DISTRIBUTED SYSTEMS
Principles and Paradigms
Second Edition
ANDREW S. TANENBAUM
MAARTEN VAN STEEN

Chapter 9
Security

Types of Threats

- *Interception.* Unauthorized party gaining access to a service or data. E.g. eavesdropping, illegal copying.
- *Interruption.* Services or data becoming unavailable, unusable, destroyed. E.g. intentional file corruption, denial of service attacks.
- *Modification.* Unauthorized changing of data or service so that it no longer adheres to its original specification.
- *Fabrication.* Additional data or activity is generated that would normally not exist. E.g., adding entry to password file or database, breaking into a system by replaying previously sent messages.

Security Policy and Mechanisms

- Security policy describes what actions the entities in a system are allowed to take and which ones are prohibited
- Security mechanisms implement security policies. The following techniques are used:
  - Encryption
  - Authentication
  - Authorization
  - Auditing

Example: The Globus Security Architecture (1)

- The environment consists of multiple administrative domains.
- Local operations are subject to a local domain security policy only.
- Global operations require the initiator to be known in each domain where the operation is carried out.

Example: The Globus Security Architecture (2)

- Operations between entities in different domains require mutual authentication.
- Global authentication replaces local authentication.
- Controlling access to resources is subject to local security only.
- Users can delegate rights to processes.
- A group of processes in the same domain can share credentials.

Example: The Globus Security Architecture (3)

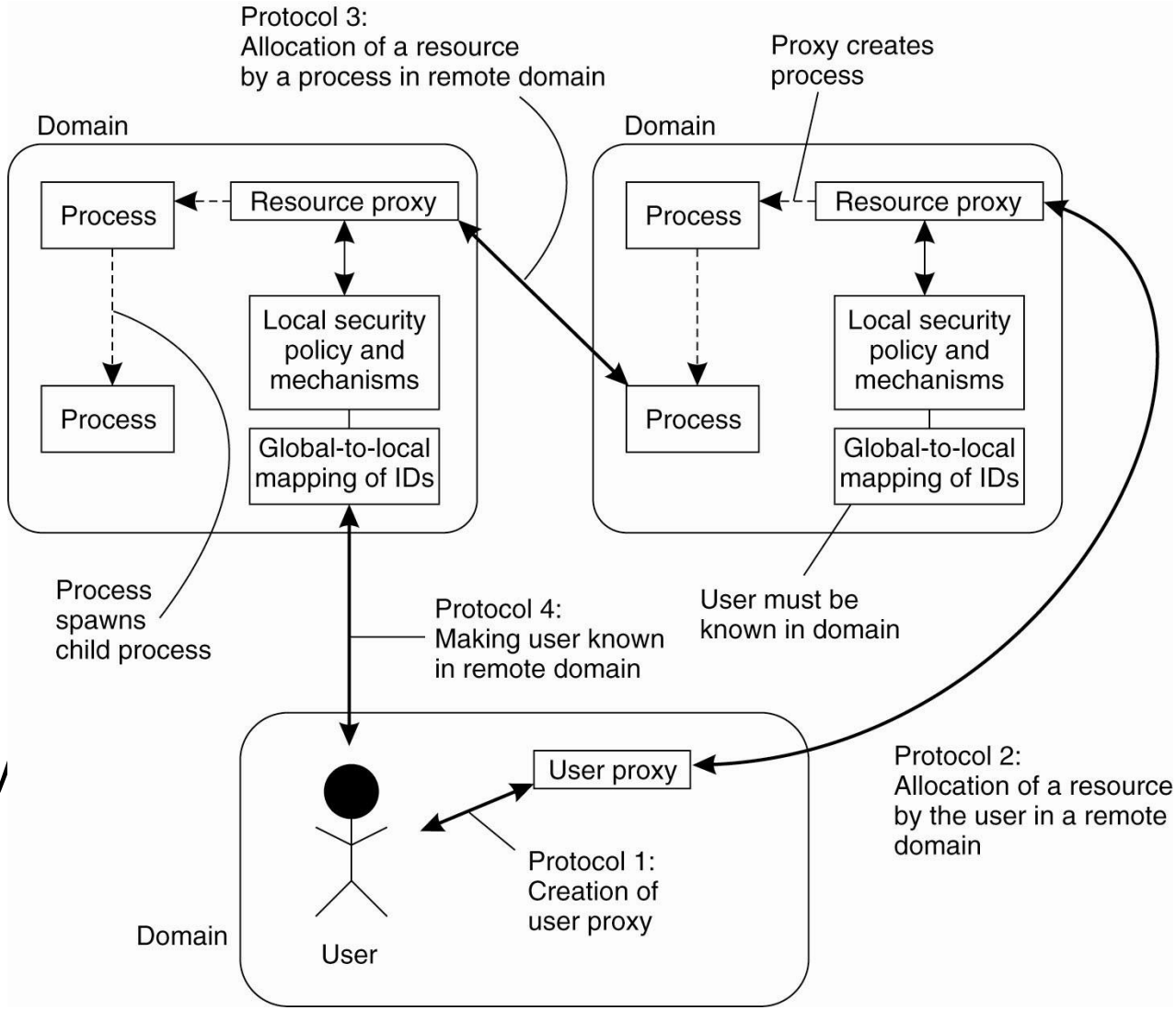


Figure 9-1. The Globus security architecture.

Design Issues

- Focus of control:
  - Data
  - Operations
  - Users
- Layering of security mechanisms
  - Security is technical; trust is emotional
  - Trusted Computing Base: set of all security mechanisms that are needed to enforce a security policy and that thus needs to be trusted
- Simplicity

Focus of Control (1)

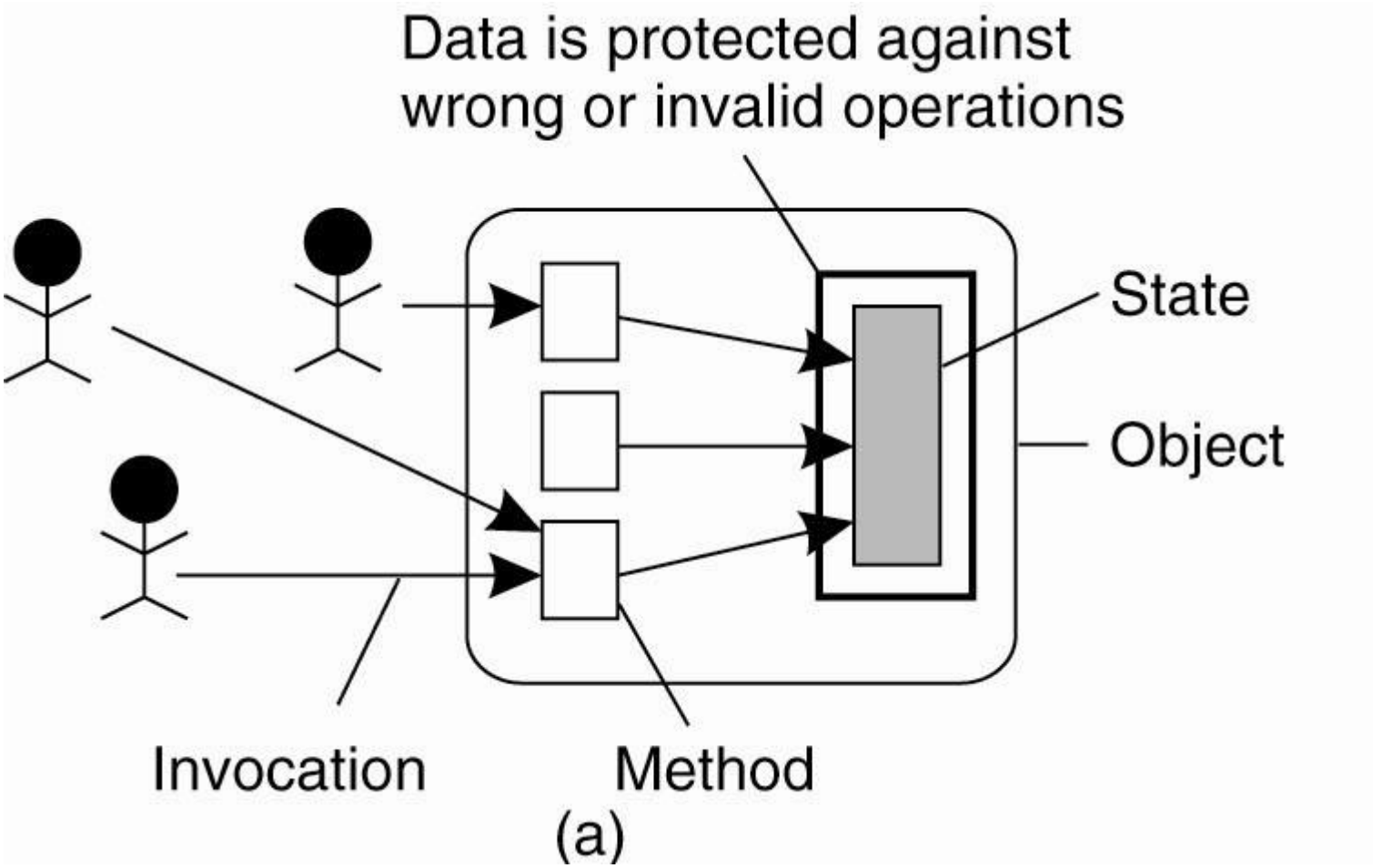


Figure 9-2. Three approaches for protection against security threats. (a) Protection against invalid operations

