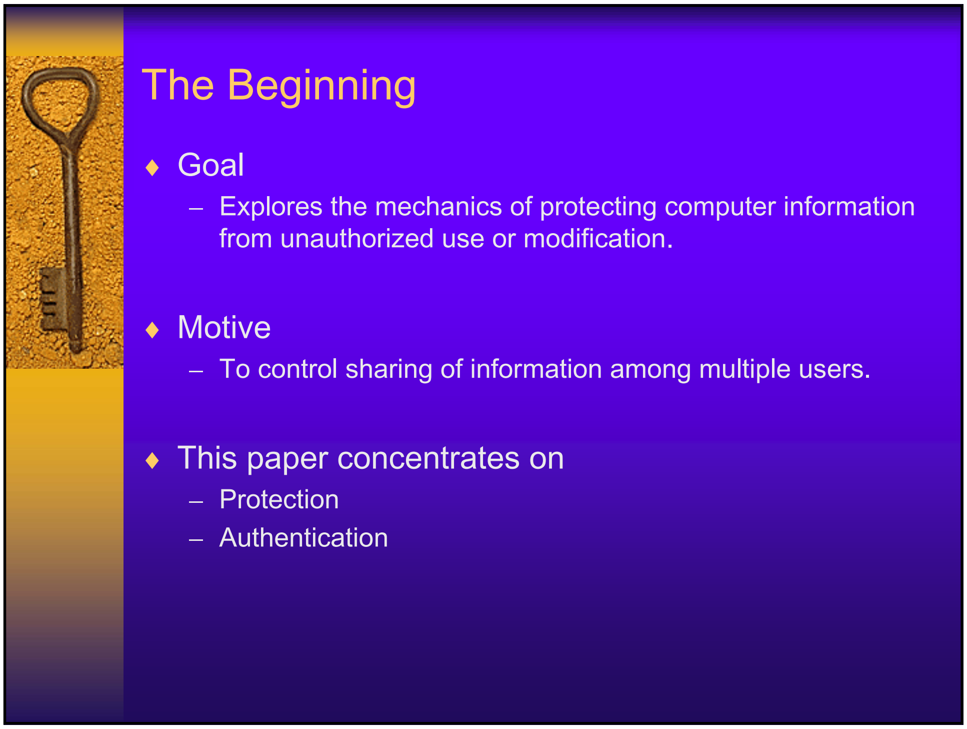
The Protection of Information in Computer Systems

Written by
Jerome H. Saltzer
Michael D. Schroeder

- Presented by KeeHong Pang

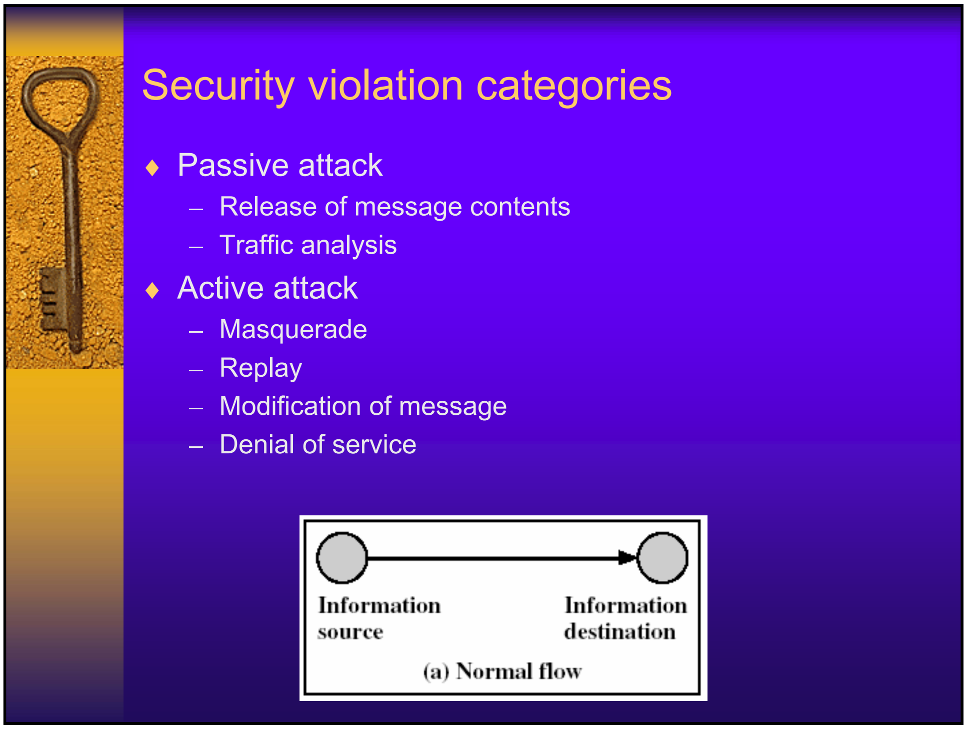
Organization

- Section I
  - Desired functions
  - Design principles
  - Examples of elementary protection and authentication mechanisms
- Section II
  - Principles of modern protection architectures
  - The relation between capability systems and access control list systems
  - Protected subsystems and protected objects
- Section III
  - Review of the state of the art and current research projects



The Beginning

- Goal
  - Explores the mechanics of protecting computer information from unauthorized use or modification.
- Motive
  - To control sharing of information among multiple users.
- This paper concentrates on
  - Protection
  - Authentication

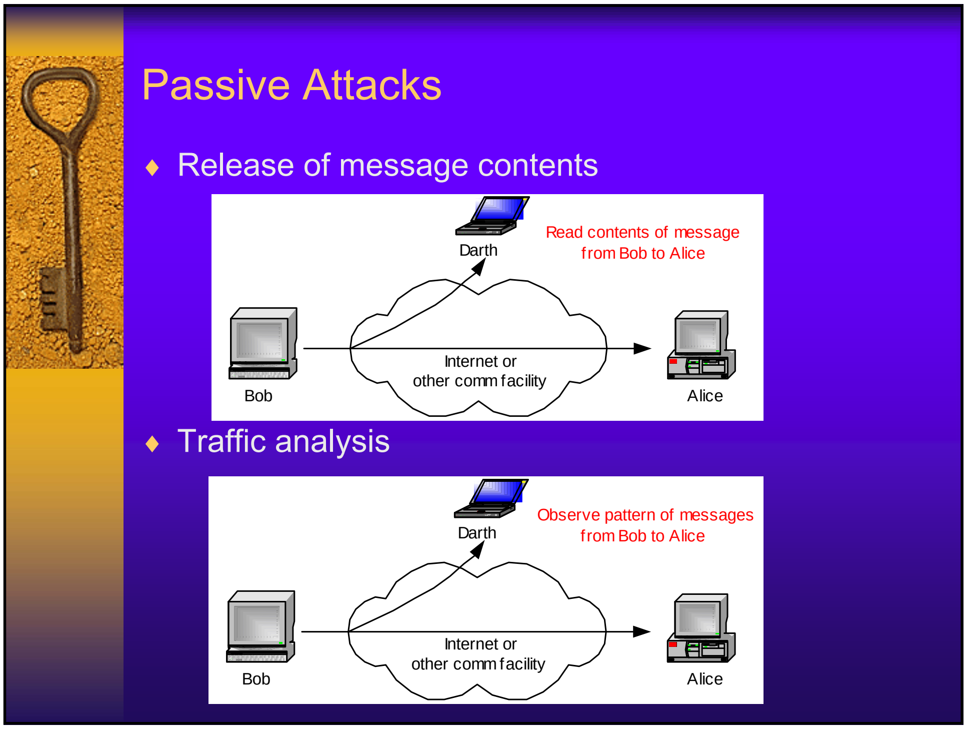


Security violation categories

- Passive attack
  - Release of message contents
  - Traffic analysis
- Active attack
  - Masquerade
  - Replay
  - Modification of message
  - Denial of service

Information source
Information destination

(a) Normal flow



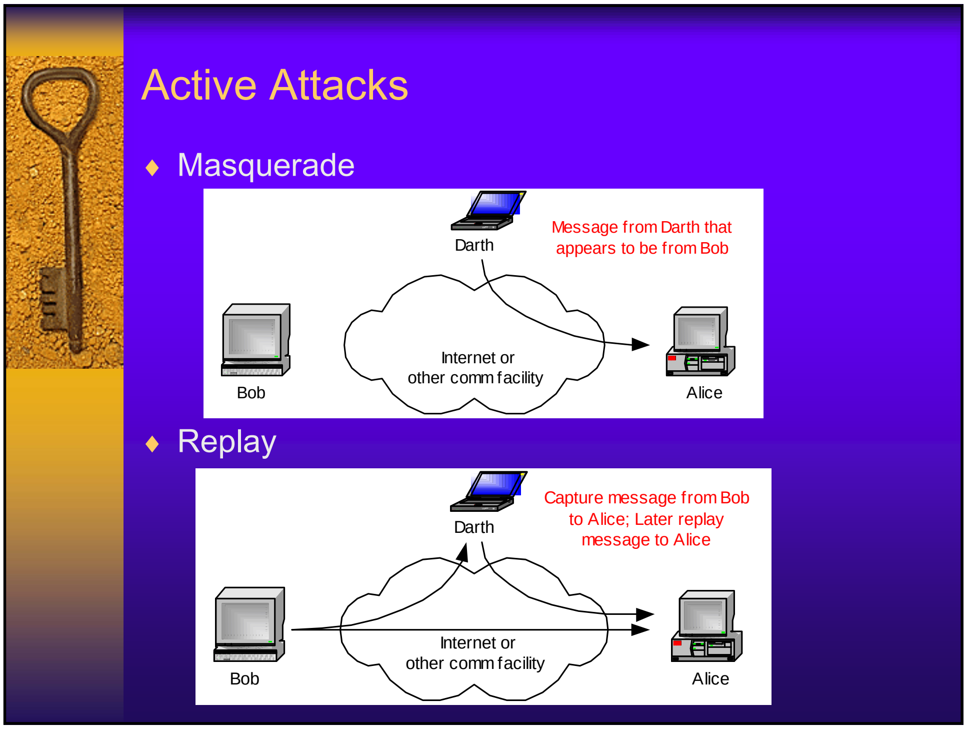
Passive Attacks

- Release of message contents

Darth
Read contents of message from Bob to Alice
Bob
Internet or other comm facility
Alice

- Traffic analysis

Darth
Observe pattern of messages from Bob to Alice
Bob
Internet or other comm facility
Alice



Active Attacks

- Masquerade

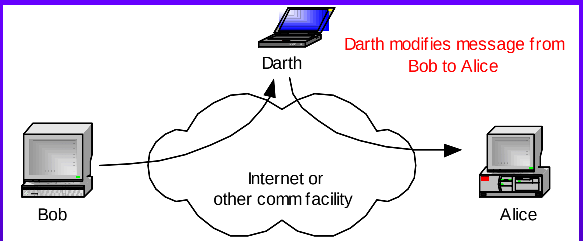
Darth
Message from Darth that appears to be from Bob
Bob
Internet or other comm facility
Alice

