Computer Science
425
Distributed Systems
Fall 2009

Lecture 26
Security in Distributed Systems

Klara Nahrstedt

Acknowledgement

- **The slides during this semester are based on ideas and material from the following sources:**
  - **Slides prepared by Professors M. Harandi, J. Hou, I. Gupta, N. Vaidya, Y-Ch. Hu, S. Mitra.**
  - **Slides from Professor S. Gosh's course at University o Iowa.**

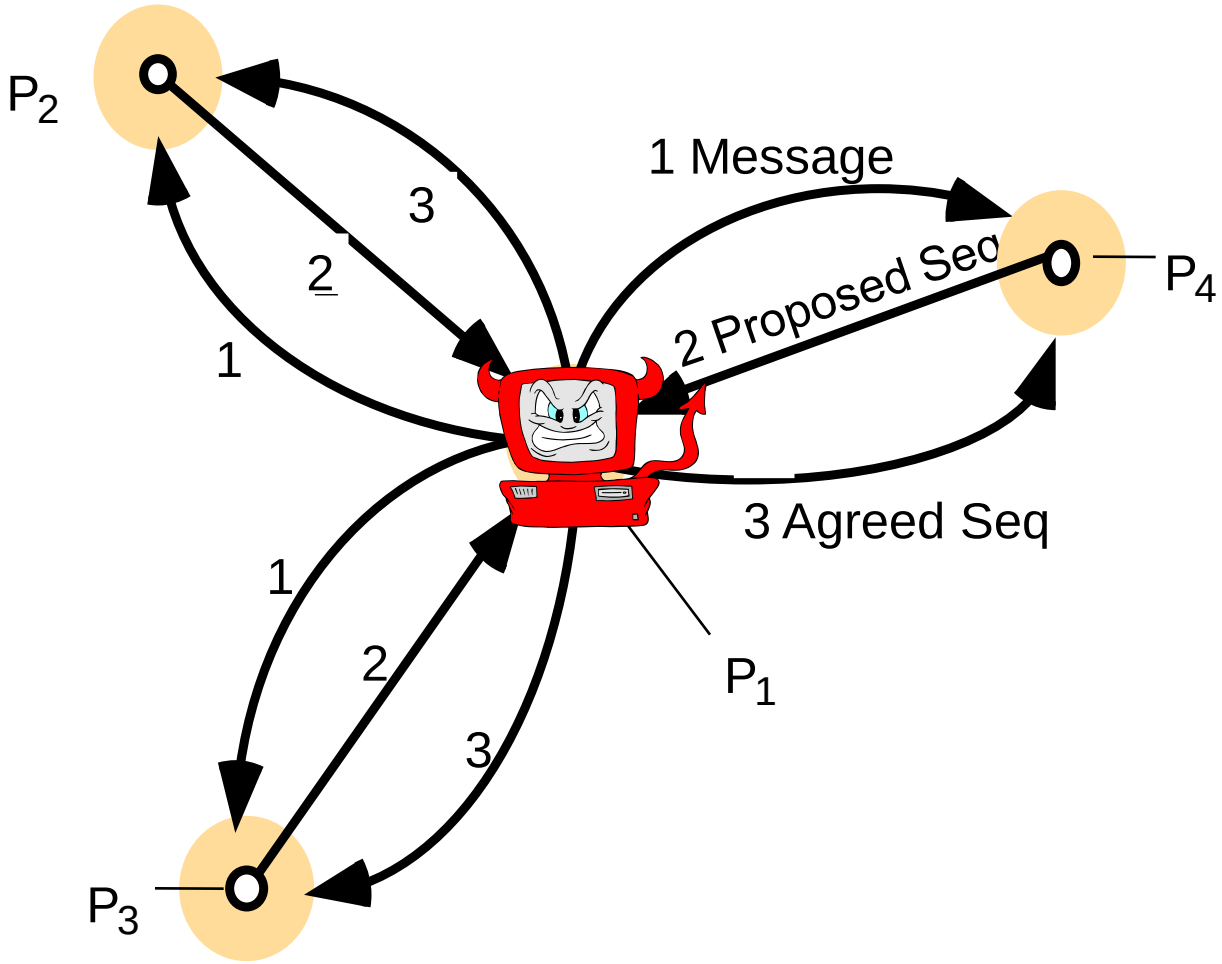
Administrative

- **MP3 posted**
  - **Deadline December 7 (Monday) – pre-competition**
    - » **Top five groups will be selected for final demonstration on Tuuesday, December 8**
  - **Demonstration Signup Sheets for Monday, 12/7, will be made available**
  - **Main Demonstration in front of the Qualcomm Representative will be on Tuesday, December 8 afternoon - details will be announced.**
- **HW4 posted**
  - **Deadline December 1, 2009 (Tuesday)**

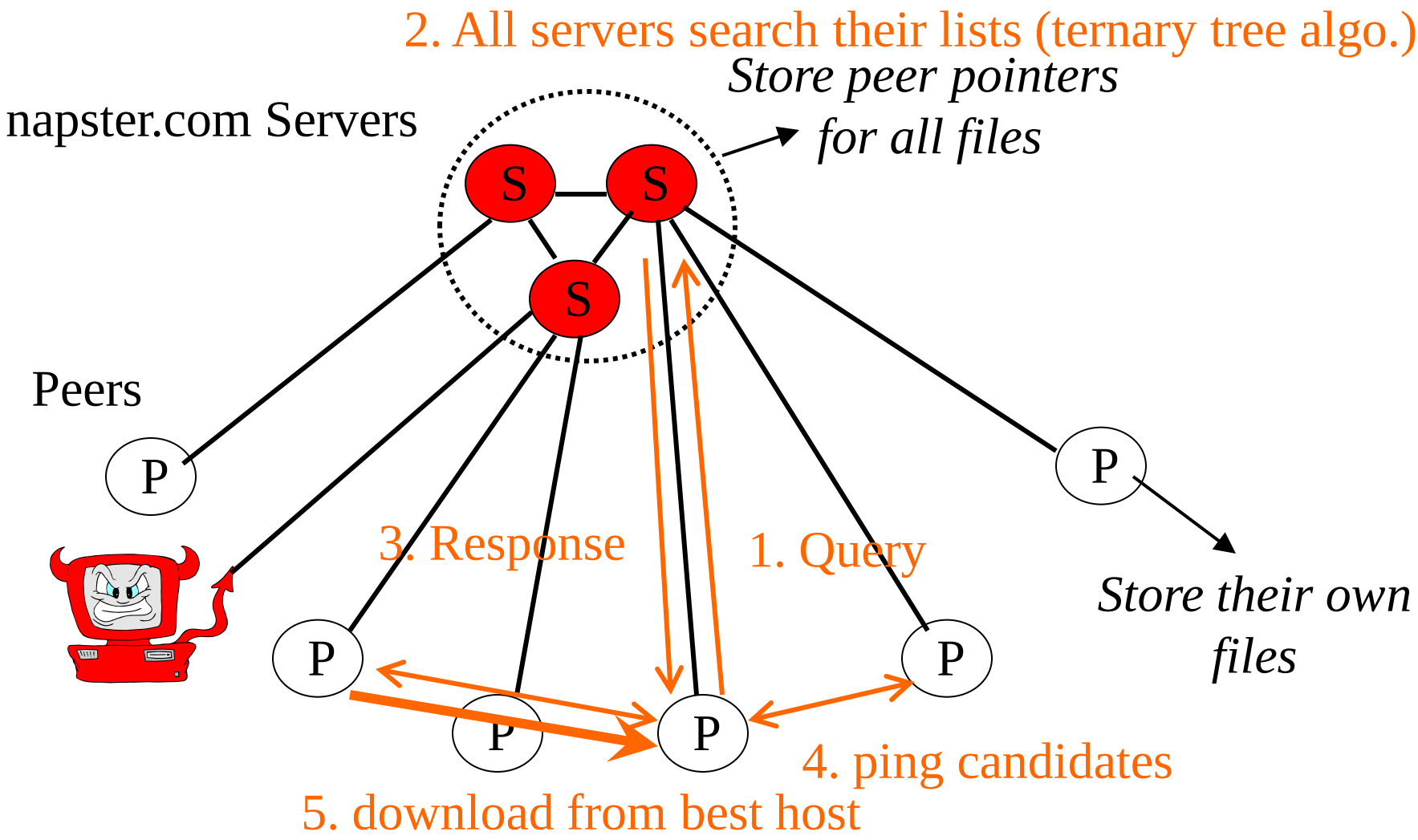
Administrative

- **MP3**
  - **Readme file must include:**
    - » **Bootstraping routine – how one install your system – developers manuscript**
    - » **How one use your system – usage prescription for users**
    - » **Known bugs, what are the issues with your system/application**
  - **Tar or zip your source code and upload it to agora wiki**
    - » **URL Information will be provided on the web/in class/on newsgroup**
  - **Fill out project template as specified**
    - » **Template Information will be provided on the web/in class/on newsgroup**

