CS323 Operating Systems Security

Yuanyuan Zhou
Lecture 36
4/28/2003

Lessons from Midterm 2

- What are the difference between
  - Virtual memory
  - Physical memory
  - Disk
  - File cache
- Memory address translation vs. page faults
- Memory accesses vs. I/O operations
  - Is Inode related to Virtual memory?
  - What's the meaning of "memory space in disks"?

Content

- 9.1 The security environment
- 9.2 Basics of cryptography
- 9.3 User authentication
- 9.4 Attacks from inside the system
- 9.5 Attacks from outside the system
- 9.6 Protection mechanisms
- 9.7 Trusted systems

Question

- How many Hollywood movies about security?

Total Approach to Security

- External Security
- User Interface Security -- Establishing user identification and access rights.
- Internal Security -- Controls built into the hardware and software to ensure:
  - Reliable and uncorrupted operation of the system.
  - Integrity of programs and data

External Security

- Physical Security --
  - Protection against disasters
  - Protection against intruders
- Operational Security -- Policies and procedures implemented by the management of a particular computer installation.
  - Authorization -- what access is allowed to what entities.
  - Classification -- parcels the problem into subproblems. Data and users are divided into classes to which particular authorizations are granted.
  - Personnel selection and assignment.
  - Division of responsibilities.
  - Built in controls -- checks and balances.
  - Personnel should be unaware of the nature of the controls, but aware of the fact that the controls are present.

Internal Threats/Security

- *Data Confidentiality*
  - have secret data remain secret.
- *Data Integrity*
  - unauthorized used should not be able modify any data without the owner's permission.
- *System Availability*
  - nobody can disturb the system to make it unusable (e.g., make sure that denial of service does not occur).
- *Privacy*
  - the system protects individuals from misuse of information
- The security system needs to protect against
  - *intruders (adversaries)*
  - *accidental data loss*

Internal Threats

- Security goals and threats

Goal	Threat
Data confidentiality	Exposure of data
Data integrity	Tampering with data
System availability	Denial of service

Intruders

- Common Categories
  - Casual prying by nontechnical users
  - Snooping by insiders
  - Determined attempt to make money
  - Commercial or military espionage

Accidental Data Loss

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

Encryption

- A common method of protecting information transmitted over unreliable links.
- Clear text → Encryption → Ciphertext
- Ciphertext → Decryption → Clear text

Encryption

E - Encryption Algorithm
D - Decryption Algorithm
k - key(s)
M- Message (clear text)

Alternative notation (public, private keys)

Encryption

- An encryption algorithm satisfies:
  - $D_k(E_k(M)))=M$
  - Both E_k and D_k can be computed efficiently.
  - The security of the system depends only on the security of the key, and not on the security of the algorithms E and D.

Encryption Systems

- Public Key Systems
- Secure Secret Key Systems

Public Key Systems

- Use two keys:
  - public key k_{PUB} which is published by the user
  - private key k_{PRIV} $k_{PRIV} \neq k_{PUB}$
- The holder of k_{PRIV} can send an authenticated message to anyone because they can read the message using k_{PUB}.
- $m = D(k_{PRIV}, E(k_{PUB}, m))$ or
  $m = D(k_{PUB}, E(k_{PRIV}, m))$

Public Key Systems

- Example:
  - B, C, D can all encrypt message for A using A's public key.
  - If B encrypted message with A's public key
    - A can read the message using A's private key
    - But C cannot decrypt it even if C knew that it was encrypted with A's public key.

Alternative: Secure Secret Key Systems

- Use single key, called secret key which is shared between encryptor and decryptor (shared key).
- These systems are called Symmetric Systems.
- Example: Data Encryption Standard DES is an example of a shared key.

Symmetric Systems

- Advantages:
  - Symmetric systems provide a two-way channel to their users;
    - Is public key system symmetric?
  - As long as the key remains secret, the system also provides authentication.
- Problems:
  - If the key is revealed, then interceptor can immediately decrypt and encrypt information.

