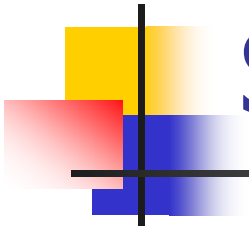
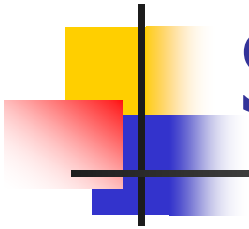


CS423 Operating Systems Security



Security

- What are examples of security goals?

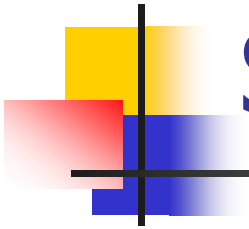


Security

- Data confidentiality: Preventing data exposure
- Data integrity: Preventing tempering with data
- System availability: Preventing denial of service
- Exclusion of outsiders: Preventing un-authorized use

Intruders

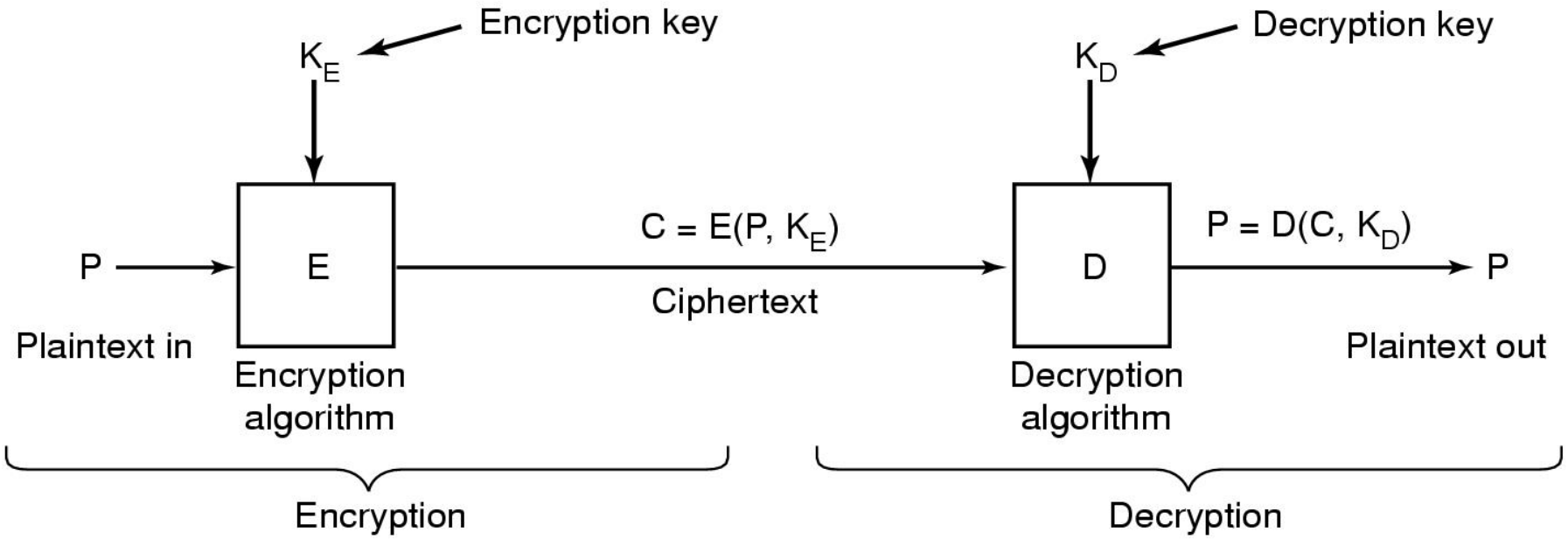
- Casual prying
- Insider snooping
- Financial attacks (e.g., programmers for a bank add code to transfer to their account)
- Commercial/military espionage
- Viruses, etc.



Security Attacks

- Authentication
  - E.g., a rogue program, pretending to be the sign-on program, asking for the user's password and then storing it.
- Buffer overflow
  - Give input to a program that will overflow a buffer. If no checks are made, buffer may overwrite stack (including return address) and cause jump to virus code (part of input)
- Operator carelessness
  - E.g., tricking the operator into executing a virus program (such as one named "ls" in a user's directory)
- Residue
  - Interesting information often turns up in wastebaskets; use paper shredders! Information is often left in central memory from a previous user, possibly a system routine; variables that contain sensitive information should be overwritten before they are deallocated!
- Shielding
  - One can inductively ``tap'' a cable, phone line, or in fact any wire over which information passes, without making physical connection to it. Electrical shielding can protect against this.
- Passwords
  - Password guessing, etc.

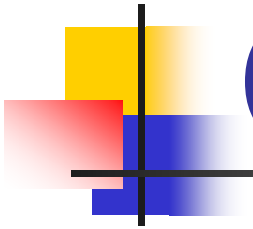
Cryptography (Data Confidentiality)



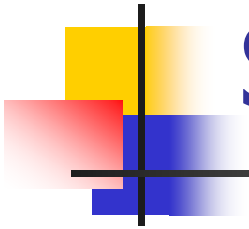
Relationship between the plaintext and the ciphertext.

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

Cryptography (Data Confidentiality)



- Symmetric key
  - Encryption key and decryption key
  - Easy to deduce one from the other
  - Think: character substitution
  - Need to share a secret between two nodes
- Public key
  - Encryption key is made publicly known
  - Decryption key is kept secret
  - Impossible to deduce decryption key from encryption key (one way function)
    - Example?



Symmetric Key Cryptography

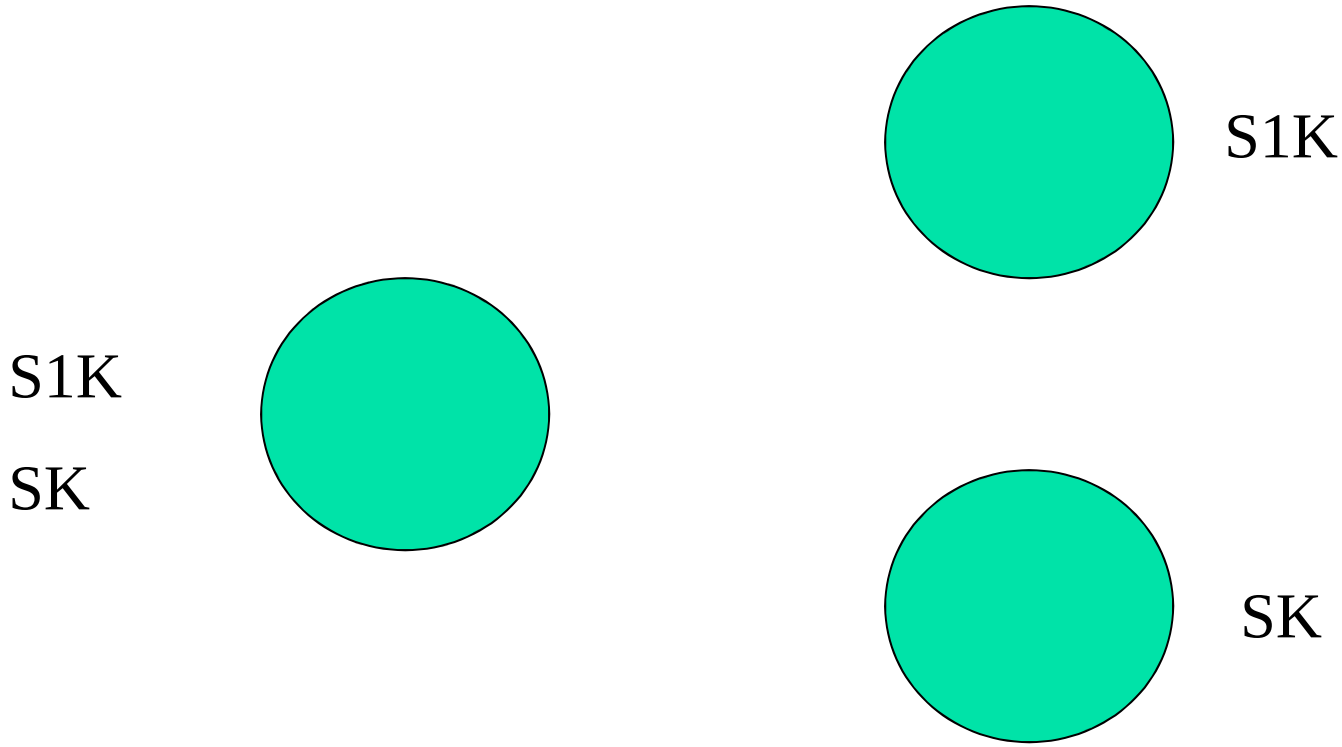
Example: Monoalphabetic substitution:

