CprE 450/550x
Distributed Systems and Middleware

Security

Yong Guan
3216 Coover
Tel: (515) 294-8378
Email: guan@ee.iastate.edu
April 29, 2004

Readings for Today's Lecture

- References
  - Chapter 8 of "Distributed Systems: Principles and Paradigms"
  - Ross Anderson, "Security Engineering"

Security Threats

- Leakage: An unauthorized party gains access to a service or data (eavesdropping).
- Tampering: Unauthorized change of data, tampering with a service
- Vandalism: Interference with proper operation, without gain to the attacker

Security Threats in Comm. Channels

- Eavesdropping – Obtaining copies of messages without authority.
- Masquerading – Sending or receiving messages with the identity of another principal.
- Message tampering – Intercepting messages and altering their contents before passing them onto the intended recipient.
- Replaying – Intercepting messages and sending them at a later time.
- Denial of Service Attack – flooding a channel or other resources with messages.

Security Policies & Mechanisms

- Security Policy indicates which actions each entity (user, data, service) is allowed or prohibited to take.
- Security Mechanism enforces the policy
  - Encryption: transform data to a form only understandable by authorized users.
  - Authentication: verify the claimed identity of a user, client, service, program, etc.
  - Authorization: verify access rights for an authenticated entity.
  - Auditing: make record of and check access to data and resources. Mainly an analysis tool to measure the success of security policies and mechanisms

Familiar Names for Principals in Security Protocols

Alice	First participant
Bob	Second participant
Carol	Participant in three- and four-party protocols
Dave	Participant in four-party protocols
Eve	Eavesdropper
Mallory	Malicious attacker
Sara	A server

7

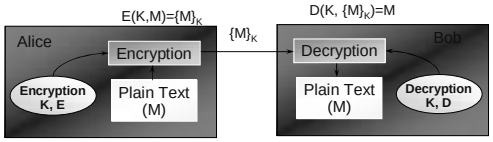
Cryptography Notations

K_A	Alice's secret key
K_B	Bob's secret key
K_{AB}	Secret key shared between Alice and Bob
K_{Apriv}	Alice's private key (known only to Alice)
K_{Apub}	Alice's public key (published by Alice for all to read)
$\{M\}_K$	Message M encrypted with key K
$[M]_K$	Message M signed with key K

8

Cryptography

- Encoding (encryption) of a message that can only be read (decryption) by a key.
- In shared key cryptography (symmetric cryptography) the sender and the recipient know the key, but no one else does.
  - **How do Alice and Bob get the shared key K_{AB} to begin with?**
- In public/private key pairs messages are encrypted with a published public key, and can only be decrypted by a secret private decryption key.



9

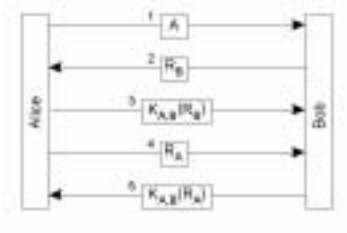
Authentication

- Use of cryptography for safeguarding communication between two principals.
- In direct authentication, the server uses a shared secret key to authenticate the client.
- In indirect authentication, a trusted authentication server provides a ticket to an authenticated client.
  - **The authentication server knows keys of principals and generates temporary shared keys.**
  - **In electronic commerce or wide area applications, public/private key pairs are used rather than shared keys.**

10

Direct Authentication

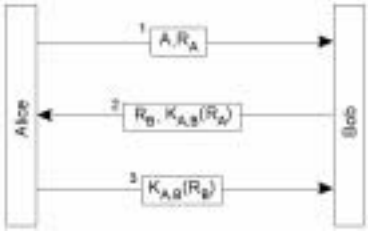
- Authentication based on a shared secret key.



11

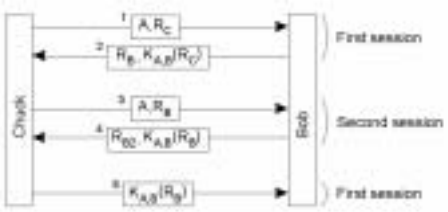
"Optimized" Direct Authentication

- Authentication based on a shared secret key, but using three instead of five messages.



12

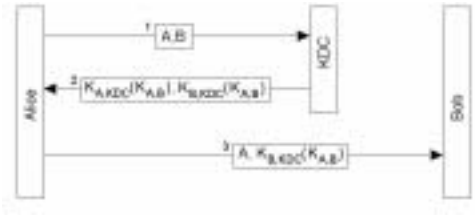
Reflection Attack



13

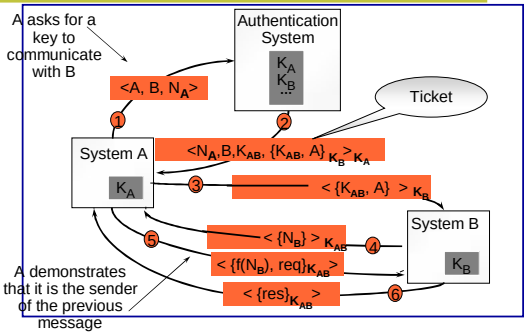
Authentication Using a Key Distribution Center

- Using a ticket and letting Alice set up a connection to Bob.