Symmetric Systems Challenges

- Distribution of keys
  - Before communication takes place, the secret keys must be sent securely to both the sender and receiver.
  - Couriers are used to distribute keys
  - Split key in pieces and distribute pieces under separate channels (Clipper program uses a 2 pieces key distribution).
- Number of keys increases in square with number of people
  - Why in square?
  - Solution: Use Clearing house or forwarding office

DES: Data Encryption Standard

- Lucifer algorithm from IBM - 1974
- It uses two main operations:
  - Substitution
  - Permutation (Transposition).
- It derives its strength from repeated application of these two techniques - total 16 cycles.
- Plaintext is encrypted as blocks of 64 bits
  - Key is 64 bits long, however in effect it is only 56 bits long because of 8 parity bits.

DES

- Algorithm is derived from two concepts of Shannon's theory of information secrecy:
  - **Confusion**: a piece of information is changed, so that the output bits have no obvious relationship to the input bits.
  - **Diffusion**: attempts to spread the effect of one plaintext bit to other bits in the ciphertext,
- Substitution provides confusion by systematically substituting some bit patterns for others.
- Permutations provide diffusion by reordering bits.

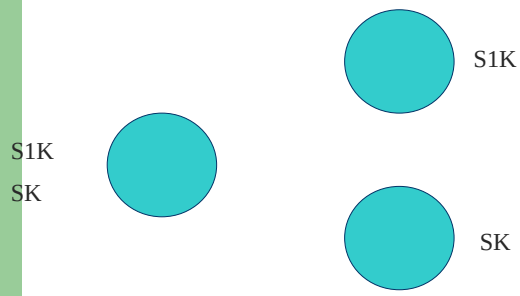
Questions about DES

- Number of Iterations (do we need 16?)
- Key length (In original Lucifer the key was 128 bits long)
- Attack: brute force, then to test all keys (2^56) one needs:
  - 228 million years if one test of the ciphertext takes 100 ms., 2,280 years if one test takes 1 ms., 2 years if one test takes 1 ns.

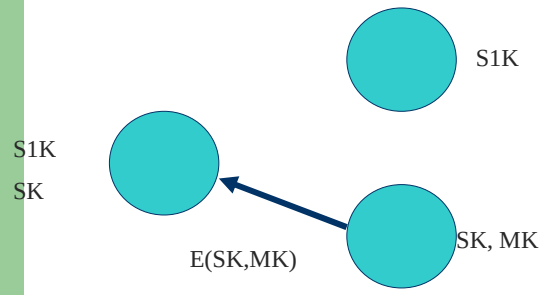
Questions about DES

- Parallel attack:
  - need 106 chips working in parallel at the rate of one key per 1 microsecond, then it takes 1 day. However, the cost of such a machine 50 Million dollars.
  - Use spare cycles in the Internet?
    - Peer-to-peer computing
    - Grid computing

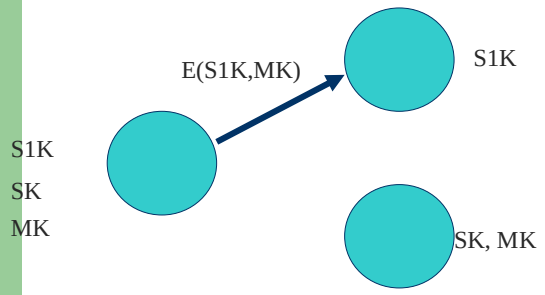
Distribution of Keys using Symmetric Keys



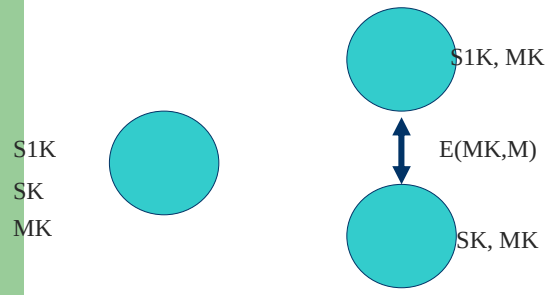
Distribution of Keys using Symmetric Keys