Plaintext: `ABCDEFGHIJKLMNOPQRSTUVWXYZ`

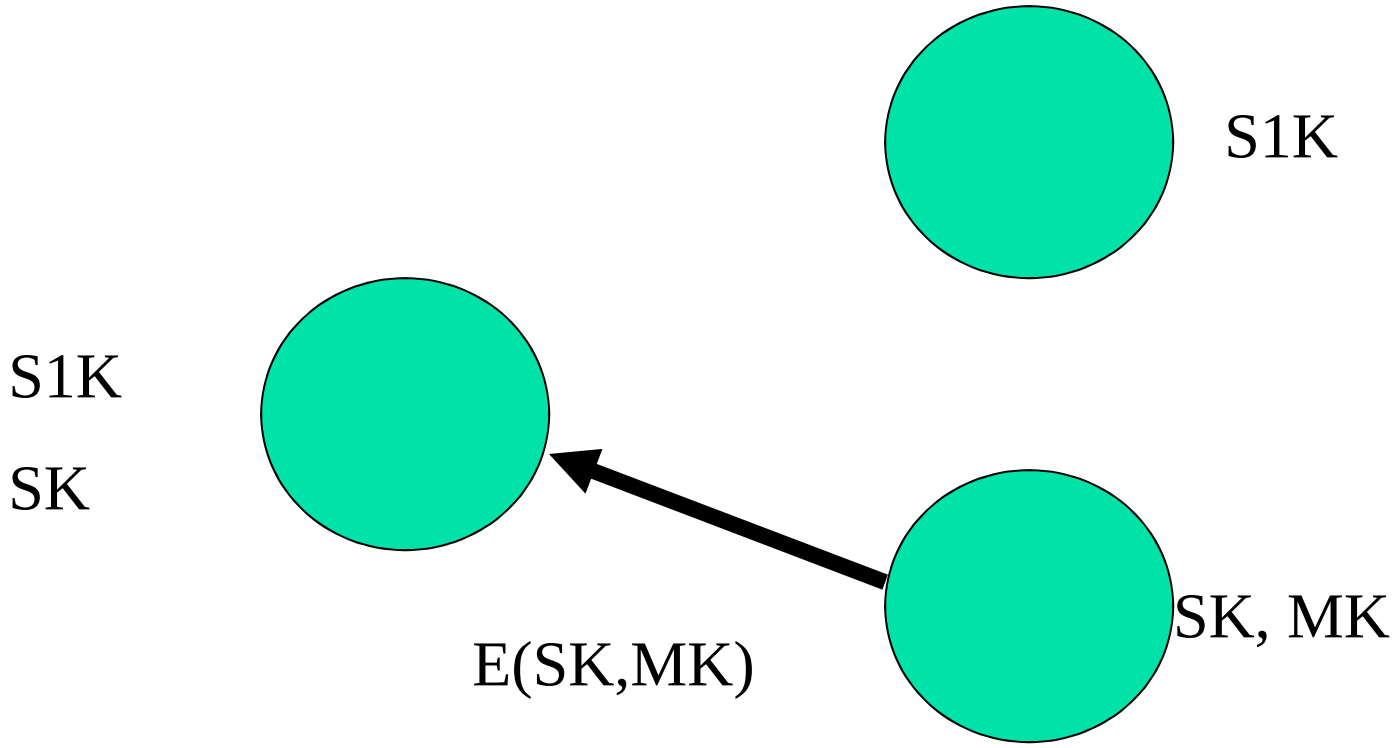
Ciphertext: `QWERTYUIOPASDFGHJKLZXCVBNM`

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**

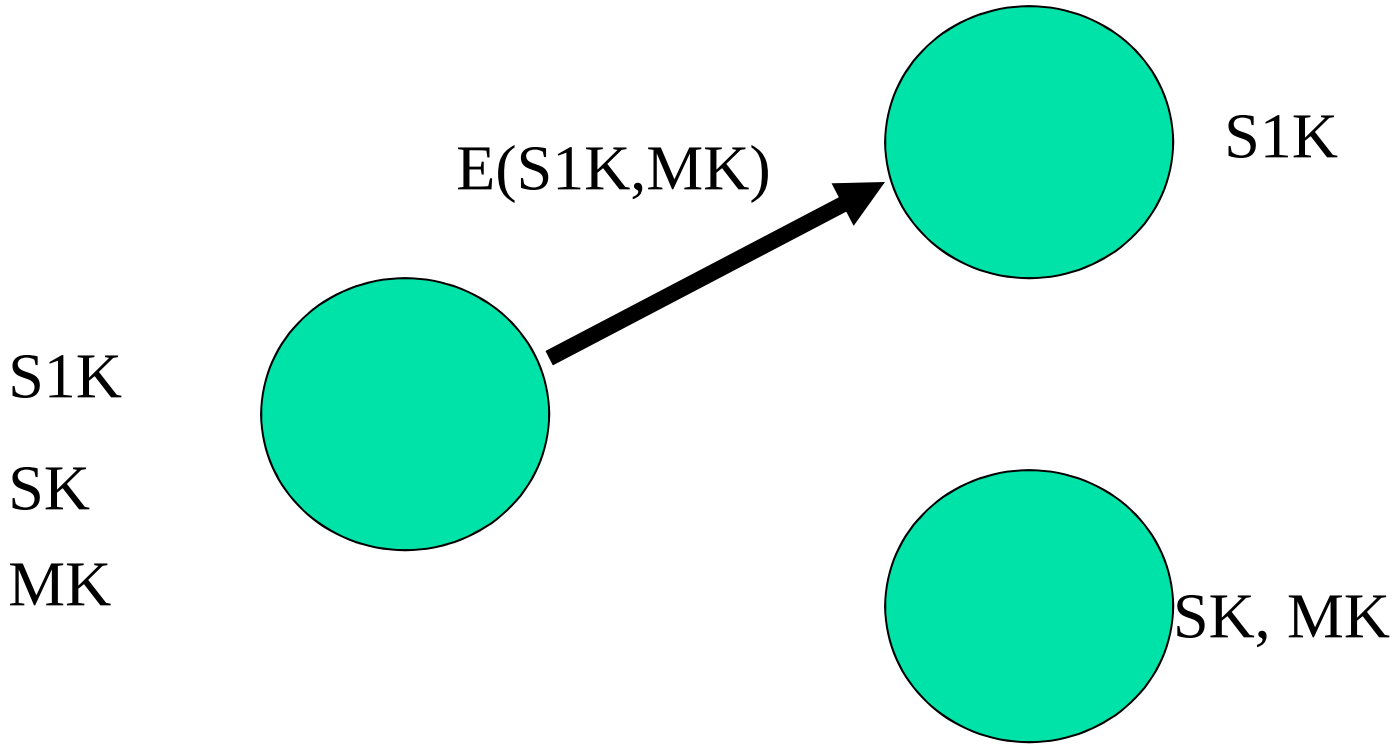
Distribution of Keys using Symmetric Keys



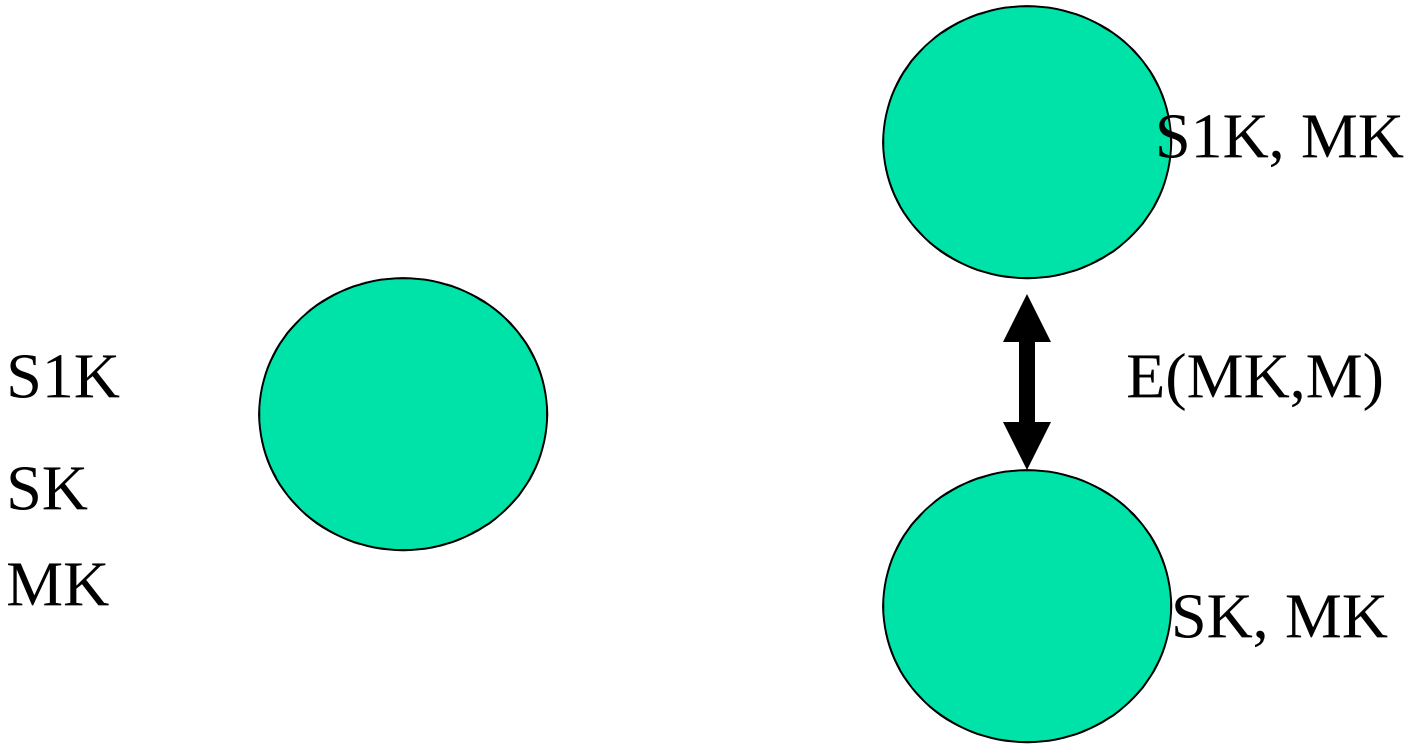
Distribution of Keys using Symmetric Keys