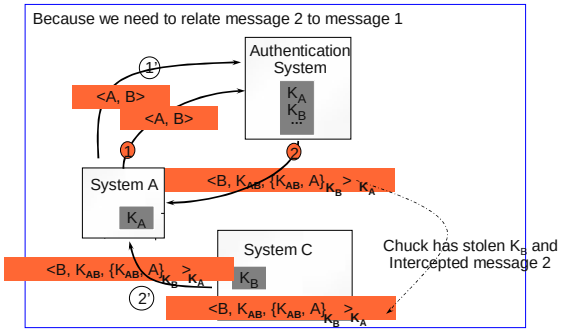
14

Needham-Schroeder Authentication



15

Why Do We Need Nonce N_A in Message 1?



16

Needham–Schroeder Secret-key Authentication Protocol

Header	*Message*	*Notes*
1. A->S:	A, B, N_A	A requests S to supply a key for communication with B.
2. S->A:	$\{N_A, B, K_{AB}, \{K_{AB}, A\}K_B\}K_A$	S returns a message encrypted in A's secret key, containing a newly generated key K_{AB} and a 'ticket' encrypted in B's secret key. The nonce N_A demonstrates that the message was sent in response to the preceding one. A believes that S sent the message because only S knows A's secret key.
3. A->B:	$\{K_{AB}, A\}K_B$	A sends the 'ticket' to B.
4. B->A:	$\{N_B\}K_{AB}$	B decrypts the ticket and uses the new key K_{AB} to encrypt another nonce N_B.
5. A->B:	$\{N_B - 1\}K_{AB}$	A demonstrates to B that it was the sender of the previous message by returning an agreed transformation of N_B.

17

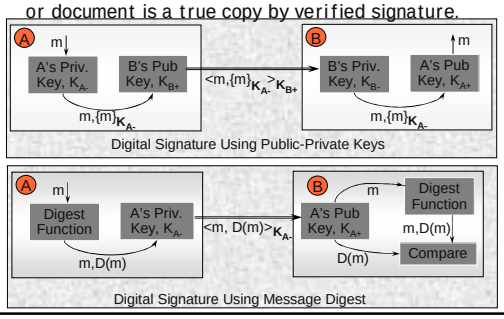
Kerberos Authentication

Read section 8.5 from text

18

Digital Signatures

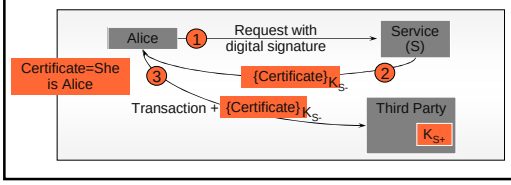
- Cryptography is also used to verify that a message or document is a true copy by verified signature.



19

Digital Certificates

- A digital certificate is a statement signed by a third party principal.
- To be useful, certificates must have:
  - **A standard format, for construction and interpretation**
  - **A protocol for constructing chains of certificates**
  - **A trusted authority at the end of the chain**



20

Alice's Bank Account Certificate

1. *Certificate type*	Account number
2. *Name*	Alice
3. *Account*	6262626
4. *Certifying authority*	Bob's Bank
5. *Signature*	{*Digest*(*field 2 + field 3*)}K_{Bpriv}

Alice may pretend to be the bank and create a new key pair, K_{B+}, K_{B-}

21

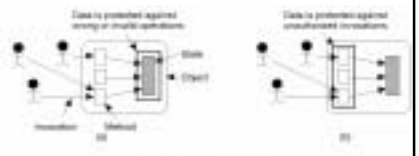
Public-Key Certificate for Bob's Bank

1. *Certificate type*	Public key
2. *Name*	Bob's Bank
3. *Public key*	K_{Bpub}
4. *Certifying authority*	Fred – The Bankers Federation
5. *Signature*	{*Digest*(*field 2 + field 3*)} K_{Fpriv}

Eventually K_{F-}, K_{F+} have to be obtained reliably.

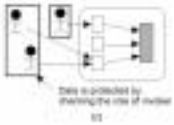
22

Focus of Access Control



- Three approaches for protection against security threats

a) Protection against invalid operations
b) Protection against unauthorized invocations
c) Protection against unauthorized users



23

Access Control

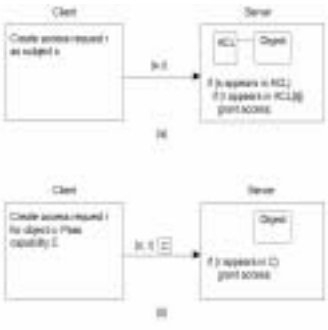
- Control of access to resources of a server.
- A basic form of access control checks <principal, op, resource> requests for:
  - Authenticity of the principal or its credentials.
  - Access rights for the requested resource & op.
- Access control matrix M.
  - Each principal is represented by a row, and each resource object is represented by a column.
  - M[s,o] lists precisely what operations principal s can request to be carried out on resource o.
  - May be sparse.
- Access control list (ACL)
  - Each object maintains a list of access rights of principals, I.e., an ACL is some column in M with the empty entries left out.