Distribution of Keys using Symmetric Keys



Distribution of Keys using Symmetric Keys



Asymmetric Keys

- Advantages:
  - Authentication –knowledge of private key
  - Privacy – use public key of recipient
  - Key distribution easy
- Problems:
  - Slower than symmetric keys

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

RSA Encryption

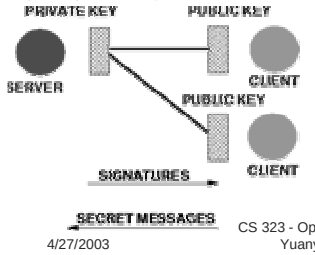
- Goal: *((m^e)^d)mod n = m*
- *n* is computed as product of two large primes *n = p * q* (100 or larger)
- *d* is random integer such that: greatest common divisor *gcd(d, (p-1)*(q-1))=1*
- *e* satisfies *e*d mod (p-1)*(q-1) = 1*
- Key *n* is publicly known, its factors are difficult to calculate.
  - What are p and q for 10403

RSA Encryption

- Example: e=11, n=35, and let assume m=3;
- Assume p=5, q = 7. Then gcd(d,24)= 1 and 11*d mod 24 = 1 are satisfied if d=11
- E(m) = m^11 mod 35, D(C) = C^11 mod 35.
- E(3) = 12 = C, D(12) = 3.
- Another example: D(5) = 10, E(10) = 5.

Private and Public Keys

- Given the public key system, if a trusted server exists in the system, then it can be can be used to provide access to other services securely.
- The server can receive secure messages from clients by publishing its ``public key''.
- The clients encrypt using the public key, but no other client can decrypt the message.



Private and Public Keys

- The server can sign messages to the clients by encrypting the message with its private key. No other client can send such a message. All clients can decrypt the signed message and know it comes from the server.
- The server can sign a message in the clear by sending a check sum of the message encrypted with its private key. Clients can authenticate the message by computing its checksum and comparing it with the encrypted value received.

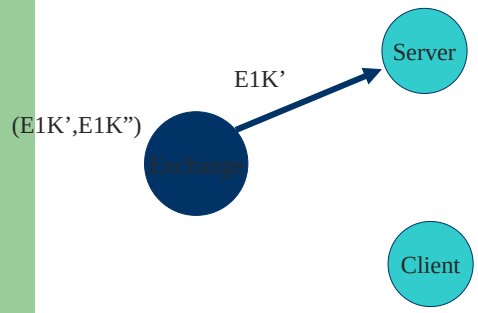
Key Exchange using Asymmetric Keys

- Given the public key system, if a trusted server exists in the system, then it can be used to provide access to other services securely.
- Send an encrypted message to the trusted server TS using its public key and include a public key for the client requesting access to service.
- Similarly for servers S, send a message to the trusted TS server including an S server public key
- The TS trusted server encrypts the S server public key with the clients key and sends a message containing it back to the client

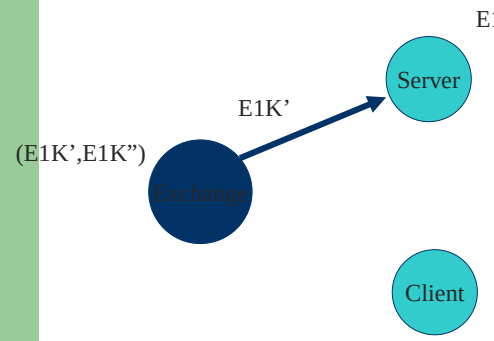
Encryption and Authentication

- Only the client can decrypt the message to get the public key of the S server. It knows it came from the TS trusted server
- The client can now encrypt a message and send it directly to the S server with the S server's public key, only known to the client and TS trusted server