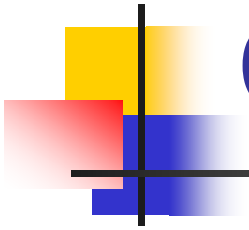
Distribution of Keys using Symmetric Keys



Distribution of Keys using Symmetric Keys



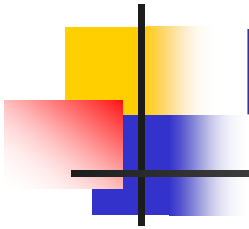
Public Key Cryptography: One Way Functions



Encryption makes use of an "easy" operation, such as how much is 314159265358979 × 314159265358979?

Decryption without the key requires you to perform a hard operation, such as what is the square root of 3912571506419387090594828508241?

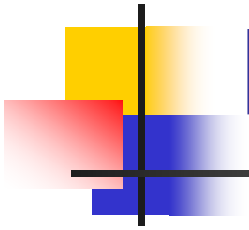
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**



RSA Public Key System

- Exploits the fact that multiplying very big numbers is easier than factoring the product.
- Private/Public key pairs:
  - Encrypt data for recipient using public key of recipient (which is a one-way function that makes it very hard to recover original data)
  - Only recipient can decrypt data using their private key

Rivest-Shamir-Adelman (RSA) Encryption



- RSA uses two keys: d and e with integer n.
  - pair (e,n) will be the public encryption key;
  - pair (d,n) will be the private key.
- Message m is encrypted as follows: E(m) = (m^e) mod n = C,
- Message m will be decrypted as follows: D(C) = (C^d) mod n

Digital Signatures (Data Integrity)

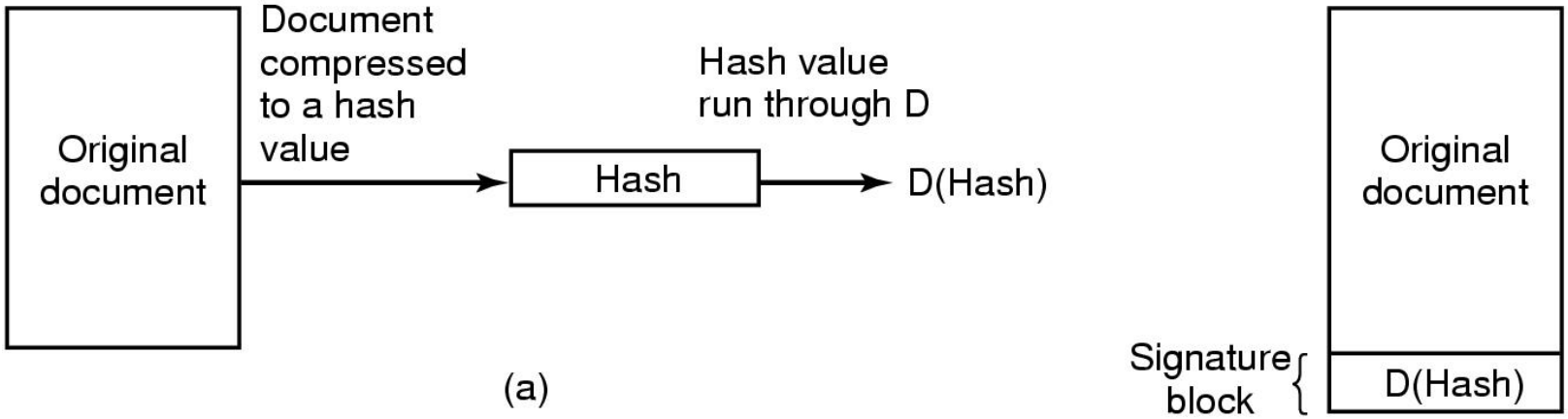
Sender

- Use a one-way function whose input is the message being signed and whose output is a digest (e.g., MD5 or SHA-1, SHA-256, …) or hash.
- Encrypt digest using owner's private key
  - Signature block

Receiver

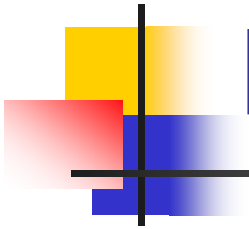
- Compute fixed length digest of the message
- Apply decryption key (sender's public key) to signature block and compare with digest

Digital Signatures (Data Integrity)



Computing a signature block.
What the receiver gets.

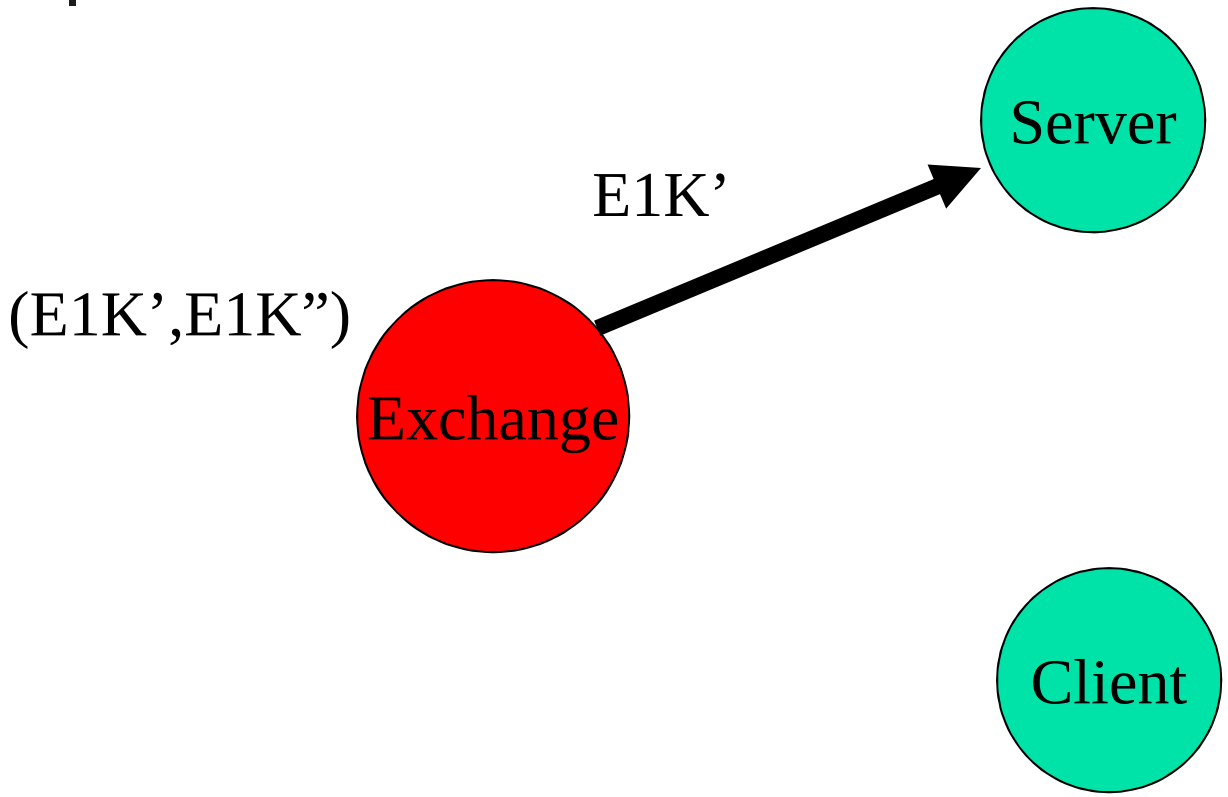
Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**