- Replay

Darth
Capture message from Bob to Alice; Later replay message to Alice
Bob
Internet or other comm facility
Alice

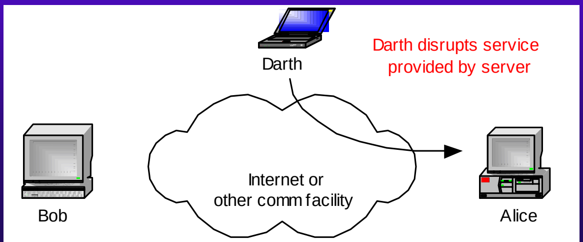
Active Attacks

- Modification of messages



Darth modifies message from Bob to Alice

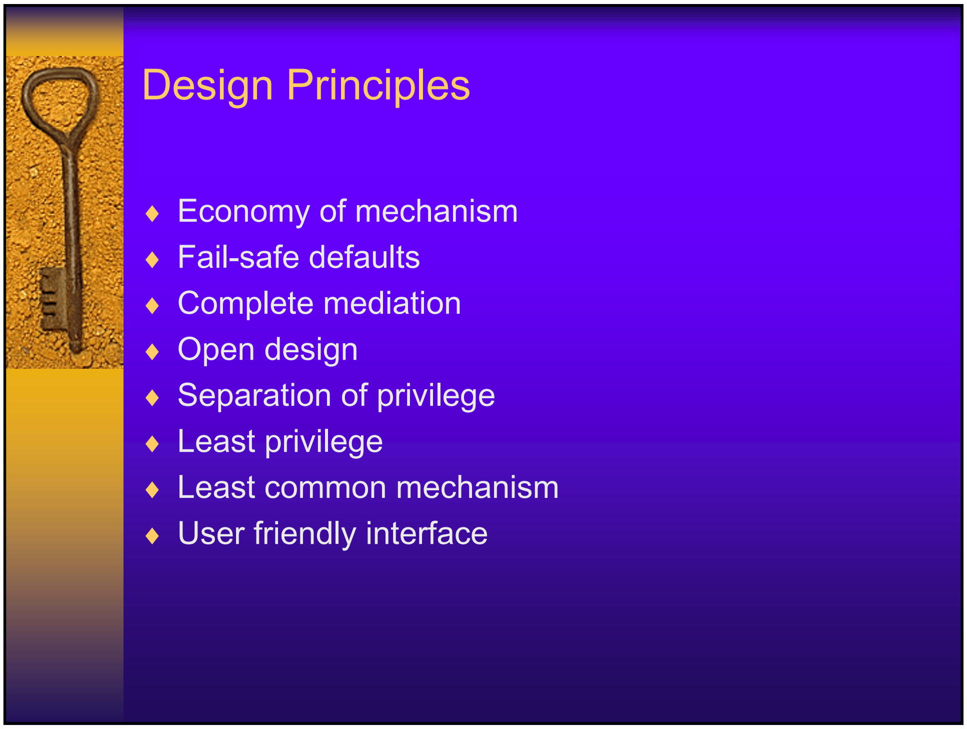
- Denial of service



Darth disrupts service provided by server

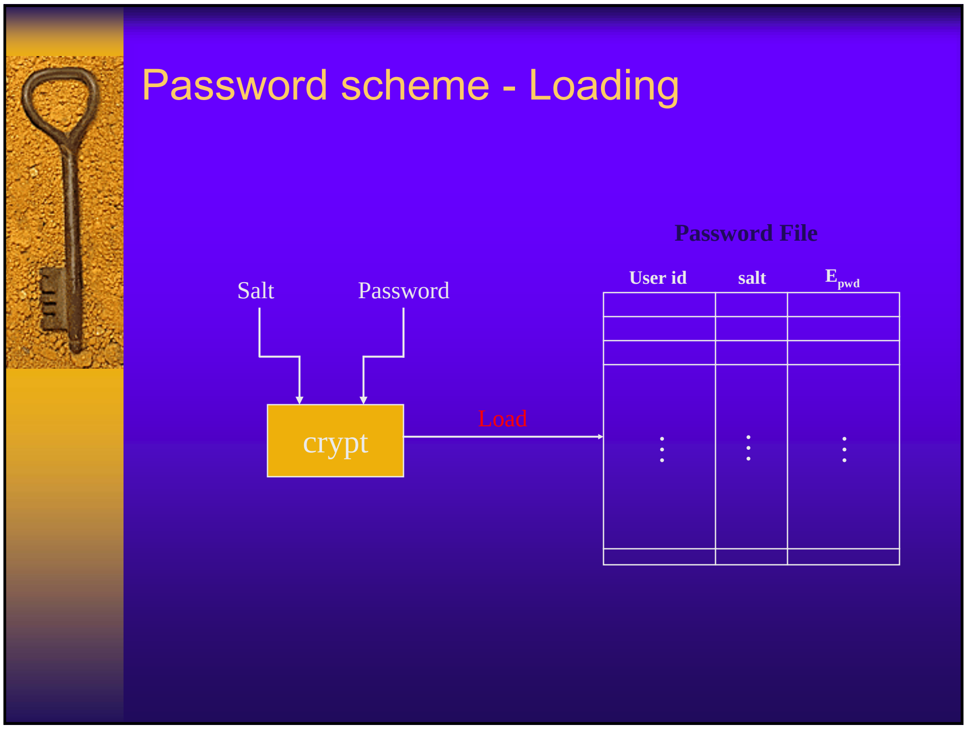
Protection schemes

- Unprotected systems
  - No provision for protection.
- All-or-nothing systems
  - Provide isolation of users or total sharing of some info.
- Controlled sharing
  - Control who may access each data item stored in the system.
- User-programmed sharing controls
  - Restrict access to a file in a way not provided in the standard.
- Putting strings on information
  - Maintain control over the user of the information even after releasing.

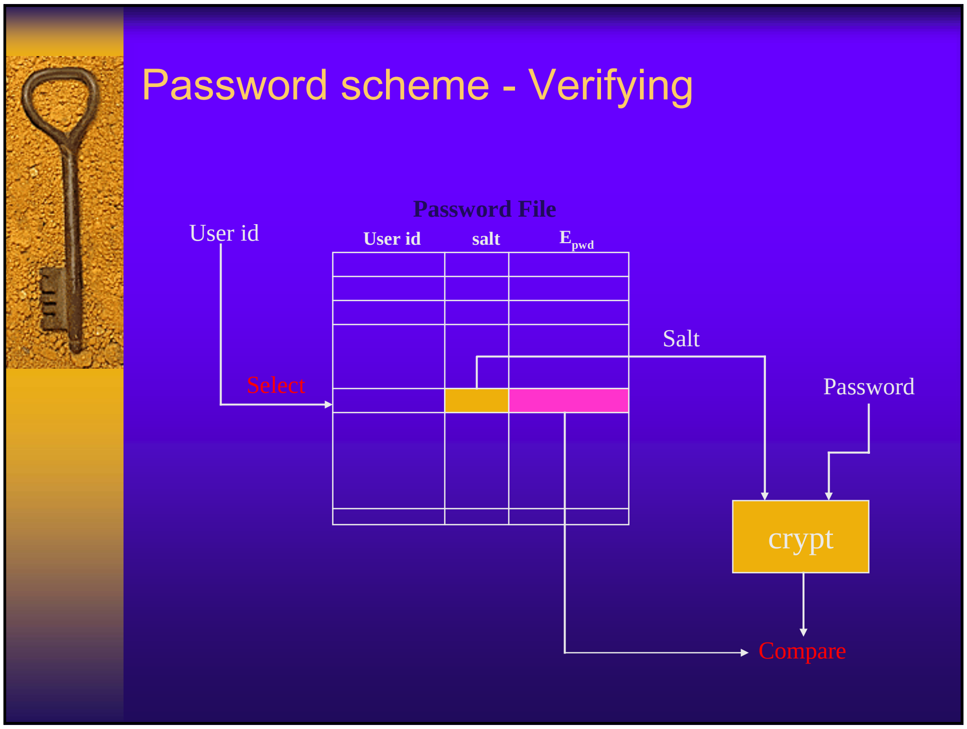


Design Principles

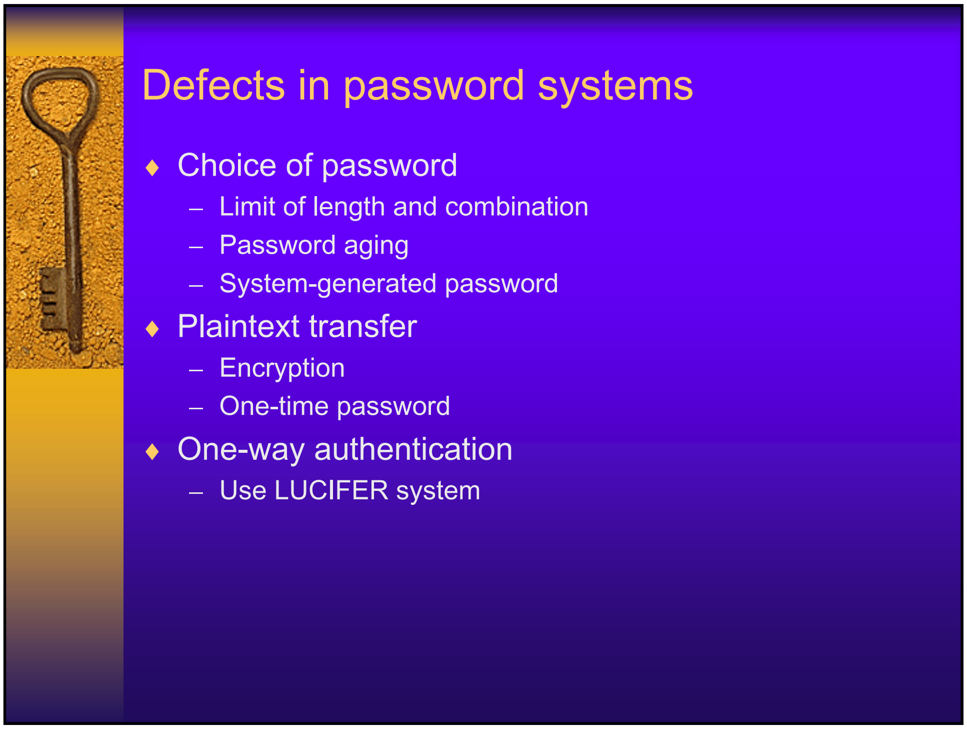
- Economy of mechanism
- Fail-safe defaults
- Complete mediation
- Open design
- Separation of privilege
- Least privilege
- Least common mechanism
- User friendly interface