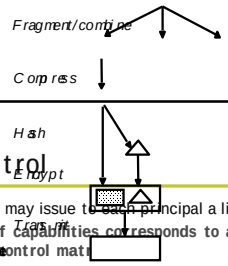
24

Access Control Matrix

Comparison between ACLs and capabilities for protecting objects.

a) Using an ACL
b) Using capabilities.





Access Control

- The server may issue to each principal a list of capabilities.
  - **A list of capabilities corresponds to an entry in the access control matrix.**
- To reduce ACLs, the notion of protection domain is introduced.
  - A protection domain is a set of (object, access rights) pairs kept by a server.
  - Whenever a principal requests an operation to be carried out on an object, the access control monitor checks if the principal belongs to that domain, and then if the request is allowed for that object.
- Each principal can carry a certificate listing the groups it belongs to.
  - The certificate should be protected by a digital signature.

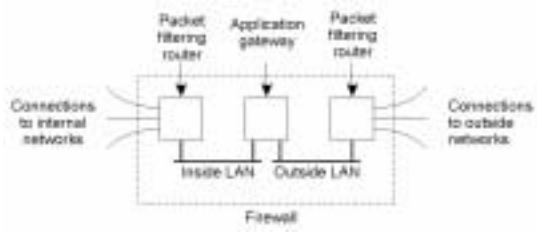
Firewalls

Firewall filtering can be done at diff. levels

- IP packet filtering: operates as a router and makes decisions as to whether or not to pass a packet based on its source/destination addresses.
  - The gateway on the outside LAN protects against incoming packets. The gateway on the inside LAN protects against outgoing packets.
- TCP gateway: checks all TCP connection requests and segment transmissions. TCP segments will be checked for correctness and may be routed to an application-level gateway for content checking.
- Application-level filtering (proxy gateway): inspects the content of incoming/outgoing messages.
  - To prevent applets to be downloaded to the inside LAN, all Web traffic could be directed through a Web proxy gateway. The gateway accepts regular HTTP requests, but may discard certain requests/pages.

Firewall Configuration

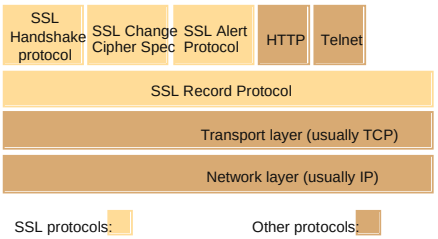
- A common implementation of a firewall.



Secure Socket Layer Protocol

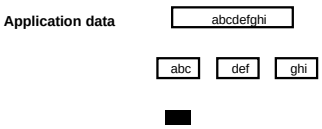
- SSL was developed by Netscape for electronic transaction security.
- A protocol layer is added below the application layer for:
  - **Negotiating encryption and authentication methods.**
  - **Bootstrapping secure communication**
- It consists of two layers:
  - **The Record Protocol Layer implements a secure channel by encrypting and authenticating messages**
  - **The Handshake Layer establishes and maintains a secure session between two nodes.**

SSL Protocol Stack



SSL Record Protocol

- The record protocol takes an application message to be transmitted,
  - fragments the data into manageable blocks,
  - optionally compresses the data,
  - computes a message authentication code (MAC),
  - encrypts and
  - adds a header.

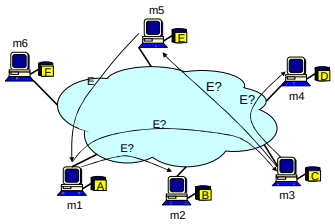


Model

- Each user stores a subset of files
- Each user has access (can download) files from all users in the system

Gnutella: Example

- Assume: m1's neighbors are m2 and m3; m3's neighbors are m4 and m5;…



Insert

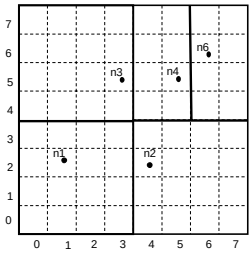
- Searching: like query, but nodes maintain state after a collision is detected and the reply is sent back to the originator
- Insertion
  - Follow the forward path; insert the file at all nodes along the path
  - A node probabilistically replace the originator with itself; obscure the true originator

Freenet Summary

- Advantages
  - Provides publisher anonymity
  - Totally decentralize architecture → robust and scalable
  - Resistant against malicious file deletion
- Disadvantages
  - Does not always guarantee that a file is found, even if the file is in the network

CAN Example: Two Dimensional Space

- Nodes n4:(5, 5) and n5:(6,6) join



Data Structure

- Assume identifier space is $0..2^m$
- Each node maintains
  - Finger table
    - Entry *i* in the finger table of *n* is the first node that succeeds or equals $n + 2^i$
  - Predecessor node
- An item identified by *id* is stored on the succesor node of *id*

Node Joining

- Node n joins the system:
  - **n picks a random identifier, id**
  - **n performs n' = lookup(id)**
  - **n->successor = n'**