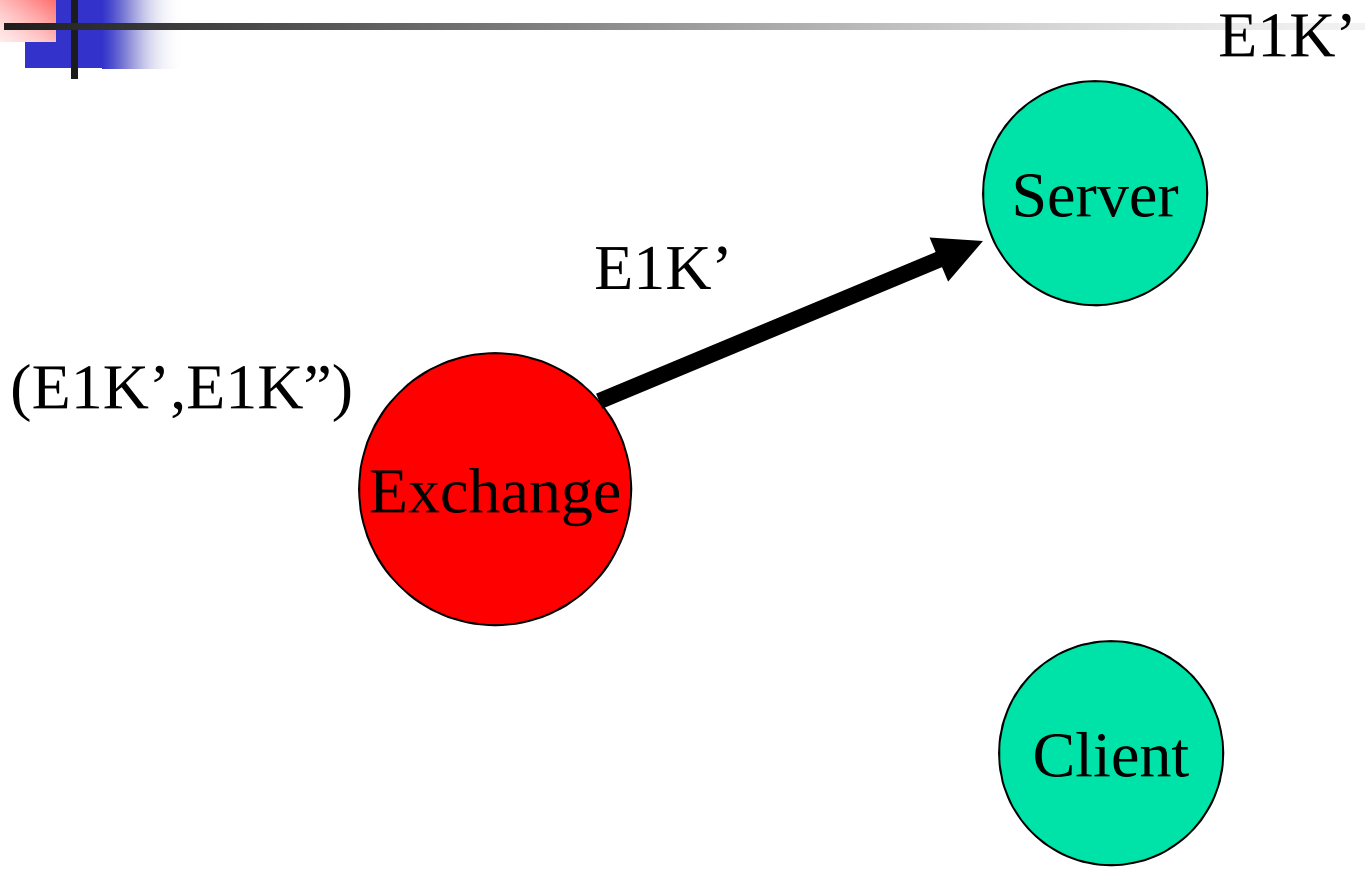
Distribution of Public Keys

- Uses certification authority (CA)
- A certificate is a user's ID and public key signed by the certification authority
- Browser's come preloaded with about 40 CA public key to decrypt certificates