ISIS algorithm for total ordering



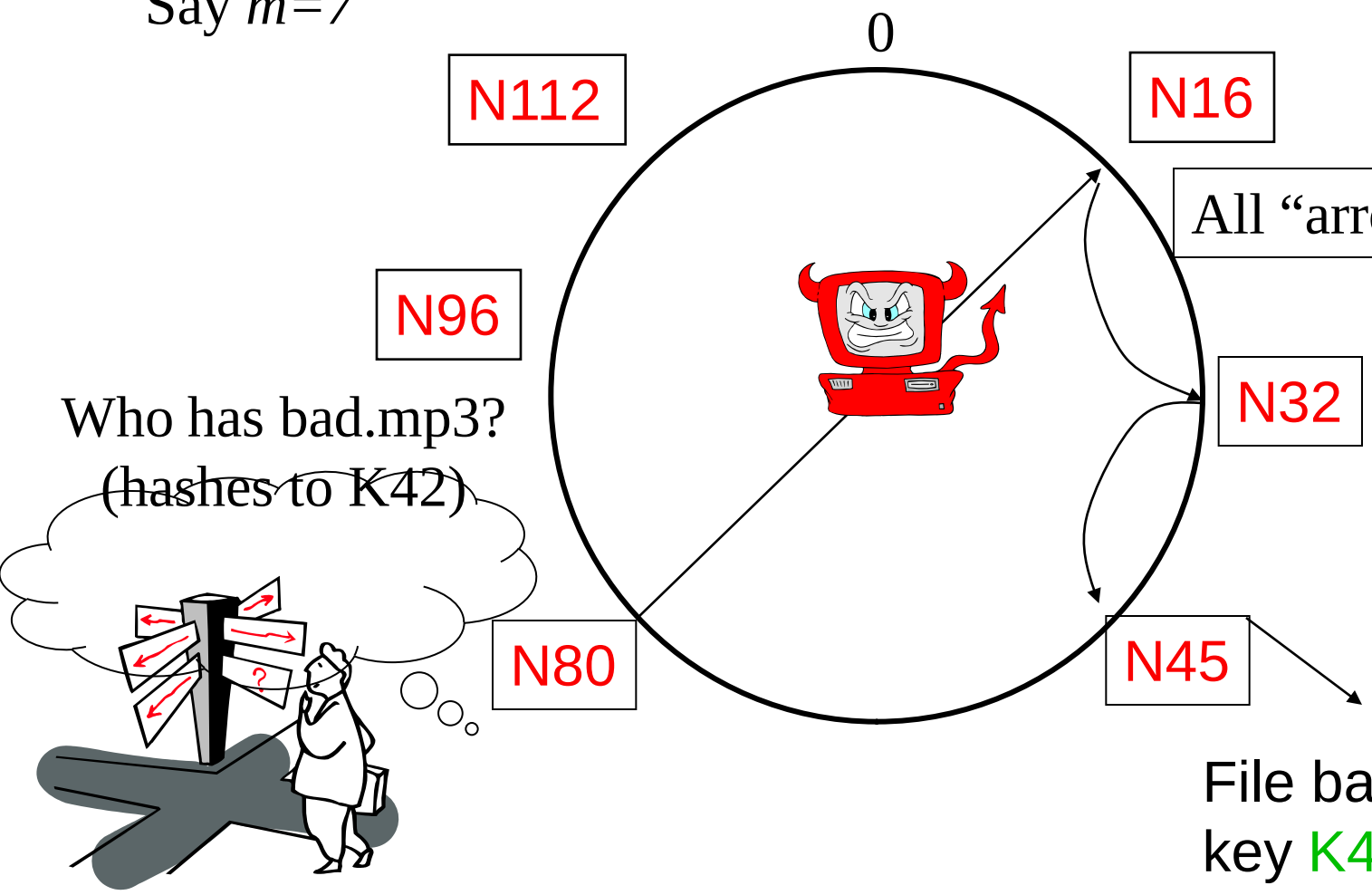
Napster


2. All servers search their lists (ternary tree algo.)
Store peer pointers
for all files
napster.com Servers
S
S
S
Peers
P
P
P
P
P
P
3. Response
1. Query
Store their own
files
4. ping candidates
5. download from best host

Chord: client to client

At node *n*, send query for key *k* to largest successor/finger entry < *k*
if none exist, return *successor(n)* to requestor

Say *m=7*



Security Threats

- **Leakage:** An unauthorized party gains access to a service or data (eavesdropping).
  - Attacker obtains knowledge of a withdrawal or account balance
- **Tampering:** Unauthorized change of data, tampering with a service
  - Attacker changes the variable holding your personal checking $$ total
- **Vandalism:** Interference with proper operation, without gain to the attacker
  - Attacker does not allow any transactions to your account
  - E.g., DOS=denial of service

More Concerns

Attacks on Communication Channel / Network

- **Eavesdropping – Obtaining copies of messages without authority.**
- **Masquerading – Sending or receiving messages with the identity of another principal (user or corporation).**
- **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 (e.g., port) with messages.**

Addressing the Challenges: Security

- **Leakage:** An unauthorized party gains access to a service or data (eavesdropping).
  - **Confidentiality : protection against disclosure to unauthorized individuals.**
- **Tampering:** Unauthorized change of data, tampering with a service
  - **Integrity : protection against alteration or corruption.**
- **Vandalism:** Interference with proper operation, without gain to the attacker
  - **Availability : protection against interference with the means to access the resources.**

Security Policies & Mechanisms

- **A Security *Policy* indicates which actions each entity (user, data, service) is allowed or prohibited to take.**
  - **E.g., Only an owner is allowed to make transactions to his account. CIA properties.**
- **A Security *Mechanism* enforces the policy**
  - ***Encryption and decryption:*** **transform data to a form only understandable by authorized users, and vice-versa.**
  - ***Authentication:*** **verify the claimed identity of a user, client, service, process, 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.**

Designing Secure Systems

- **Make worst-case assumptions about attackers:**
  - **exposed interfaces, insecure networks, algorithms and program code available to attackers, attackers may be computationally very powerful**
  - **Tradeoff between security and performance impact/difficulty**
  - **Typically design system to withstand a known set of attacks**
- **Designing Secure Systems**
  - **Traditionally done as a layer on top of existing protocols.**

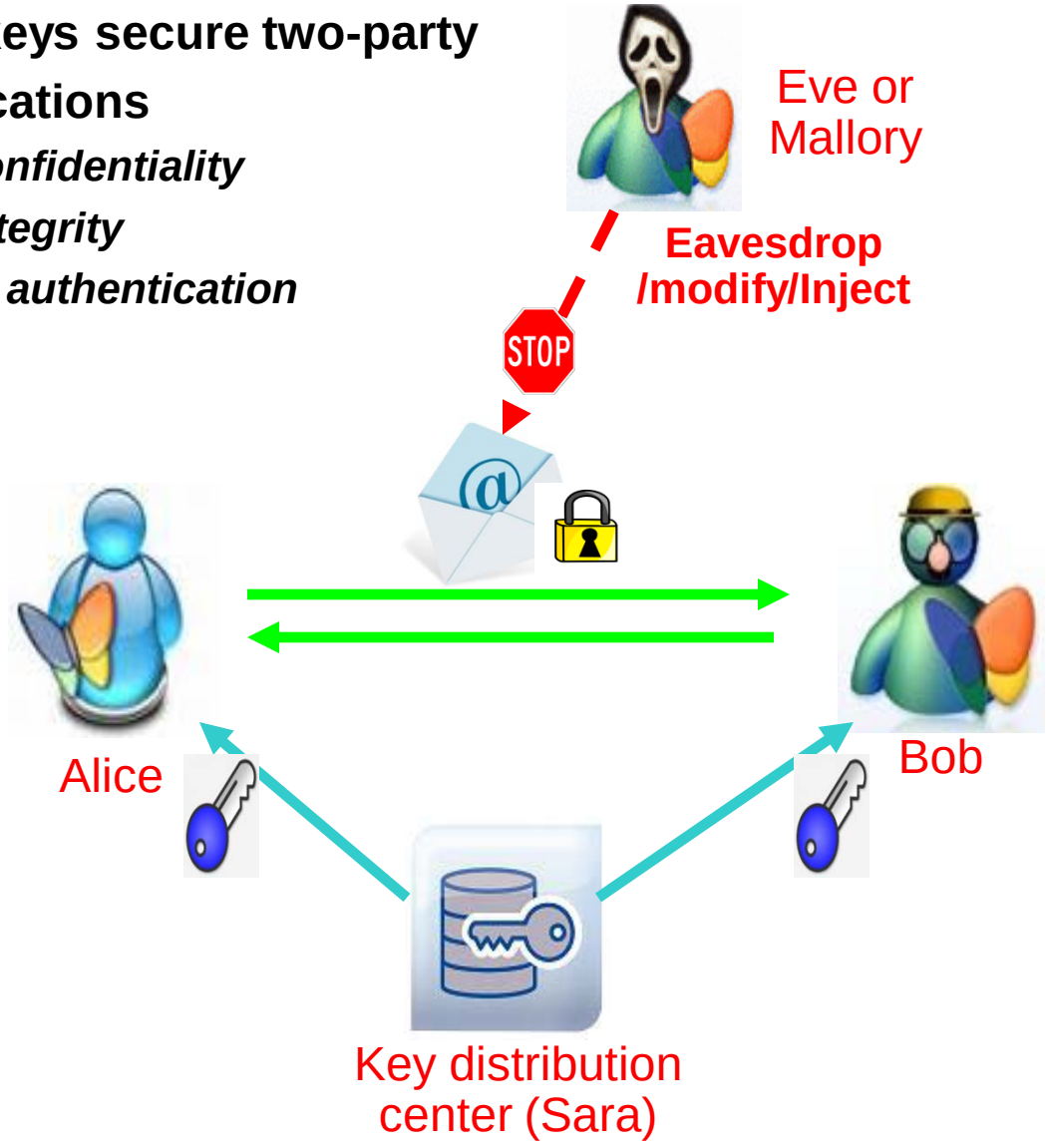
  Three phases:
  - **Specification of Protocols for Security to satisfy a policy**
  - **Analysis of Protocol Behavior when under attacks**
  - **Effect on overall performance if there were no attacks**