Password scheme - Loading

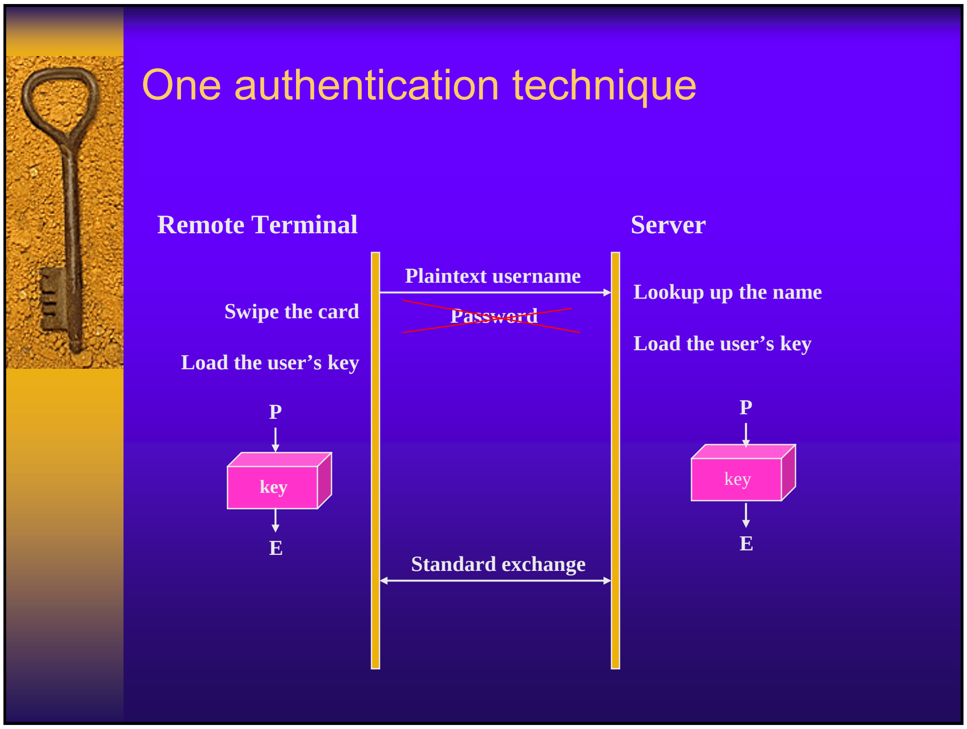


Password scheme - Verifying



Defects in password systems

- Choice of password
  - Limit of length and combination
  - Password aging
  - System-generated password
- Plaintext transfer
  - Encryption
  - One-time password
- One-way authentication
  - Use LUCIFER system



One authentication technique

Remote Terminal

Server

Swipe the card

Load the user's key

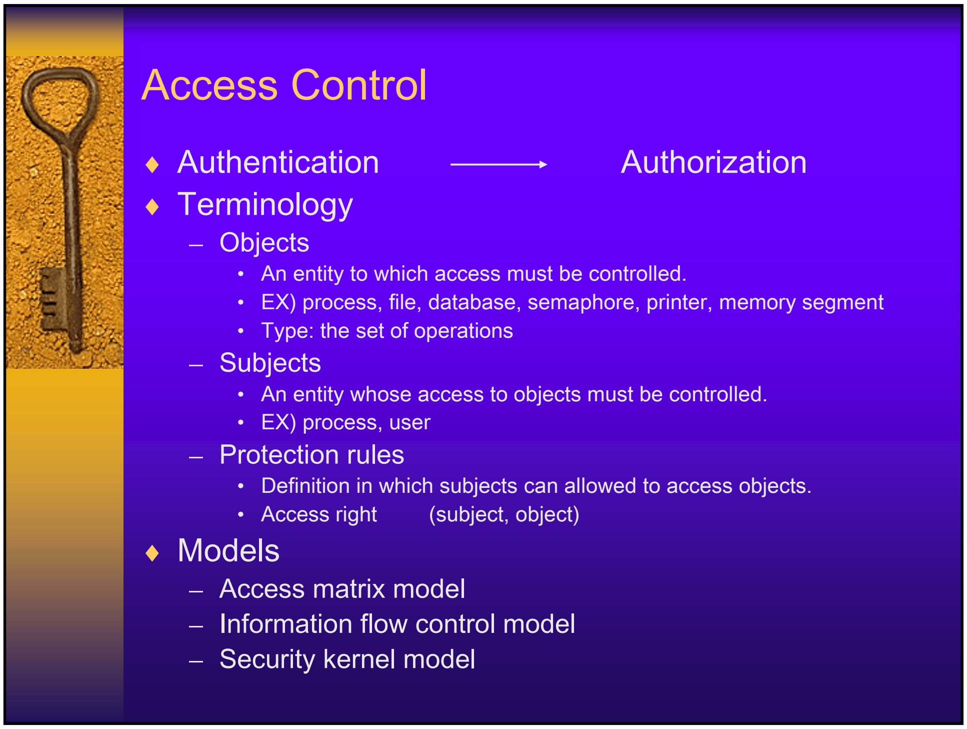
Plaintext username

~~Password~~

Lookup up the name

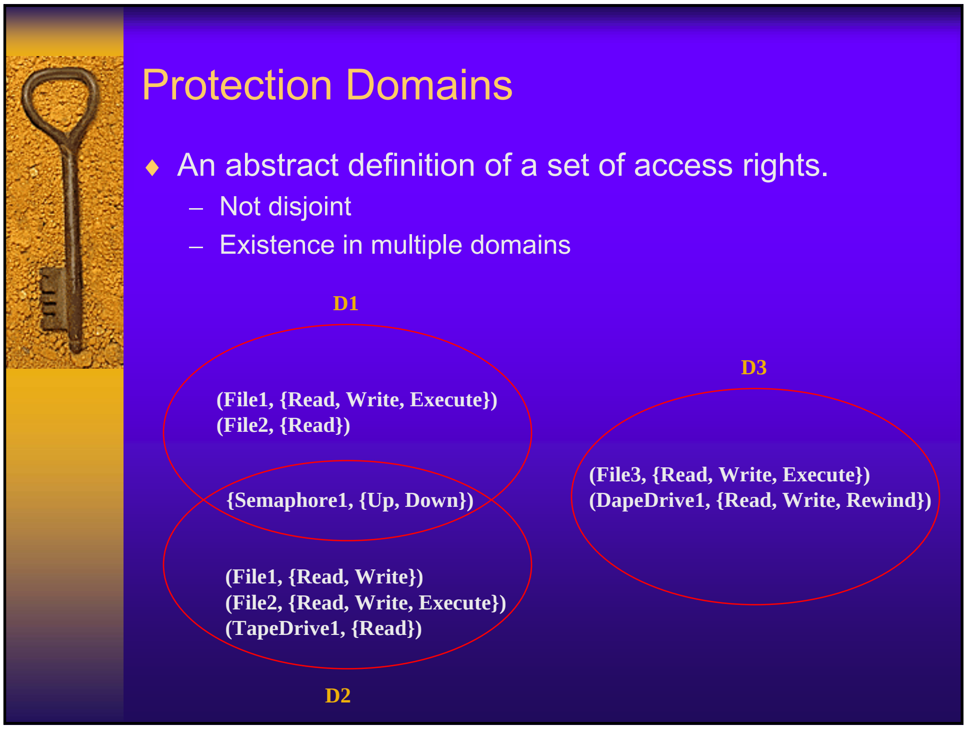
Load the user's key

Standard exchange



Access Control

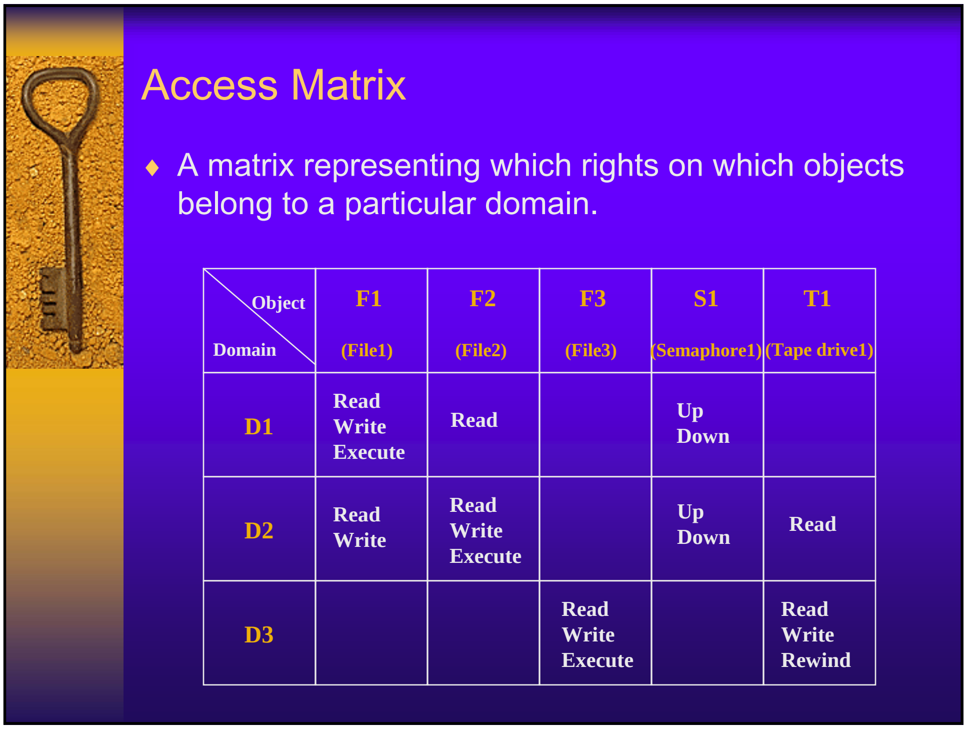
- Authentication ⟶ Authorization
- Terminology
  - Objects
    - An entity to which access must be controlled.
    - EX) process, file, database, semaphore, printer, memory segment
    - Type: the set of operations
  - Subjects
    - An entity whose access to objects must be controlled.
    - EX) process, user
  - Protection rules
    - Definition in which subjects can allowed to access objects.
    - Access right (subject, object)
- Models
  - Access matrix model
  - Information flow control model
  - Security kernel model



Protection Domains

- An abstract definition of a set of access rights.
  - Not disjoint
  - Existence in multiple domains