Focus of Control (2)

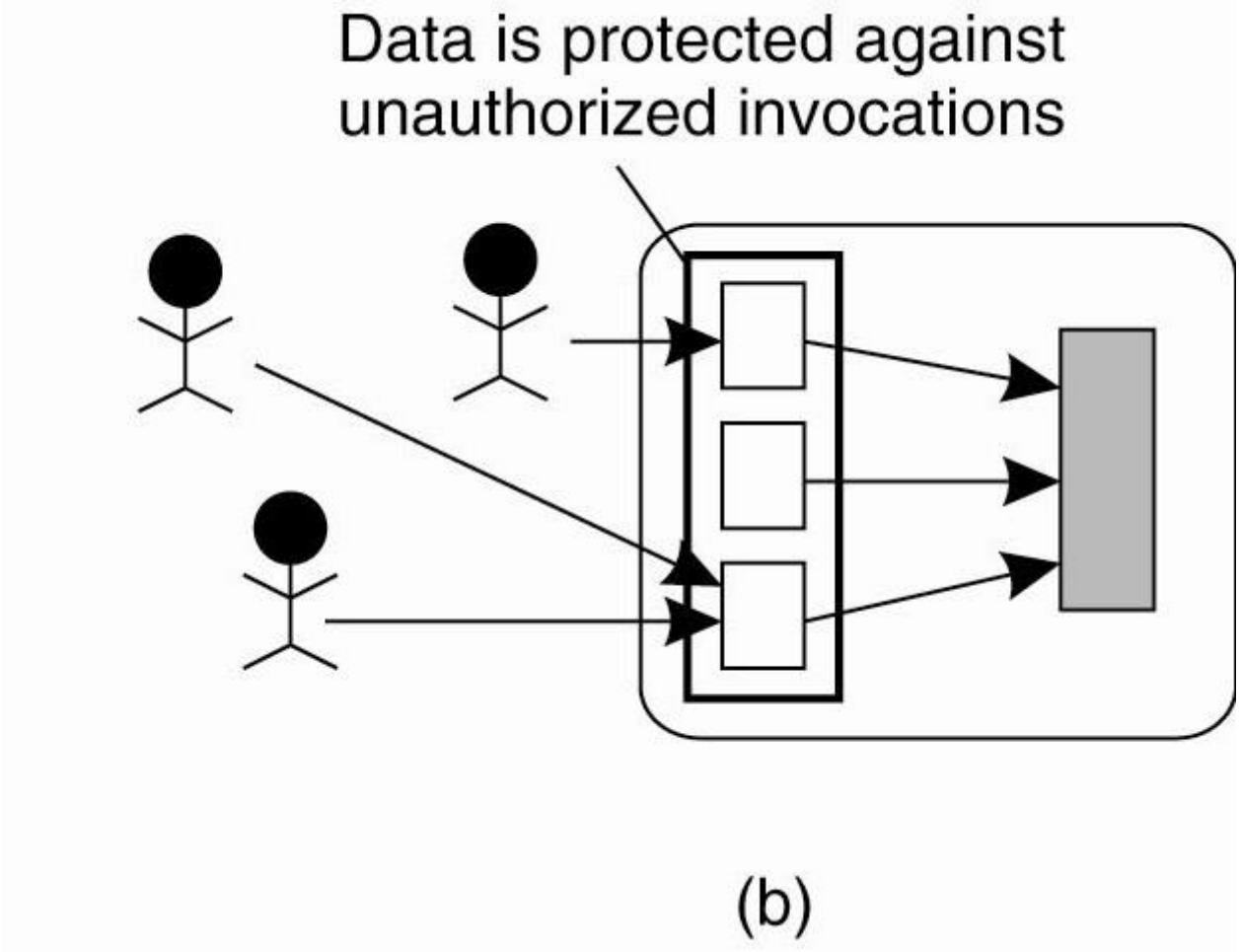


Figure 9-2. Three approaches for protection against security threats. (b) Protection against unauthorized invocations.

Focus of Control (3)

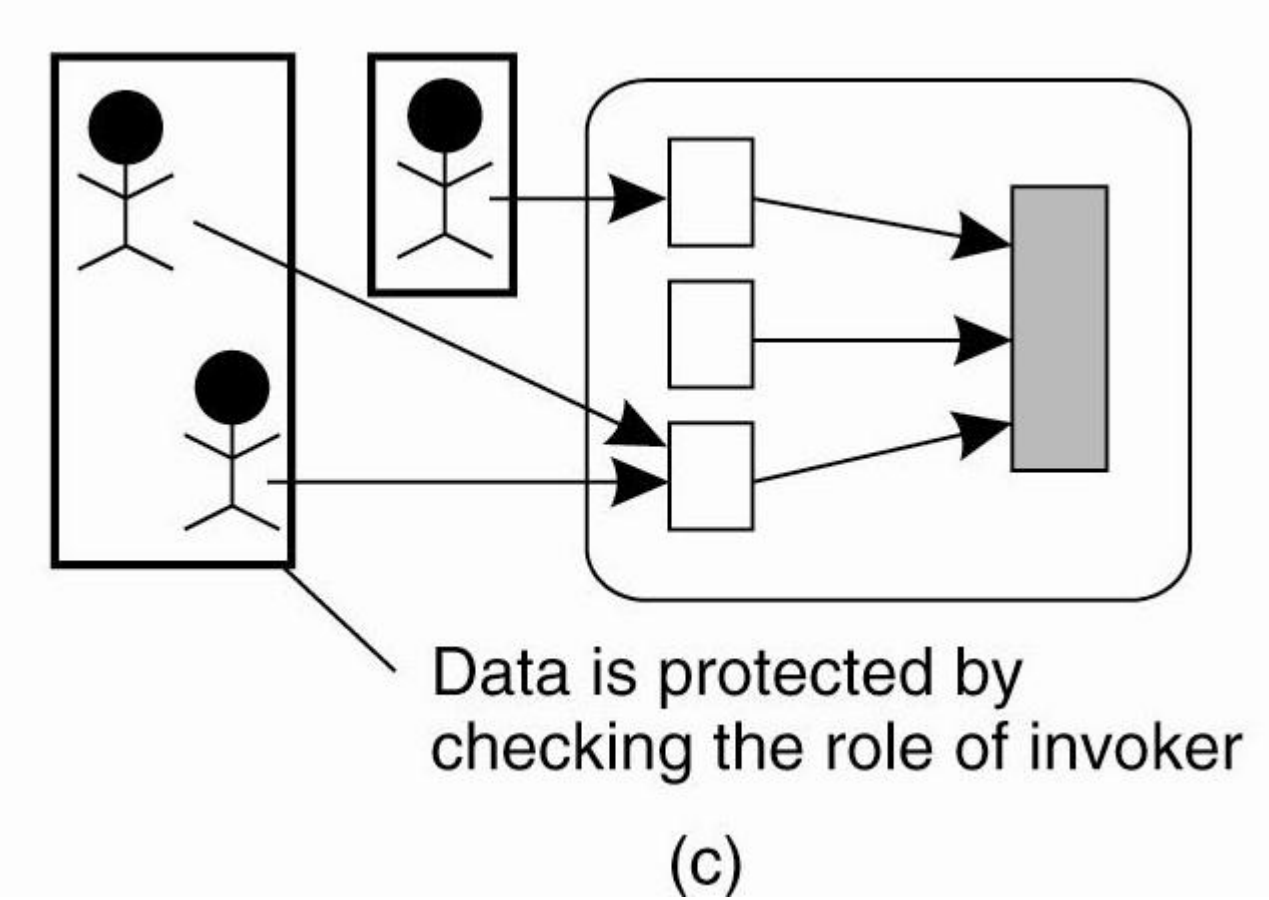


Figure 9-2. Three approaches for protection against security threats. (c) Protection against unauthorized users.