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

Two-party Communication, Pair-wise Key

- **Pairwise keys secure two-party communications**
  - ***Data confidentiality***
  - ***Data integrity***
  - ***Source authentication***



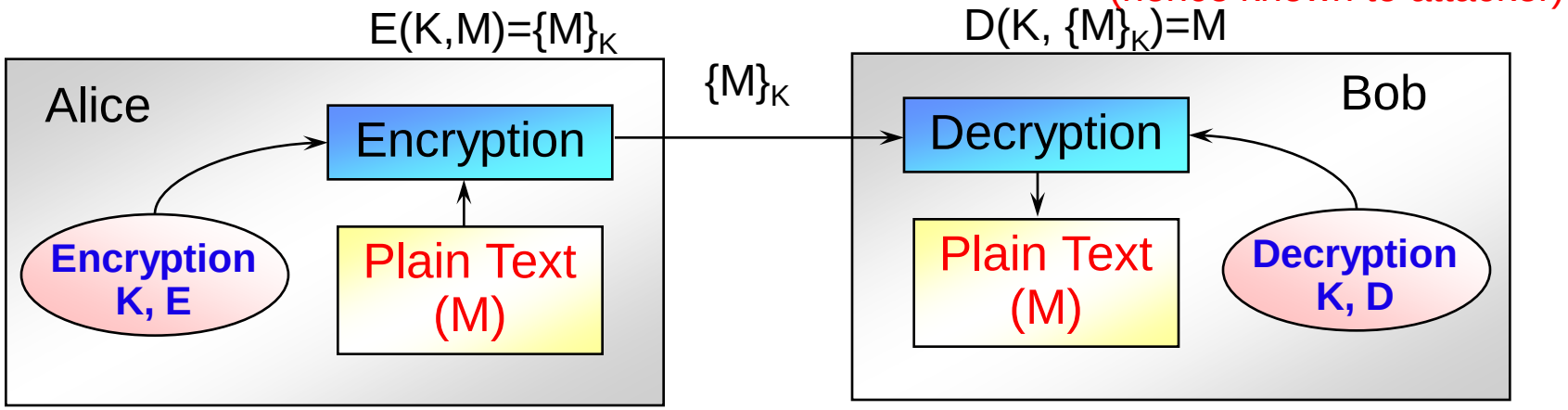
Cryptography Notations

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)
$\{M\}_K$	(Typical) Message M encrypted with key K
$[M]_K$	(Typical) Message M signed with key K

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.
  - E.g., DES (Data Encryption Standard) – 56 bit key operates on 64 bit blocks of data. Notation: $K_{AB}(M)$.
  - 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.
  - E.g., RSA / PGP keys – at least 512 b long

Code for E,D
is "open-source"
(hence known to attacker)

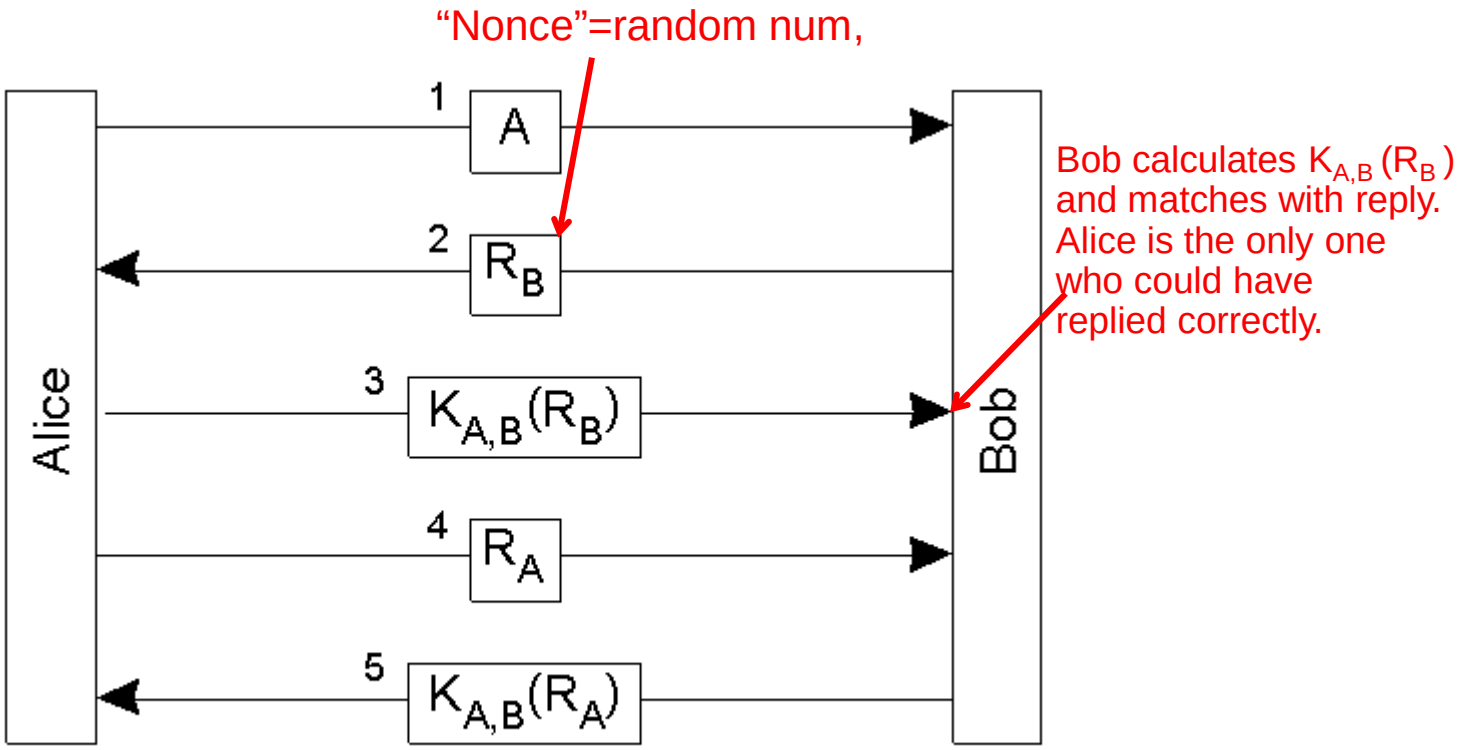


Authentication

- Use of cryptography to have two principals verify each others' identities.
  - Direct authentication: the server uses a shared secret key to authenticate the client.
  - Indirect authentication: a trusted authentication server (third party) authenticates and provides a ticket to an authenticated client.
  - The authentication server knows keys of principals and generates temporary shared keys.
    - E.g., Verisign servers
  - Shared versus public/private:
    - Shared reveals information to too many principles; may need key distribution and repudiation mechanisms
    - In electronic commerce or wide area applications, public/private key pairs are used rather than shared keys.
    - Public/private key encrypt/decrypt ops are costly
    - May use hybrid: pub/pri generates a shared key.
  - Presentation of next few protocols independent of which keying scheme, viz., shared or pub/priv

