Module 9 - Security

Issues

- Separation of
  - Security policies
    - → Precise definition of which entities in the system can take what actions
  - Security mechanism
    - → Means of enforcing that policy
- Distributed system security
  - Communication between users or processes that may be on different machines
    - → Secure channel (authentication, message integrity, confidentiality)
  - Authorization to ensure that a user or process performs only those actions that is allowed under the security policy
    - → Access Control
- Security services
  - What any distributed system should provide as part of its infrastructure to enable the implementation of different policies.

Basic Concepts

- Security: attempt to protect the services and data it offers against security threats.
- Confidentiality: the property of a computer system whereby its information is disclosed only to authorized parties
- Integrity: the characteristic that alterations to a system's assets can be made only in an authorized way.

Types of Threats

- Interception
  - An unauthorized party has gained access to a service or data
- Interruption
  - Services or data become unavailable, unusable, destroyed and so on.
- Modification
  - Unauthorized changing of data or tampering with a service so that it no longer adheres to its original specifications
- Fabrication
  - Refers to the situation in which additional data or activity are generated that would normally not exist

Methods of Attack

- Eavesdropping
  - Obtaining copies of messages without authority
- Masquerading
  - Sending or receiving messages using the identity of another principal without their authority
- Message tampering
  - Intercepting messages and altering their contents before passing them on to the intended recipient
  - Man-in-the-middle attack
- Replaying
  - Storing intercepted messages and sending them out at a later time
- Denial of service
  - Flooding a communication channel or a system resource with messages in order to deny access for others

Historical context: the evolution of security needs

	1965-75	*1975-89*	*1990-99*	*Current*
Platforms	Multi-user timesharing computers	Distributed systems based on local networks	The Internet, wide-area services	The Internet + mobile devices
Shared resources	Memory, files	Local services (e.g. NFS), local networks	Email, web sites, Internet commerce	Distributed objects, mobile code
Security requirements	User identification & authentication	Protection of services	Strong security for commercial transactions	Access control for individual objects, secure mobile code
Security management environment	Single authority, single authorization database (e.g. /etc/ passwd)	Single authority, delegation, replicated authorization databases (e.g. NIS)	Many authorities, no network-wide authorities	Per-activity authorities, groups with shared responsibilities

From Coulouris, Dollimore and Kindberg, Distributed Systems: Concepts and Design, 3rd ed.

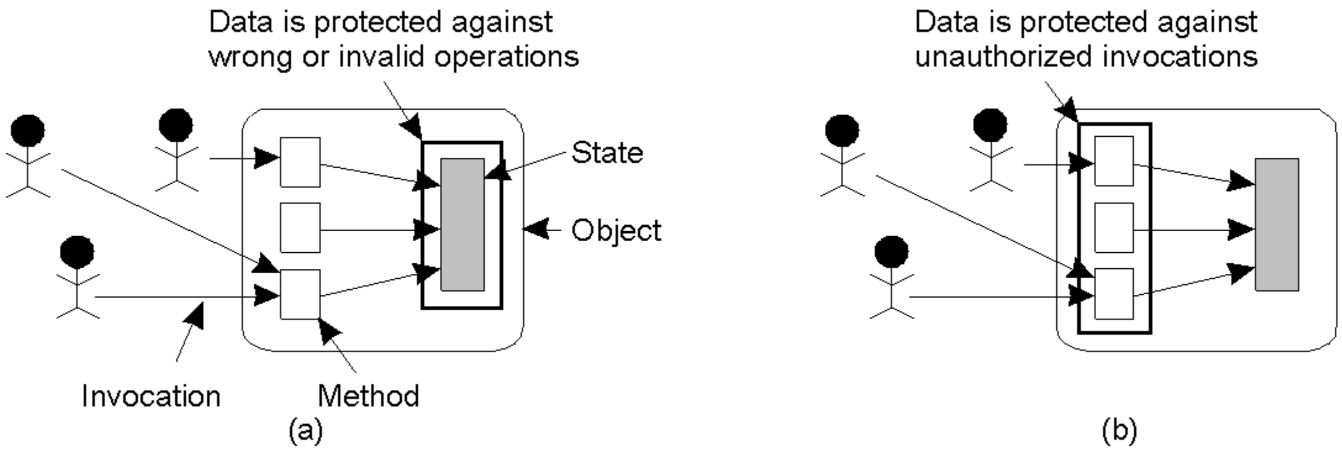
Security Mechanisms

- Encryption
  - transform data into something an attacker cannot understand
  - provides a means to implement confidentiality
  - provides support for integrity
- Authentication
  - verify the claimed identity of a user, client, server and so on
- Authorization
  - check whether the client is authorized to perform the action requested
- Auditing
  - auditing tools are used to trace which clients accessed what, and which way

Security Services - Design Issues

- Focus of Control
  - Decide the focus of control: data, operations or users
- Layering of security mechanisms
  - Decide at which level security mechanisms should be placed
- Simplicity
  - Simplicity will contribute to the trust that end users will put into the application and, more importantly, will contribute to convincing the designers that the system has no security holes.

Focus of Control

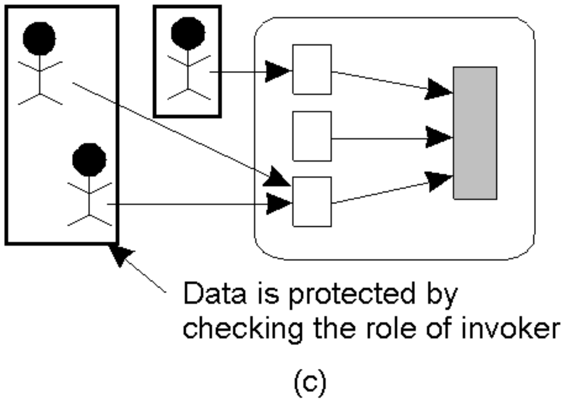

Data is protected against wrong or invalid operations
State
Object
Invocation
Method
(a)
Data is protected against unauthorized invocations
(b)

Three approaches for protection against security threats

a) Protection against invalid operations

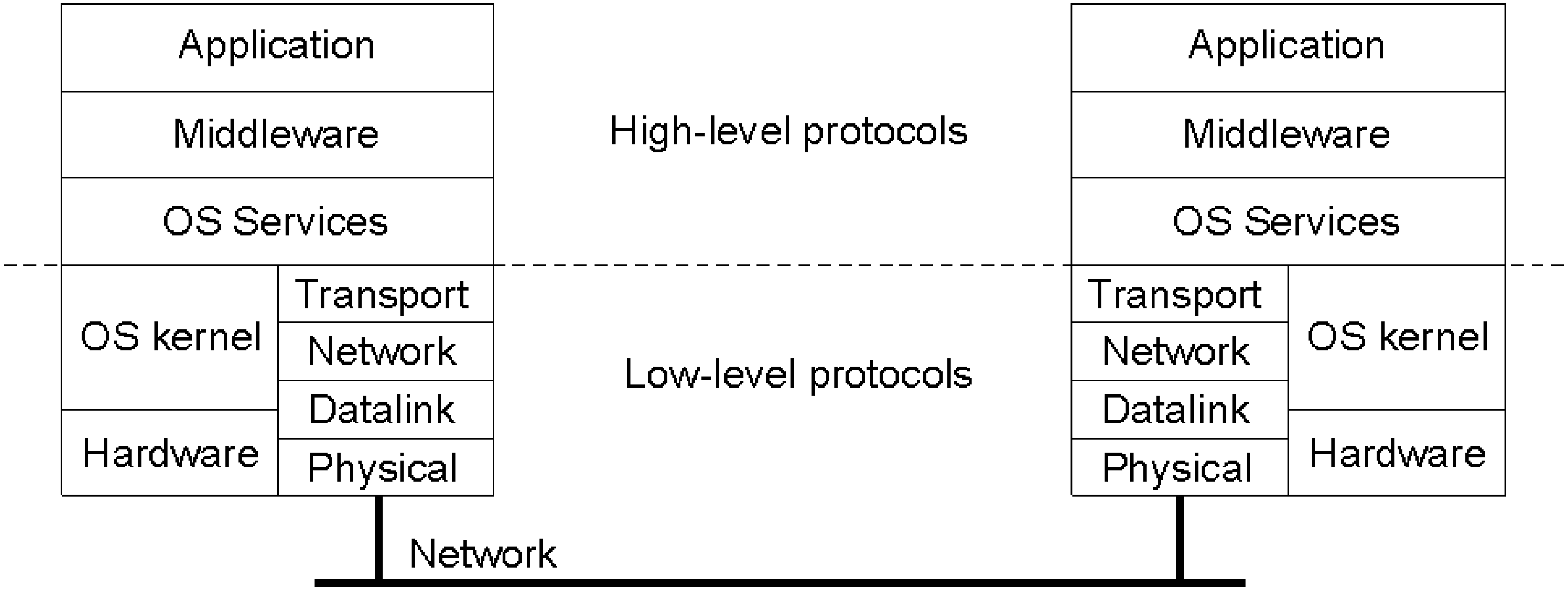
b) Protection against unauthorized invocations

c) Protection against unauthorized users



From Tanenbaum and van Steen, Distributed Systems: Principles and Paradigms
© Prentice-Hall, Inc. 2002

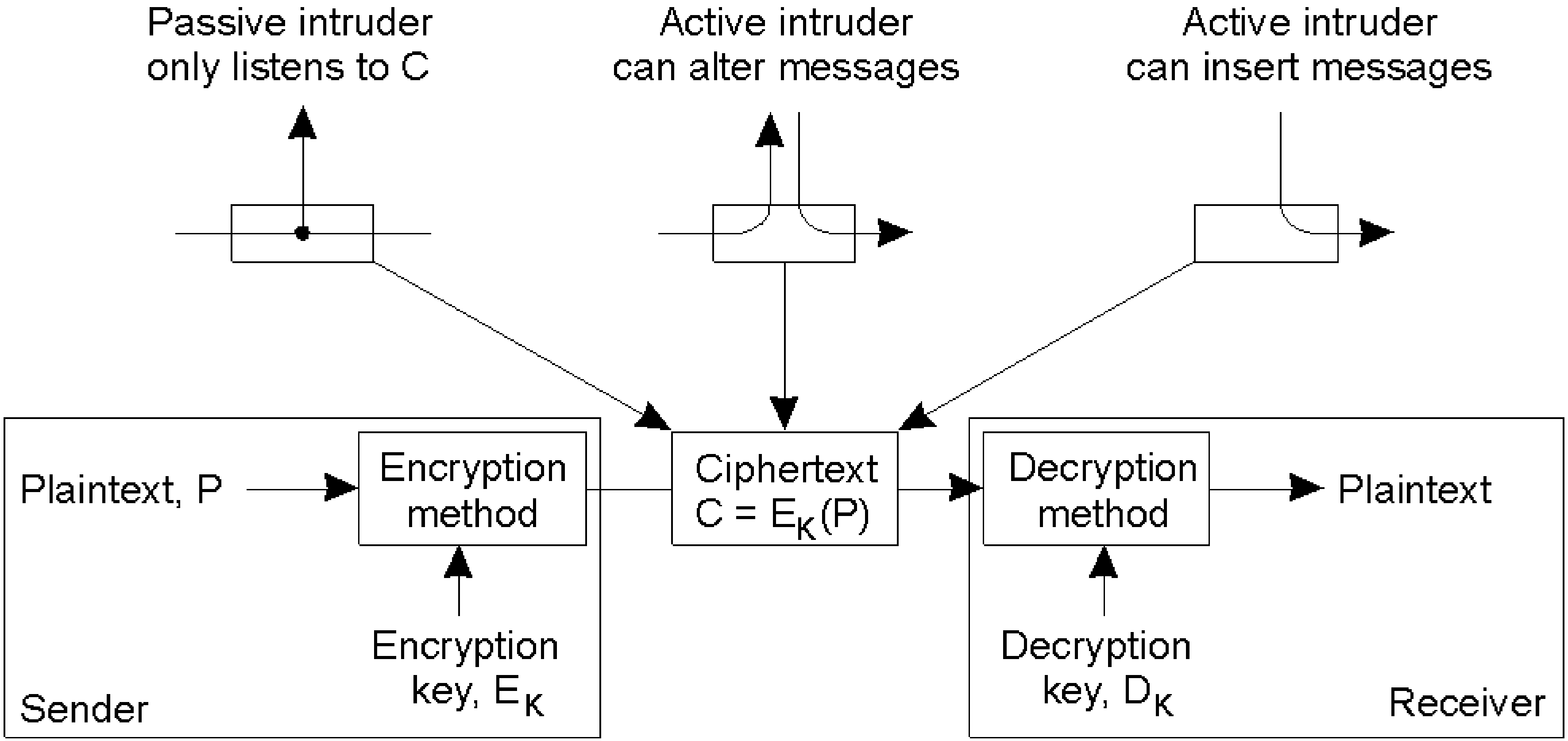
Layering of Security Mechanisms



The logical organization of a distributed system into several layers.