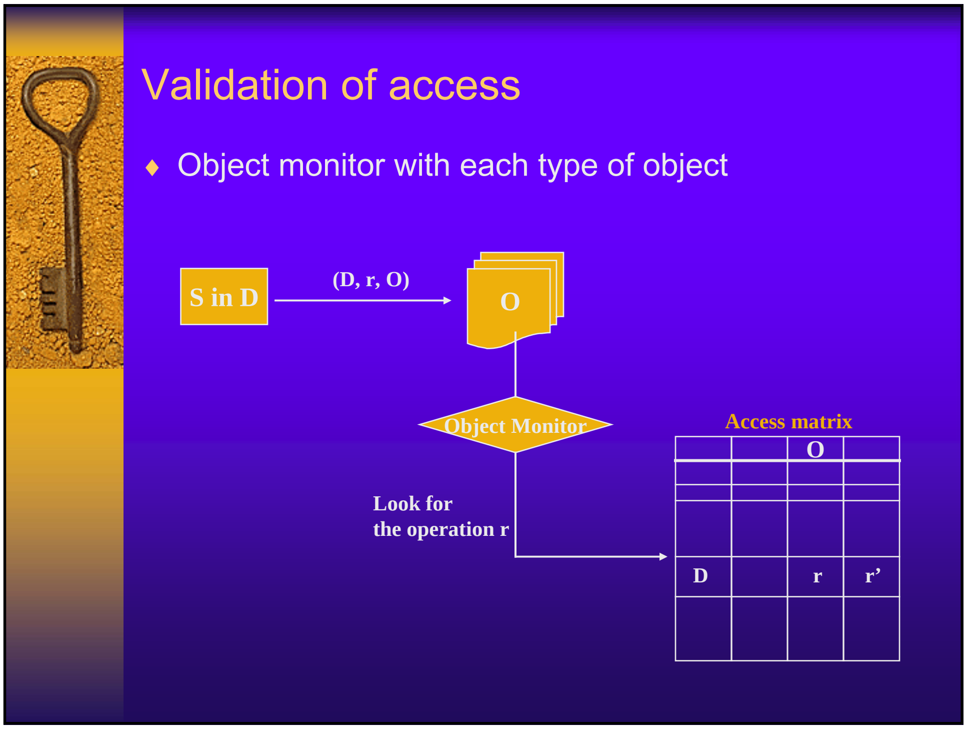
D1
(File1, {Read, Write, Execute})
(File2, {Read})
{Semaphore1, {Up, Down})
(File1, {Read, Write})
(File2, {Read, Write, Execute})
(TapeDrive1, {Read})
D2
D3
(File3, {Read, Write, Execute})
(DapeDrive1, {Read, Write, Rewind})



Access Matrix

- A matrix representing which rights on which objects belong to a particular domain.

Object Domain	F1 (File1)	F2 (File2)	F3 (File3)	S1 (Semaphore1)	T1 (Tape drive1)
D1	Read Write Execute	Read		Up Down	
D2	Read Write	Read Write Execute		Up Down	Read
D3			Read Write Execute		Read Write Rewind



Validation of access

- Object monitor with each type of object

S in D

(D, r, O)

O

Object Monitor

Look for
the operation r

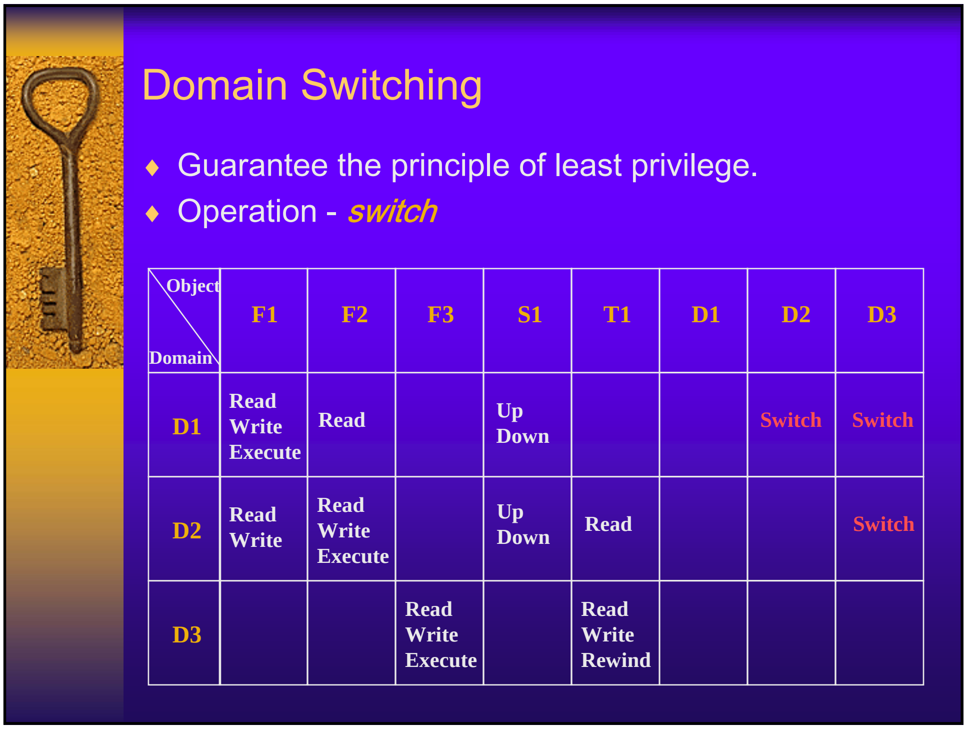
Access matrix

O

D

r

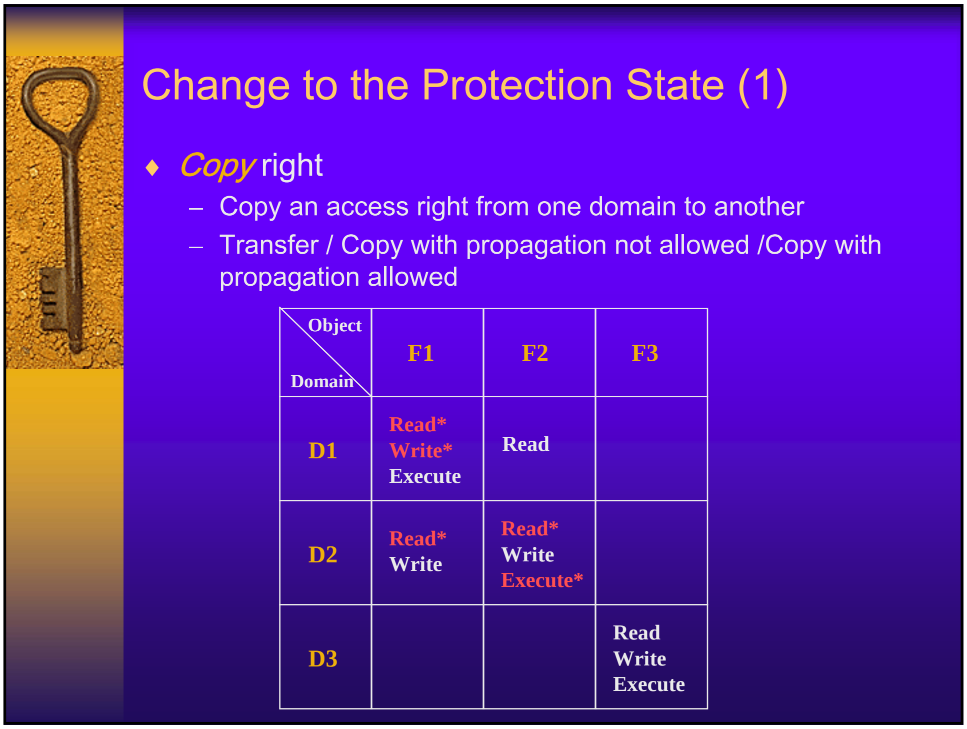
r'



Domain Switching

- Guarantee the principle of least privilege.
- Operation - *switch*

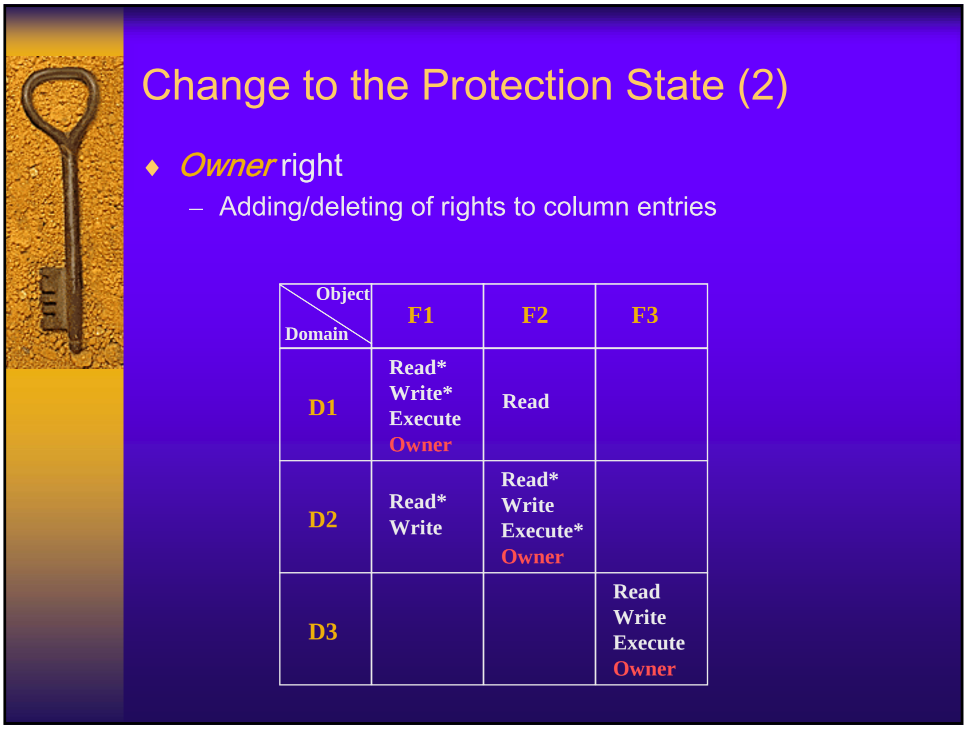
Object / Domain	F1	F2	F3	S1	T1	D1	D2	D3
D1	Read Write Execute	Read		Up Down			Switch	Switch
D2	Read Write	Read Write Execute		Up Down	Read			Switch
D3			Read Write Execute		Read Write Rewind			



Change to the Protection State (1)

- *Copy* right
  - Copy an access right from one domain to another
  - Transfer / Copy with propagation not allowed /Copy with propagation allowed