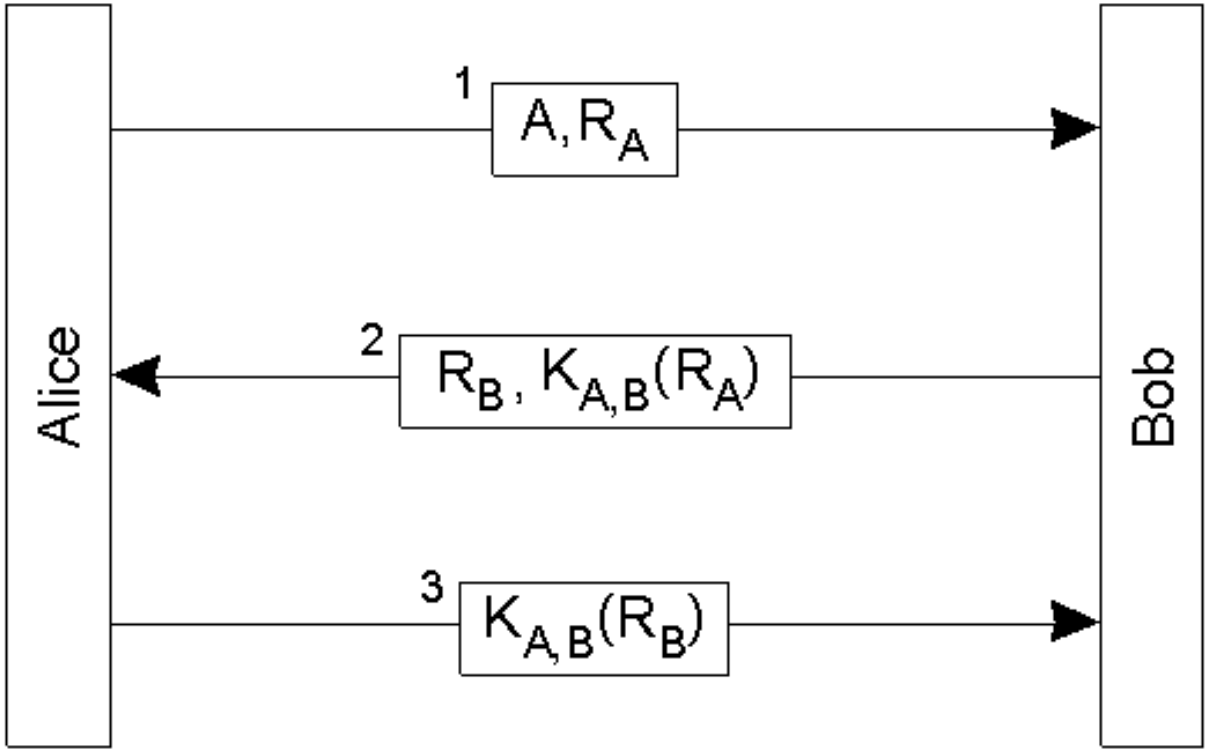
Direct Authentication

- **Authentication based on a shared secret key.**



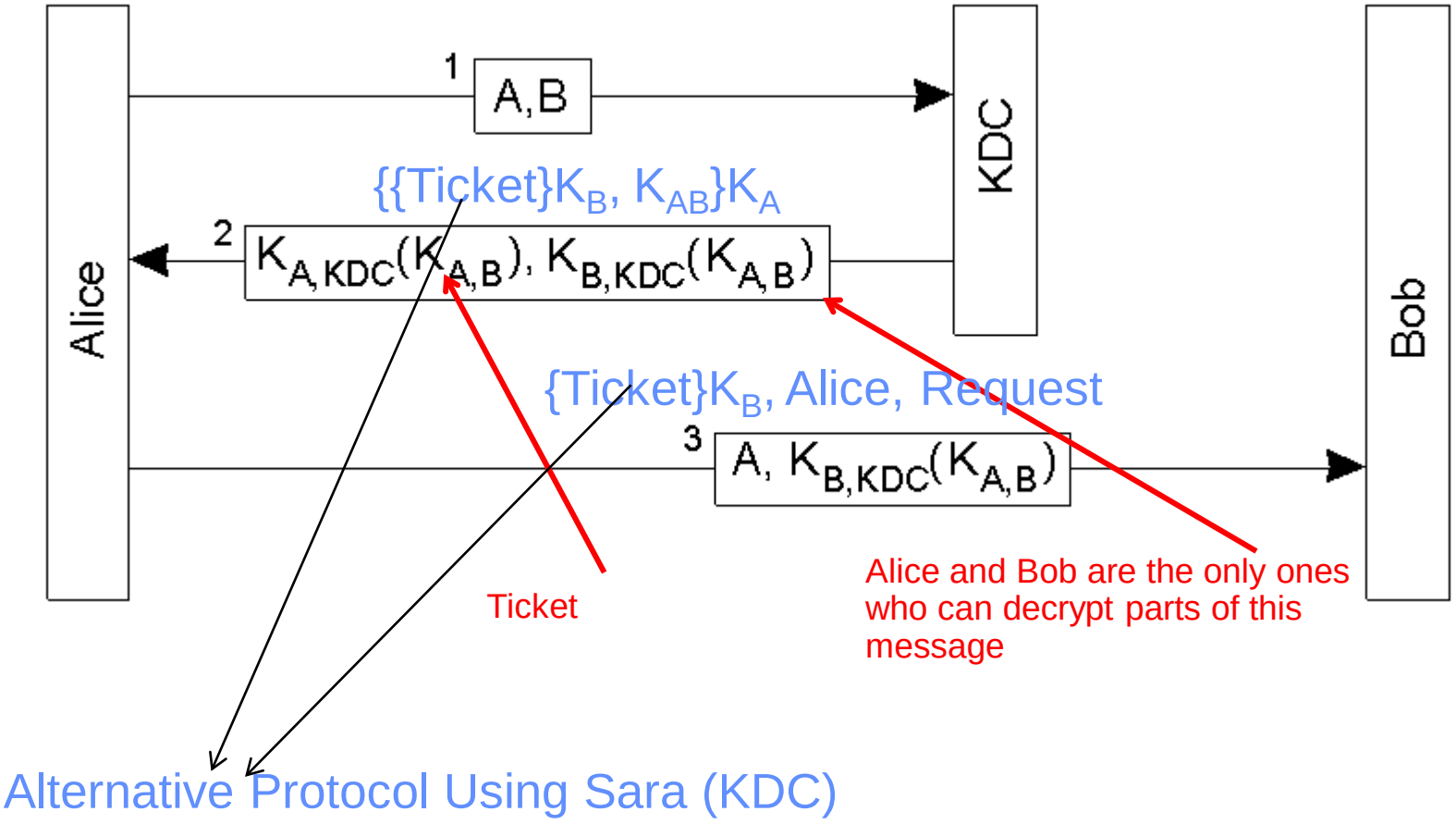
"Optimized" Direct Authentication

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

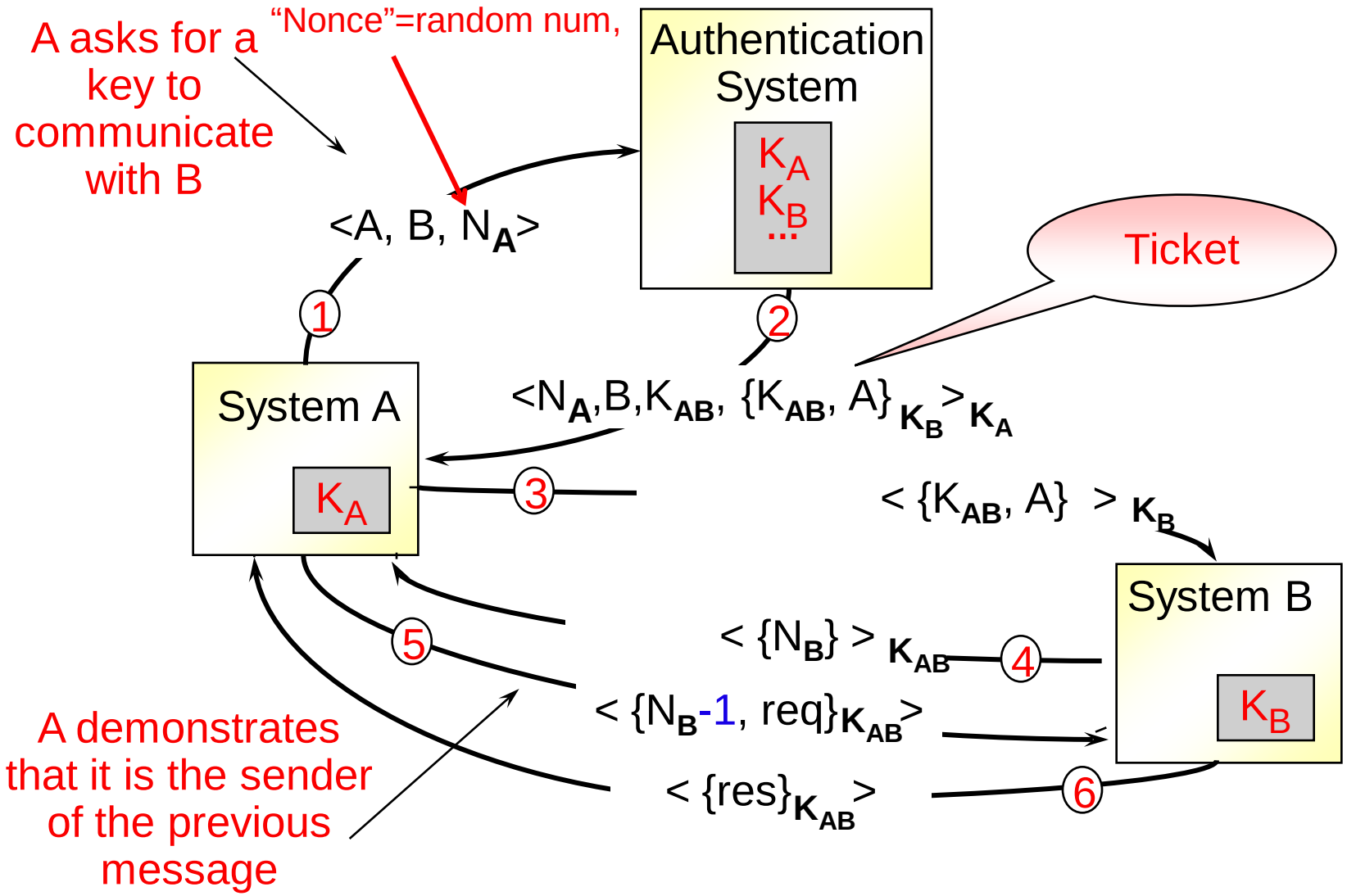