Layering of Security Mechanisms (1)

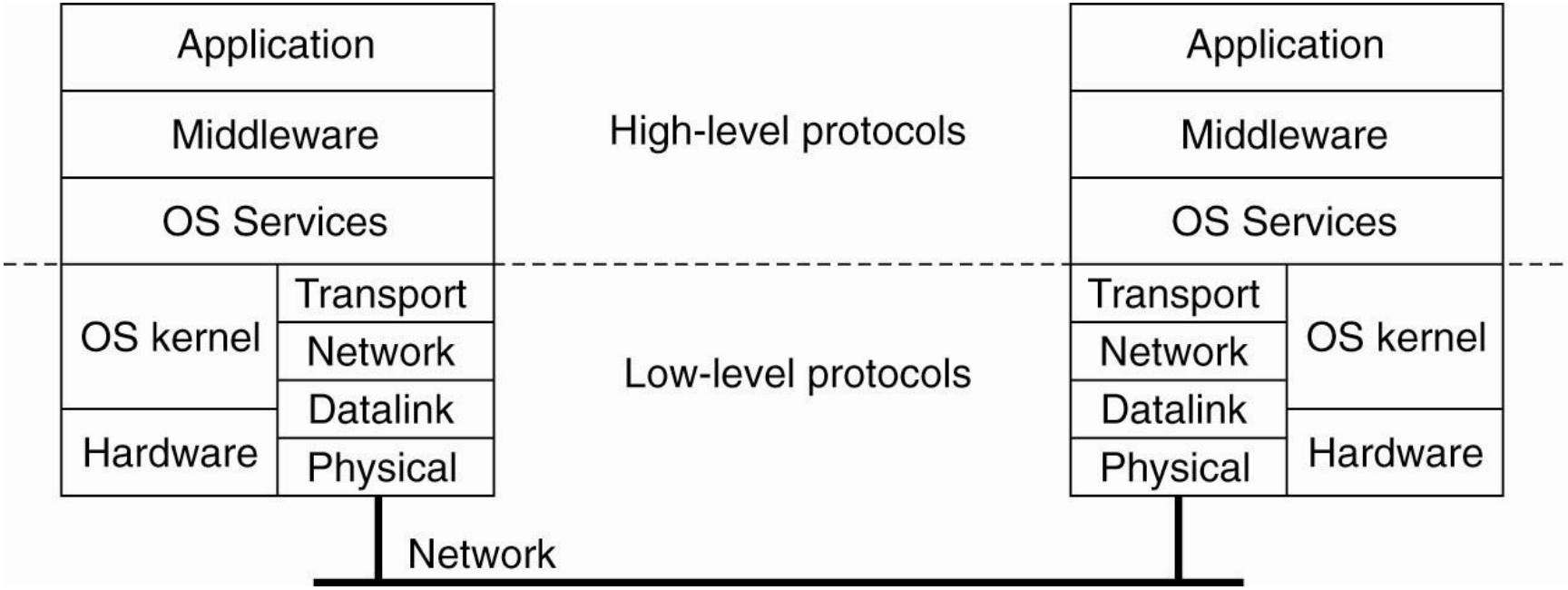


Figure 9-3. The logical organization of a distributed system into several layers.

Layering of Security Mechanisms (2)

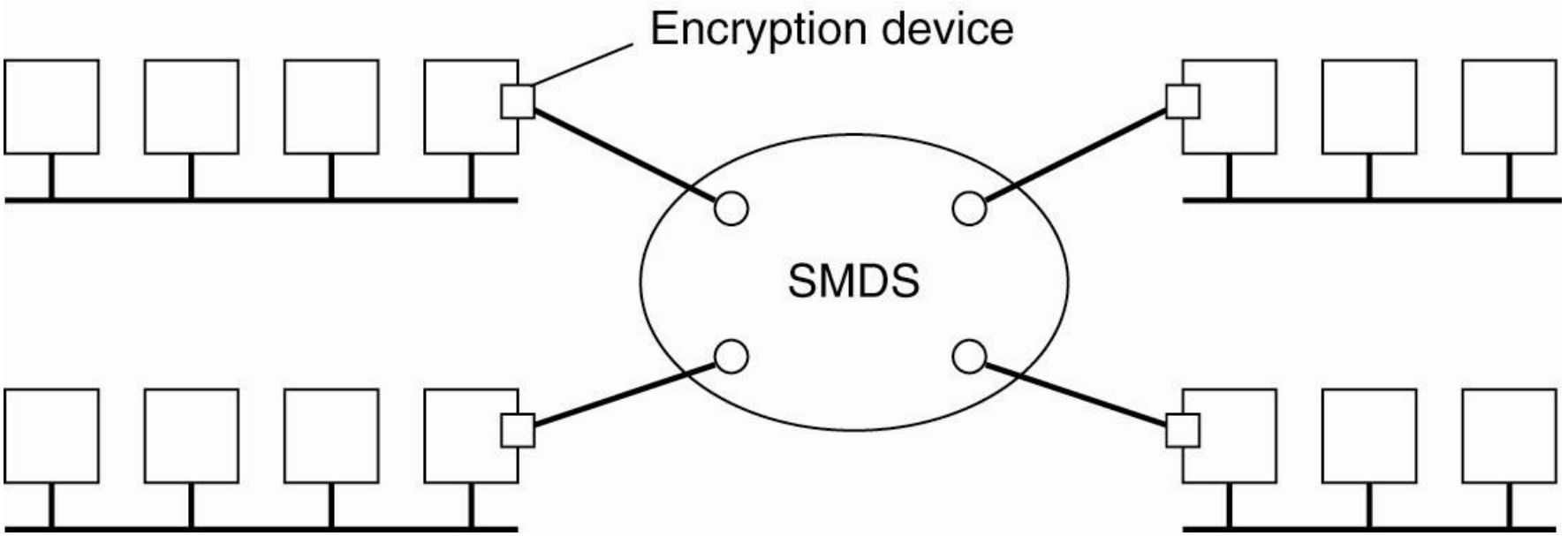


Figure 9-4. Several sites connected through a wide-area backbone service.

Distribution of Security Mechanisms

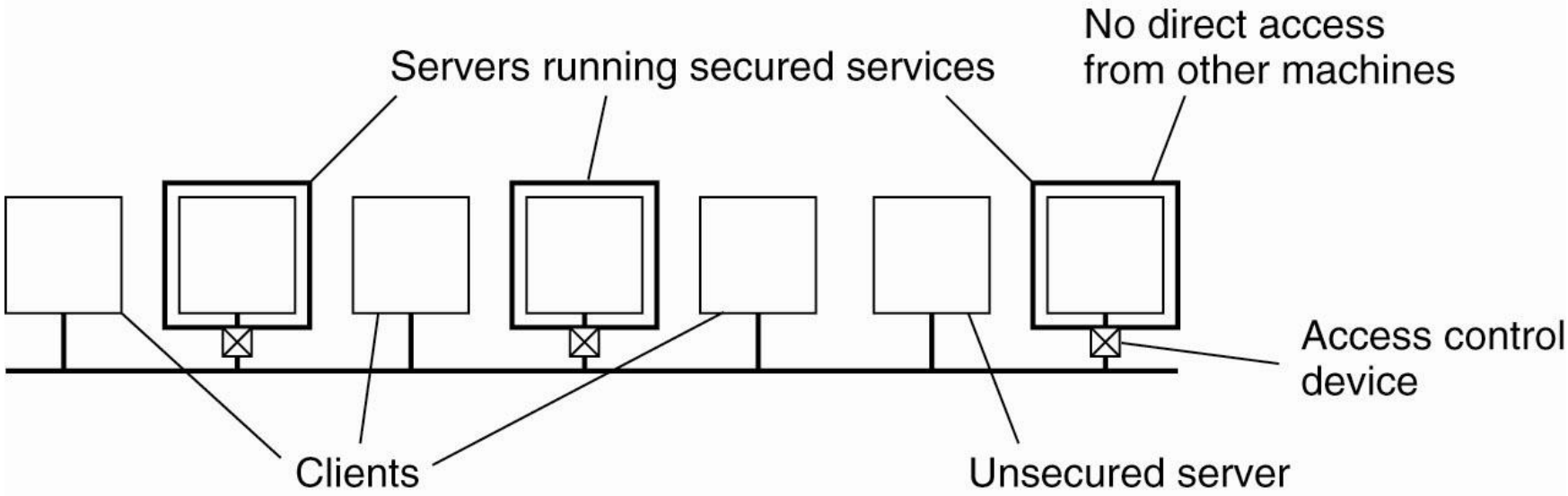


Figure 9-5. The principle of RISSC (Reduced Interfaces for Secure System Components) as applied to secure distributed systems.

Cryptography (1)