From Tanenbaum and van Steen, Distributed Systems: Principles and Paradigms
© Prentice-Hall, Inc. 2002

Cryptography



The three different attacks that we need to protect against, and for which encryption helps.

From Tanenbaum and van Steen, Distributed Systems: Principles and Paradigms
© Prentice-Hall, Inc. 2002

Cryptography

- Symmetric cryptosystem
  - The same key is used to encrypt and decrypt a message
    - → $P = D_K(E_K(P))$
  - Also referred to as secret-key or shared-key systems
  - Example: DES (Data Encryption Standard)
- Asymmetric cryptosystem
  - The keys for encryption and decryption are different, but together form a unique pair
    - → $P = D_{K_D}(E_{K_E}(P))$
  - One of the keys is kept private the other is made public (which is public and which is private depends on how the keys are used)
  - Also referred to as public-key systems
  - Example: RSA (Rivest, Shamir, and Adleman)

Cryptography (2)

- Hash functions
  - A hash function takes a message *m* of arbitrary length as input and produces a bit string *h* having a fixed length as output
    - → $h = H(m)$
  - Have a number of properties
    - → One-way functions: computationally infeasible to find *m* that corresponds to a known *h*.
    - → Weak collision resistance: given *m* and $h = H(m)$, it is computationally infeasible to find $m' \neq m$ such that $H(m) = H(m')$
    - → Strong collision resistance: given only *H*, it is computationally infeasible to find two different input values *m* and *m'*, such that $H(m) = H(m')$
  - Example: MD5

Secure Channels

- Secure communication requires authentication of the communicating parties, but also ensuring message integrity and possibly confidentiality as well.
- A secure channel protects senders and receivers against interception, modification, and fabrication of messages.
- It does not necessarily protect against interruption.
- Protecting messages against interception is done by ensuring confidentiality
- Protecting messages against modification and fabrication is done through protocols for mutual authentication and message integrity.

Authentication

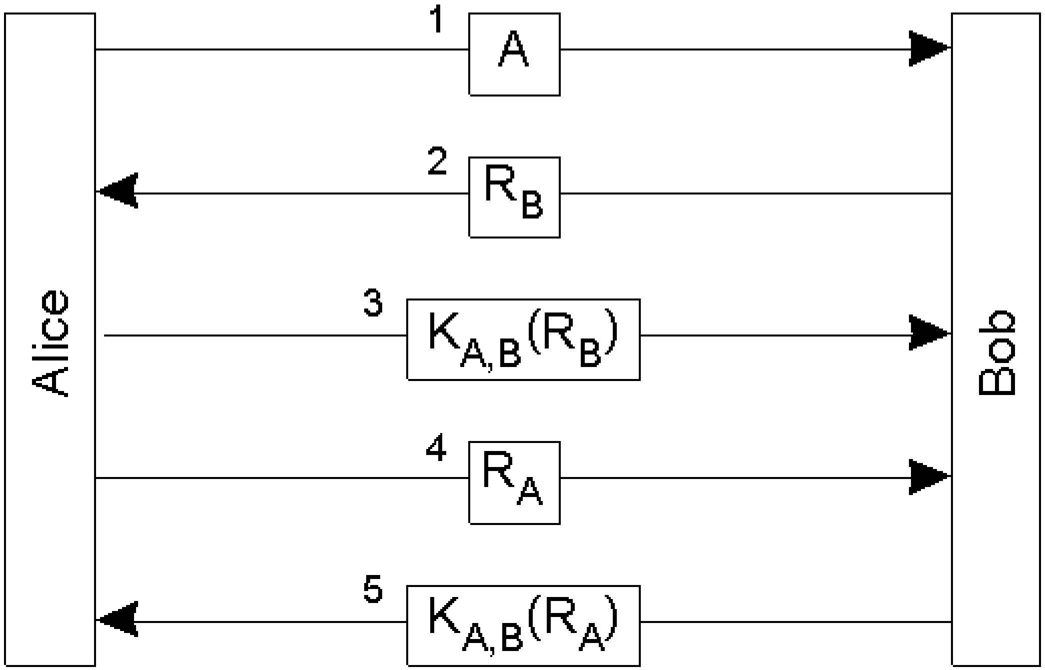
- Authentication and message integrity cannot do without each other.
- The combination works as follows:
  - Alice starts by sending a message to Bob to set up a channel
  - Once the channel has been set up, Alice knows for sure that she is talking to Bob, and Bob knows for sure that he is talking to Alice, they can exchange messages
  - To subsequently ensure integrity it is common practice to use secret-key cryptography by means of session keys.

Notation	**Description**
$K_{A,B}$	Secret key shared by A and B
K_A^+	Public key of A
K_A^-	Private key of A

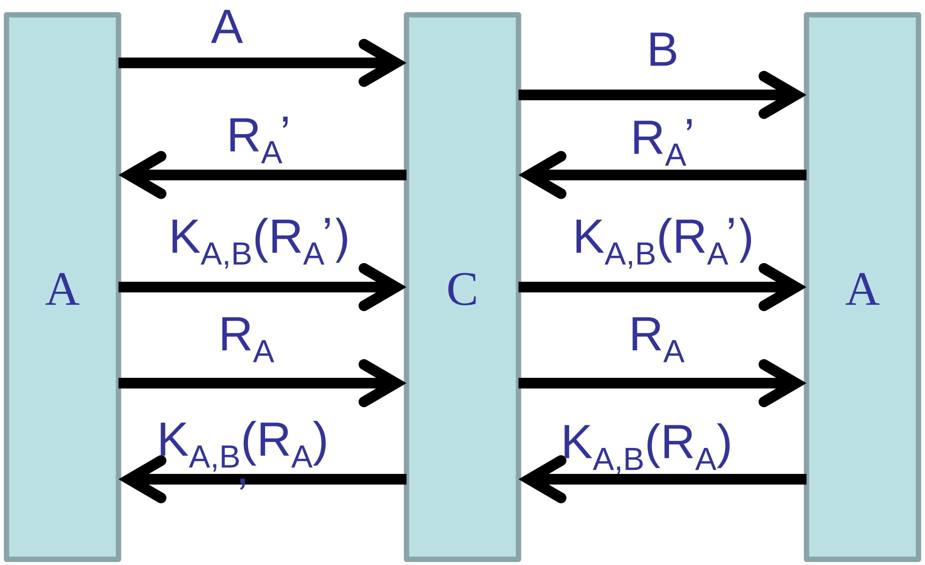
From Tanenbaum and van Steen, Distributed Systems: Principles and Paradigms
© Prentice-Hall, Inc. 2002

Authentication

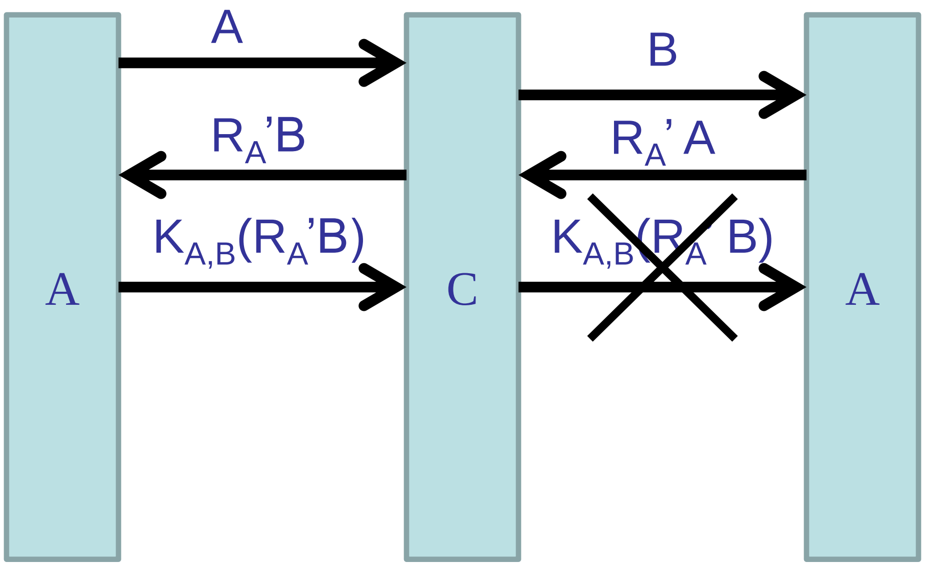
- Authentication based on a shared secret key
- Also known as challenge-response protocol



From Tanenbaum and van Steen, Distributed Systems: Principles and Paradigms
© Prentice-Hall, Inc. 2002



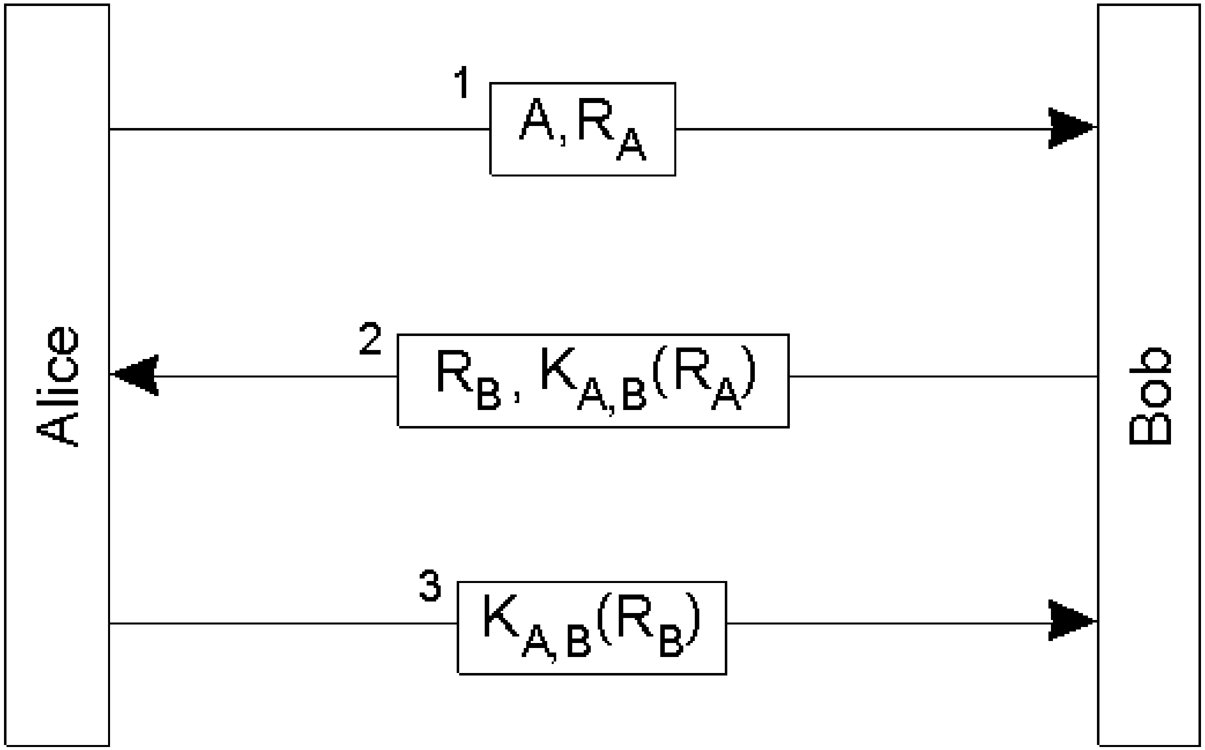
- Chuck can impersonate Bob by initiating a second channel with Alice.
- Requires that Alice is willing to accept channel requests.
- Can be defeated by including the challenger's name as part of the challenge.