Authentication Using a Key Distribution Center

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

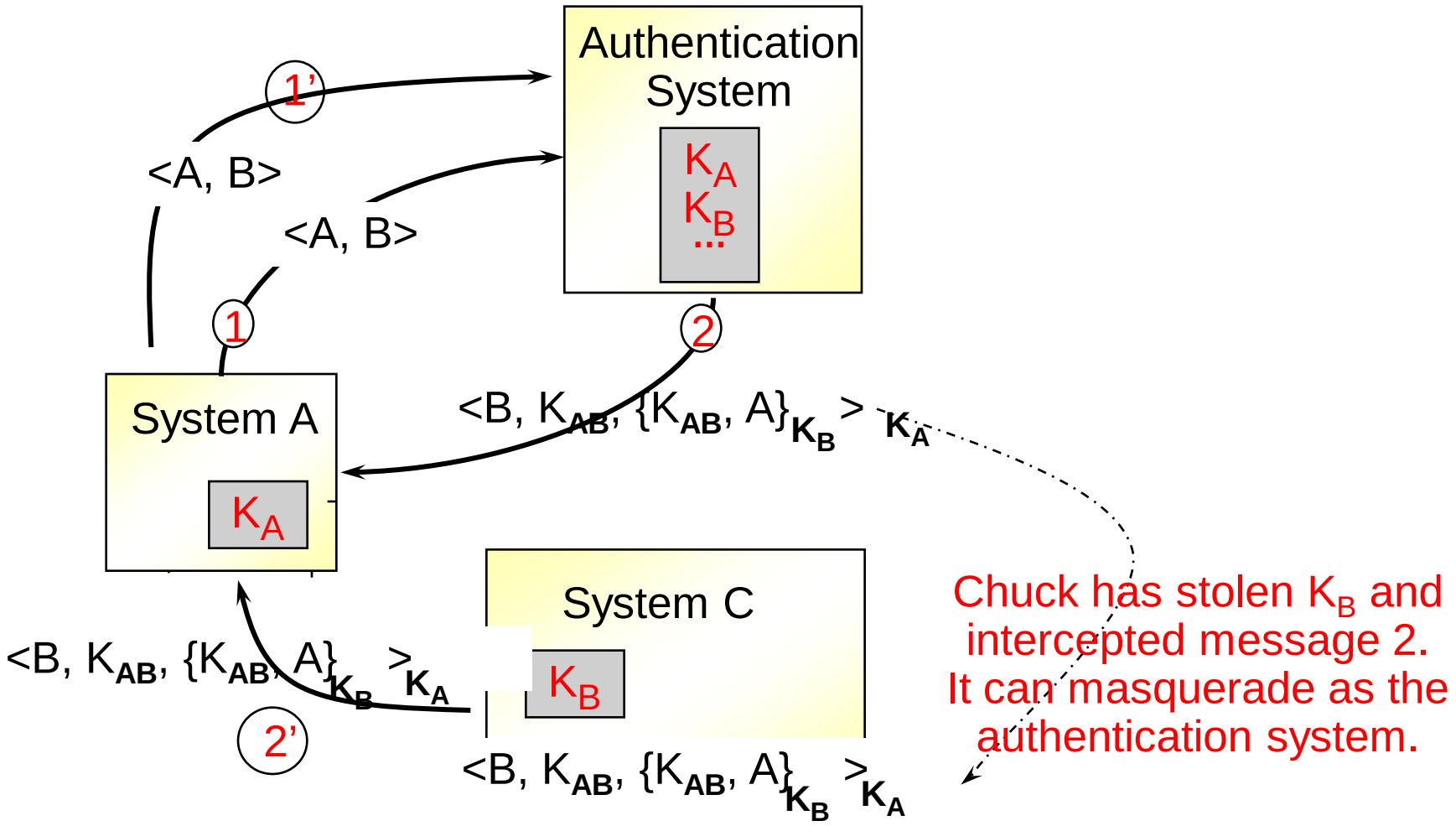


Needham-Schroeder Authentication



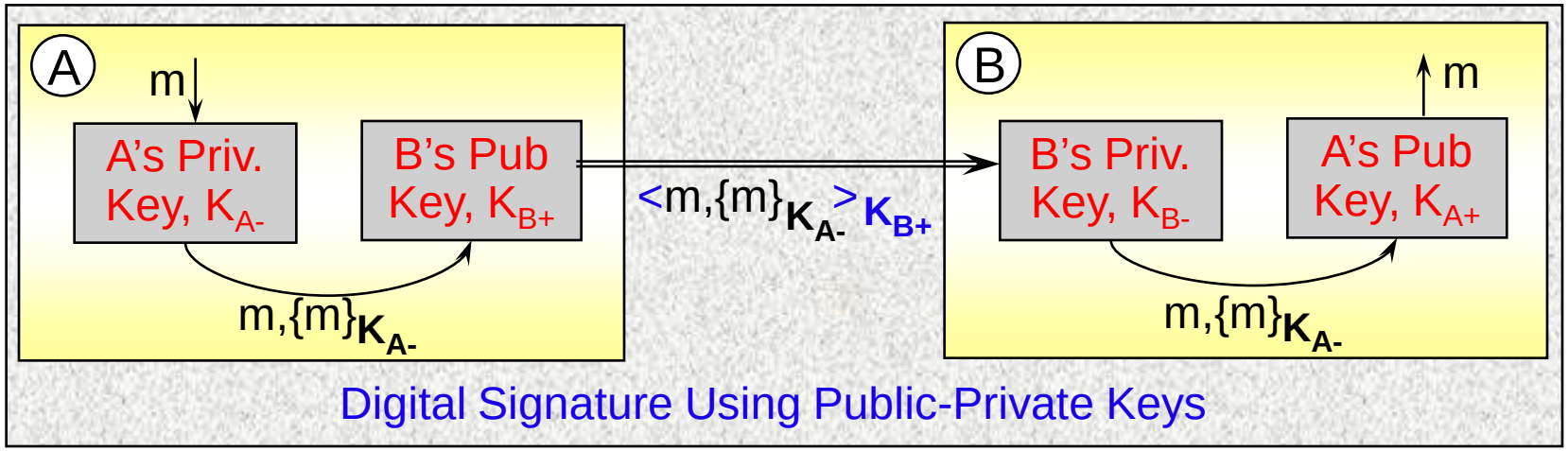
Why Do We Need Nonce N_A in Message 1?