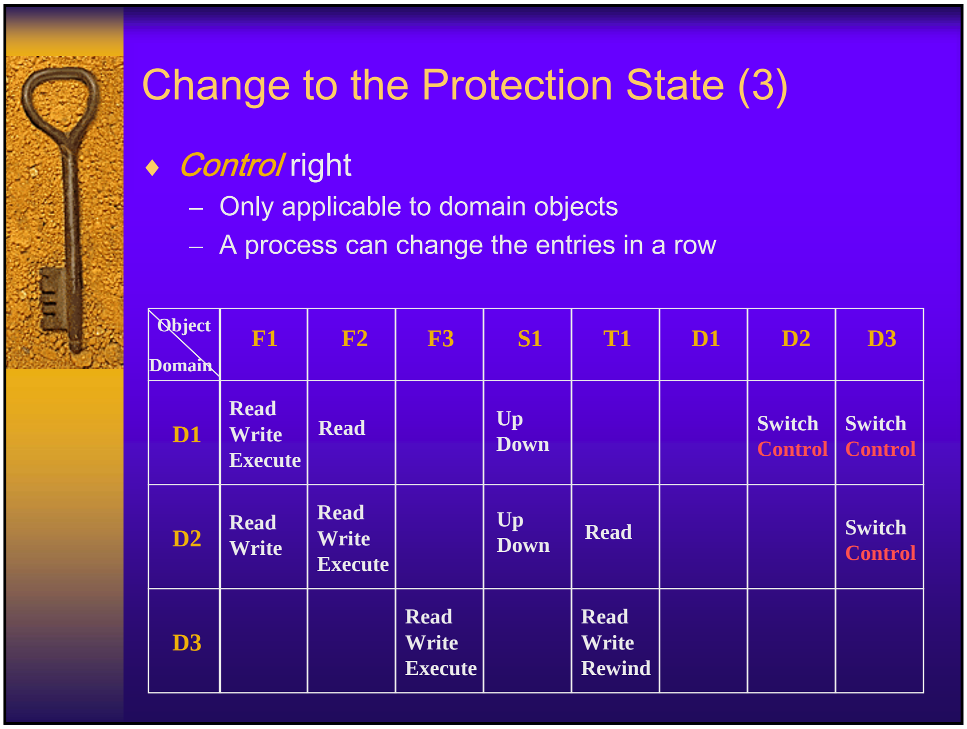
Object / Domain	F1	F2	F3
D1	Read* Write* Execute	Read	
D2	Read* Write	Read* Write Execute*	
D3			Read Write Execute



Change to the Protection State (2)

- *Owner* right
  - Adding/deleting of rights to column entries

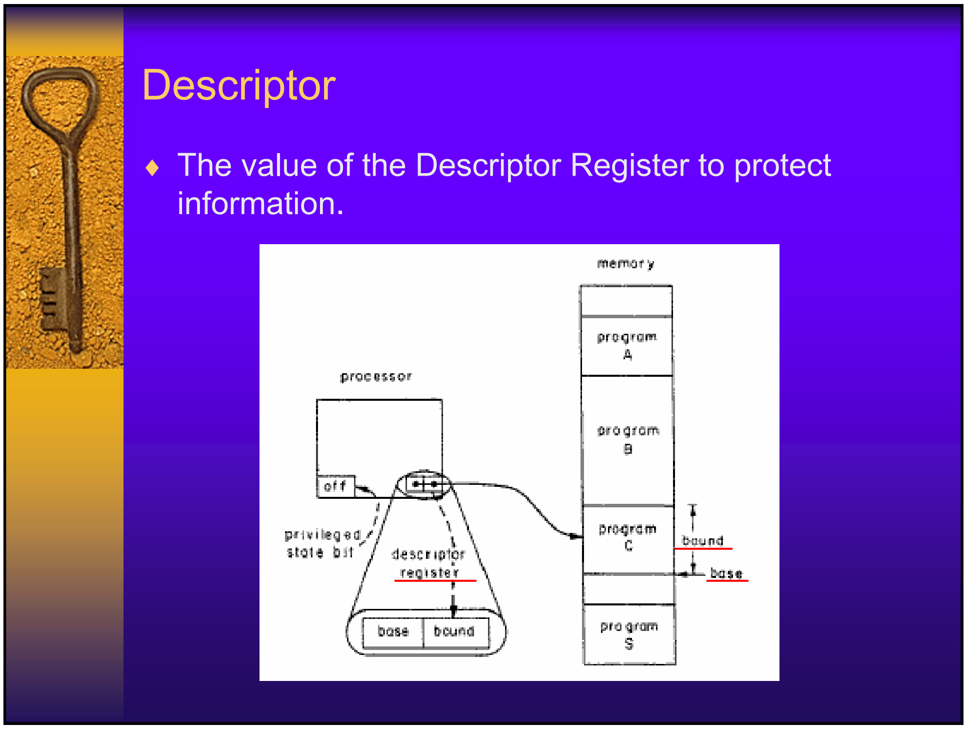
Object / Domain	F1	F2	F3
D1	Read* Write* Execute Owner	Read	
D2	Read* Write	Read* Write Execute* Owner	
D3			Read Write Execute Owner



Change to the Protection State (3)

- *Control* right
  - Only applicable to domain objects
  - A process can change the entries in a row

Object Domain	F1	F2	F3	S1	T1	D1	D2	D3
D1	Read Write Execute	Read		Up Down			Switch Control	Switch Control
D2	Read Write	Read Write Execute		Up Down	Read			Switch Control
D3			Read Write Execute		Read Write Rewind			

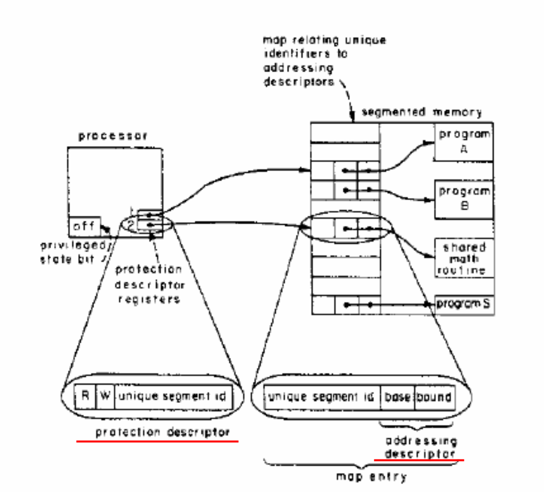


Descriptor

- The value of the Descriptor Register to protect information.

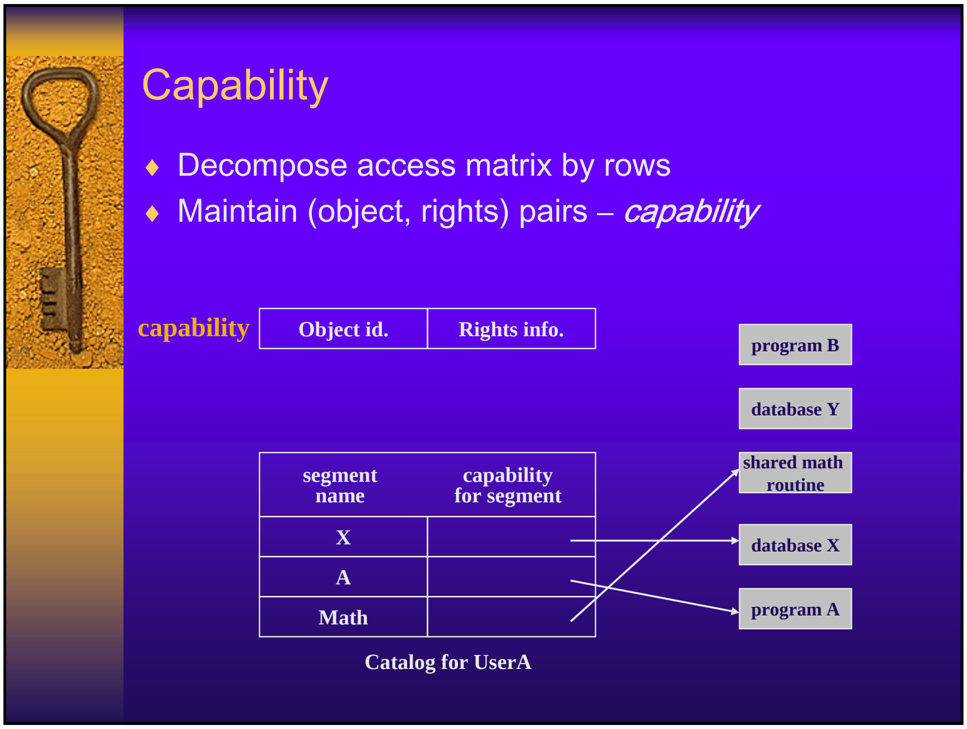
Separation of Addressing and Protection

- All memory accesses were divided into two levels of descriptors → *protection* and *addressing*



Approaches

- Concept
- Validation
- Sharing
- Revocation



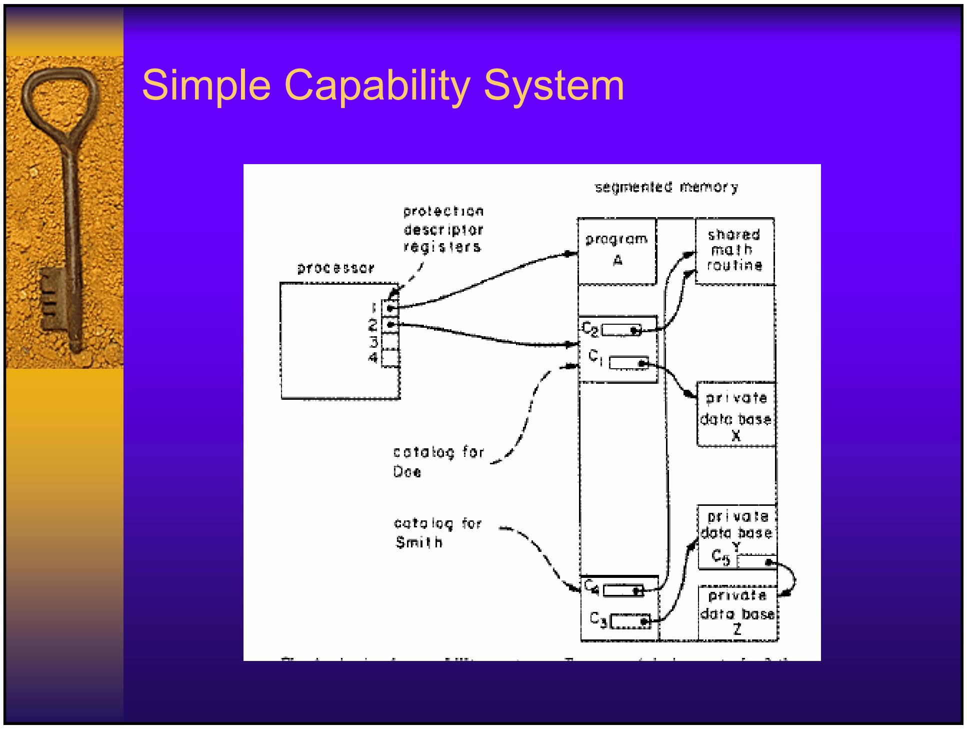
Capability

- Decompose access matrix by rows
- Maintain (object, rights) pairs – *capability*