- Chuck can impersonate Bob by initiating a second channel with Alice.
- Requires that Alice is willing to accept channel requests.
- Can be defeated by including the challenger's name as part of the challenge.

Authentication (2)

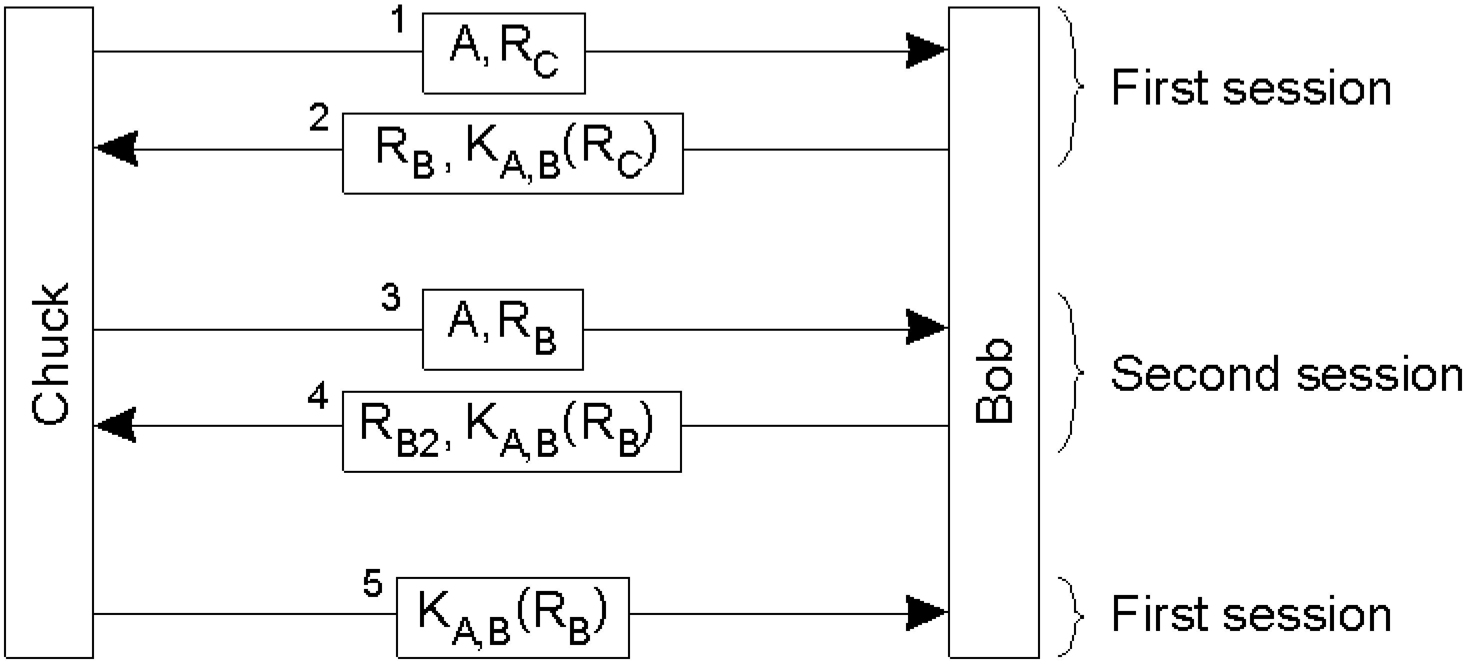
- Consider this “optimization”: Authentication based on a shared secret key, but using three instead of five messages.



From Tanenbaum and van Steen, Distributed Systems: Principles and Paradigms
© Prentice-Hall, Inc. 2002

Authentication (3)

- The reflection attack.

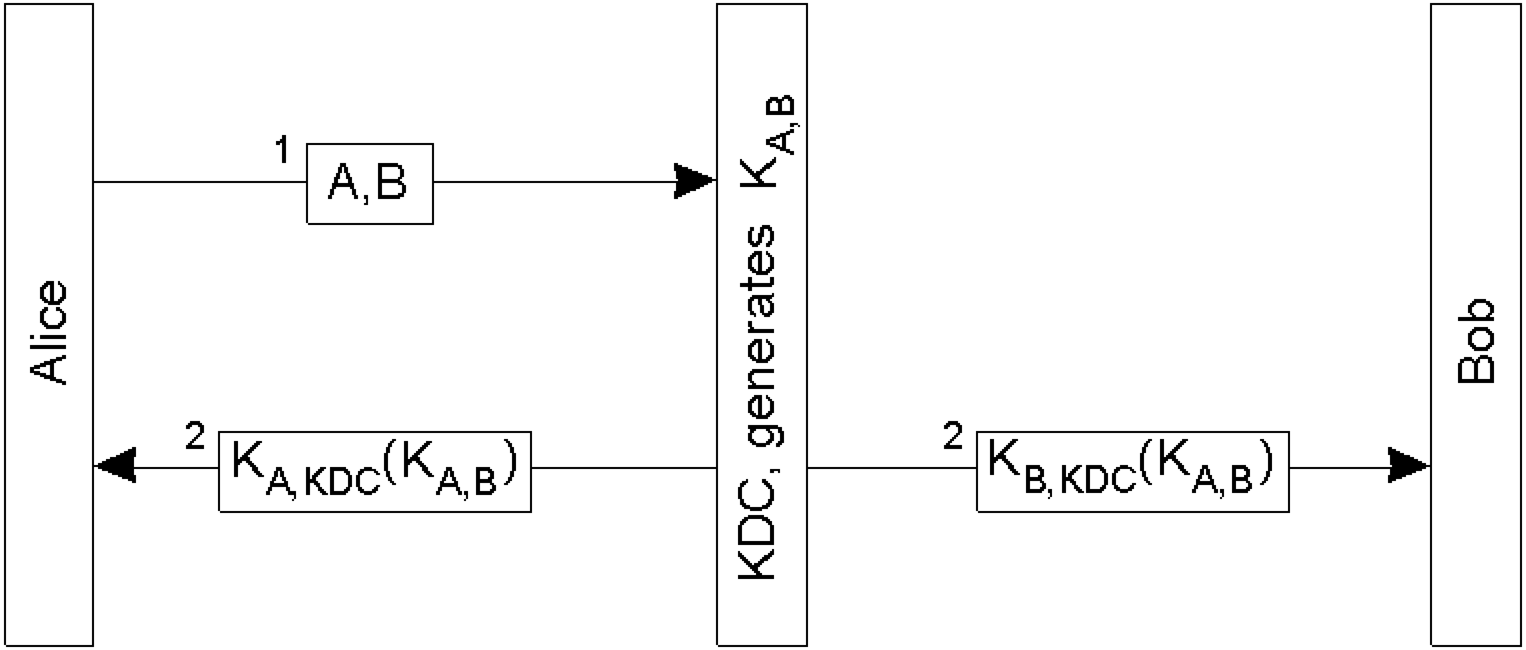


- Tweaking an existing protocol to improve its performance, can easily affect its correctness

From Tanenbaum and van Steen, Distributed Systems: Principles and Paradigms
© Prentice-Hall, Inc. 2002

Authentication Using a KDC

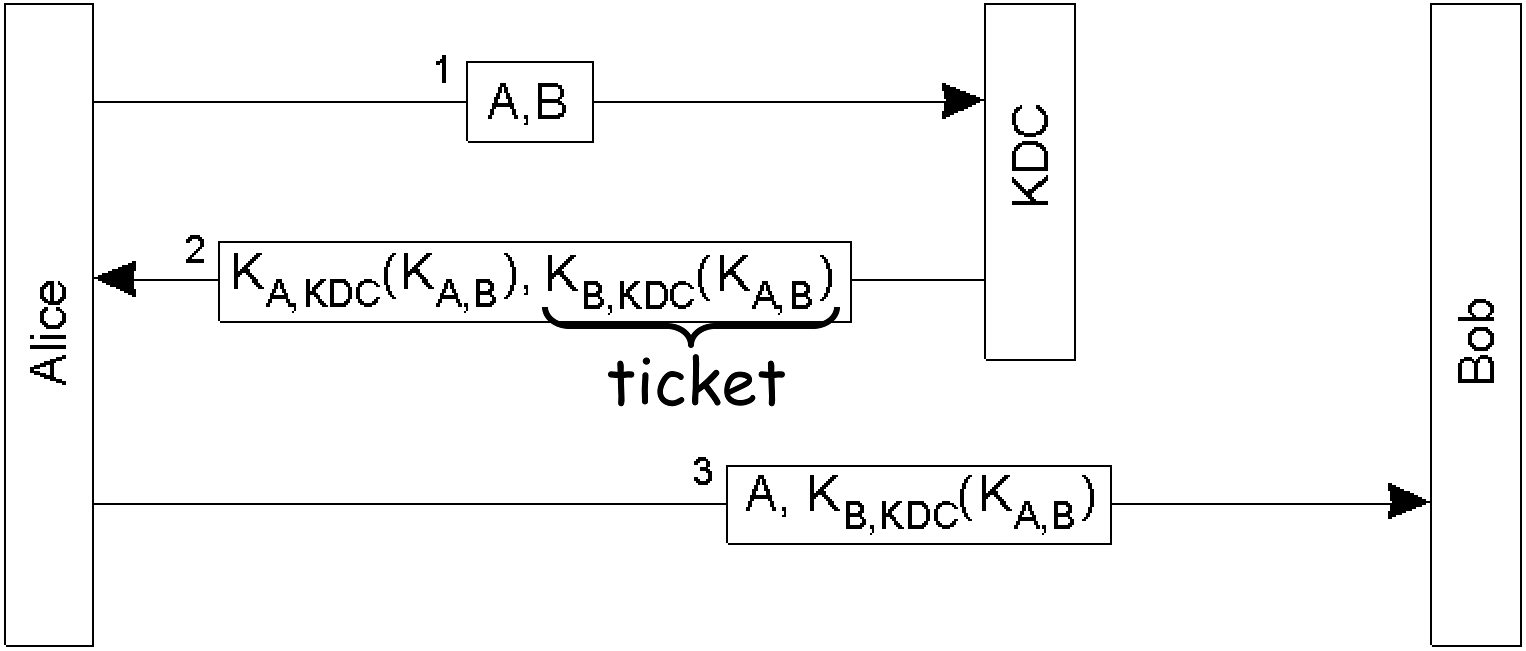
- One of the problems with using a shared secret key for authentication is scalability
- For a system with *N* hosts: KDC requires *N* keys instead of *N*(*N*-1)/2



From Tanenbaum and van Steen, Distributed Systems: Principles and Paradigms
© Prentice-Hall, Inc. 2002