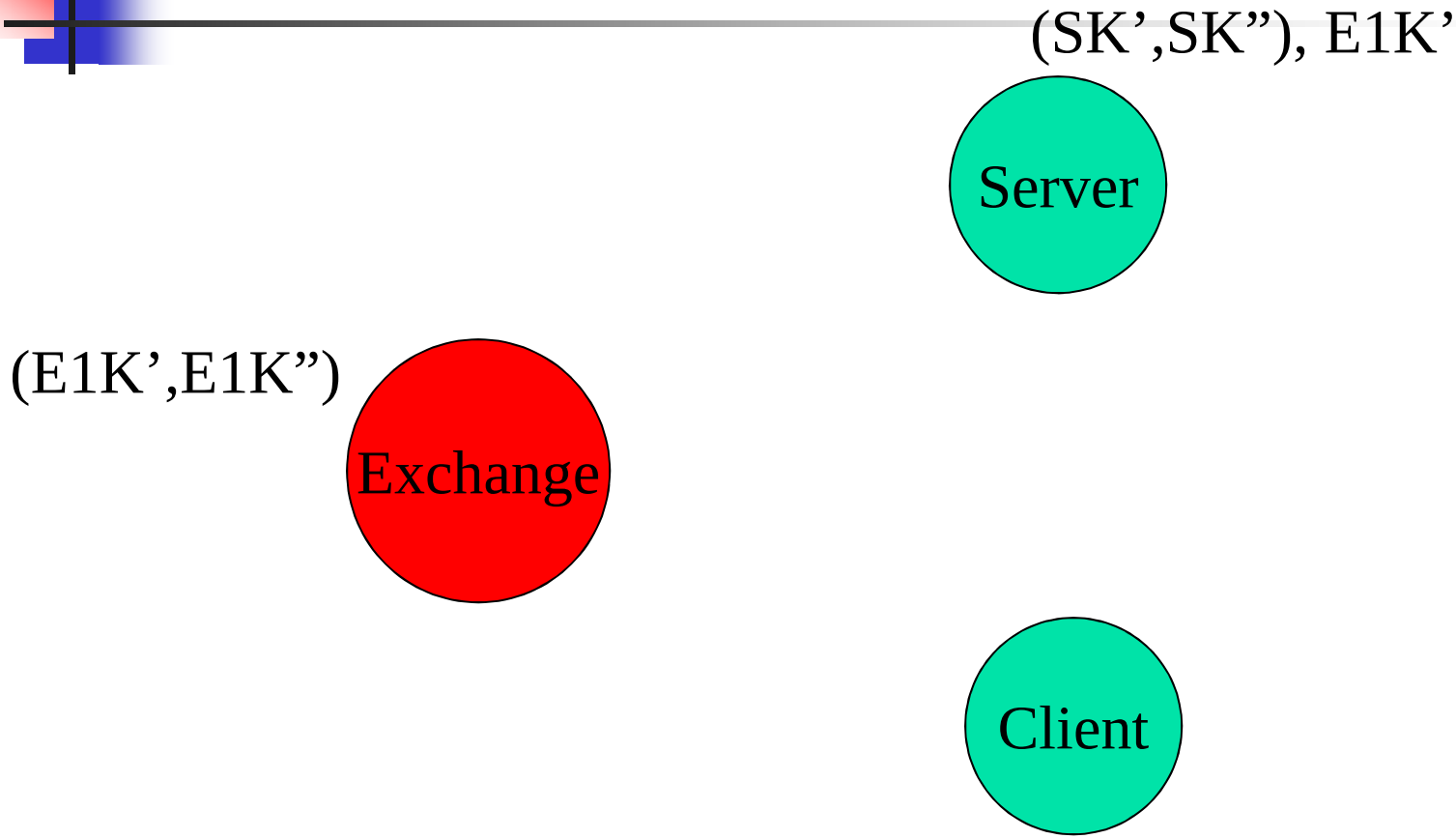
Key Exchange



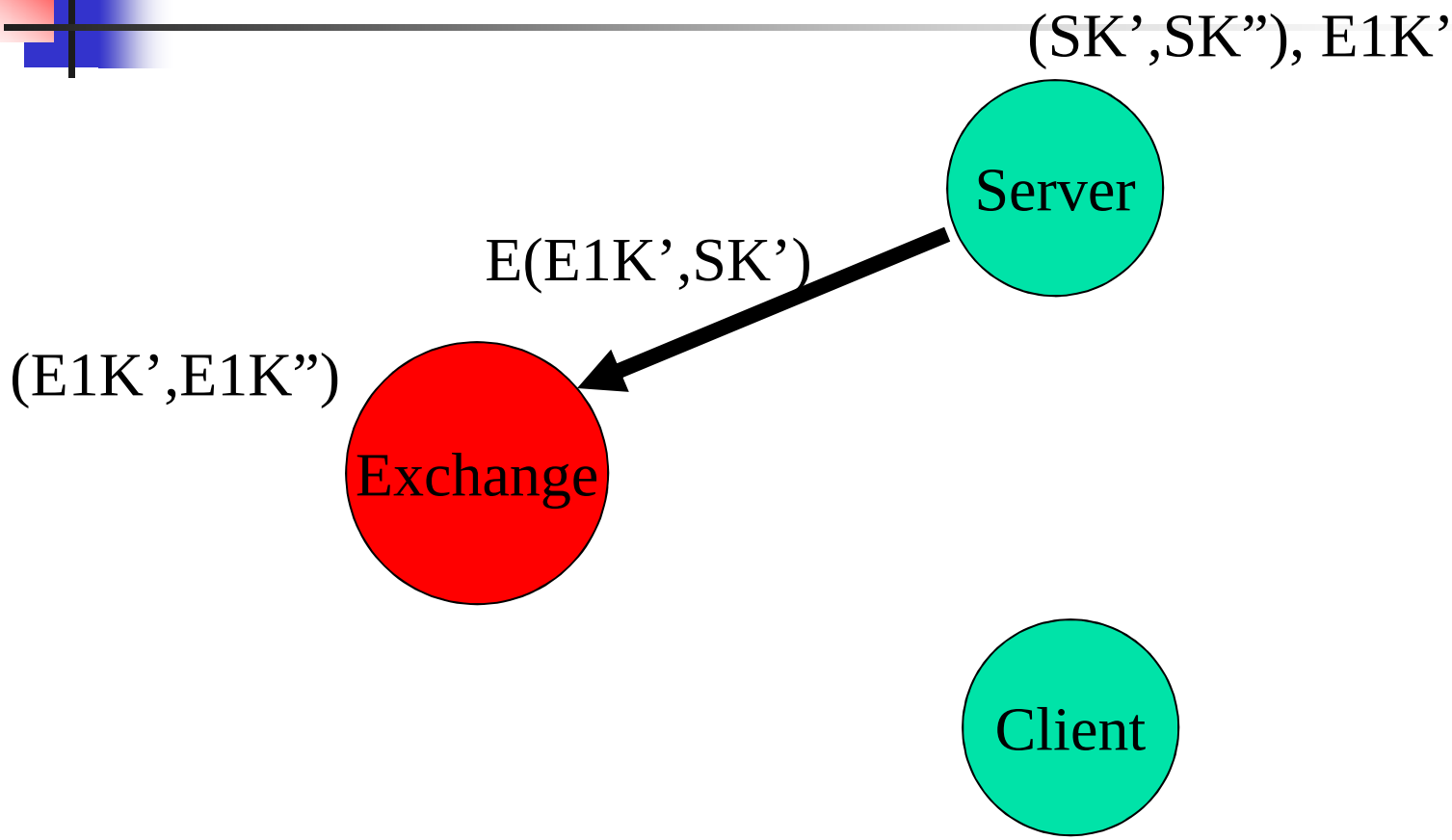
Key Exchange



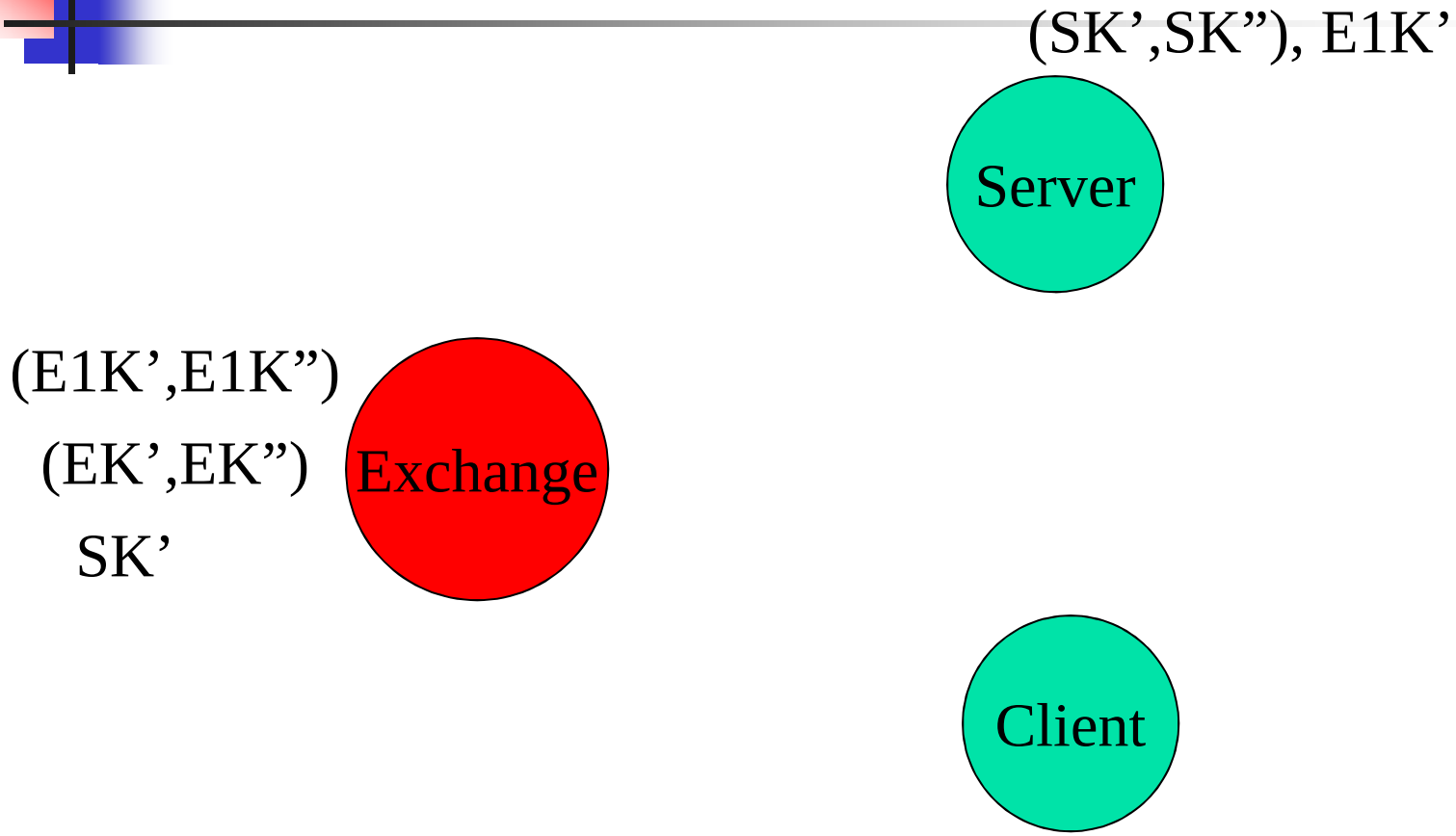
Key Exchange



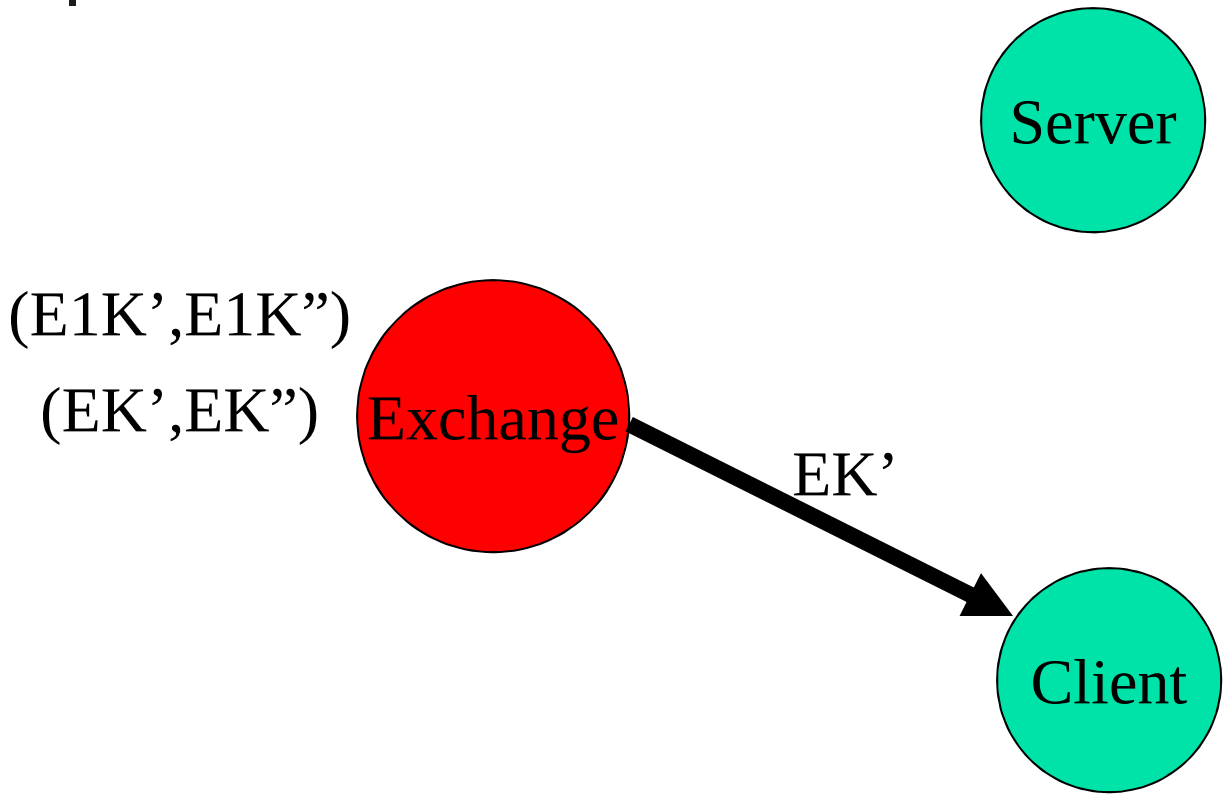
Key Exchange



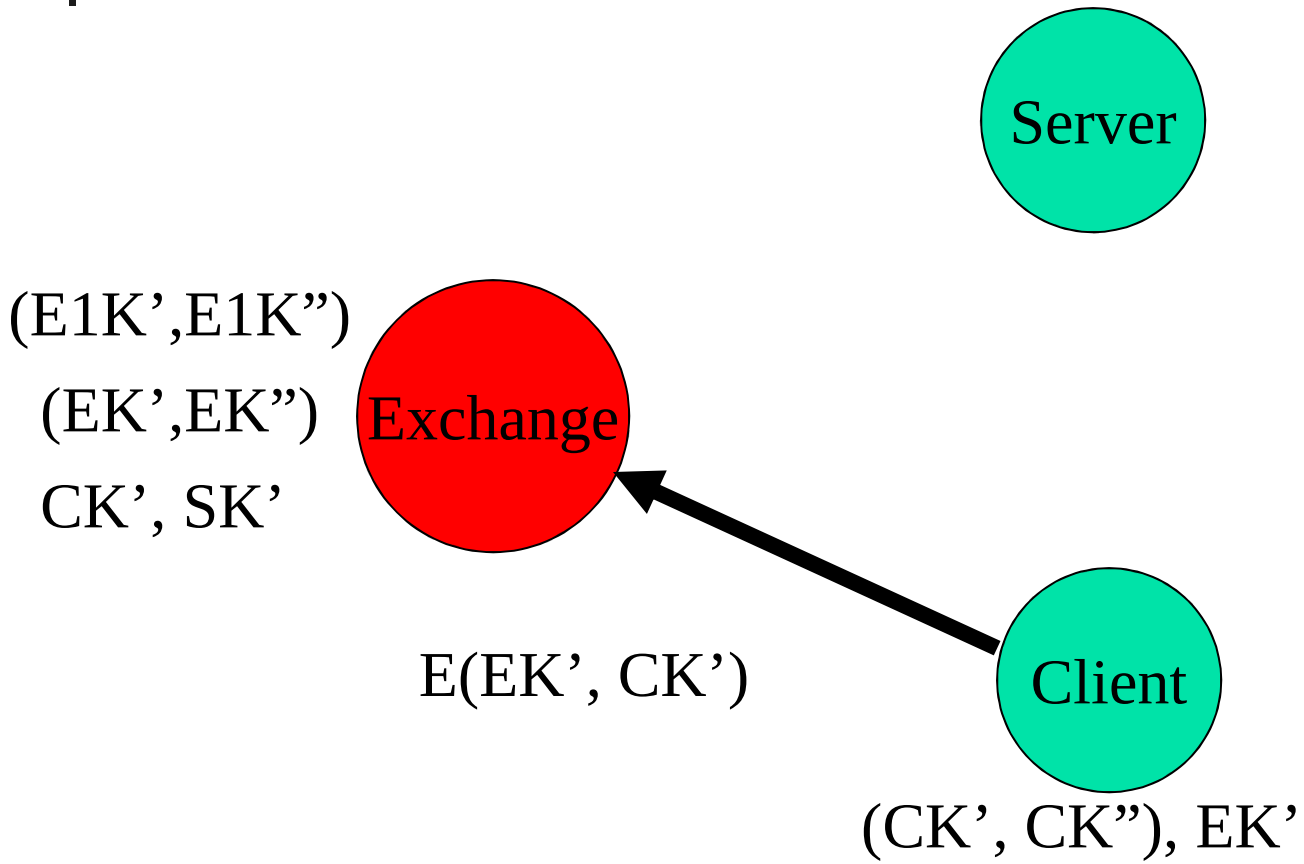
Key Exchange



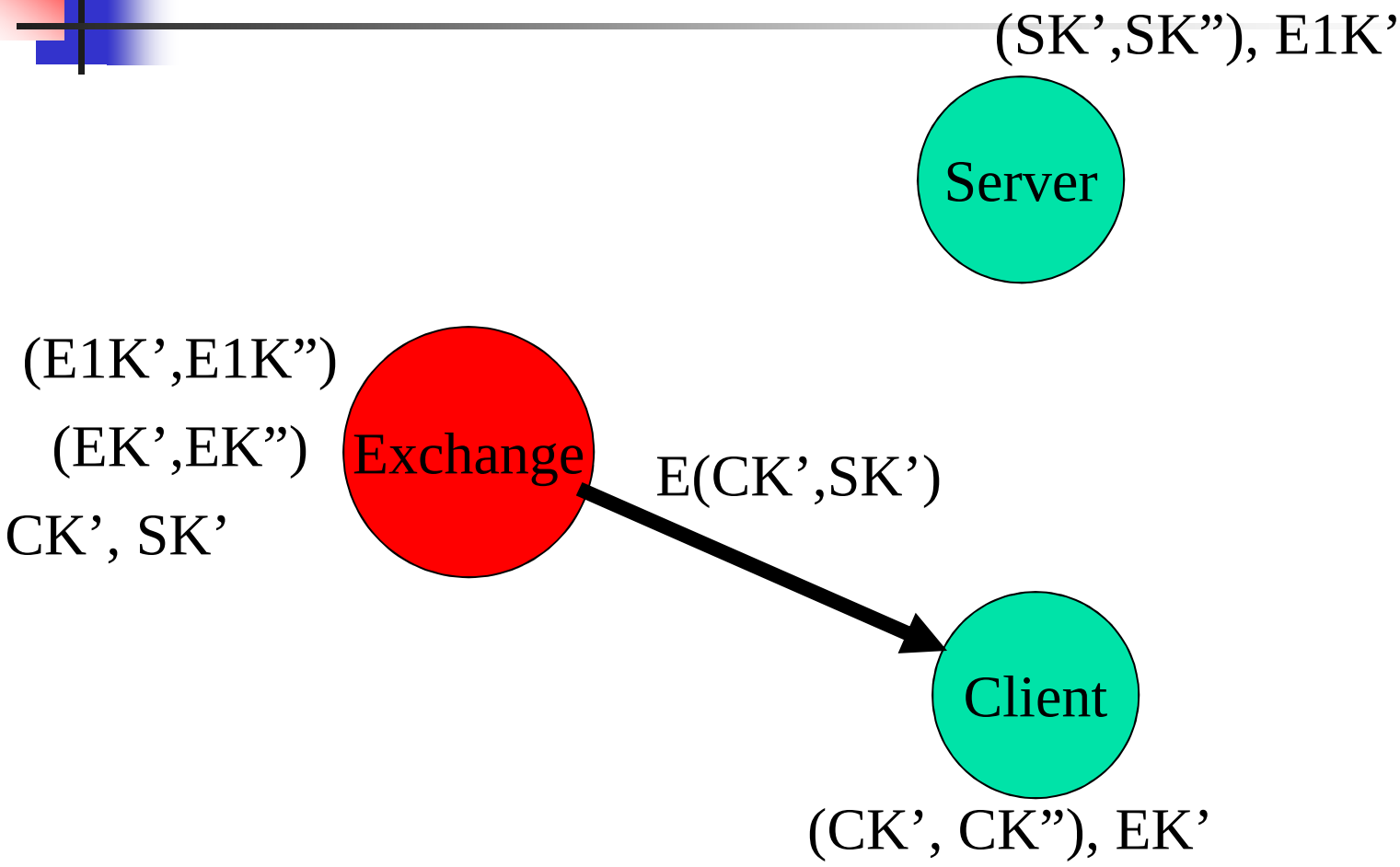
Key Exchange