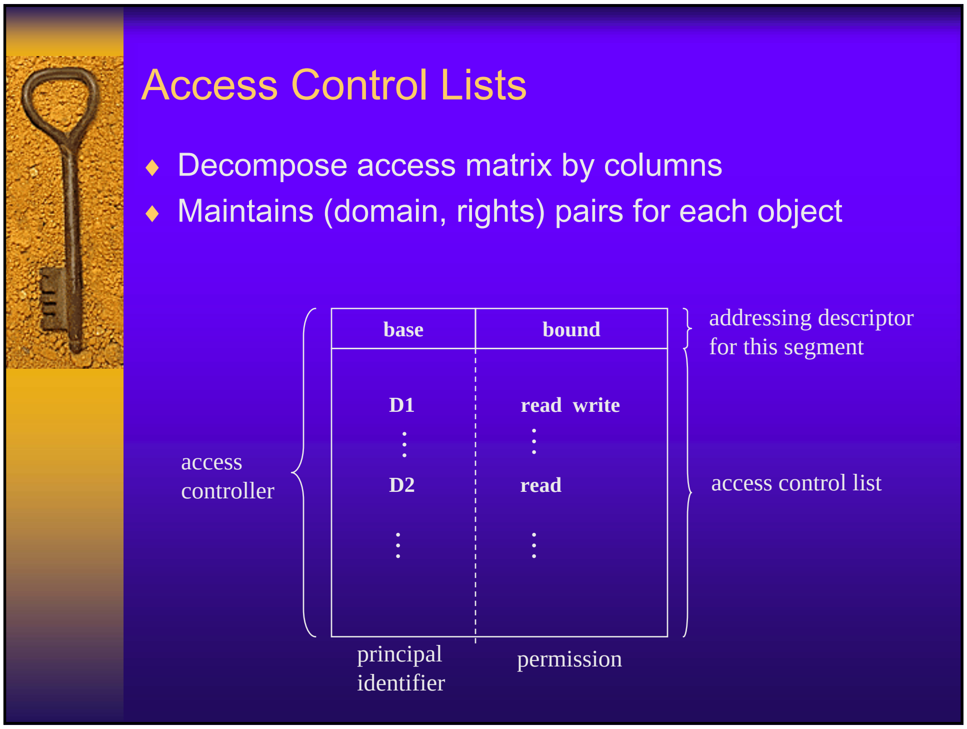
capability	Object id.	Rights info.

segment name	capability for segment
X	
A	
Math	

Catalog for UserA



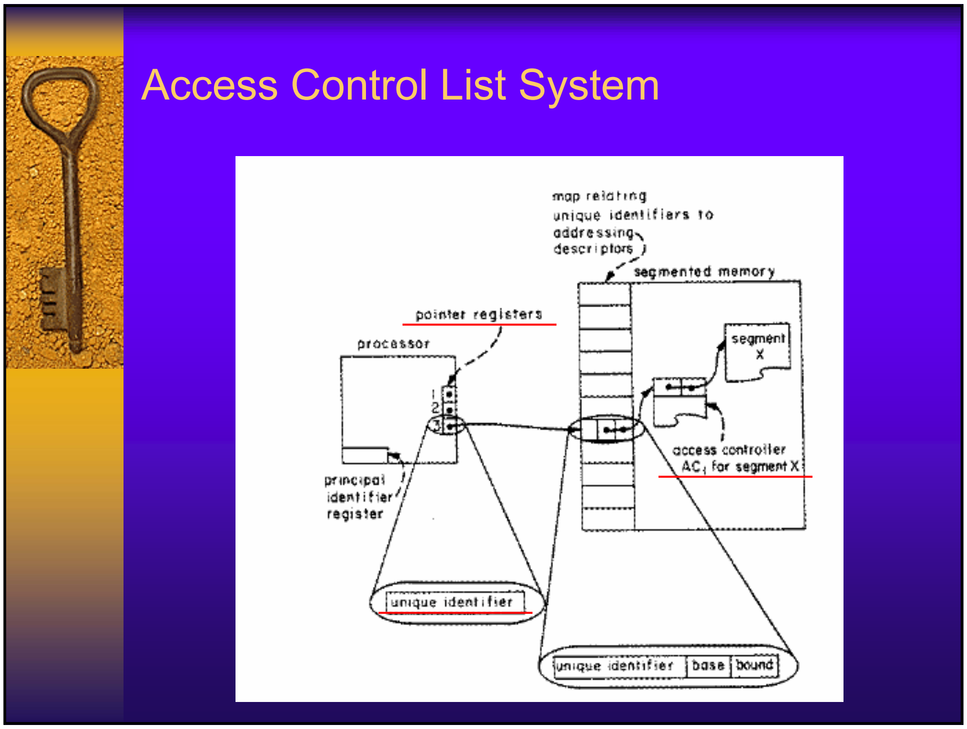
Simple Capability System



Access Control Lists

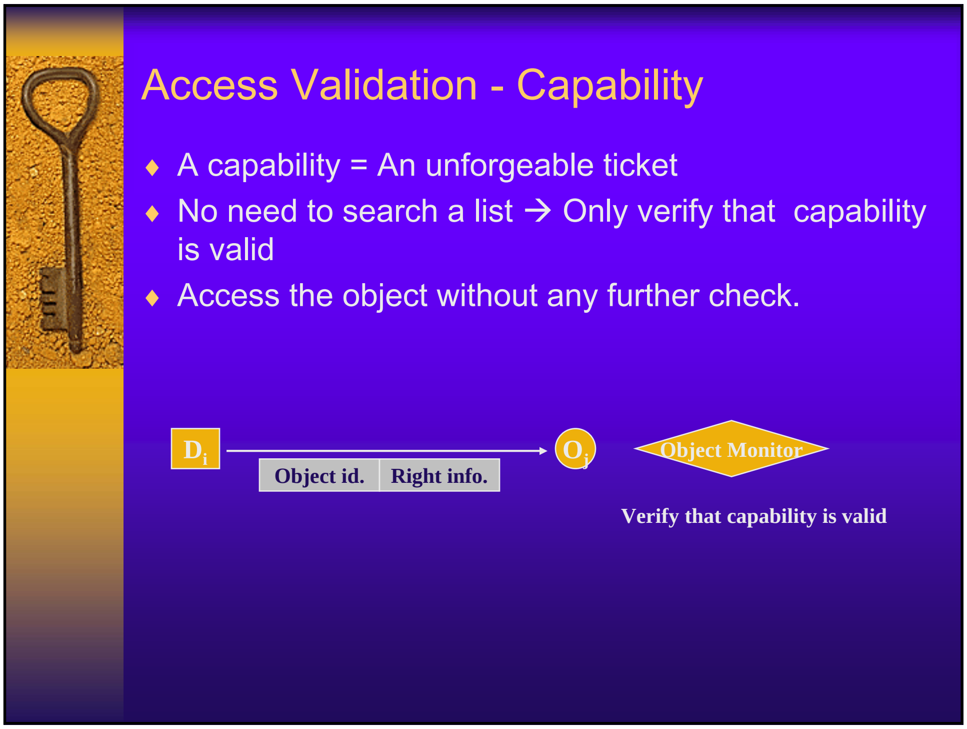
- Decompose access matrix by columns
- Maintains (domain, rights) pairs for each object

base	bound
D1	read write
⋮	⋮
D2	read
⋮	⋮



Access Control List System

map relating unique identifiers to addressing descriptors
segmented memory
pointer registers
processor
1
2
3
segment X
access controller AC_1 for segment X
principal identifier register
unique identifier
unique identifier
base
bound



Access Validation - Capability

- A capability = An unforgeable ticket
- No need to search a list → Only verify that capability is valid
- Access the object without any further check.

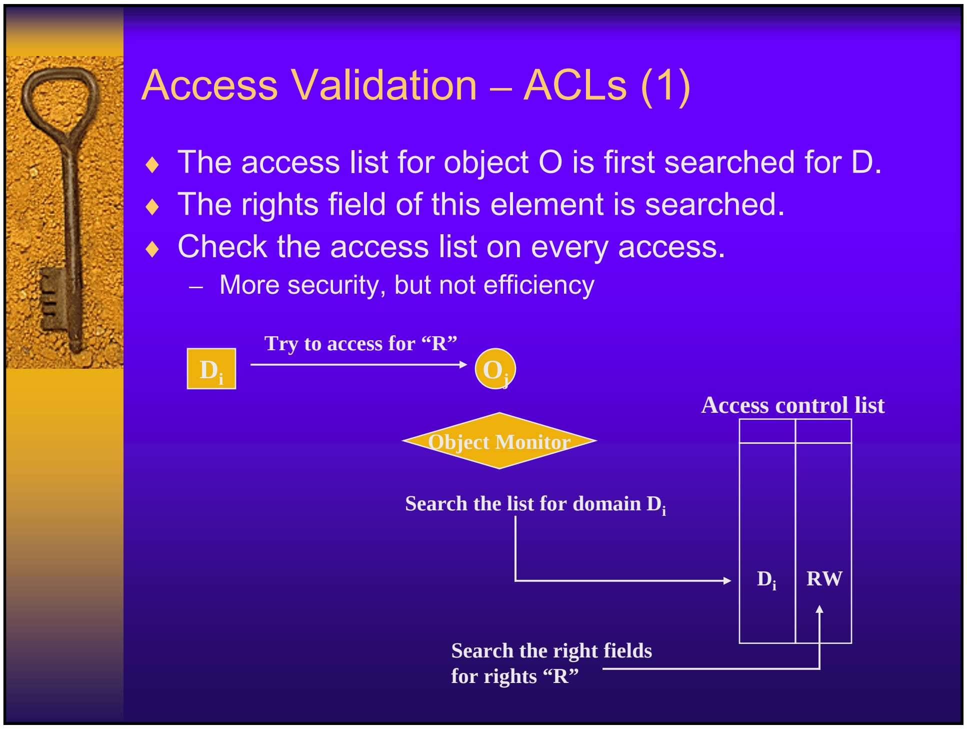
D_i

Object id. | Right info.