Because we need to relate message 2 to message 1

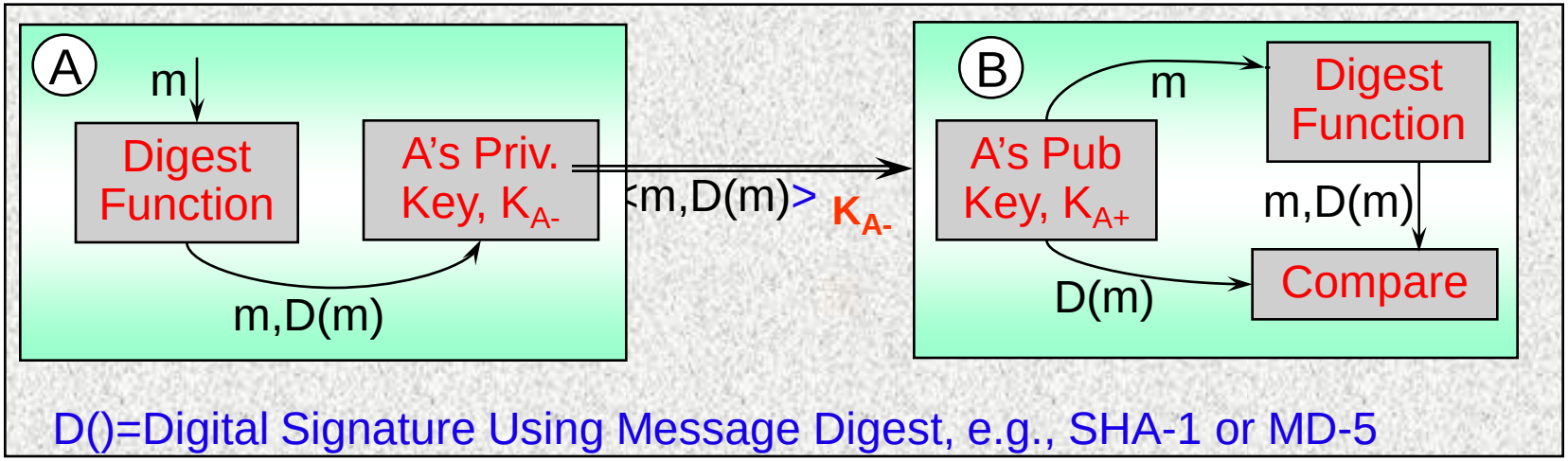


Digital Signatures

- Signatures need to be authentic, unforgeable, and non-repudiable.



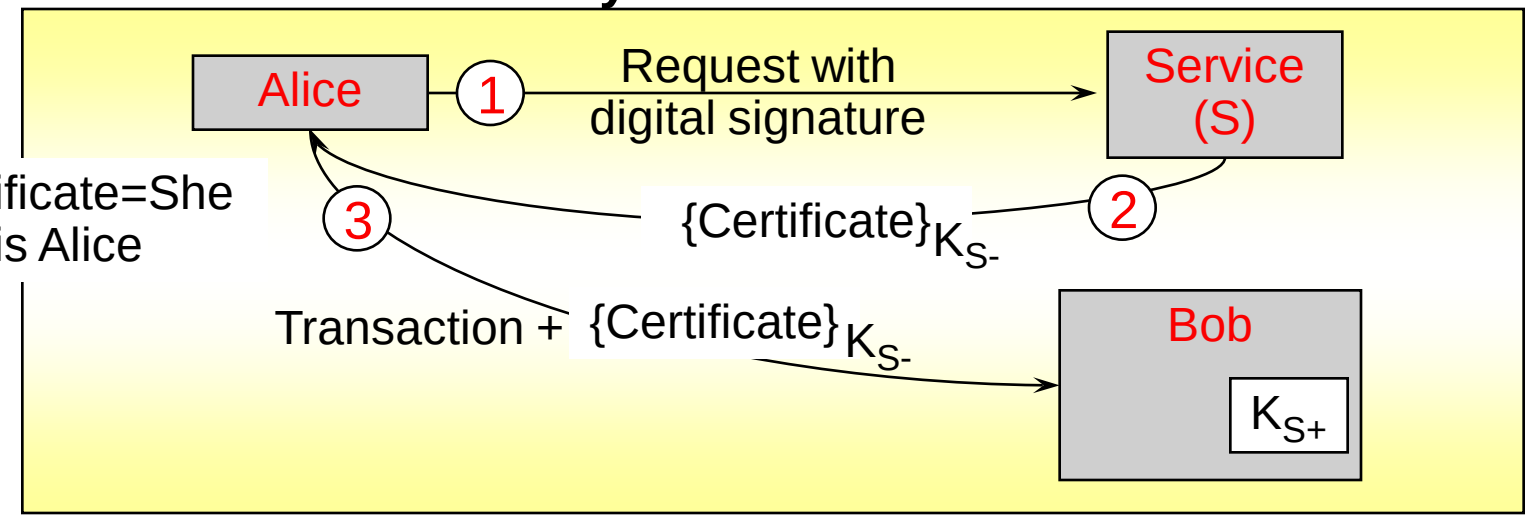
Digital Signature Using Public-Private Keys



D()=Digital Signature Using Message Digest, e.g., SHA-1 or MD-5

Digital Certificates

- A digital certificate is a statement signed by a third party principal,
  - e.g., Verisign Certification Authority (CA)
- 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



Alice's Bank Account Certificate

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

Public-Key Certificate for Bob's Bank

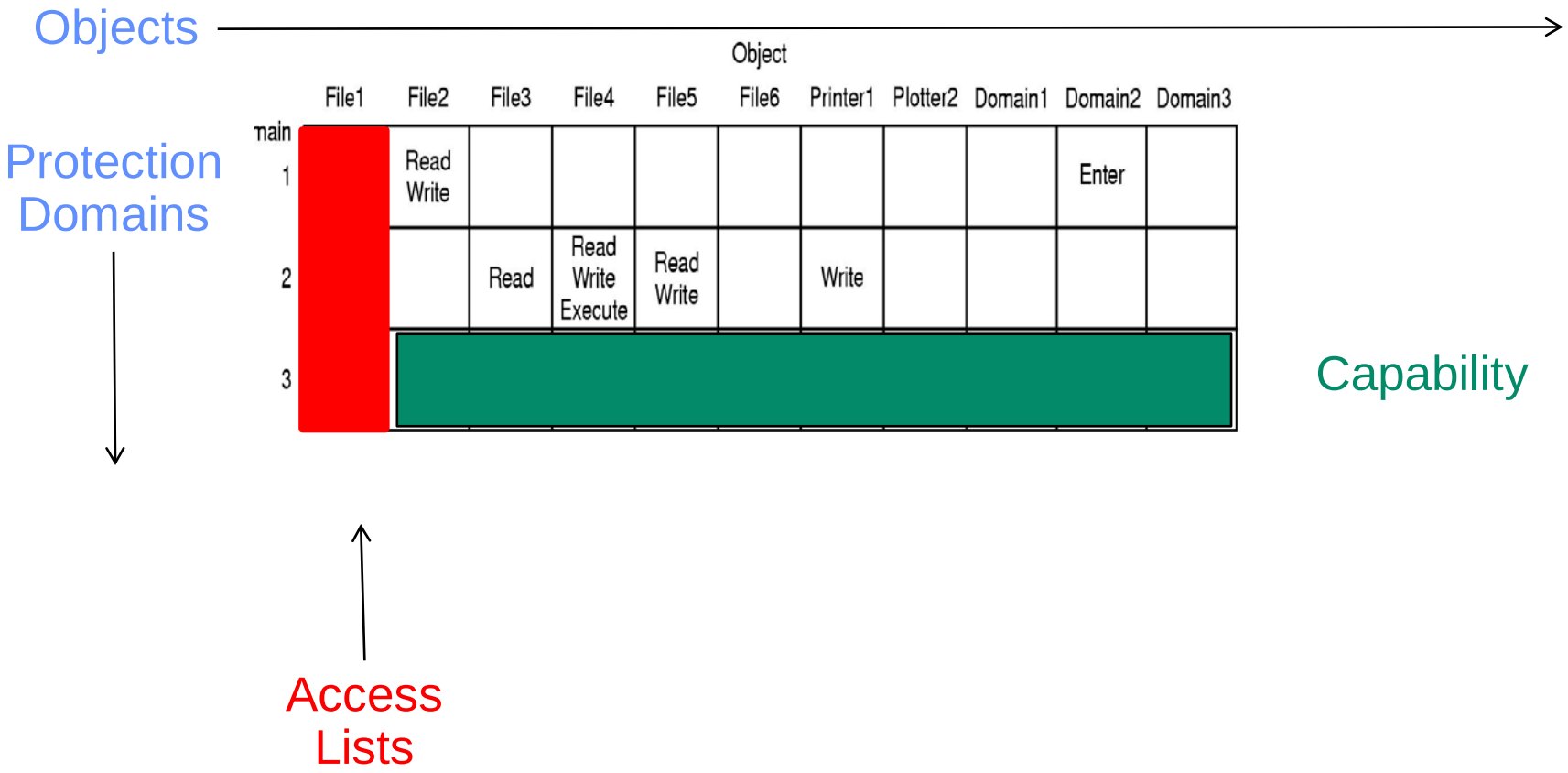
1. *Certificate type*	Public key
2. *Name*	Bob's Bank
3. *Public key*	K_{Bpub}
4. *Certifying authority*	Fred – The Bankers Federation
5. *Signature*	$\{Digest(field\ 2 + field\ 3)\}\ K_{Fpriv}$

Eventually K_F-, K_F+ have to be obtained reliably.