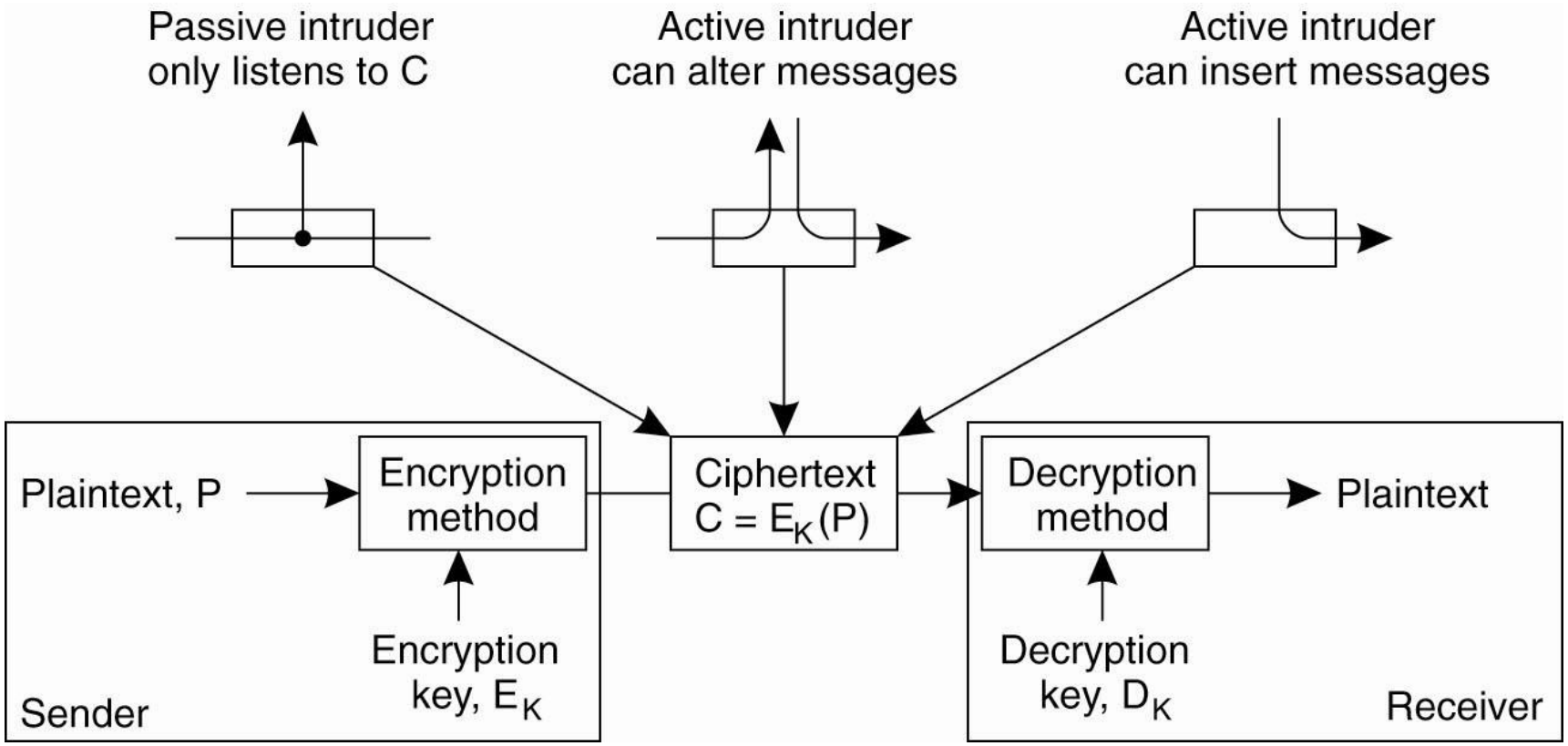


Figure 9-6. Intruders and eavesdroppers in communication.

Cryptography (2)

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

Figure 9-7. Notation used in this chapter.

Cryptography (3)

Symmetric cryptosystem. Encryption and decryption keys are the same.

Asymmetric cryptosystem (Public-key systems). The keys for encryption and decryption are different, but form a unique pair together.

For any encryption function, it should be computationally infeasible to find the *key K* when given the *plaintext P* and associated *ciphertext* $C=E_K(P)$.

When given a plaintext P and a key K, it should be infeasible to find another key K' such that $E_K(P) = E_{K'}(P)$.

Symmetric Cryptosystems: DES (1)

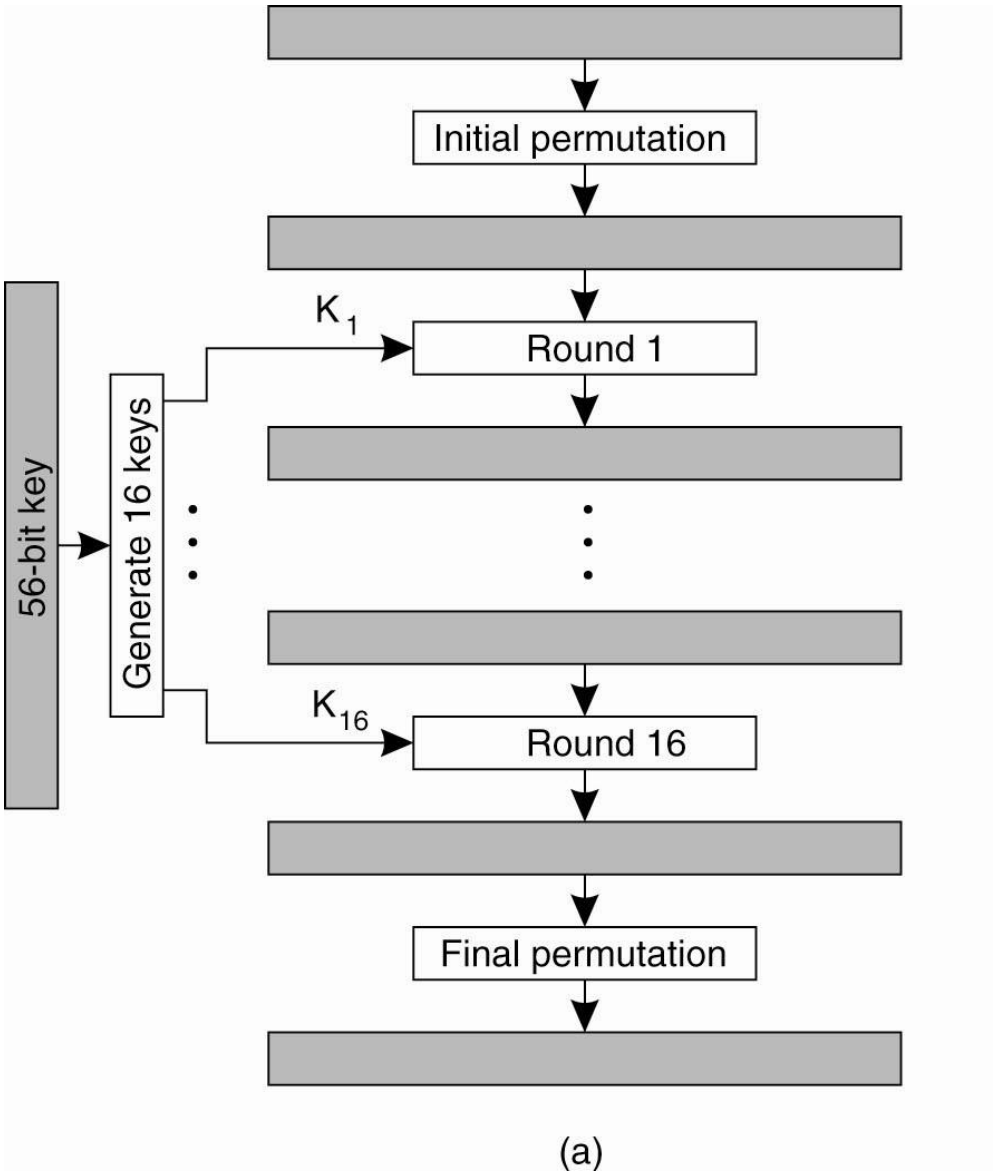


Figure 9-8. (a) The principle of DES.

Symmetric Cryptosystems: DES (2)

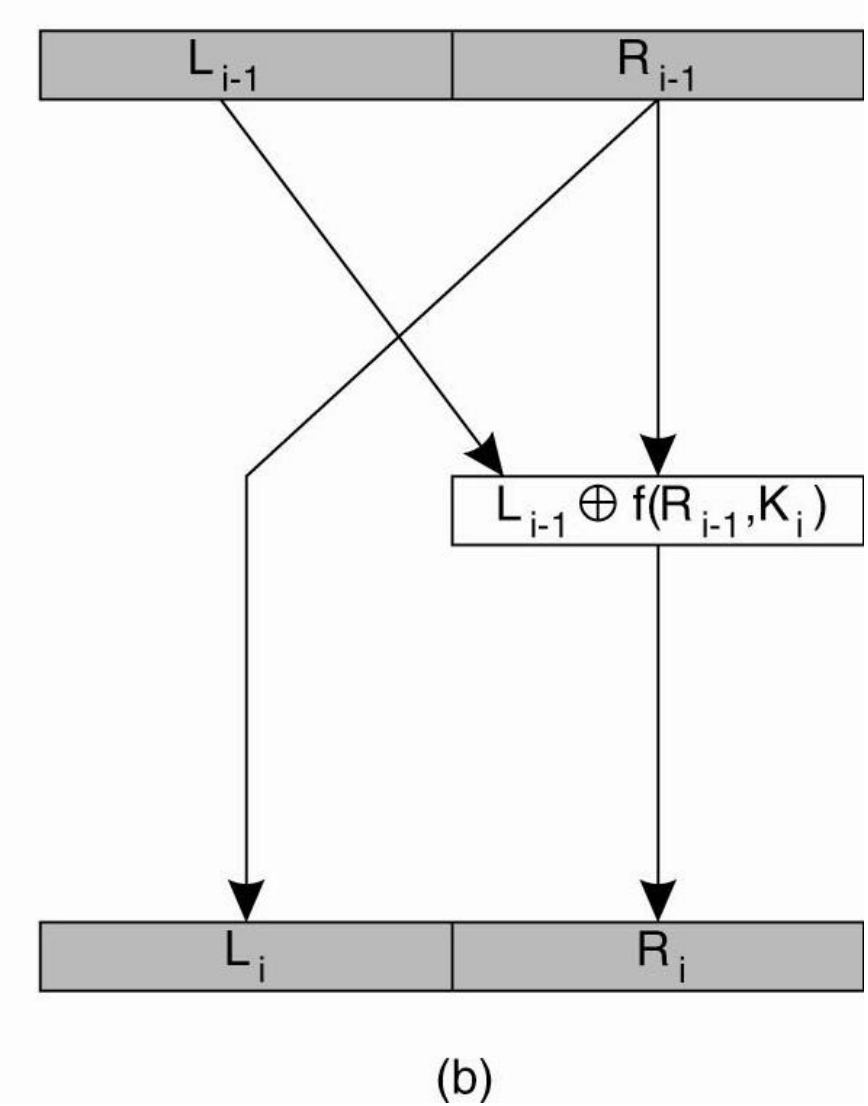


Figure 9-8. (b) Outline of one encryption round.