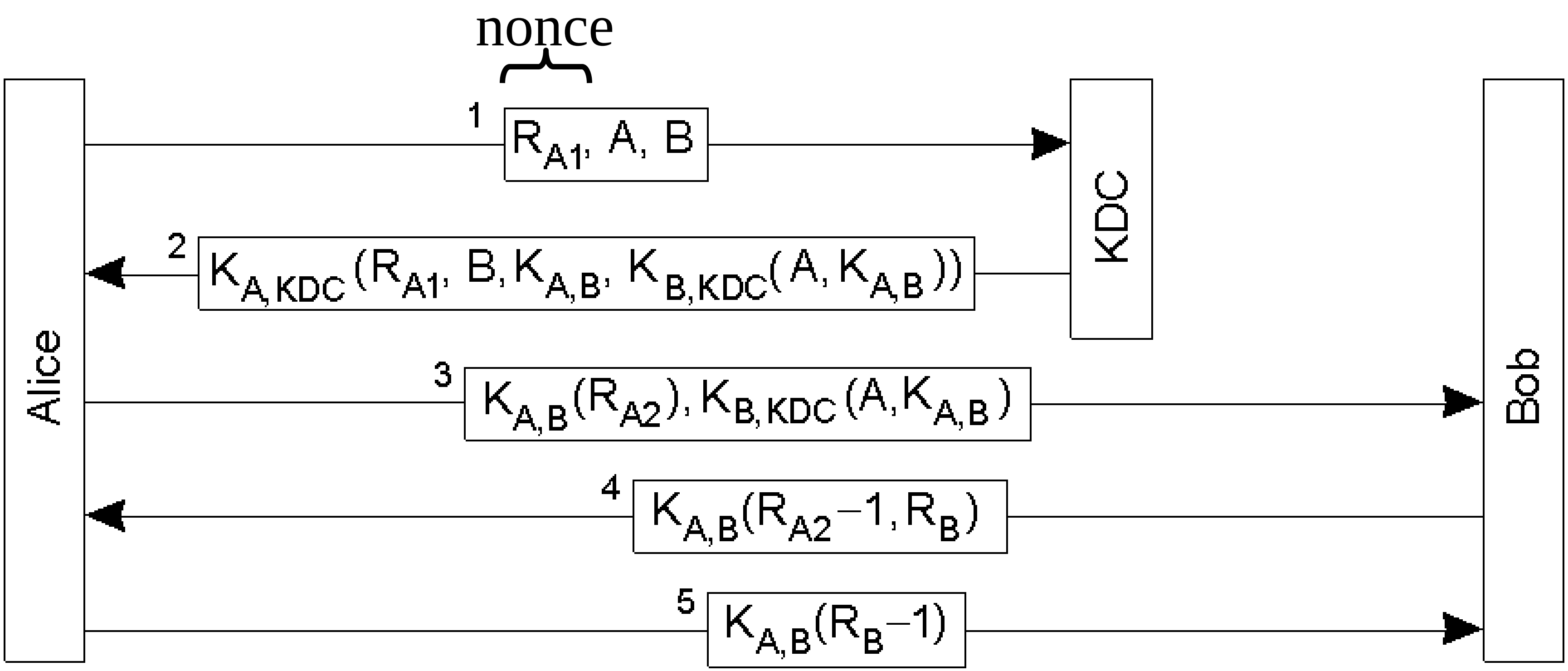
Authentication Using a KDC (2)

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


Alice
KDC
Bob
1
A,B
2
$K_{A,KDC}(K_{A,B}), \underbrace{K_{B,KDC}(K_{A,B})}_{\text{ticket}}$
3
$A, K_{B,KDC}(K_{A,B})$

From Tanenbaum and van Steen, Distributed Systems: Principles and Paradigms
© Prentice-Hall, Inc. 2002

The Needham-Schroeder Authentication Protocol

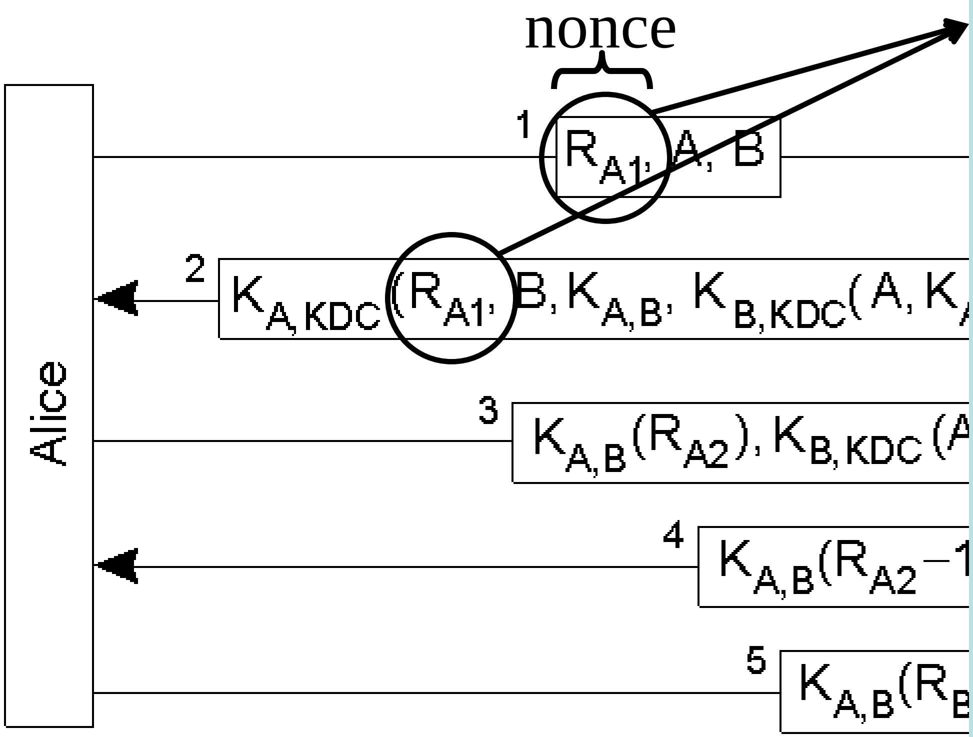


nonce
1 R_{A1}, A, B
2 $K_{A,KDC}(R_{A1}, B, K_{A,B}, K_{B,KDC}(A, K_{A,B}))$
3 $K_{A,B}(R_{A2}), K_{B,KDC}(A, K_{A,B})$
4 $K_{A,B}(R_{A2}-1, R_B)$
5 $K_{A,B}(R_B-1)$
Alice
KDC
Bob

As presented here the algorithm still has a weak point.

From Tanenbaum and van Steen, Distributed Systems: Principles and Paradigms
© Prentice-Hall, Inc. 2002

The Needham-Schroeder Authentication Protocol


nonce
Alice
1
R_{A1}, A, B
2
$K_{A,KDC}(R_{A1}, B, K_{A,B}, K_{B,KDC}(A, K$
3
$K_{A,B}(R_{A2}), K_{B,KDC}(A$
4
$K_{A,B}(R_{A2}-1$
5
$K_{A,B}(R_B$

The nonce couples message 1 with message 2 and prevents replays of an older message 2.

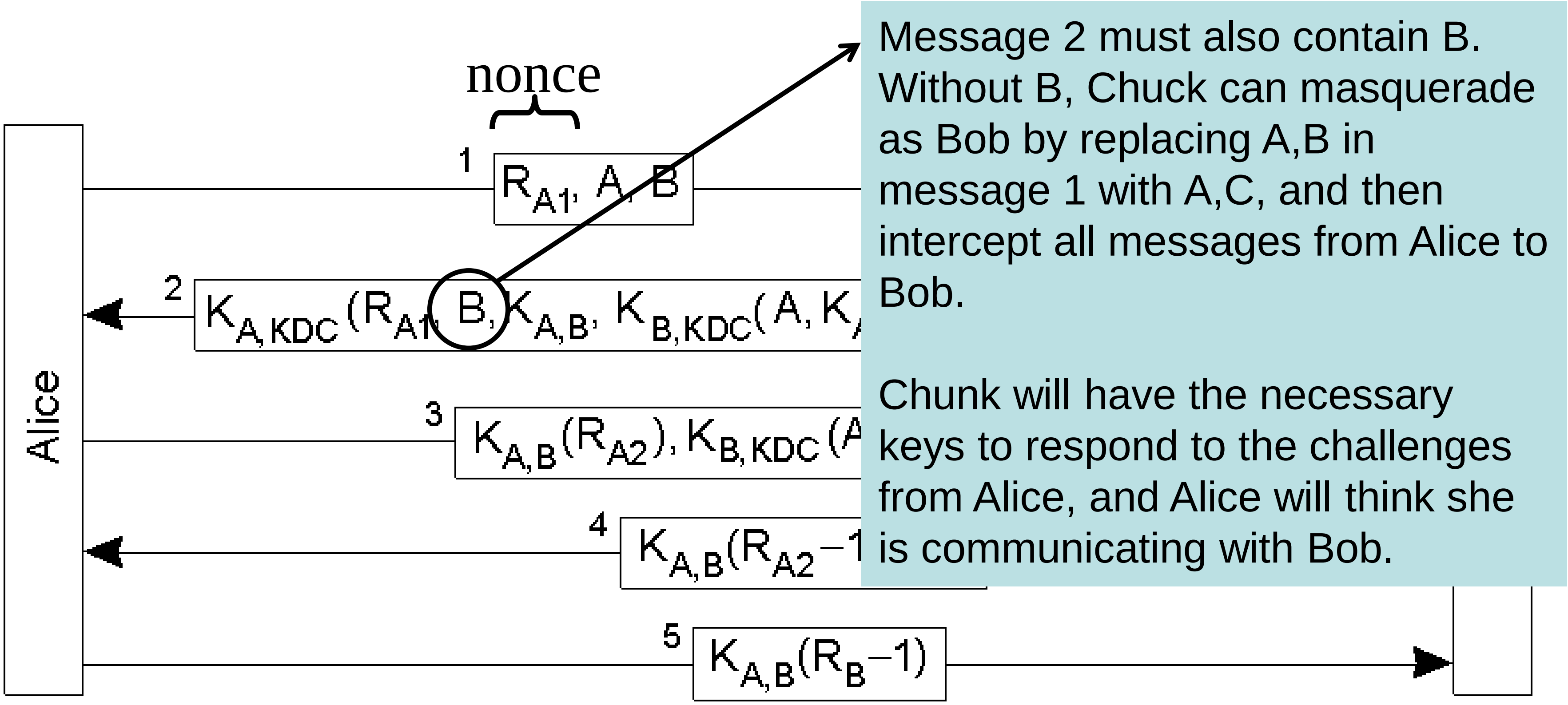
Without a nonce, if Chuck steals one of Bob's old keys ($K_{B,KDC}^{old}$) and has intercepted a copy of message 2 that uses the old key.

Chuck then waits until Alice tries to set up a channel with Bob, intercepts all of her messages, and replays the old message. Now he can decrypt the ticket to get the session key $K_{A,B}$ and pretend to be Bob.