Access Control: Protection Domain

- **A computer system is a set of processes and objects**
- **Processes and objects have unique names**
- **Objects are abstract data types with well-defined operations**
- **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 Privilege *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.**

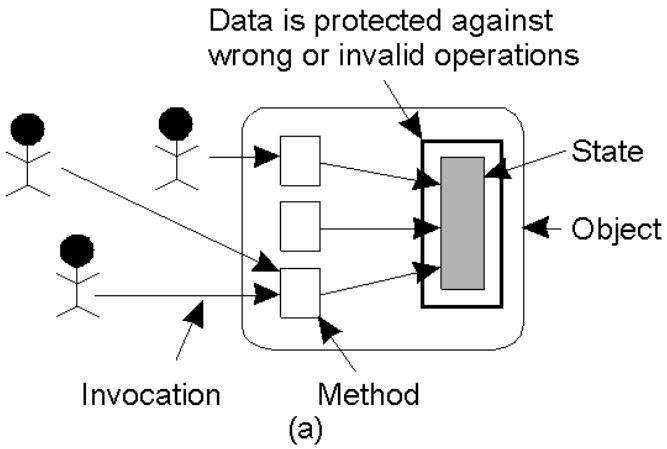
Access Matrix

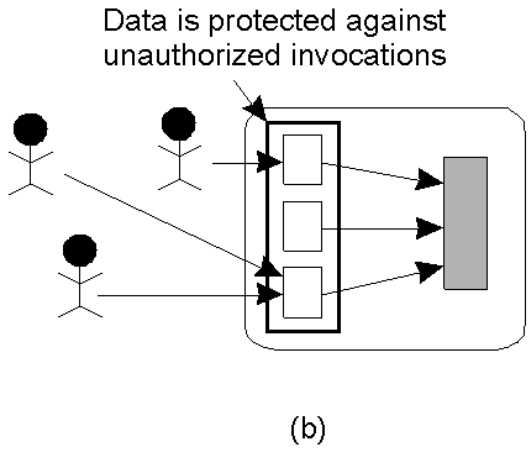


Authorization: Access Control

- Control of access to resources of a remote server.
- A basic form of remote 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 (e.g., maintained at server)
  - 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.
  - Capability List = row in access control matrix, i.e., per-principal list.

Focus of Access Control



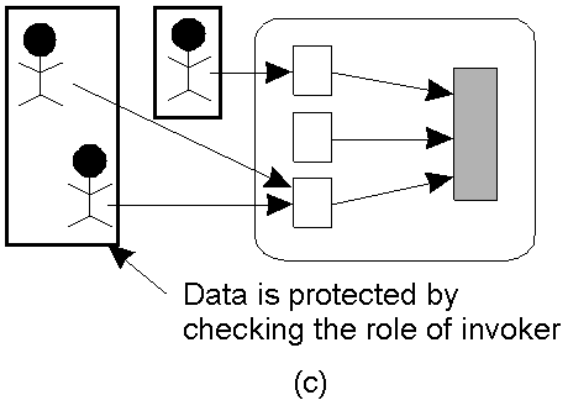


- **Three approaches for protection against security threats**

a) **Protection against invalid operations**

b) **Protection against unauthorized invocations**

c) **Protection against unauthorized users**

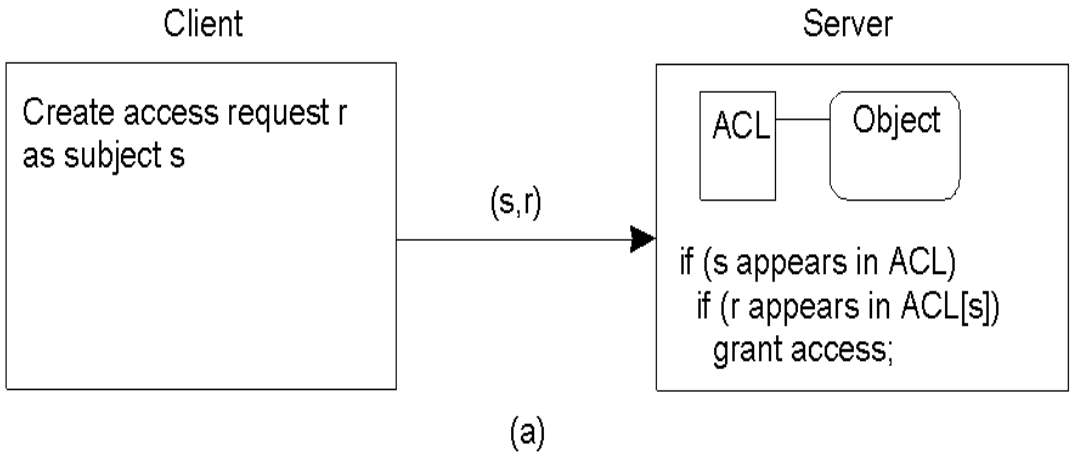


Access Control Matrix

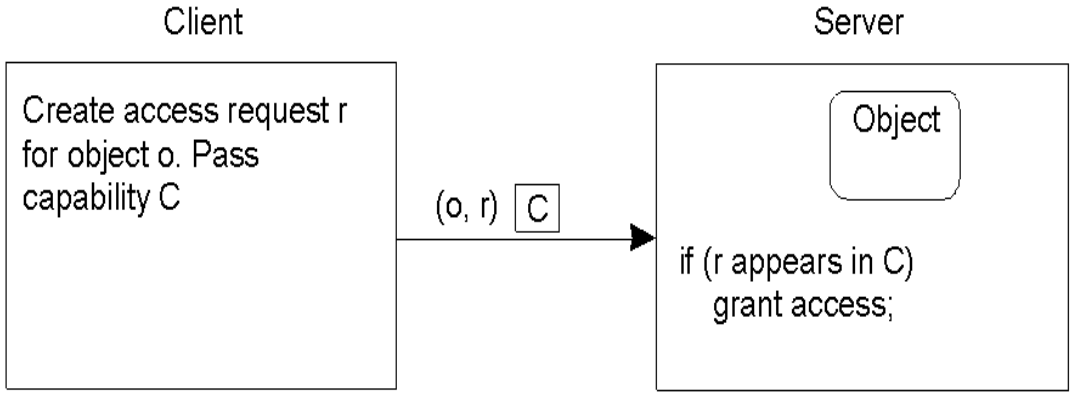
Comparison between ACLs and capabilities for protecting objects.

a) Using an ACL

b) Using capabilities.



(a)



(b)

Summary

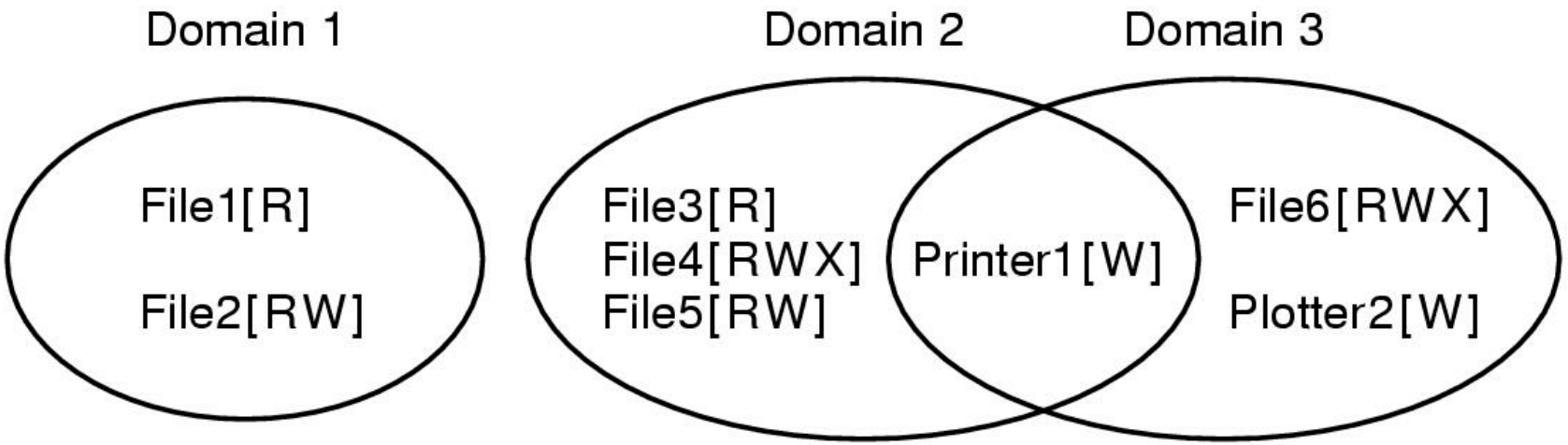
- **Variety of attacks – C. I. A.**
- **Policy vs. Mechanism**
- **Cryptography – symmetric or public-private**
- **Authentication, Authorization, Auditing.**
- **Needham-Schroeder Authentication Protocol**
- **Access Control**