Symmetric Cryptosystems: DES (3)

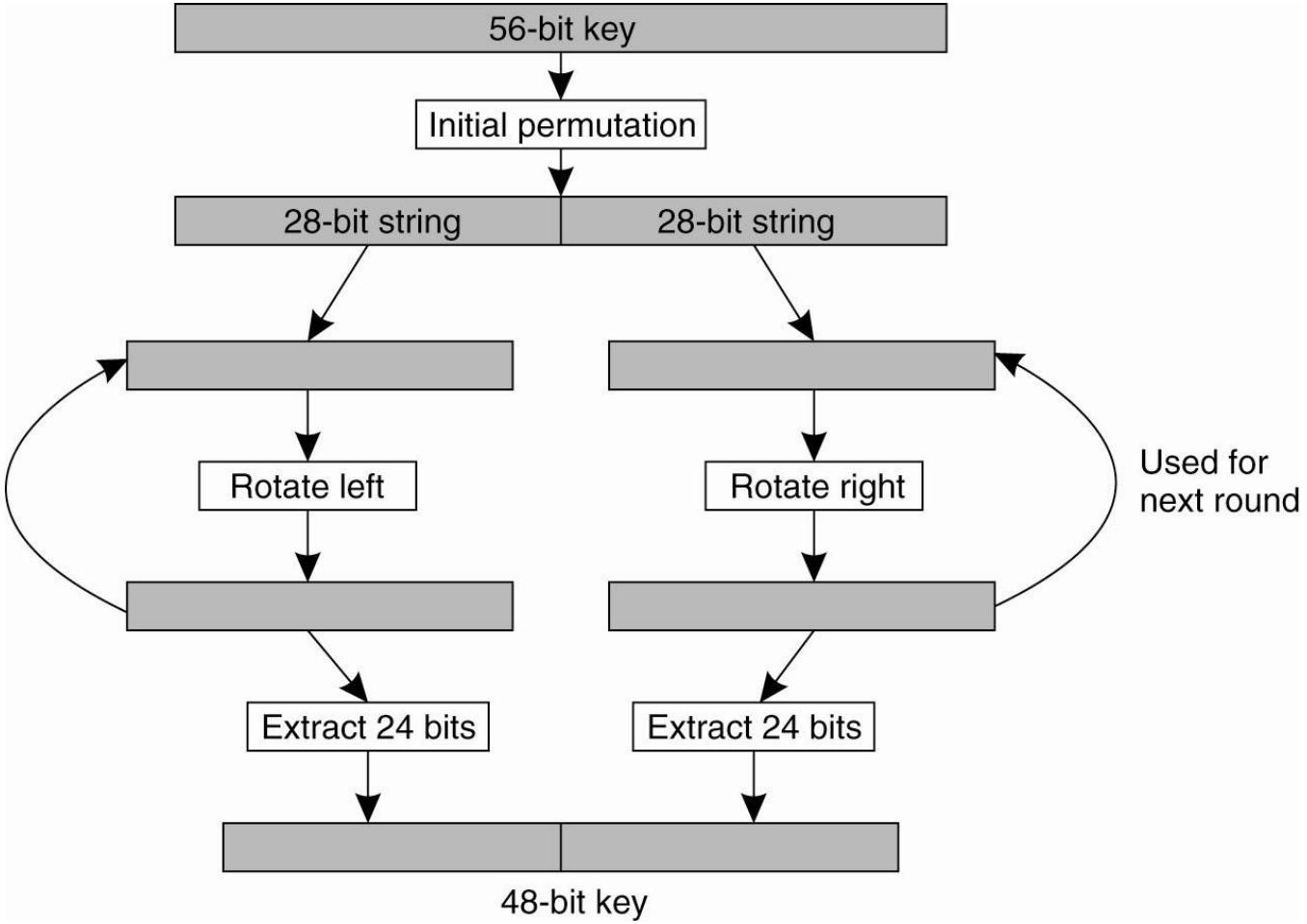


Figure 9-9. Details of per-round key generation in DES.

Advanced Encryption Standard

The Advanced Encryption Standard (AES) (also known as Rijndael), is a block cipher adopted as an encryption standard by the US government. It has been analyzed extensively and is now used worldwide.

Unlike DES, AES is a substitution-permutation network. AES is fast in both software and hardware, is relatively easy to implement and requires little memory.

Public-Key Cryptosystems: RSA

Generating the private and public keys requires four steps:

- Choose two very large prime numbers, p and q.
- Compute $n = p \times q$ and $z = (p - 1) \times (q - 1)$.
- Choose a number d that is relatively prime to z.
- Compute the number e such that
  $e \times d = 1 \bmod z$.

Hash Functions: MD5 (1)

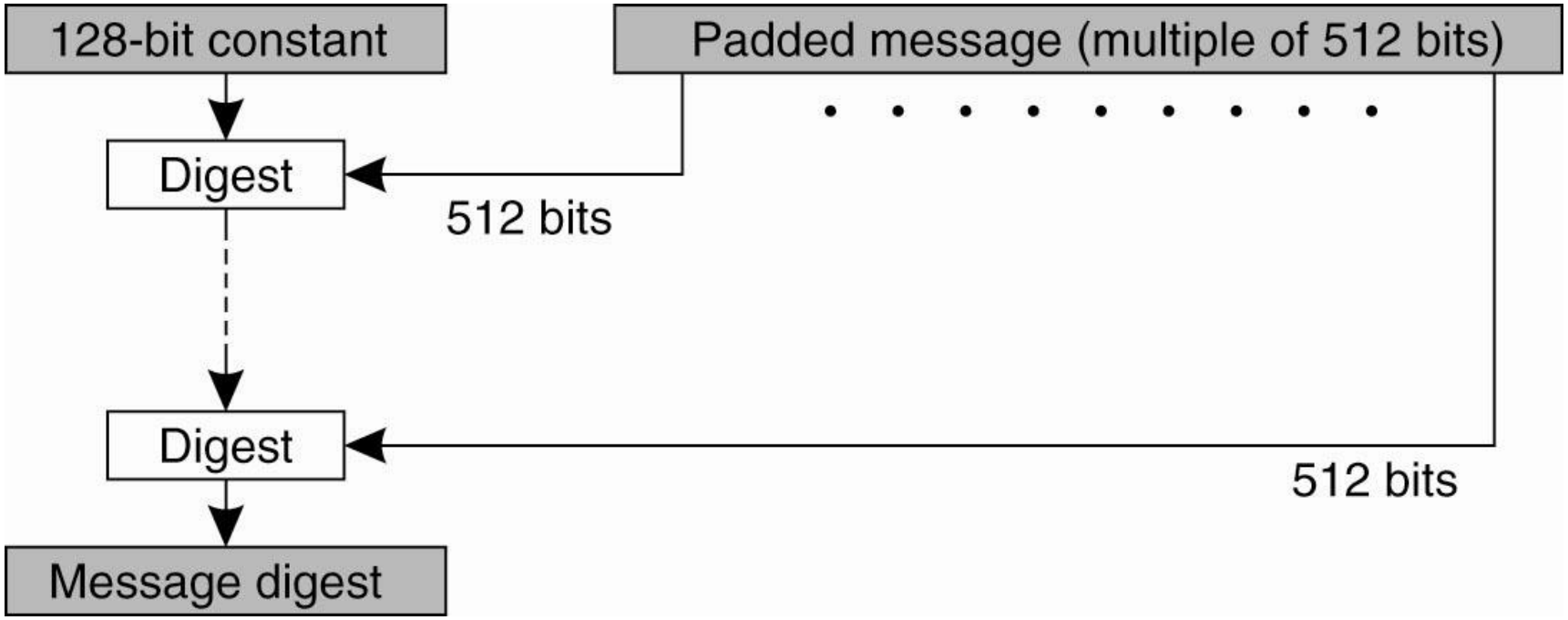


Figure 9-10. The structure of MD5.

Hash Functions: MD 5 (2)

Each phase in MD5 consists of four rounds of computations, where each
round uses one of the following four functions.

F(x,y,z) = (x AND y) OR ((NOT x) AND z)
G(x,y,z) = (x AND z) OR (y AND (NOT z))
H(x,y,z) = x XOR y XOR z
I(x,y,z) = y y XOR (x OR (NOT z))

Each of the above functions operate on 32-bit variables.