As presented here the algorithm still

From Tanenbaum and van Steen, Distributed Systems: Principles and Paradigms
© Prentice-Hall, Inc. 2002

The Needham-Schroeder Authentication Protocol



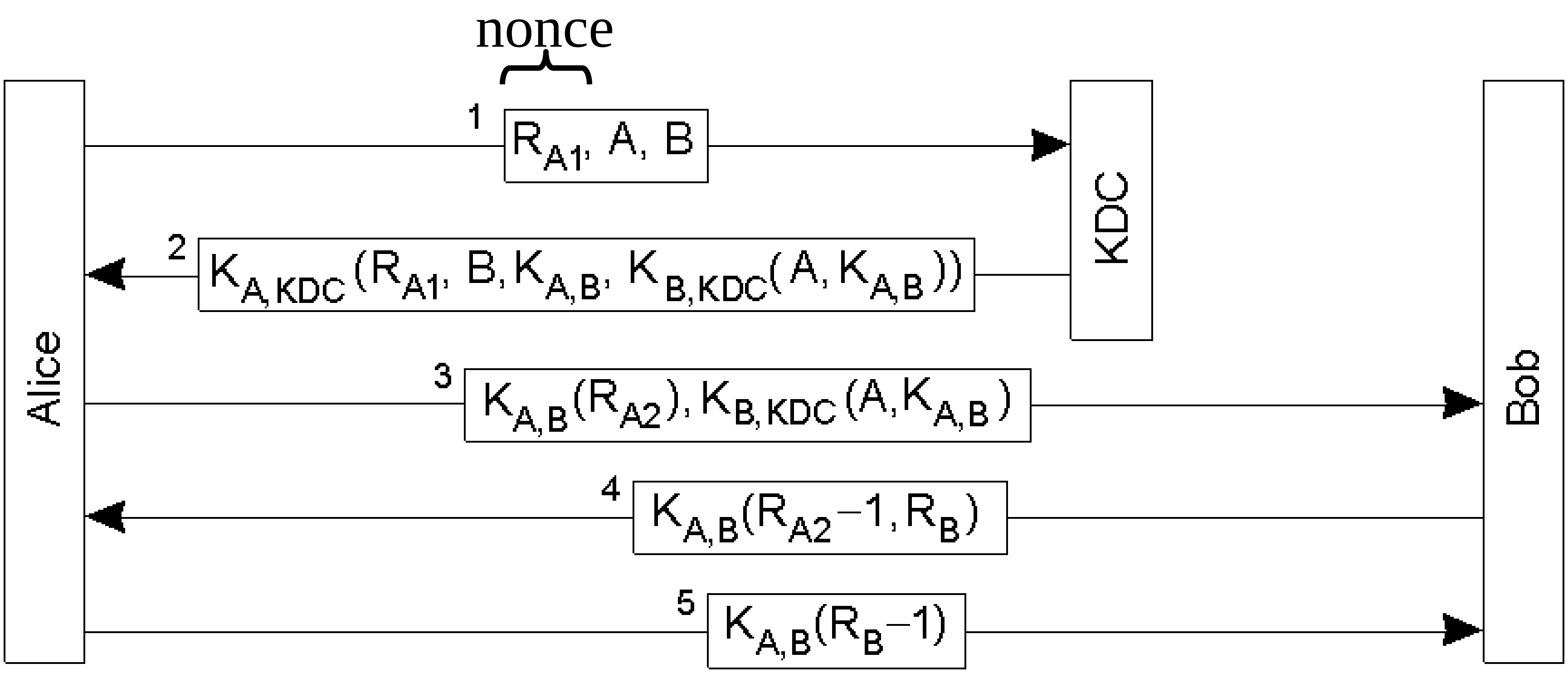
Message 2 must also contain B. Without B, Chuck can masquerade as Bob by replacing A,B in message 1 with A,C, and then intercept all messages from Alice to Bob.

Chunk will have the necessary keys to respond to the challenges from Alice, and Alice will think she is communicating with Bob.

As presented here the algorithm still has a weak point.

From Tanenbaum and van Steen, Distributed Systems: Principles and Paradigms
© Prentice-Hall, Inc. 2002

The Needham-Schroeder Authentication Protocol



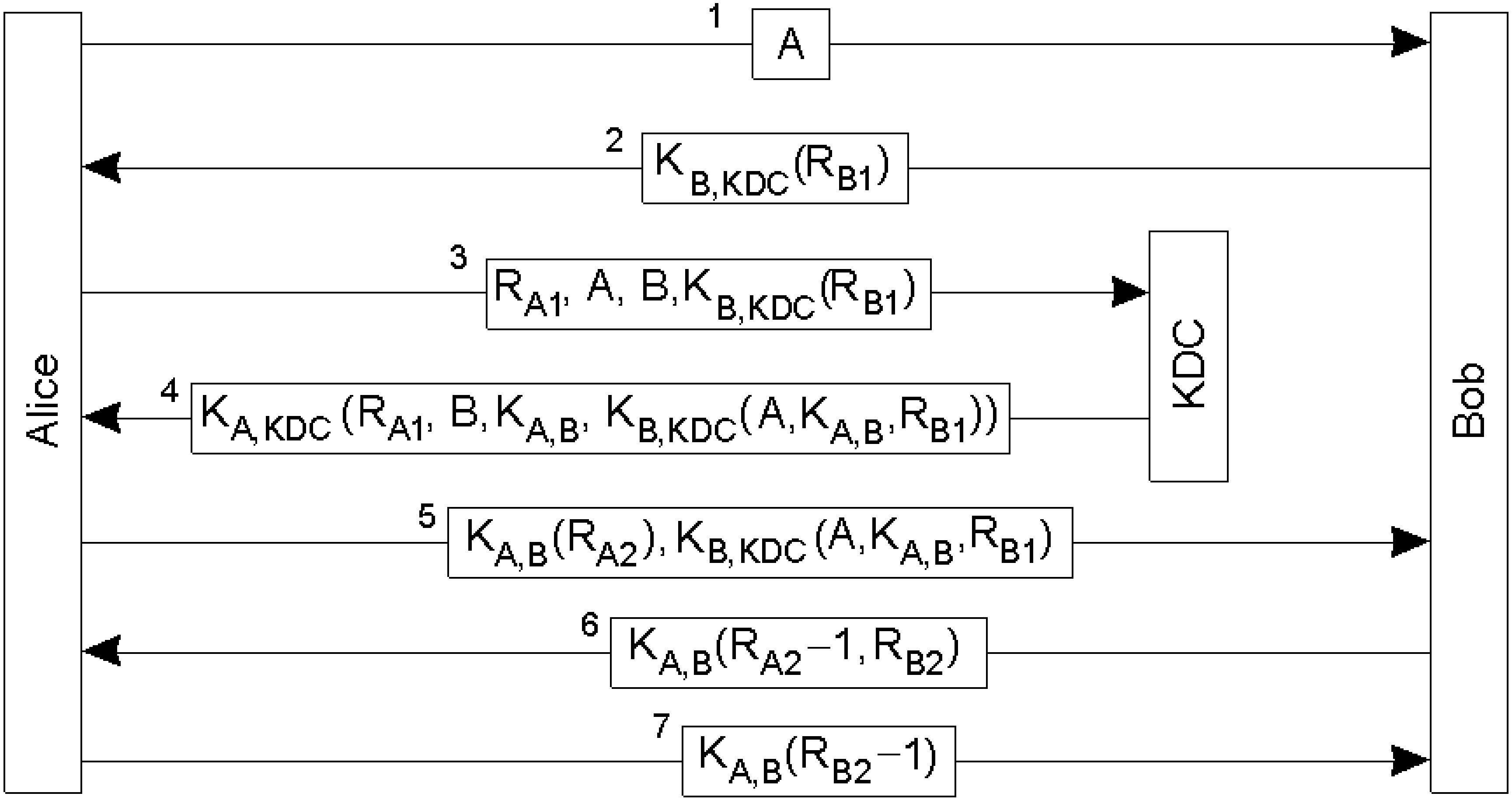
nonce
Alice
KDC
Bob
1
R_{A1}, A, B
2
$K_{A,KDC}(R_{A1}, B, K_{A,B}, K_{B,KDC}(A, K_{A,B}))$
3
$K_{A,B}(R_{A2}), K_{B,KDC}(A, K_{A,B})$
4
$K_{A,B}(R_{A2}-1, R_B)$
5
$K_{A,B}(R_B-1)$

If Chuck ever gets a hold of an old key $K_{A,B}^{old}$ he can replay an old copy of message 3 and masquerade as Alice.

From Tanenbaum and van Steen, Distributed Systems: Principles and Paradigms
© Prentice-Hall, Inc. 2002

The Needham-Schroeder Authentication Protocol

- Protection against malicious reuse of a previously generated session key.

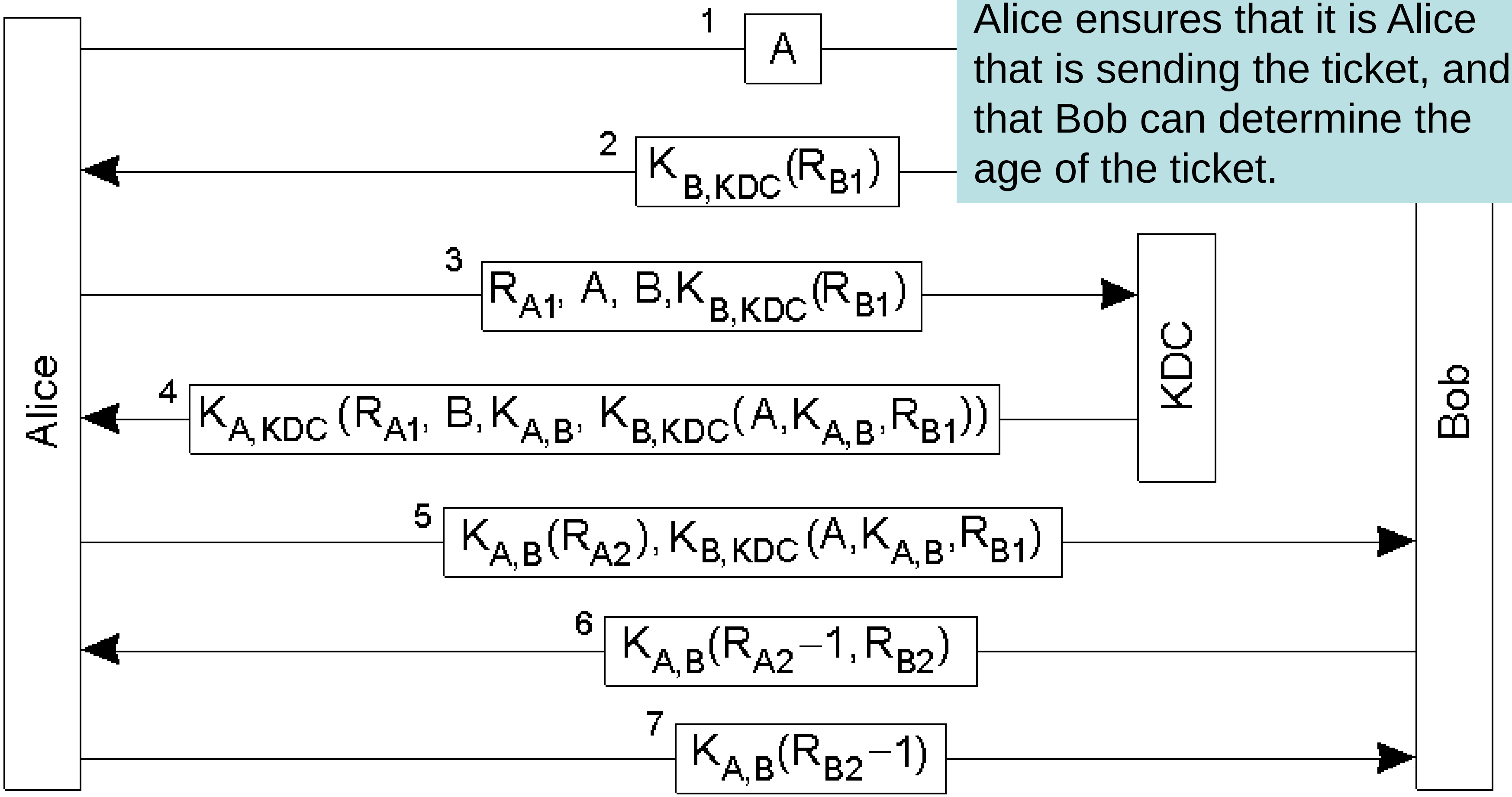

Alice
KDC
Bob
1
A
2
$K_{B,KDC}(R_{B1})$
3
R_{A1}, A, B, $K_{B,KDC}(R_{B1})$
4
$K_{A,KDC}(R_{A1}, B, K_{A,B}, K_{B,KDC}(A, K_{A,B}, R_{B1}))$
5
$K_{A,B}(R_{A2}), K_{B,KDC}(A, K_{A,B}, R_{B1})$
6
$K_{A,B}(R_{A2}-1, R_{B2})$
7
$K_{A,B}(R_{B2}-1)$

From Tanenbaum and van Steen, Distributed Systems: Principles and Paradigms
© Prentice-Hall, Inc. 2002