O_j

Object Monitor

Verify that capability is valid



Access Validation – ACLs (1)

- The access list for object O is first searched for D.
- The rights field of this element is searched.
- Check the access list on every access.
  - More security, but not efficiency

D_i

Try to access for "R"

O_j

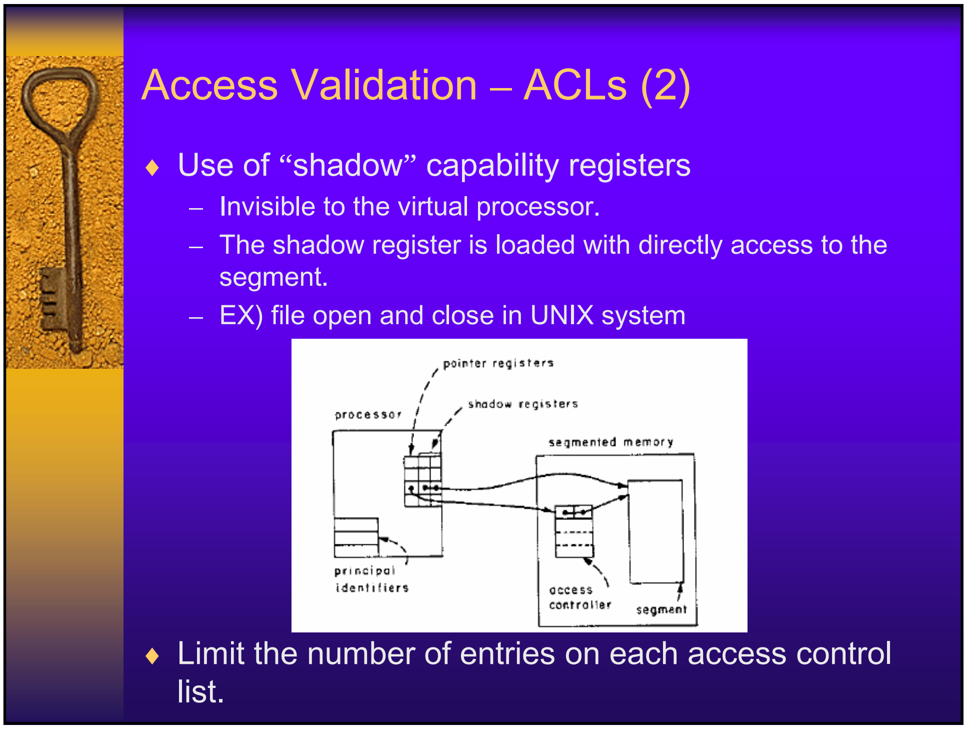
Object Monitor

Search the list for domain D_i

Access control list

D_i	RW

Search the right fields for rights "R"

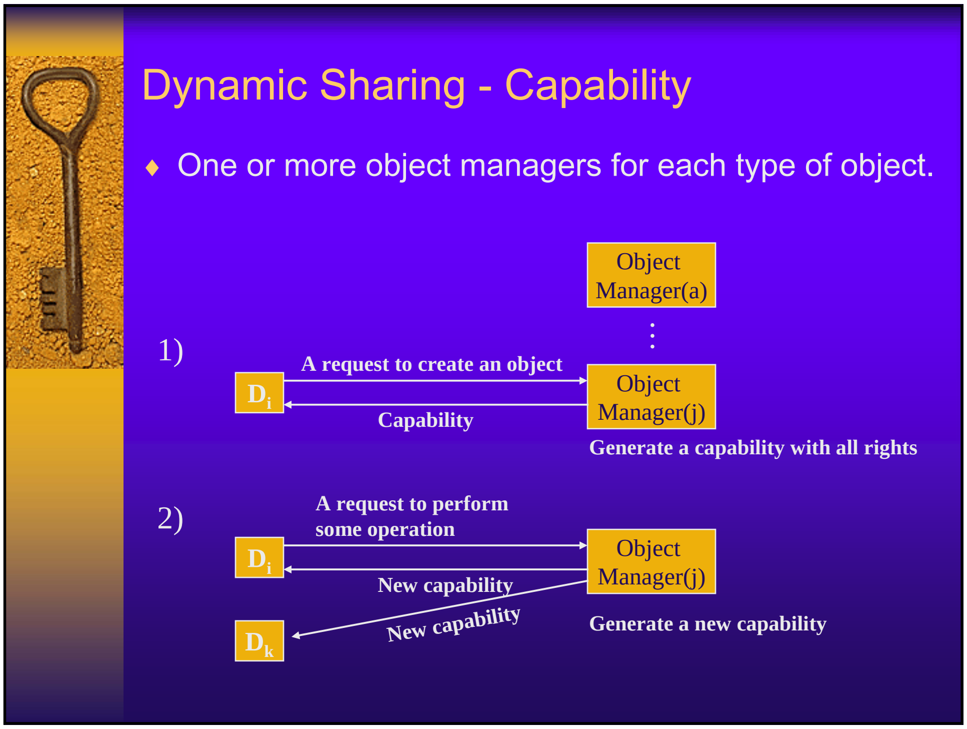


Access Validation – ACLs (2)

- Use of “shadow” capability registers
  - Invisible to the virtual processor.
  - The shadow register is loaded with directly access to the segment.
  - EX) file open and close in UNIX system

pointer registers
shadow registers
processor
segmented memory
principal identifiers
access controller
segment

- Limit the number of entries on each access control list.



Dynamic Sharing - Capability

- One or more object managers for each type of object.

Object Manager(a)

⋮

1) D_i — A request to create an object → Object Manager(j)

Capability

Generate a capability with all rights

2) D_i — A request to perform some operation → Object Manager(j)

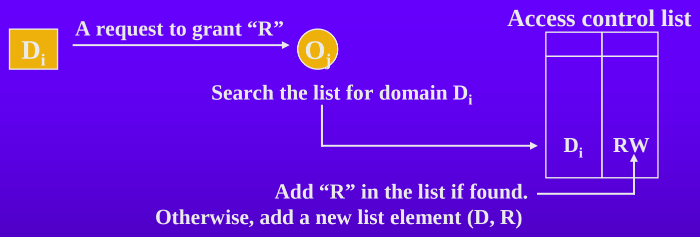
New capability

D_k — New capability

Generate a new capability

Dynamic Sharing – ACLs (1)

- To grant access right *r* for object O to domain D



D_i

A request to grant "R"

O_j

Access control list

Search the list for domain D_i

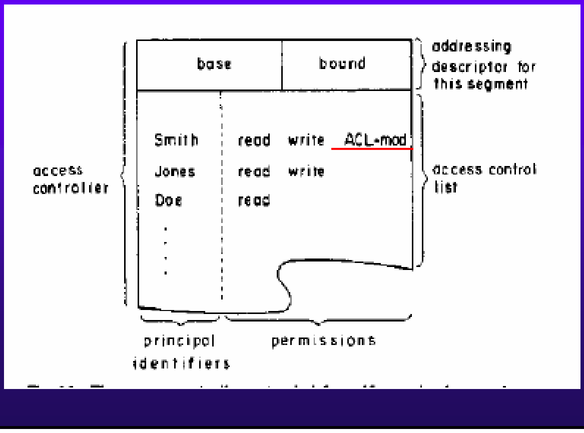
D_i RW

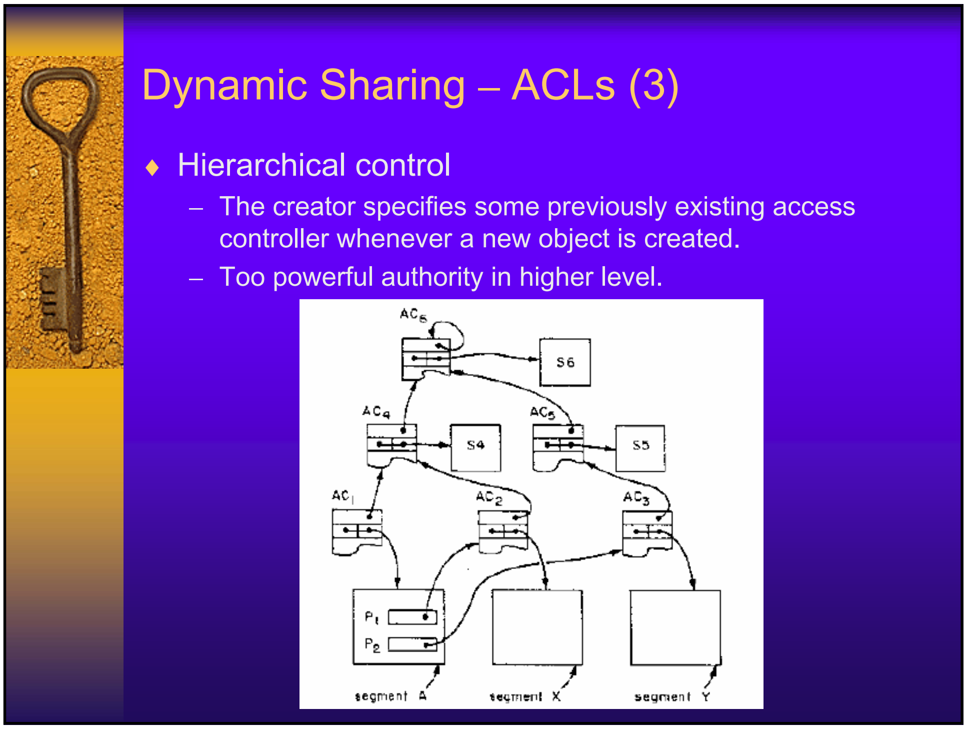
Add "R" in the list if found.
Otherwise, add a new list element (D, R)

- To pass access right *r* from a domain D1 to another domain D2
  - Check if D1 possesses either *owner* right or *copy* right for access right *r*.

Dynamic Sharing – ACLs (2)

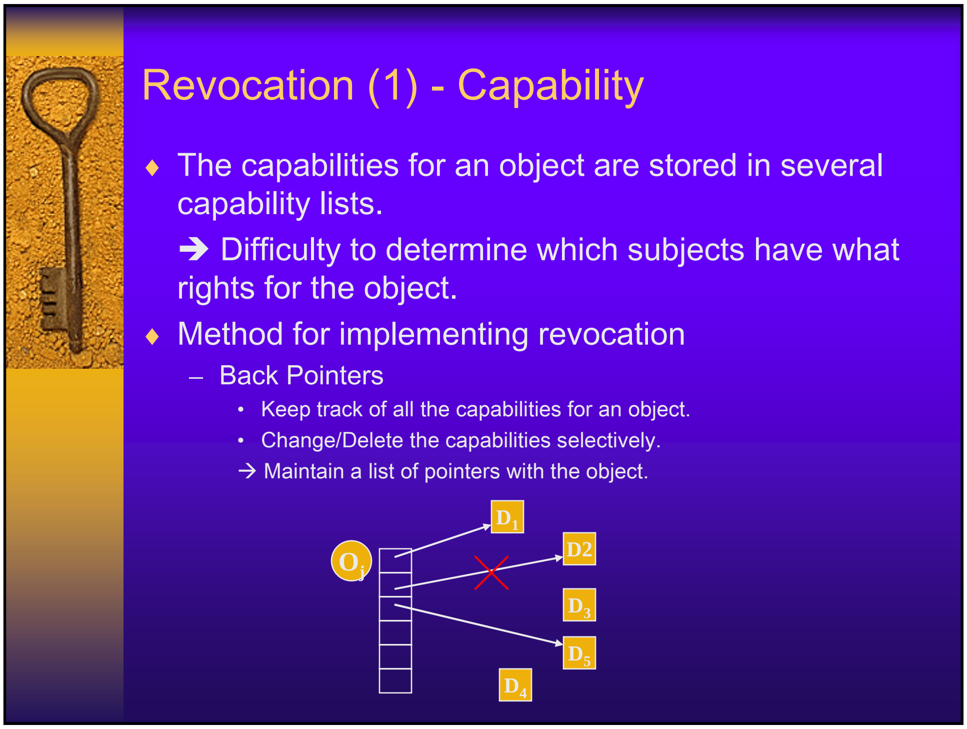
- Self control
  - Permission to modify the access control list.
  - Too absolute – no provision for another way to control.