Hash Functions : MD5 (3)

Iterations 1-8	Iterations 9-16
$p \leftarrow (p + F(q,r,s) + b_0 + C_1) \lll 7$	$p \leftarrow (p + F(q,r,s) + b_8 + C_9) \lll 7$
$s \leftarrow (s + F(p,q,r) + b_1 + C_2) \lll 12$	$s \leftarrow (s + F(p,q,r) + b_9 + C_{10}) \lll 12$
$r \leftarrow (r + F(s,p,q) + b_2 + C_3) \lll 17$	$r \leftarrow (r + F(s,p,q) + b_{10} + C_{11}) \lll 17$
$q \leftarrow (q + F(r,s,p) + b_3 + C_4) \lll 22$	$q \leftarrow (q + F(r,s,p) + b_{11} + C_{12}) \lll 22$
$p \leftarrow (p + F(q,r,s) + b_4 + C_5) \lll 7$	$p \leftarrow (p + F(q,r,s) + b_{12} + C_{13}) \lll 7$
$s \leftarrow (s + F(p,q,r) + b_5 + C_6) \lll 12$	$s \leftarrow (s + F(p,q,r) + b_{13} + C_{14}) \lll 12$
$r \leftarrow (r + F(s,p,q) + b_6 + C_7) \lll 17$	$r \leftarrow (r + F(s,p,q) + b_{14} + C_{15}) \lll 17$
$q \leftarrow (q + F(r,s,p) + b_7 + C_8) \lll 22$	$q \leftarrow (q + F(r,s,p) + b_{15} + C_{16}) \lll 22$

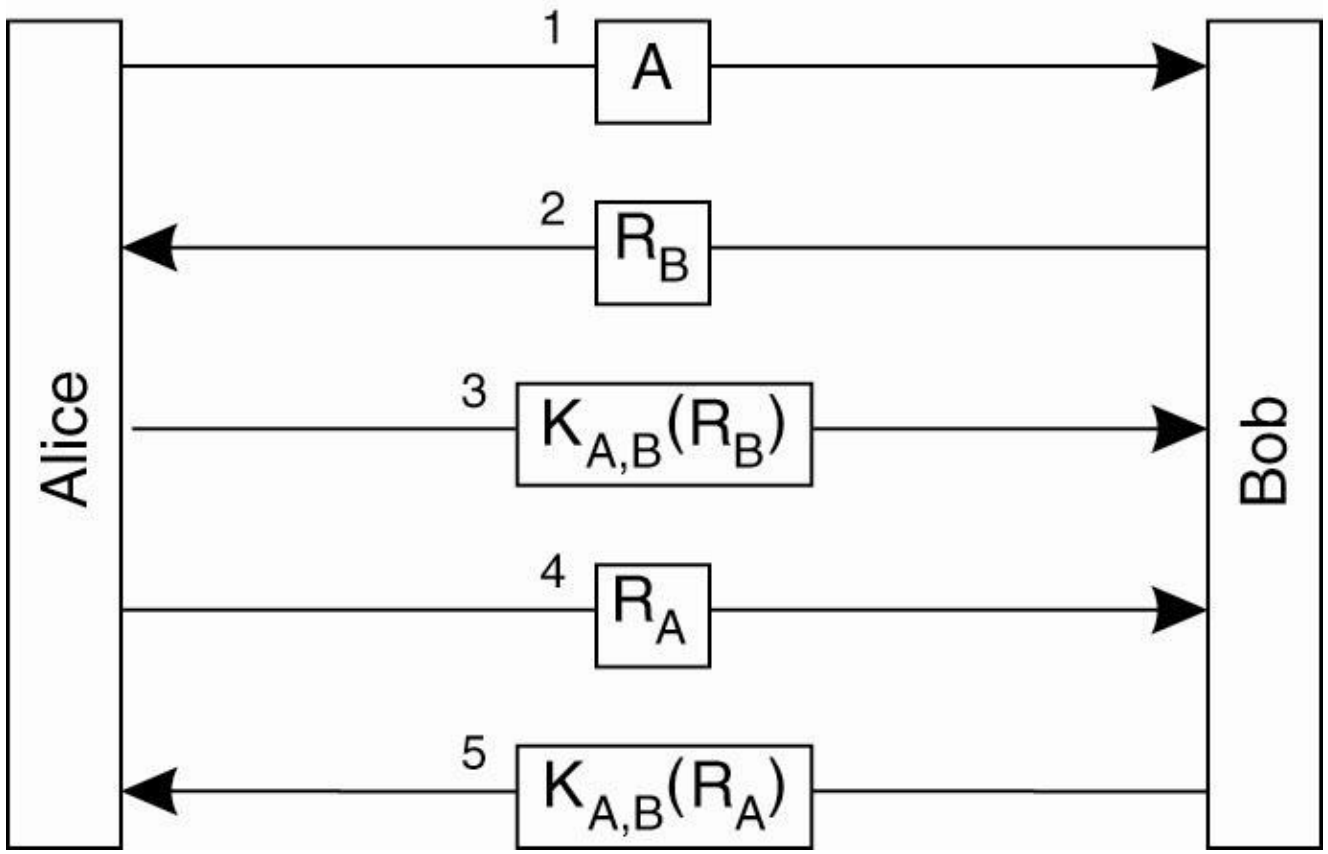
The 16 iterations during the first round in a phase in MD5. The C's are predefined constants. A 512-bit block is divided into 16 32-bit blocks $b_0 ... b_{15}$ for processing.

Secure Channels

A *secure channel* protects senders and receivers against interception, modification and fabrication of messages. It does not also necessarily protect against interruption.

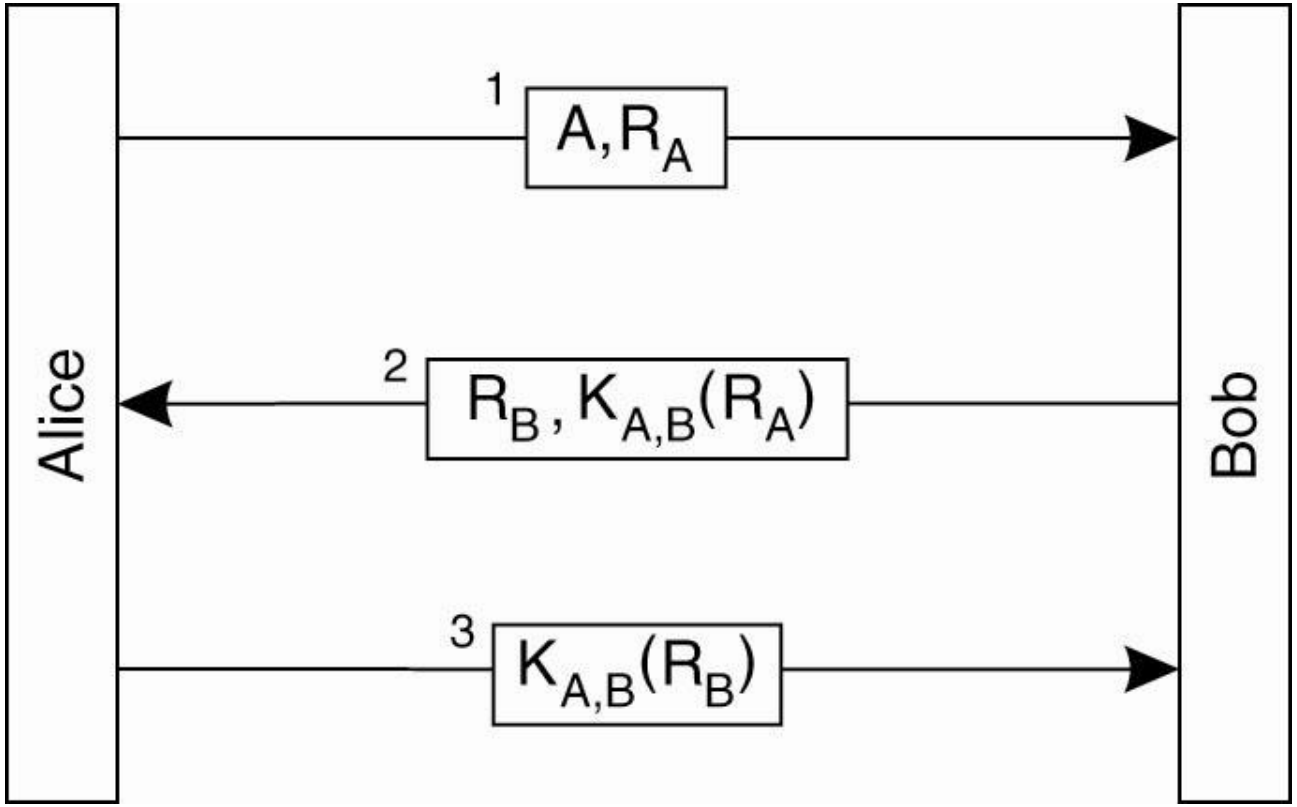
A secure channel provides for *authentication, confidentiality and message integrity.*

Authentication Based on a Shared Secret Key (1)



Authentication based on a shared secret key. An example of a *challenge-response protocol*.

Authentication Based on a Shared Secret Key (2)



In correct authentication based on a shared secret key, but using three instead of five messages.