The Needham-Schroeder Authentication Protocol

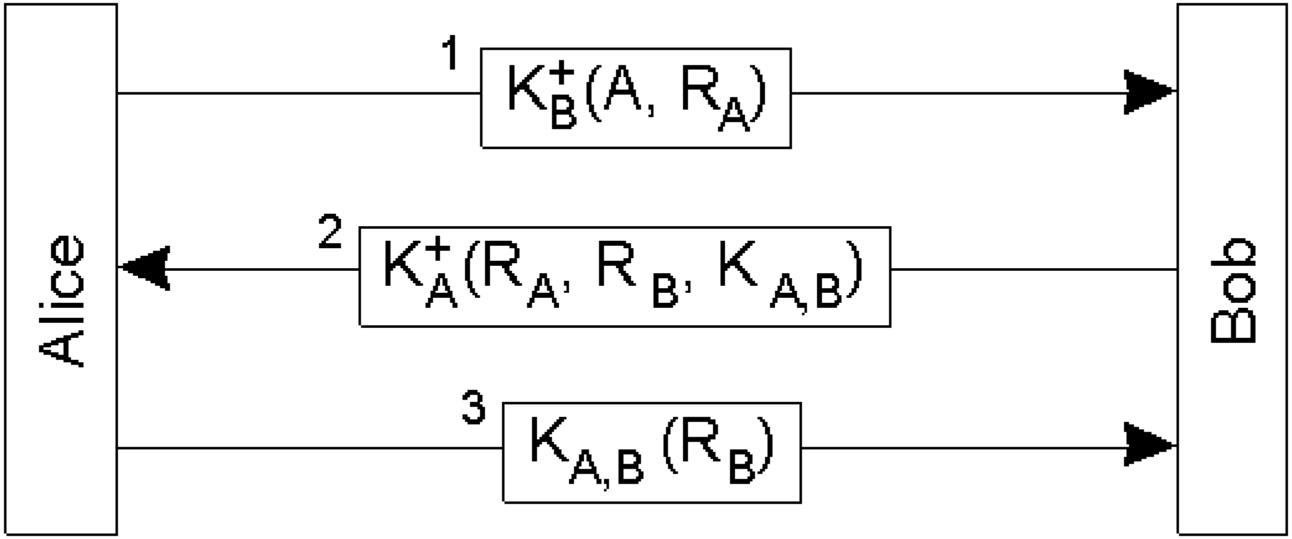
- Protection against malicious reuse of a previous

The extra nonce from Bob to Alice ensures that it is Alice that is sending the ticket, and that Bob can determine the age of the ticket.


Alice
KDC
Bob
1
A
2
$K_{B,KDC}(R_{B1})$
3
$R_{A1}, A, B, K_{B,KDC}(R_{B1})$
4
$K_{A,KDC}(R_{A1}, B, K_{A,B}, K_{B,KDC}(A, K_{A,B}, R_{B1}))$
5
$K_{A,B}(R_{A2}), K_{B,KDC}(A, K_{A,B}, R_{B1})$
6
$K_{A,B}(R_{A2}-1, R_{B2})$
7
$K_{A,B}(R_{B2}-1)$

From Tanenbaum and van Steen, Distributed Systems: Principles and Paradigms
© Prentice-Hall, Inc. 2002

Authentication Using Public-Key Cryptography



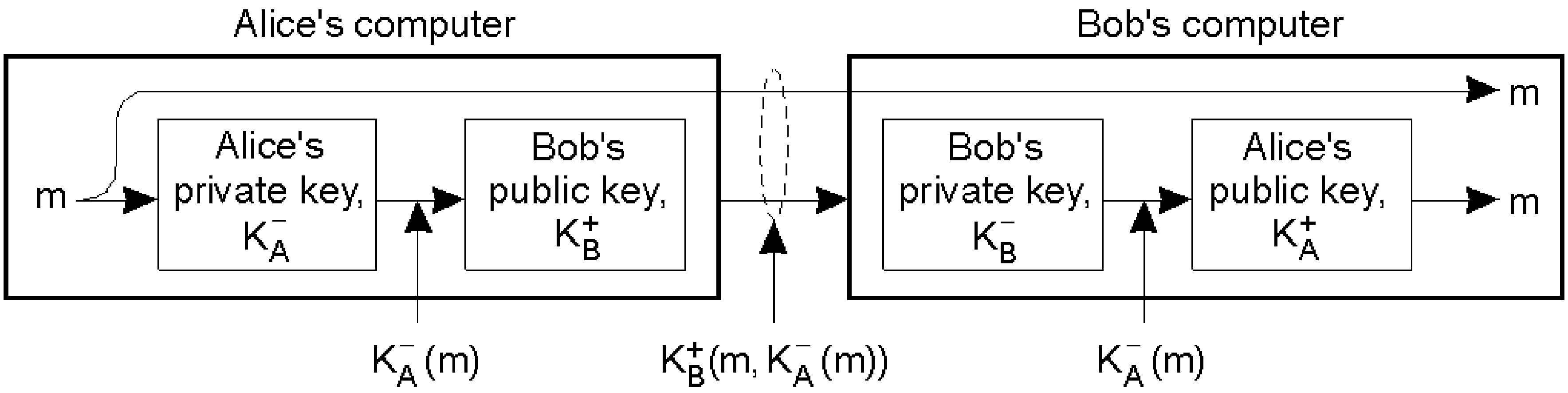
From Tanenbaum and van Steen, Distributed Systems: Principles and Paradigms
© Prentice-Hall, Inc. 2002

Message Integrity and Confidentiality

- Besides authentication, a secure channel should also provide guarantees for message integrity and confidentiality
- Confidentiality is easily established by simply encrypting a message before sending it.
- Protecting a message against modifications is somewhat more complicated
- Digital Signatures:
  - Digitally sign a message in such a way that the signature is uniquely tied to its content
  - Several ways to place digital signatures:
    - → Use a public-key cryptosystem such as RSA
    - → Use a message digest

Digital Signatures

Digital signing a message using public-key cryptography.


Alice's computer
Bob's computer
m
Alice's private key, K_A^-
Bob's public key, K_B^+
Bob's private key, K_B^-
Alice's public key, K_A^+
m
m
$K_A^-(m)$
$K_B^+(m, K_A^-(m))$
$K_A^-(m)$

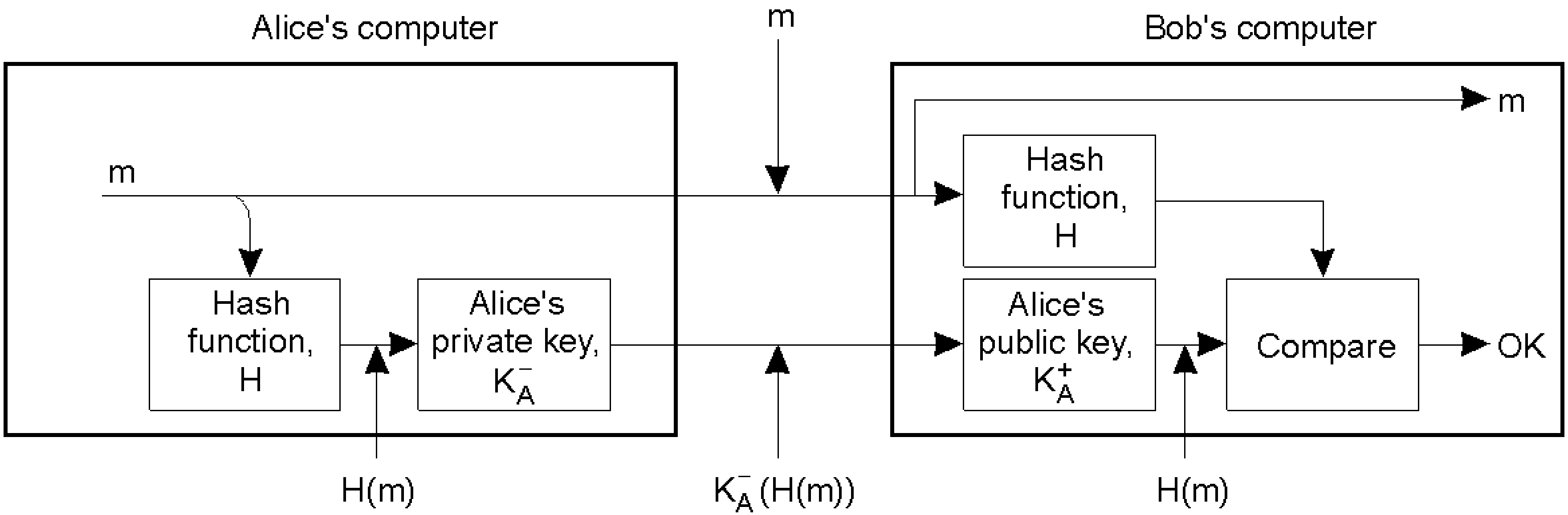
Problems with this scheme:

- the validity of the signature holds only as long as the private key remains secret
- what if Alice decides to change her private key
- encryption of the entire message may be costly in terms of processing requirements and is actually unnecessary

From Tanenbaum and van Steen, Distributed Systems: Principles and Paradigms

Digital Signatures

Digitally signing a message using a message digest.



From Tanenbaum and van Steen, Distributed Systems: Principles and Paradigms
© Prentice-Hall, Inc. 2002

Session Keys

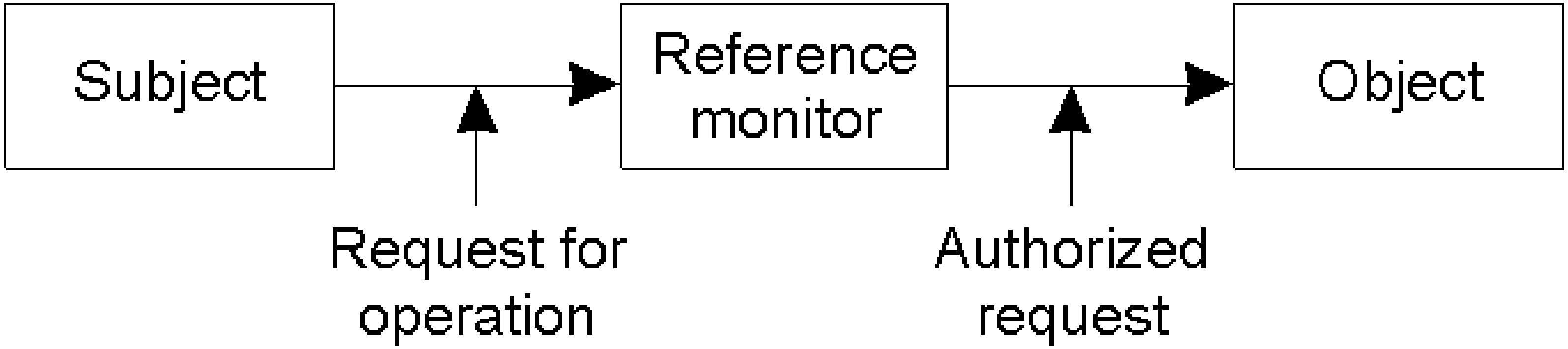
- During the establishment of a secure channel, after the authentication phase has completed, the communicating parties generally use a unique shared session key for confidentiality.
- The session key is safely discarded when the channel is no longer used.
- Why not using the same keys for confidentiality as those that are used for setting up the secure channel?
  - Cryptographic keys are subject to "wear and tear" just like ordinary keys.
  - Protection against replay attacks
  - If a key is compromised, only a single session is affected
- ☒ The combination of a long-lasting keys with the much cheaper and more temporary session keys is often a good choice for implementing secure channels for exchanging data

Access Control

- In the client-server model, once a client and a server have set up a secure channel, the client can issue requests that are to be carried out by the server.
- A request involve carrying out operations on resources that are controlled by the server.
- Such a request can be carried out only if the client has sufficient access rights for that request.
- Verifying access rights is referred to as access control, whereas authorization is about granting access rights.