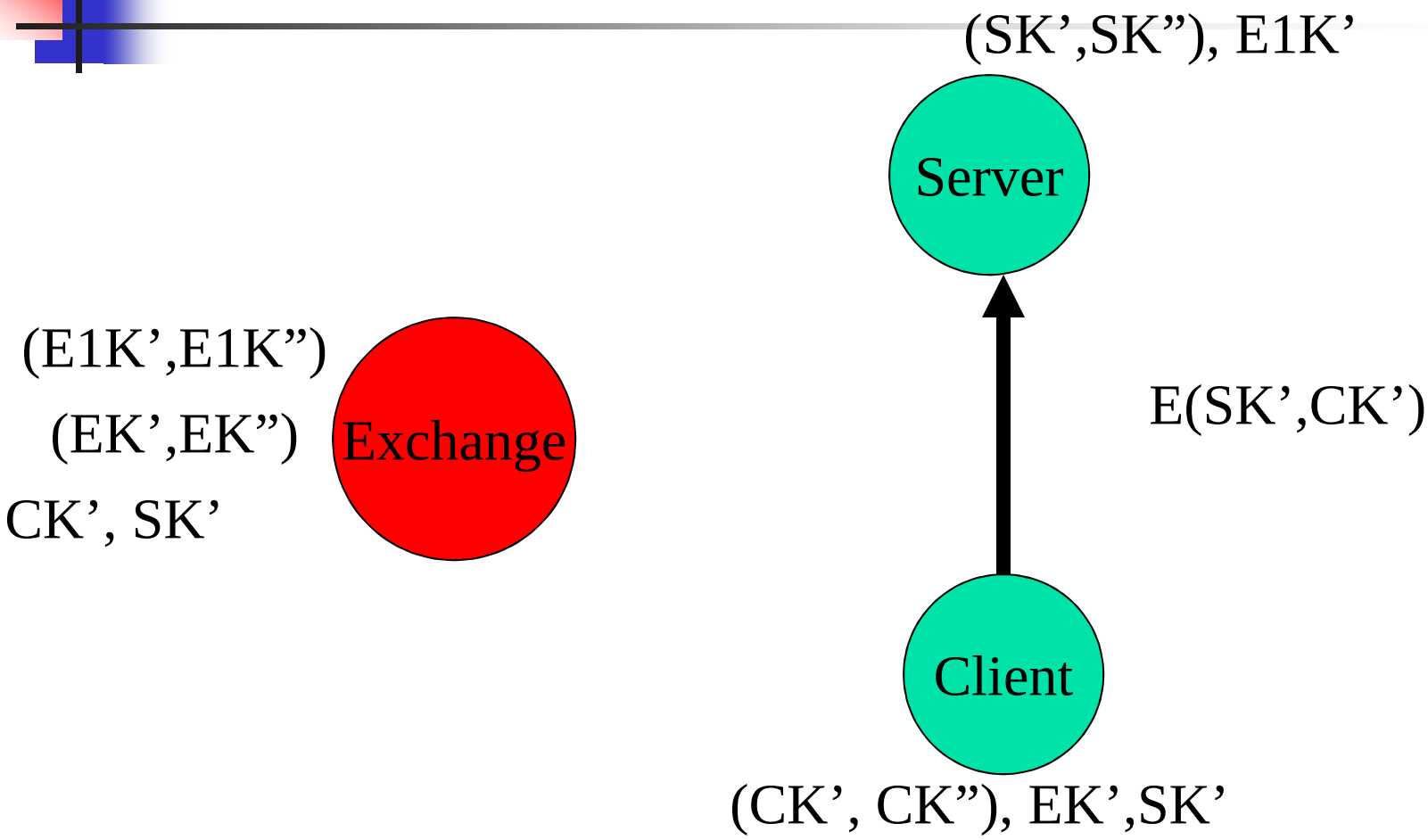
Key Exchange



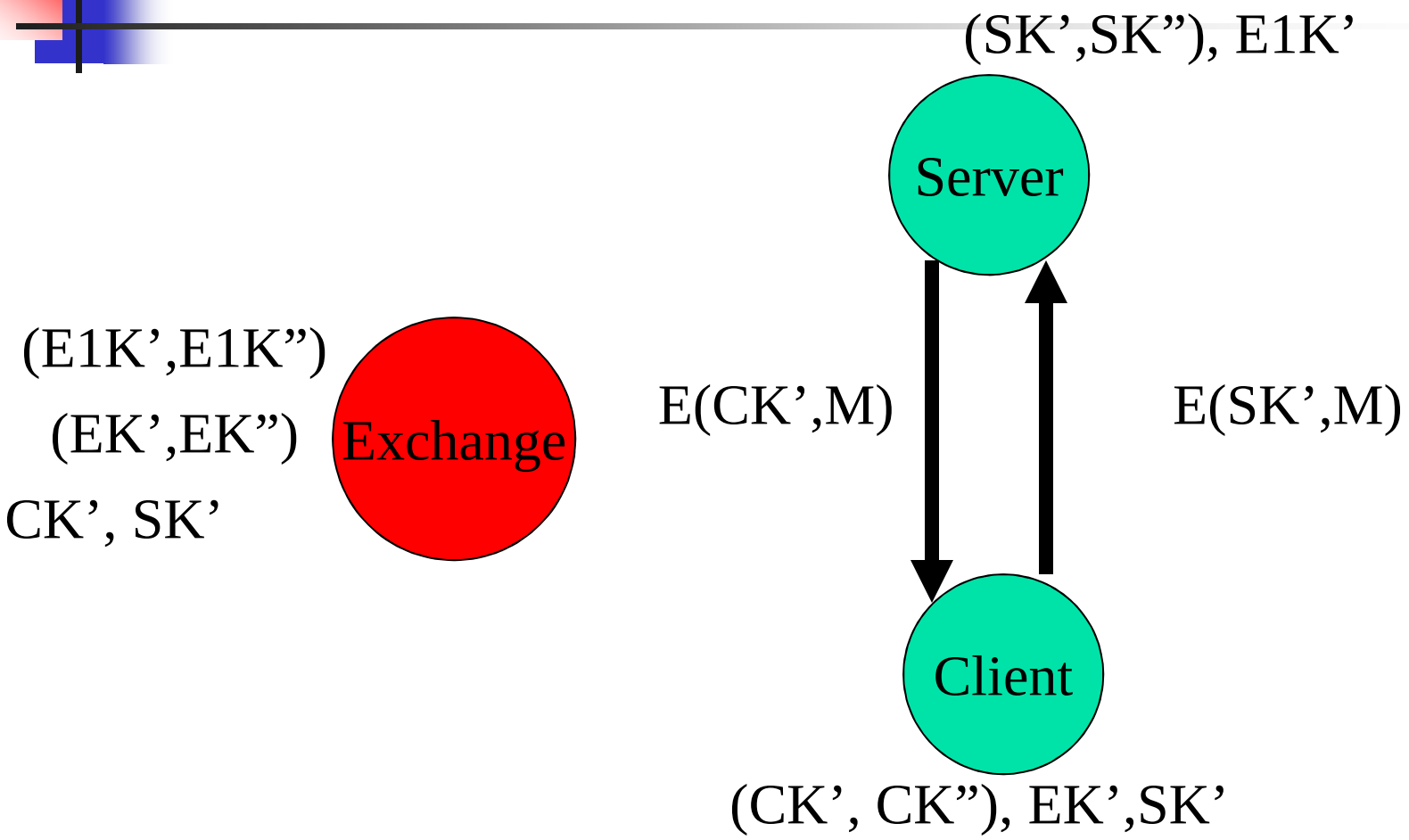
Key Exchange



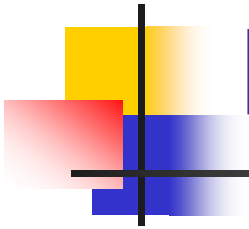
Key Exchange



Key Exchange



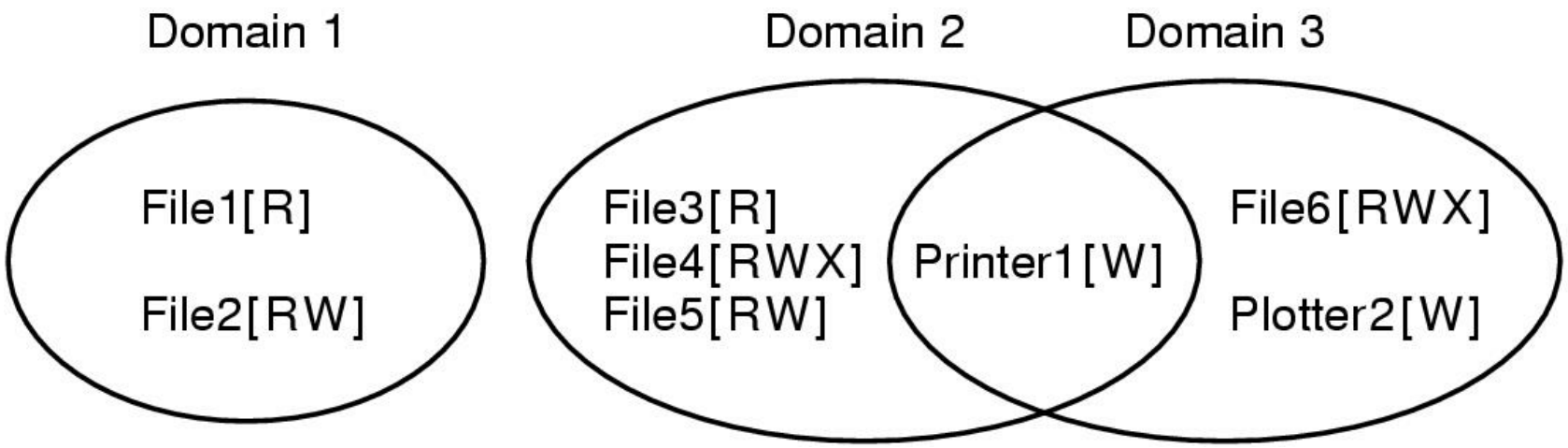
Access Control and Protection Domains



- A computer system is a set of processes and objects
- A process operates within a protection domain
- A protection domain specifies the resources a process may access and the types of operations that may be invoked on the objects.
- **The Principle of Least Authority (*Need to know*): The protection domain of a process should be as small as possible consistent with the need of that process to accomplish its assigned task.**