Dynamic Sharing – ACLs (3)

- Hierarchical control
  - The creator specifies some previously existing access controller whenever a new object is created.
  - Too powerful authority in higher level.

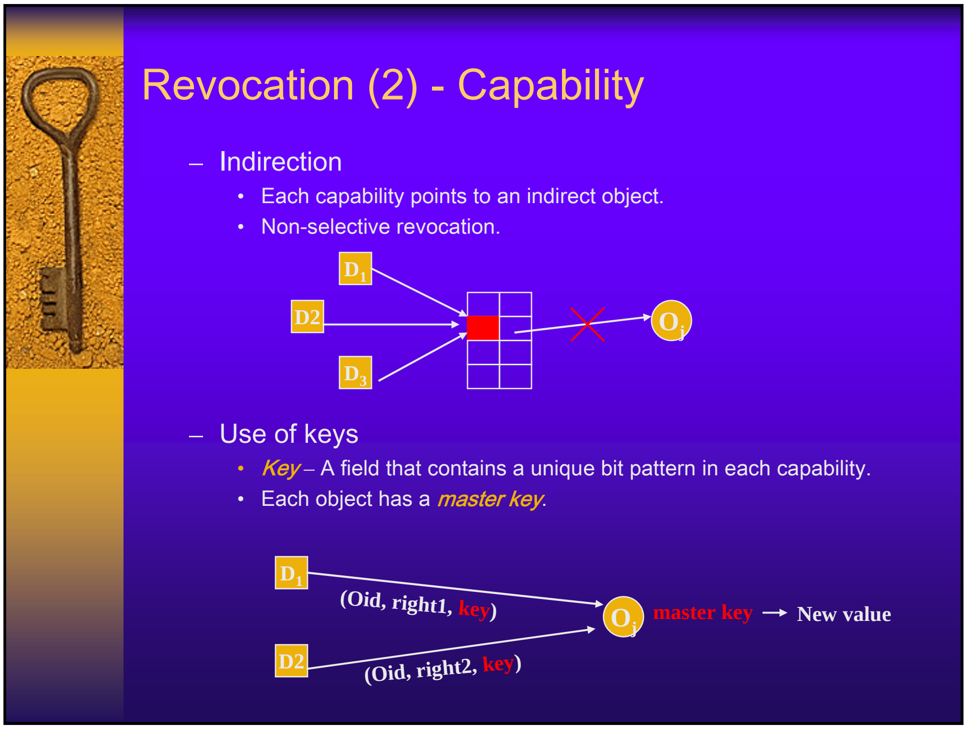


Revocation (1) - Capability

- The capabilities for an object are stored in several capability lists.

  ➔ Difficulty to determine which subjects have what rights for the object.
- Method for implementing revocation
  - Back Pointers
    - Keep track of all the capabilities for an object.
    - Change/Delete the capabilities selectively.

    → Maintain a list of pointers with the object.



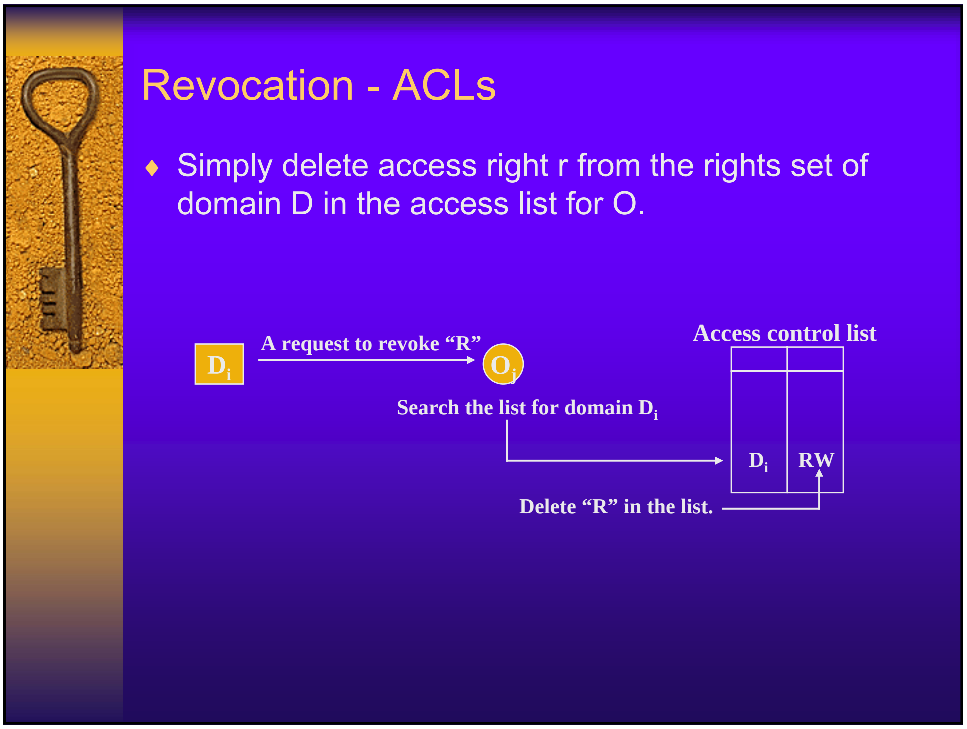
Revocation (2) - Capability

- Indirection
  - Each capability points to an indirect object.
  - Non-selective revocation.

D_1
D2
D_3
O_j

- Use of keys
  - *Key* – A field that contains a unique bit pattern in each capability.
  - Each object has a *master key*.

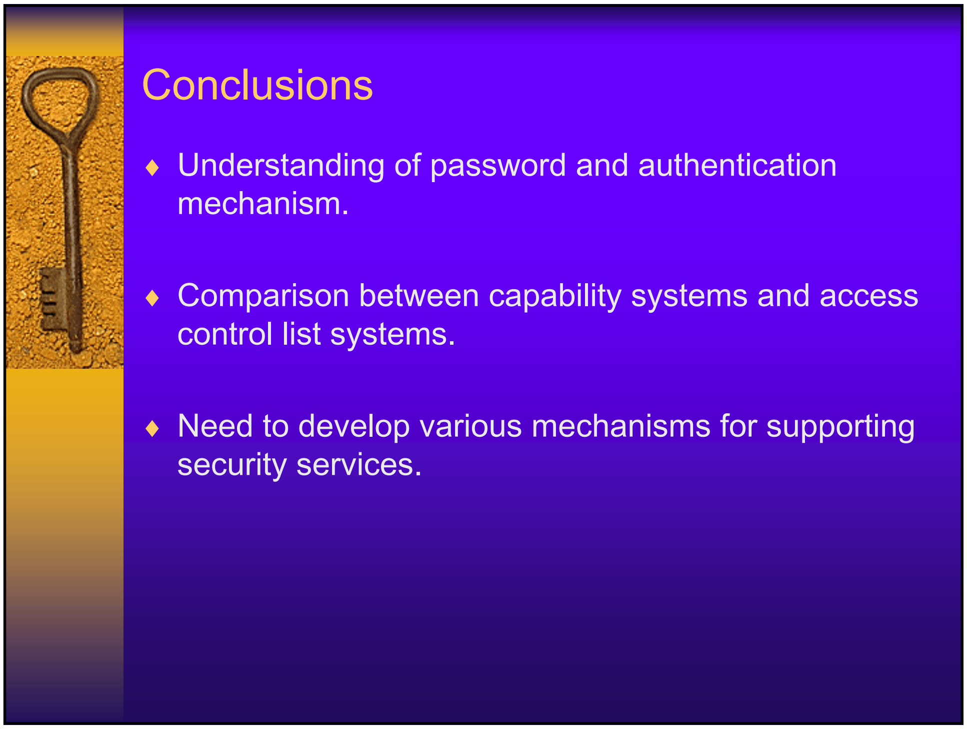
D_1
(Oid, right1, key)
D2
(Oid, right2, key)
O_j
master key → New value



Revocation - ACLs

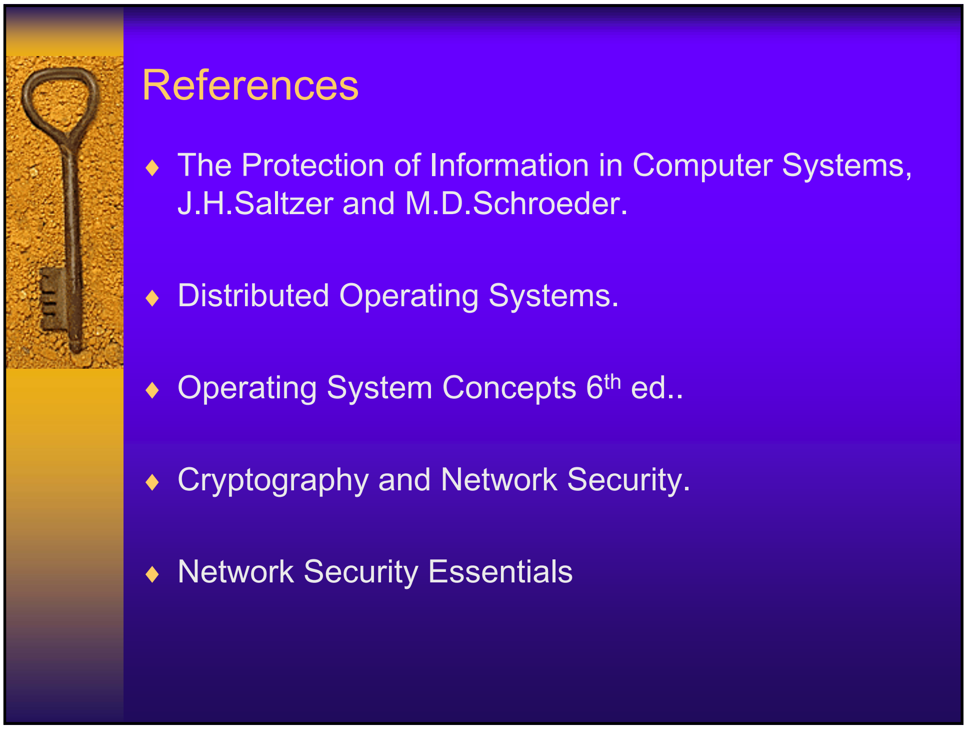
- Simply delete access right r from the rights set of domain D in the access list for O.

D_i
A request to revoke “R”
O_j
Search the list for domain D_i
Access control list
D_i
RW
Delete “R” in the list.



Conclusions

- Understanding of password and authentication mechanism.
- Comparison between capability systems and access control list systems.
- Need to develop various mechanisms for supporting security services.



References

- The Protection of Information in Computer Systems, J.H.Saltzer and M.D.Schroeder.
- Distributed Operating Systems.
- Operating System Concepts 6th ed..
- Cryptography and Network Security.
- Network Security Essentials