General Issues in Access Control

- General model of controlling access to objects.

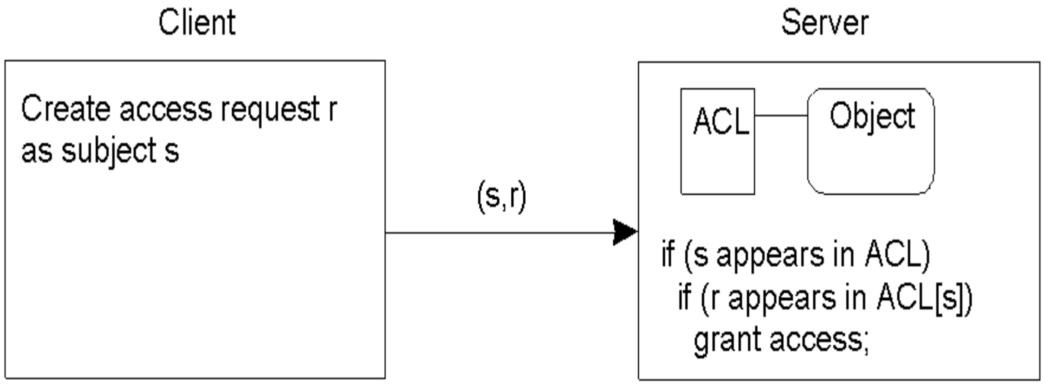


- Controlling the access to an object is all about protecting the object against invocations by subjects that are not allowed to have specific methods carried out
- Also, protection may include object management issues

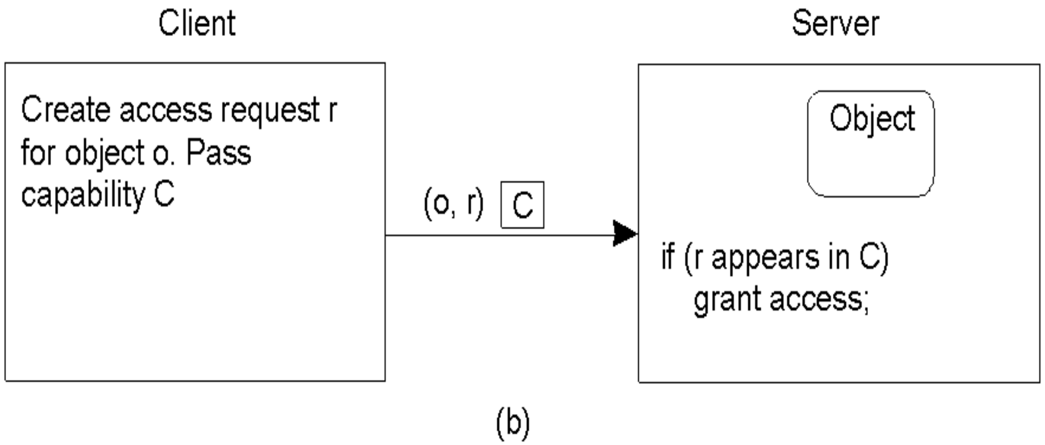
From Tanenbaum and van Steen, Distributed Systems: Principles and Paradigms

Access Control Matrix

- A common approach to modeling the access rights of subjects with respect to objects, is to construct an access control matrix $M[s,o]=\{m_1,m_2,\ldots\}$
  - Access Control List: The matrix is distributed column-wise
  - Capabilities: The matrix is distributed row-wise



Using an ACL



Using capabilities

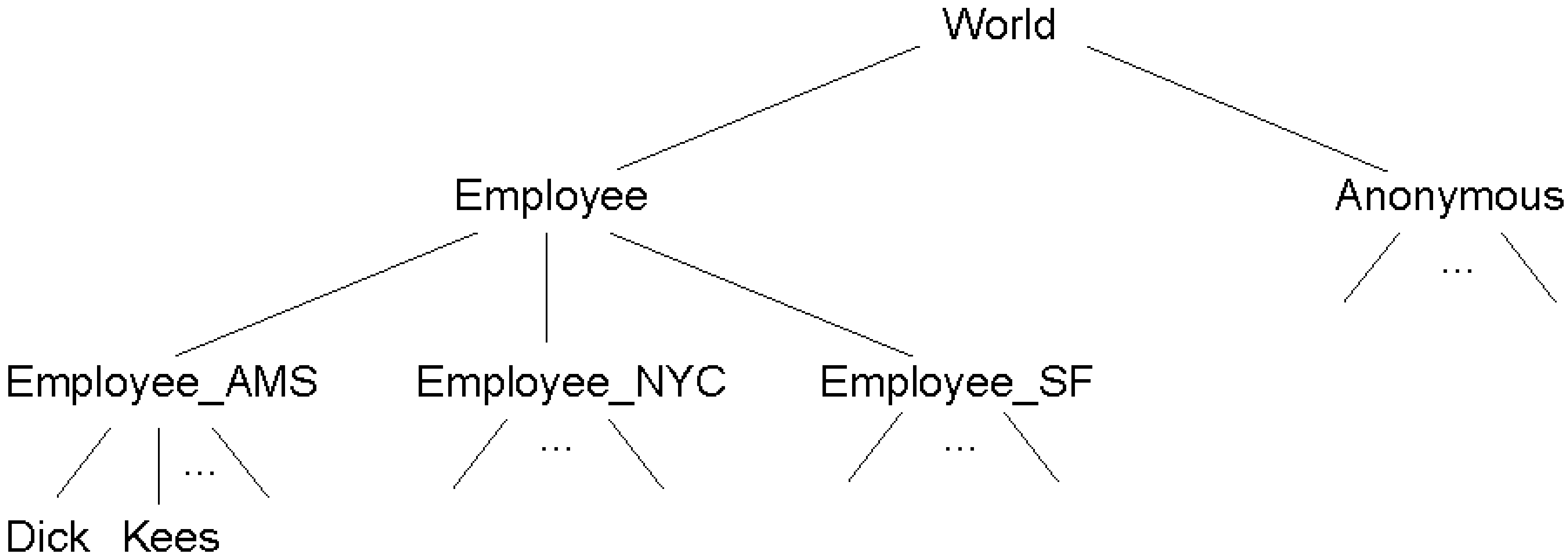
From Tanenbaum and van Steen, Distributed Systems: Principles and Paradigms
© Prentice-Hall, Inc. 2002

Protection Domains

- ACLs and capabilities help in efficiently implementing an access control matrix by ignoring all empty entries.
- An ACL or a capability list can still become quite large if no further measures are taken.
- One general way of reducing ACLs is to make use of protection domains.
- A protection domain is a set of (object, access rights) pairs. Each pair specifies for a given object exactly which operations are allowed to be carried out.
- Requests for carrying out an operation are always issued within a domain.

Protection Domains

- The hierarchical organization of protection domains as groups of users.
  - Advantage: managing group membership is relatively easy
  - Disadvantage: looking up a member can be quite costly if the membership database is distributed.



From Tanenbaum and van Steen, Distributed Systems: Principles and Paradigms
© Prentice-Hall, Inc. 2002

Role-Based Access Control

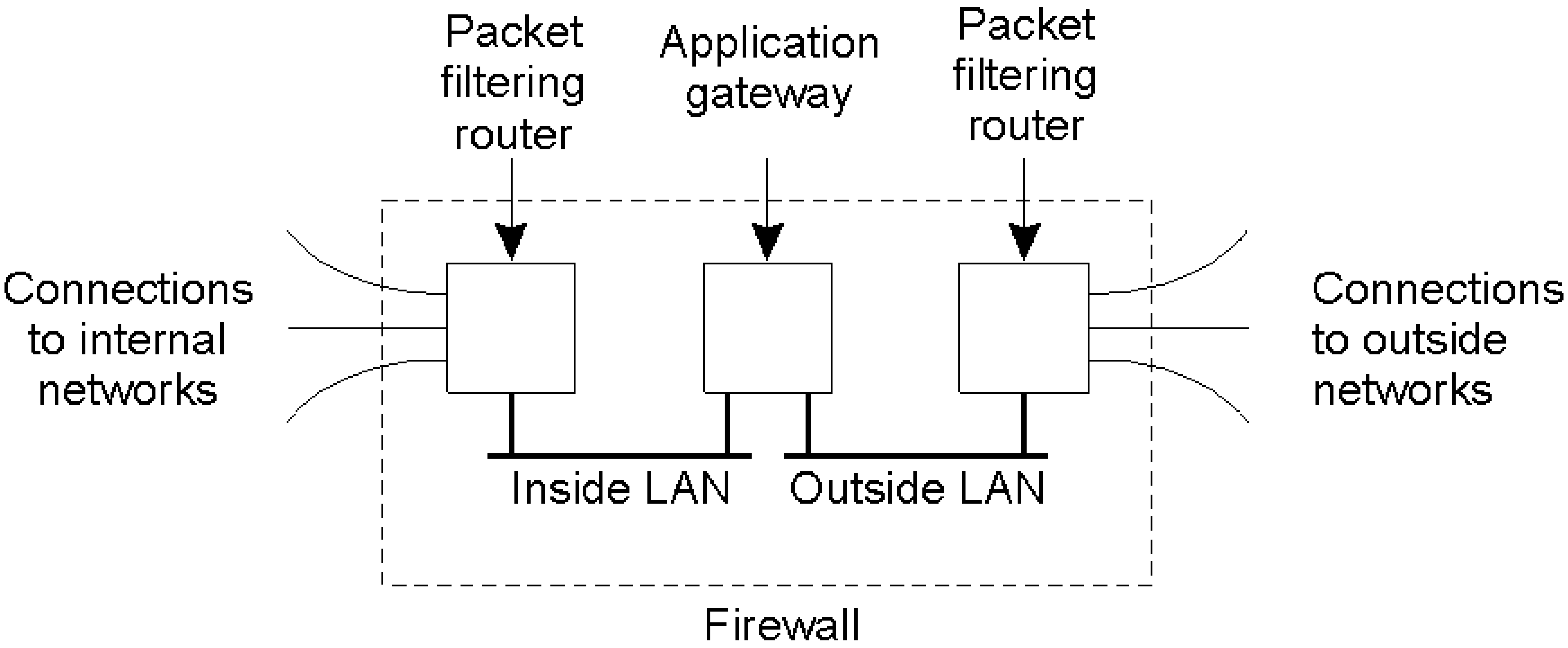
- Related to having groups as protection domains, it is also possible to implement protection domains as roles.
- In role-based access control, a user always logs into the system with a specific role, which is often associated with a function the user has in an organization.
- A user may have several functions.
- Depending on the role the user takes when logging in, he may be assigned different privileges (i.e. his role determines the protection domain in which he will operate).

Firewalls

- When outsiders are allowed to access the resources controlled by a DS: protection using cryptographic techniques and access control matrix is not enough.
- External access to any part of a DS is controlled by a special kind of reference monitor known as a firewall.
- A firewall disconnects any part of a DS from the outside world.
- All outgoing, but especially all incoming packets are routed through a special computer and inspected before they are passed.
- Unauthorized traffic is discarded and not allowed to continue.