Needham–Schroeder Secret-key Authentication Protocol

Variant used for authentication in Windows 2K (slightly modified – nonce added to msg 3)

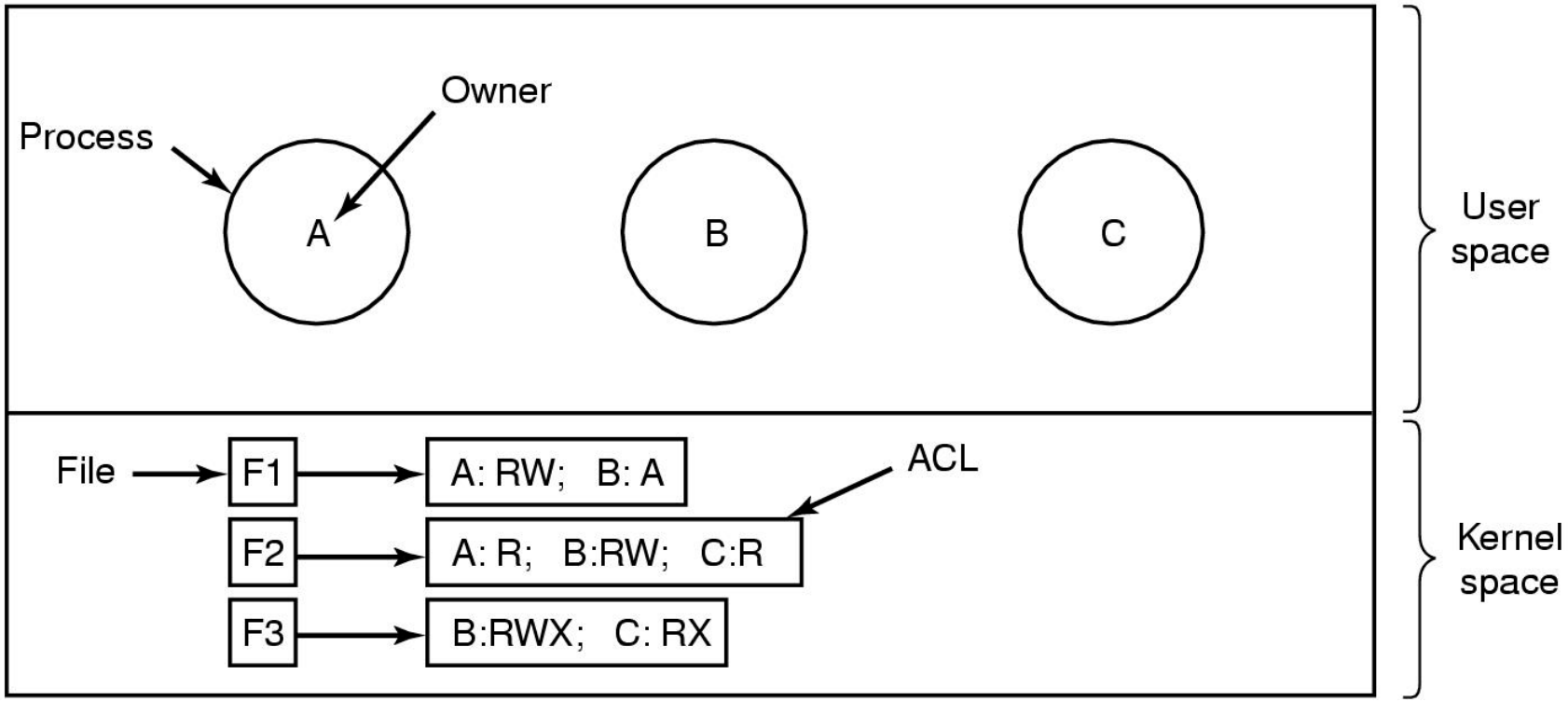
Header	*Message*	*Notes*
1. A->S:	A, B, N_A	A requests S to supply a key for communication with B.
2. S->A:	$\{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_B}\}_{K_A}$	S returns a message encrypted in A's secret key, containing a newly generated key K_{AB} and a 'ticket' encrypted in B's secret key. The nonce N_A 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.

Protection Mechanisms



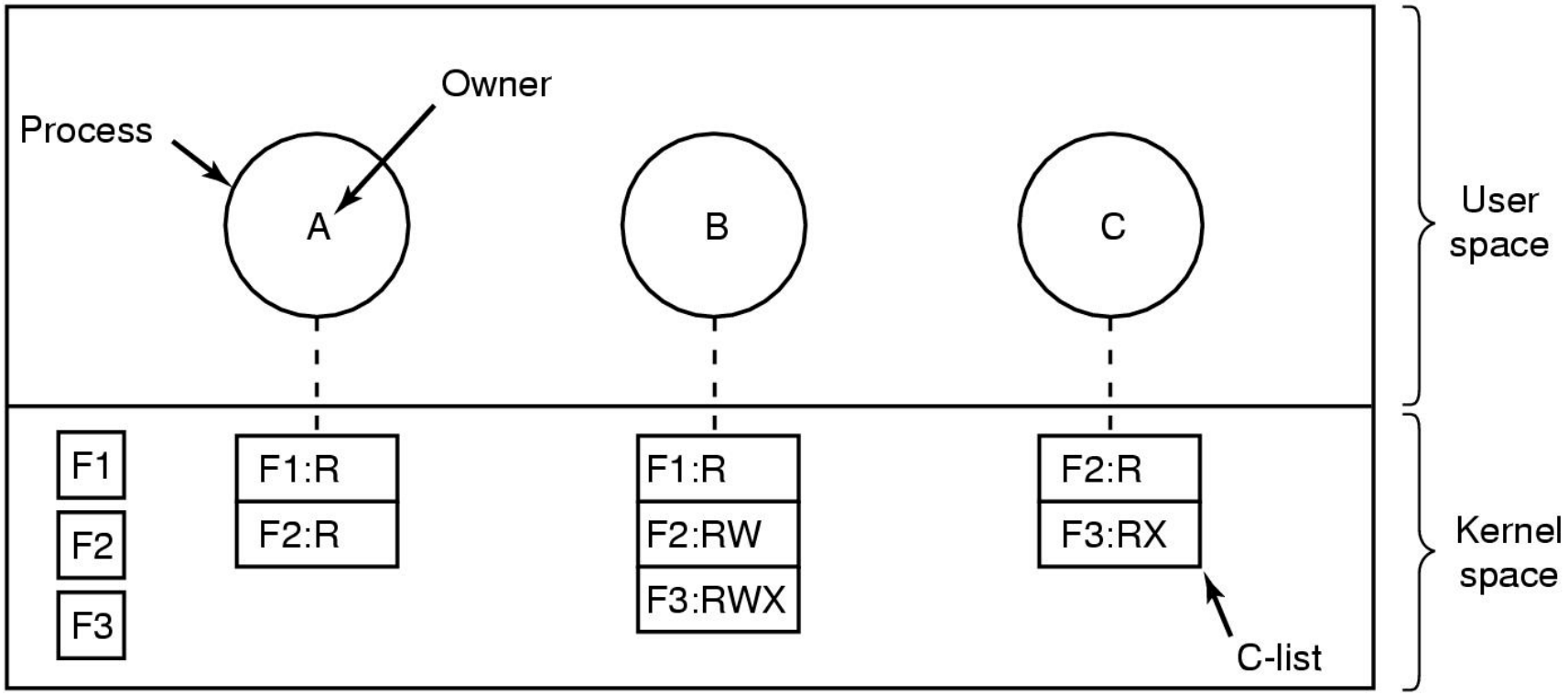
Examples of three protection domains

Access Control Lists (Example)



Use of access control lists to manage file access

Capabilities (Example)



Each process has a capability list

Access Control

- The server may issue to each principal a list of capabilities.
  - A capability list is a row in the Access Control Matrix.
- Notion of protection domain for a collection of processes:
  - A protection domain is a set of (object, access rights) pairs kept by a server.
  - A protection domain is created for each principal when it starts
  - Unix: each (uid,gid) pair spans a protection domain, e.g., user parts of two processes with same (uid, gid) pair have identical access rights.
  - Whenever a principal requests an operation to be carried out on an object, the access control monitor checks if the principal belongs to the object's 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.