Authentication Based on a Shared Secret Key (3)

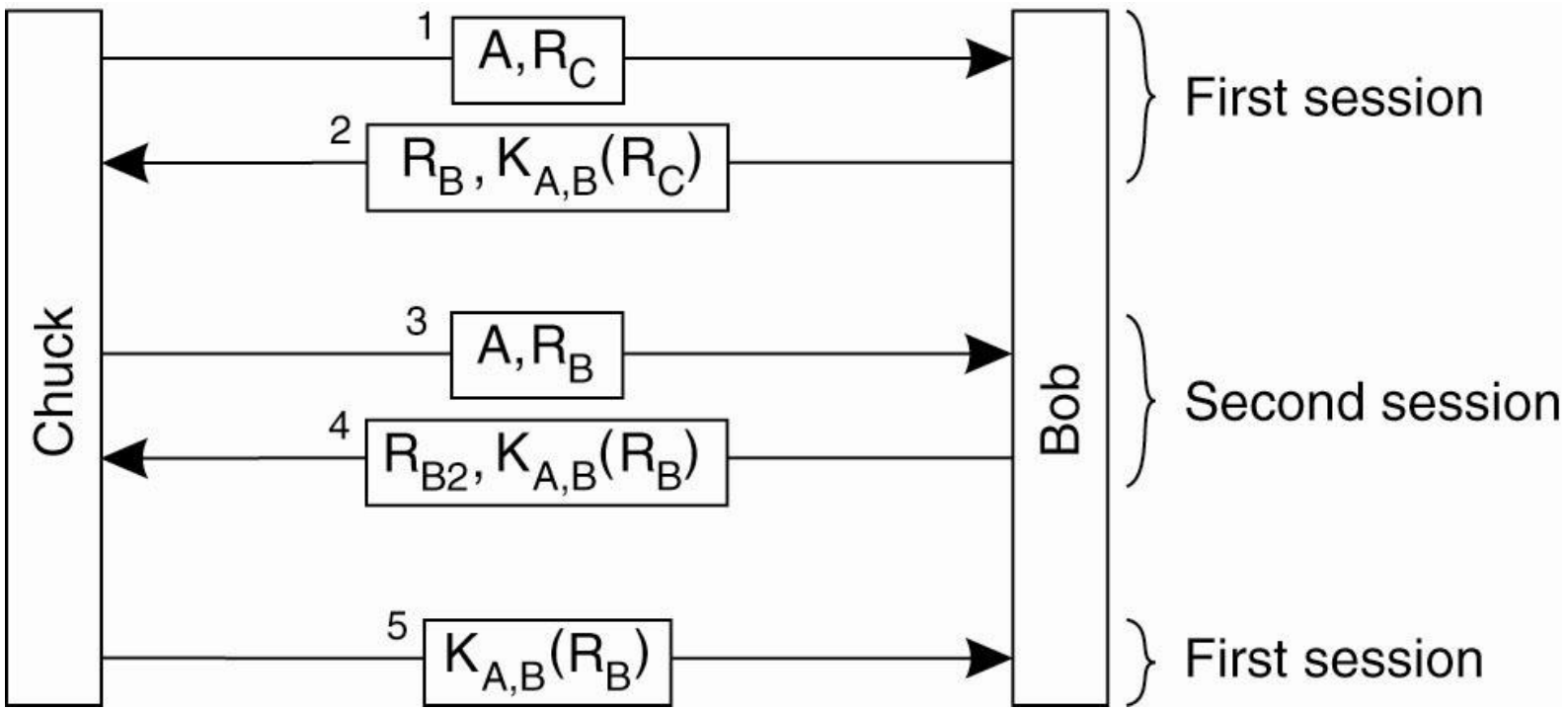
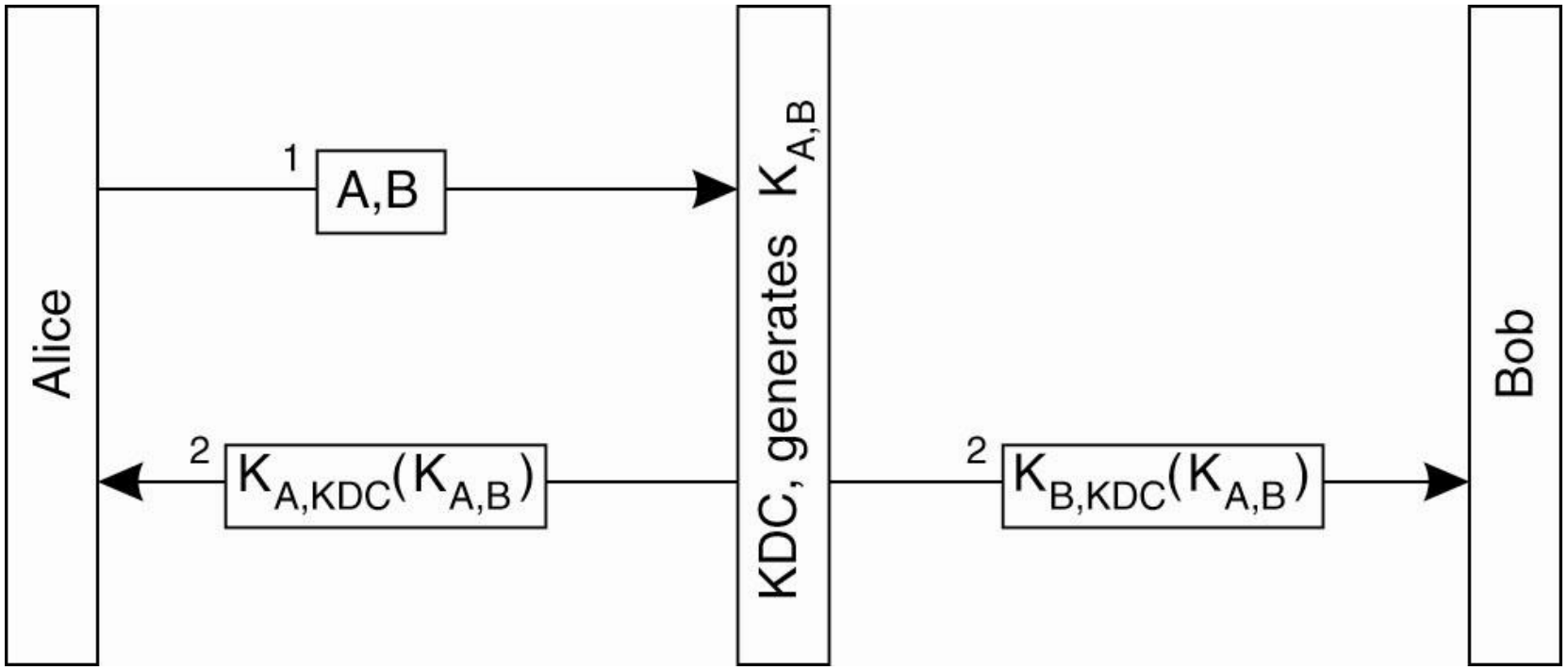


Figure 9-14. The reflection attack.

Scalability

N hosts would need *N(N-1)/2* keys and each host would have to manage N-1 keys. A better scheme is to use a centralized Key Distribution Center.

Authentication Using a Key Distribution Center (1)



The principle of using a KDC. What if Alice starts opening a connection with Bob before Bob receives his message from the KDC?

Authentication Using a Key Distribution Center (2)

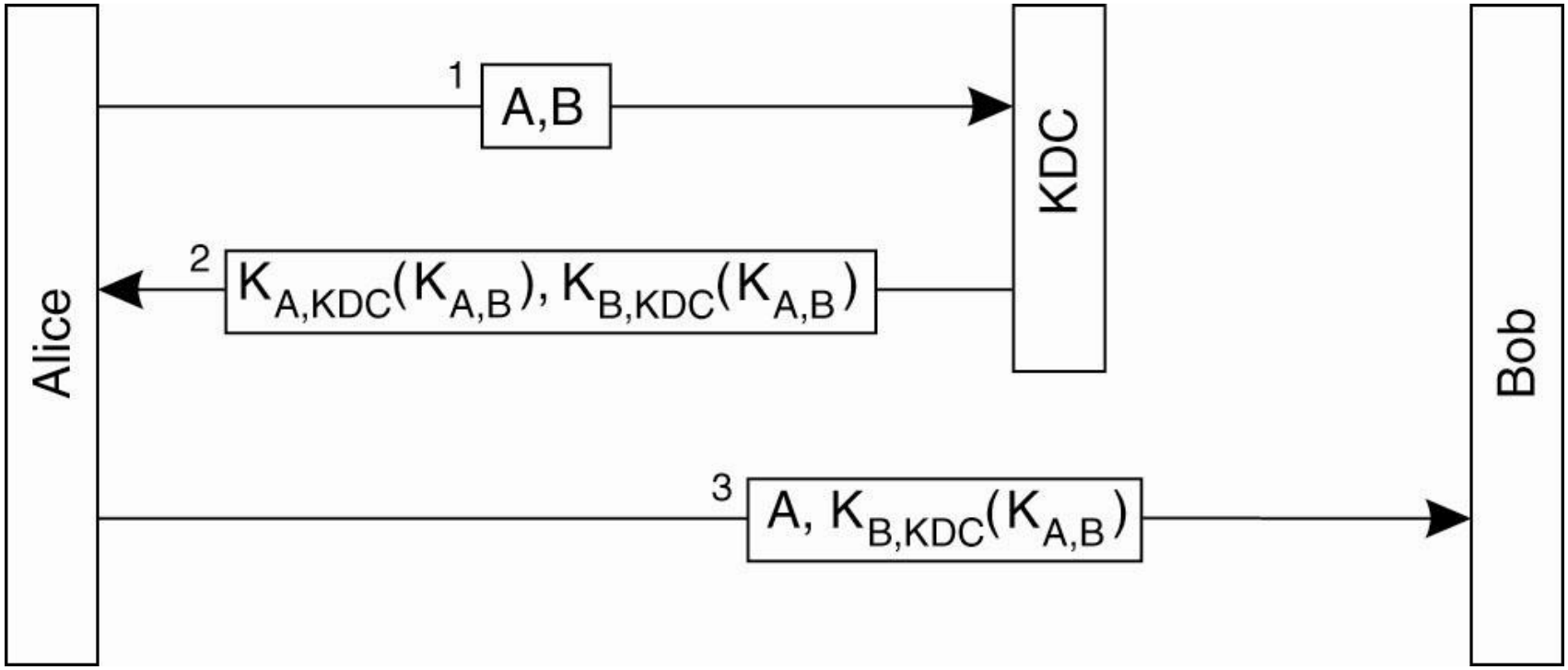


Figure 9-16. Using a ticket and letting Alice set up a connection to Bob.