Firewalls

- Firewalls come in two different flavors that are often combined:
  - packet-filtering gateway
  - application-level gateway



From Tanenbaum and van Steen, Distributed Systems: Principles and Paradigms
© Prentice-Hall, Inc. 2002

Security Management

- Key Management
- Secure Group Management
- Authorization Management

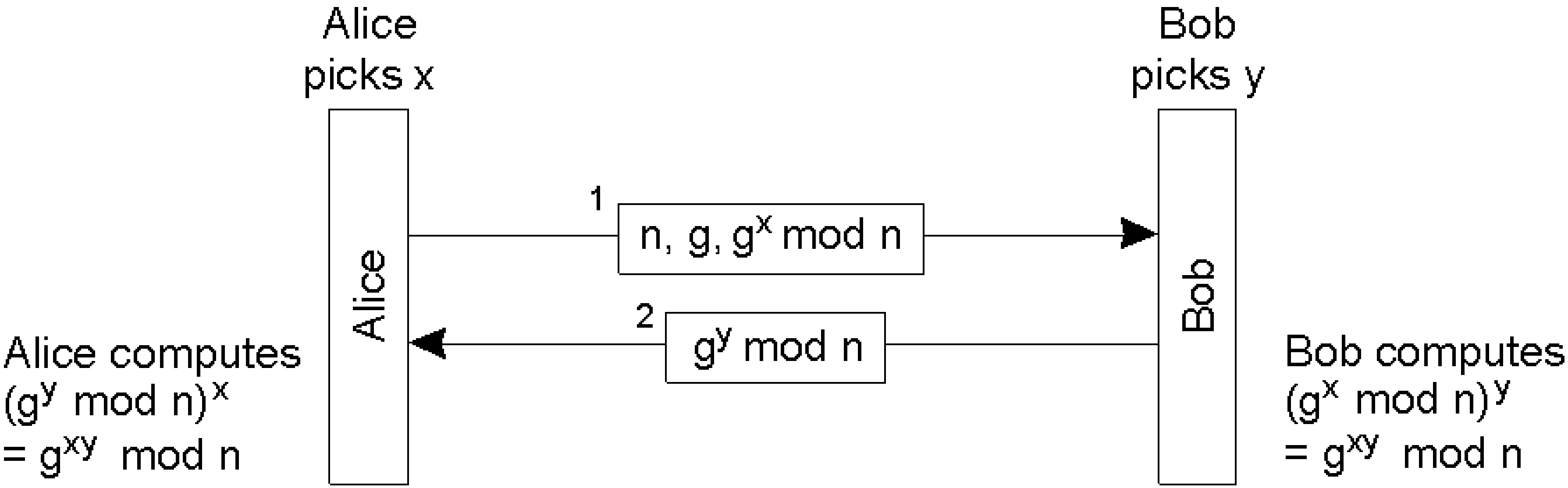
Key Management

- So far, we have described various cryptographic protocols in which we implicitly assumed that various keys were readily available.
- For example, in the case of public-key cryptosystems, we assumed that a sender of a message had the public key of the receiver at its disposal so that it could encrypt the message to ensure confidentiality.
- However, establishing and distributing keys is not a trivial matter.
- Also, mechanisms are needed to revoke keys, that is, prevent a key from being used after it has been compromised or invalidated.

Key Establishment

- How session keys can be established?
- Easy, if using public-key or shared-key to establish a secure channel
- The method requires that the communicating parties already have the means available to establish a secure channel.
- In other words, some form of key establishment and distribution must already have taken place.
- The same applies when a shared secret key is established by means of a trusted third party, such as a KDC.
- An elegant and widely applied scheme for establishing a shared key across an insecure channel, is the Diffie-Hellman key exchange (1976).

Diffie-Hellman Key Exchange

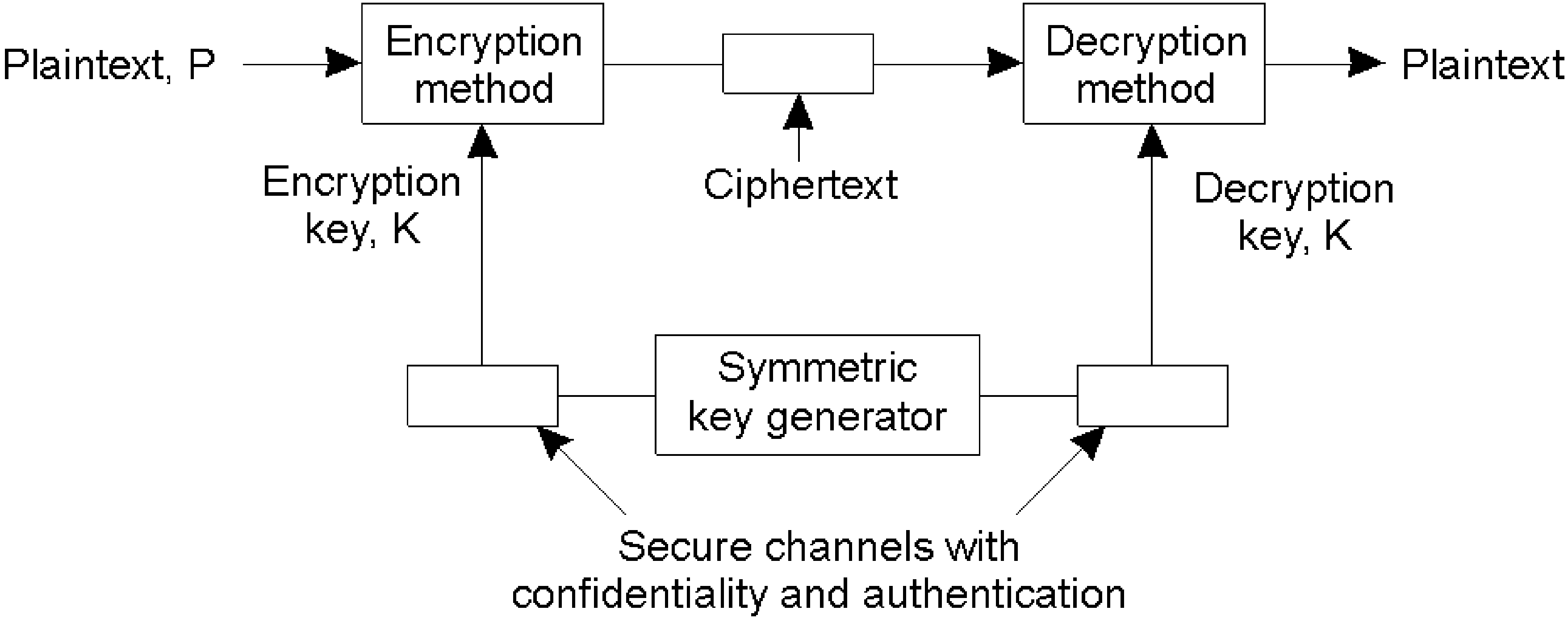


- Diffie-Hellman can be viewed as a public-key cryptosystem. In the case of Alice, x is her private key, where as g^x *mod* n is her public key
- Securely distributing the public key is essential to making Diffie-Hellman work in practice

From Tanenbaum and van Steen, Distributed Systems: Principles and Paradigms
© Prentice-Hall, Inc. 2002

Key Distribution

- In a symmetric cryptosystem, the initial shared secret key must be communicated along a secure channel that provides authentication as well as confidentiality.

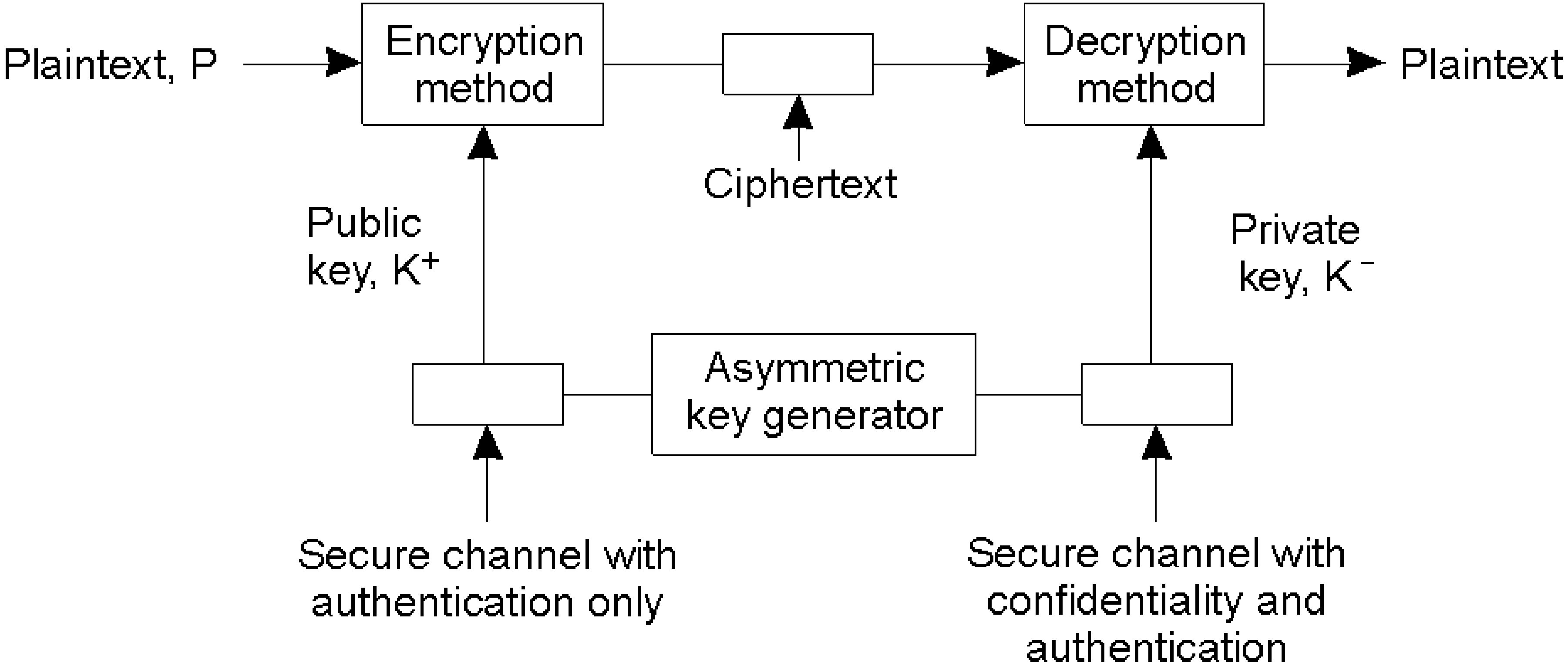


- If there are no keys available to Alice and Bob to set up such a secure channel, it is necessary to distribute the key out-of-band.

From Tanenbaum and van Steen, Distributed Systems: Principles and Paradigms
© Prentice-Hall, Inc. 2002

Key Distribution

- In the case of a public-key cryptosystem, we need to distribute the public key in such a way the the receivers can be sure that the key is indeed paired to a claimed private key.



From Tanenbaum and van Steen, Distributed Systems: Principles and Paradigms
© Prentice-Hall, Inc. 2002

Public-Key Certificates

- In practice, public-key distribution takes place by means of public-key certificates.
- Public-key certificate consists of:
  - a public key
  - an entity
  - certification authority
  - signature of the certification authority
- Signing takes place by means of a private key K^{-}_{CA} that belongs to the certification authority.
- The corresponding public key K^{+}_{CA} is assumed to be well known (e.g. the public keys of various certification authorities are built into most Web browsers and shipped with the binaries)

Public-Key Certificates

- The client must assume that the public key K^{+}_{CA} indeed belongs to the associated certification authority.
- It should be possible to verify the validity of K^{+}_{CA} through another certificate coming from a different, possibly more trusted certification authority.
- Such hierarchical trust models in which the highest-level certification authority must be trusted by everyone, are not uncommon.
- Example: Privacy Enhanced Mail (PEM) uses a three-level trust model in which lowest-level certification authorities can be authenticated by Policy Certification Authorities (PCA), which in turn can be authenticated by the Internet Policy Registration Authority (IPRA).

Lifetime of Certificates

- We need a mechanism to revoke the certificate by making it publicly known that the certificate is no longer valid.
- One common approach is with a Certificate Revocation List (CRL) published regularly by the certification authority.
- An alternative approach is to restrict the lifetime of a certificate (analogous to handing out leases).
- A final extreme case is to reduce the lifetime of a certificate to nearly zero.
- Practice indicates that client applications hardly ever consult CRLs and simply assume a certificate to be valid until it expires.
- When it comes to Internet security in practice, there is still much room for improvement.