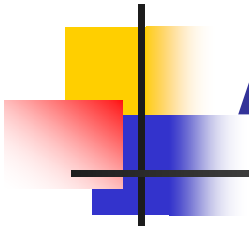
Protection Mechanisms

Protection Domains



Examples of three protection domains

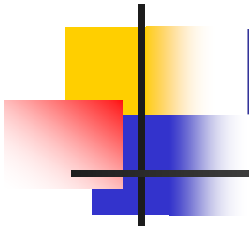
In UNIX: Protection based on UID, GID



Access Control Lists

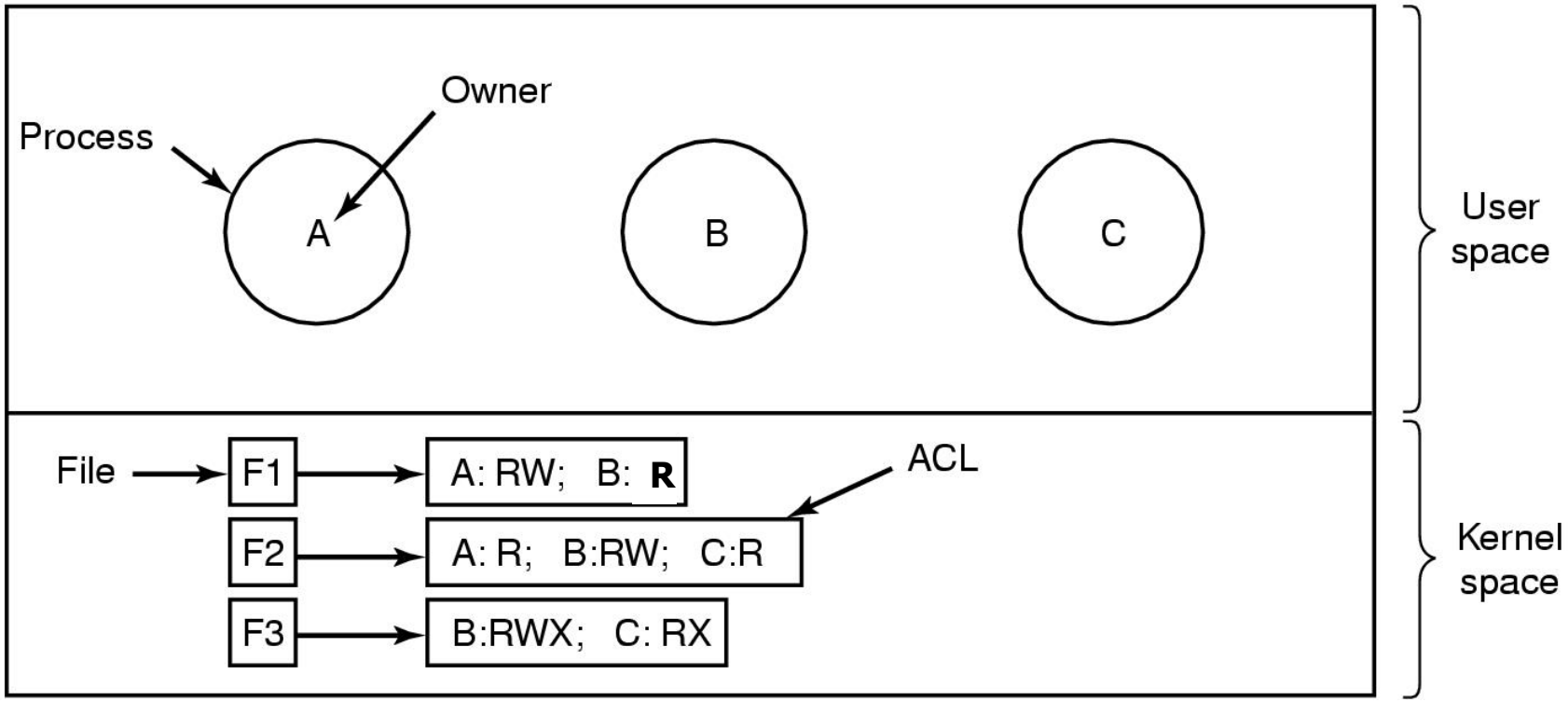
- Each column in the access matrix is implemented as an access list for one Object.
  - On invocation of a method R on an object O by a process running in a domain D,
  - The access control list is searched to check whether D is allowed to perform method R on object O (e.g., allowed to read the file or execute the program)
- A default (e.g., "rest of the world") can be associated with an access list so that any Domain not specified in the list can access the Objects using default methods.
- It is easy for the owner of the Object to grant access to another Domain or revoke access.
- ACL entries can be for individual users or for a group of users.
- In UNIX:

Examples of Access Control Lists

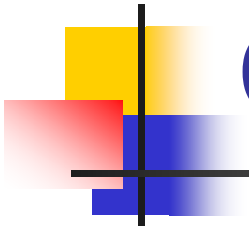


- File Systems
  - Opening a file is checked against an access list to determine if a process may open the file with a given set of access methods.
- Login Shells
  - The login to a system is checked against an access list (usually the password file owned by root).
  - Rlogins are checked against an .rhost file that contains the names of machines from which a rlogin is permitted.

Access Control Lists



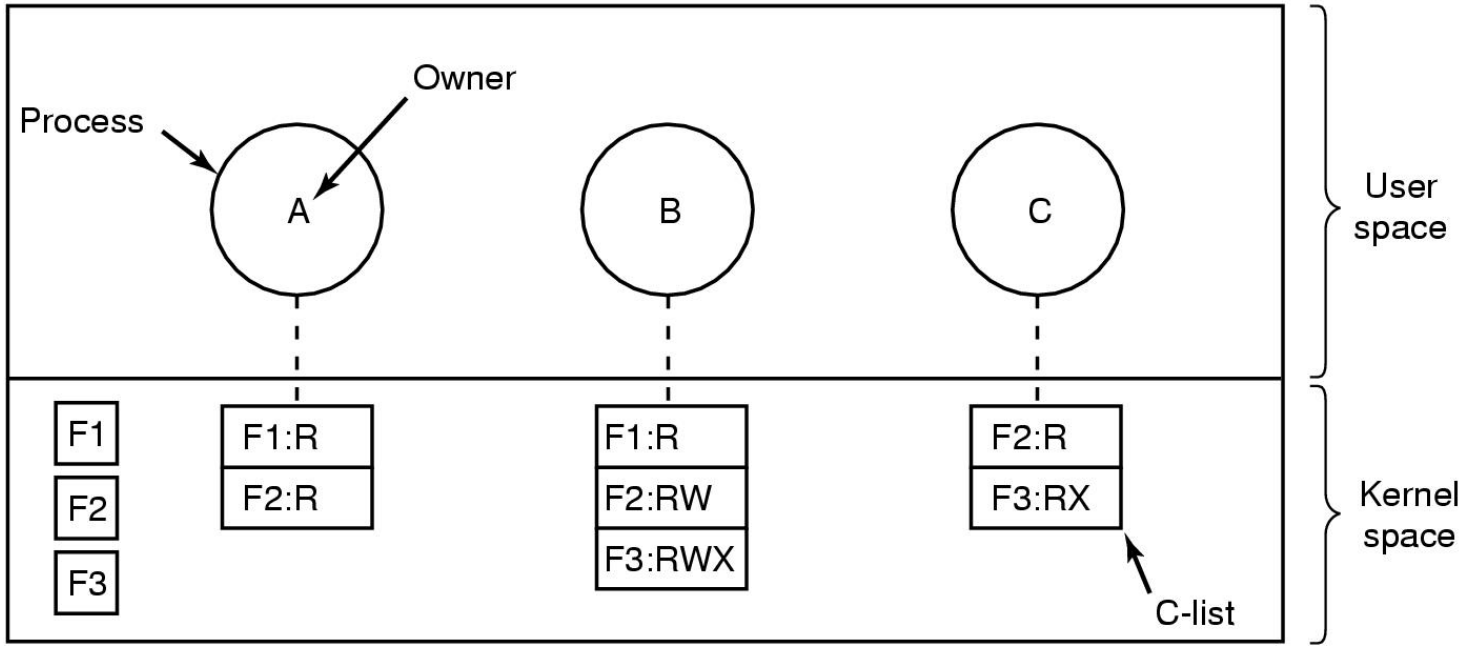
Use of access control lists to manage file access in UNIX



Capability Lists / C-Lists

- Capabilities are keys that allow the holder different types of access
- Each raw in the access matrix is a capability not an identity
- Whoever has the capability (key) can perform corresponding access
- Capabilities can be passed from user to user
  - Object does not control who has access

Capability Lists / C-Lists



When capabilities are used,
each process has a capability list.

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-**6006639**