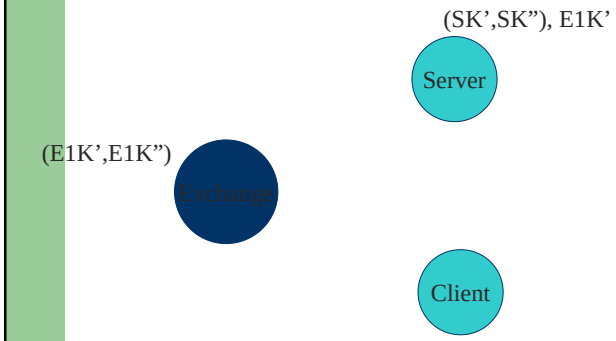
Key Exchange



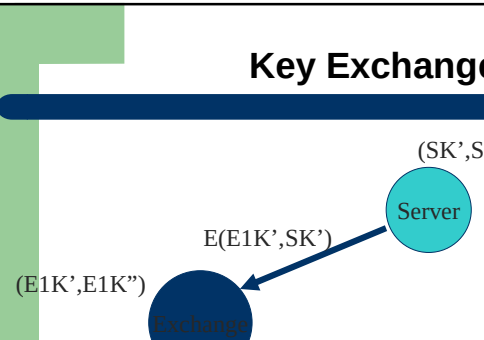
Key Exchange



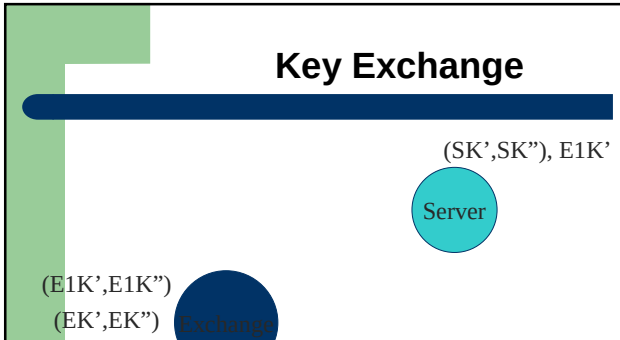
Key Exchange



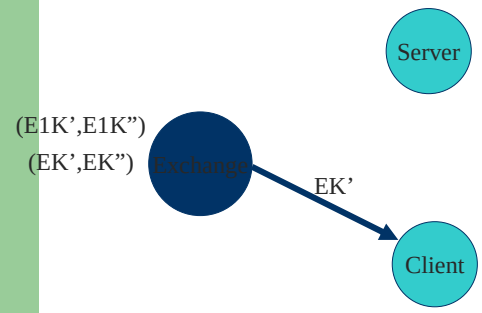
Key Exchange



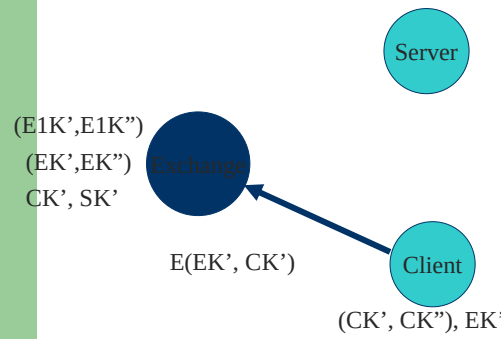
Key Exchange



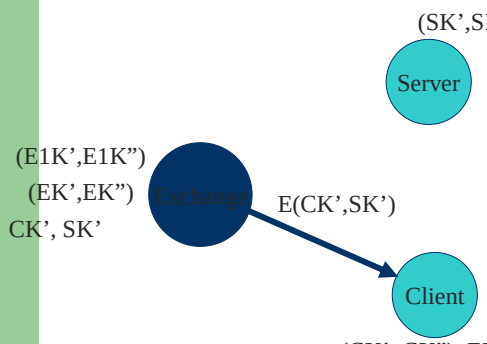
Key Exchange



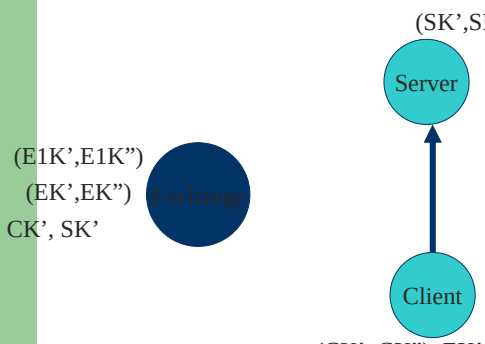
Key Exchange



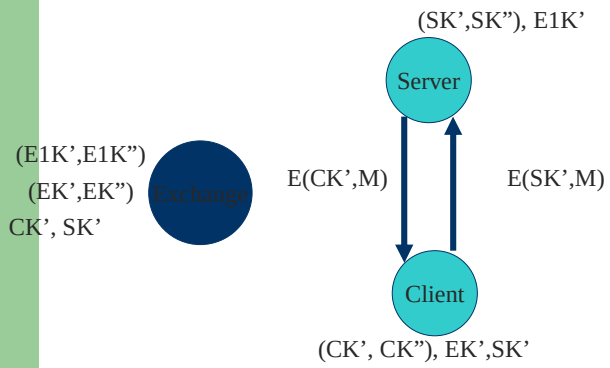
Key Exchange



Key Exchange



Key Exchange

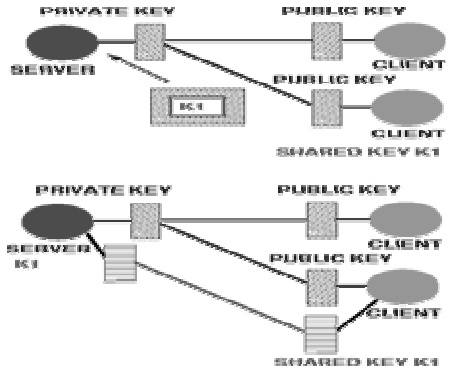


Issues

- Any client can replay an earlier communication, even if the client cannot decode the message in the communication.
  - Some mechanism is needed to make sure communications are fresh! A signed timestamp or message count is often used.
- The server has no way of knowing who a client is.
  - The client needs to have some password or secret it can show to the server. A signed message indicating who the client is and using a different public key for the signature is often used. See CS423.

Distribution of Shared Keys Using Public Keys

- A client may have a two way secure communication with a trusted server if the trusted server publishes a public key.
- The client sends a shared key to the server, encrypted with the public key.
- The client and server may exchange secret messages using the shared key.

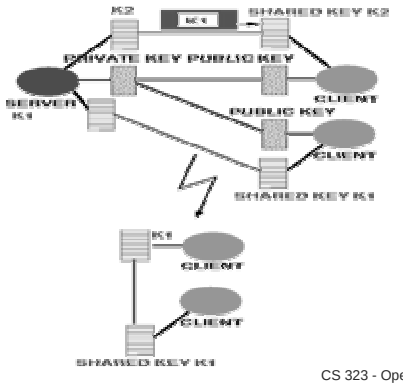


Issues

- Once again freshness and client identification is an issue.
- Also, using the same key over and over again may allow other clients to decode the messages.

Setting up private communications between Clients

- A client may establish two way secure communication with another client through a server
- The client sends a shared key to the server using the server's public key
- The other client sends a shared key to the server using the server's public key
- If the first client wants to talk to the second client, the server sends the first client's shared key to the second client



Issues

- Once again, freshness and client identification is an issue.
- Also, both the server and the other client know the key.
- It is better to use a third key for the communication rather than the two shared keys used to communicate with the server.

Summary

- Symmetric
- Asymmetric
- Authentication
- Privacy
- Key Exchange