Authentication Using a Key Distribution Center (3)

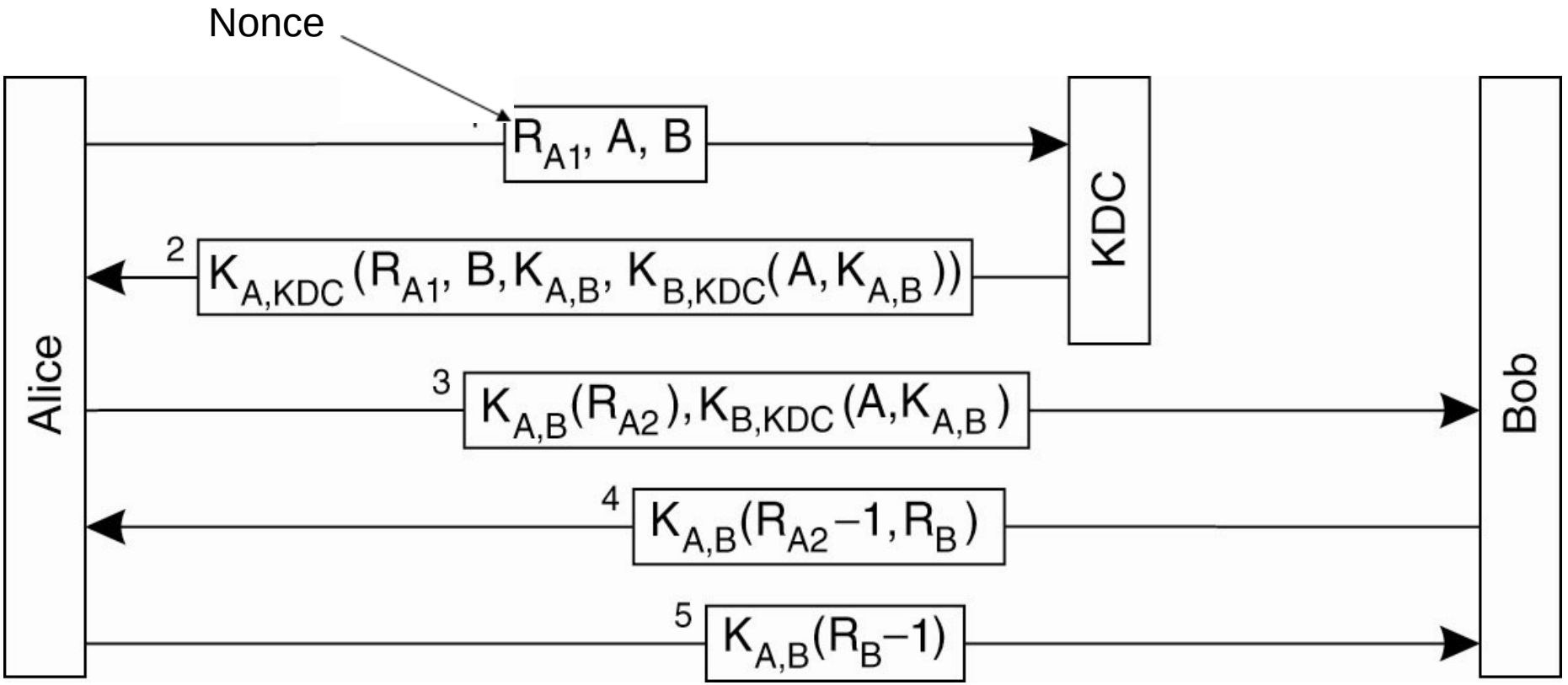


Figure 9-17. The Needham-Schroeder authentication protocol.

Observations

Need for nonces. Suppose Alice does not send a nonce. Also suppose Chuck has stolen one of Bob's old keys and intercepted an old response from the KDC. When Alice requests to set up a secure channel with Bob. Chuck replays the old message, fooling Alice into thinking that he is Bob. And Chuck can also decrypt the ticket to confirm that he is Bob.

Need for sending B in message 1 and 2. Without it, Chuck can intercept message 1 from Alice by replacing Bob's identity with Chuck's. The KDC would then think that Alice wants to set up a secure channel with Chuck and responds accordingly. As soon as Alice wants to contact Bob, Chuck intercepts the message and fools Alice into believing that she is talking to Bob.

Authentication Using a Key Distribution Center (4)

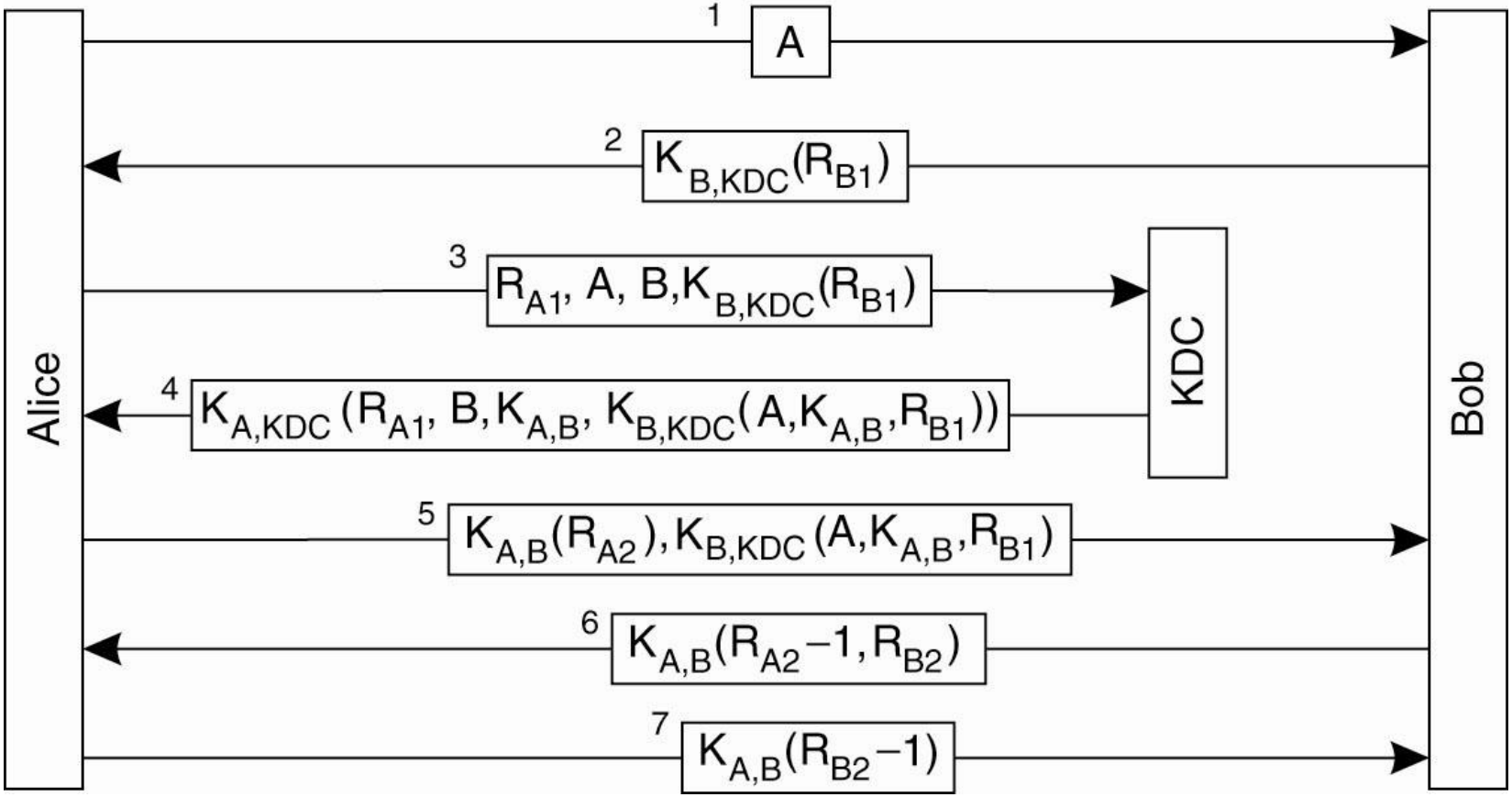


Figure 9-18. Protection against malicious reuse of a previously generated session key in the Needham-Schroeder protocol.

Authentication Using a Key Distribution Center (5)

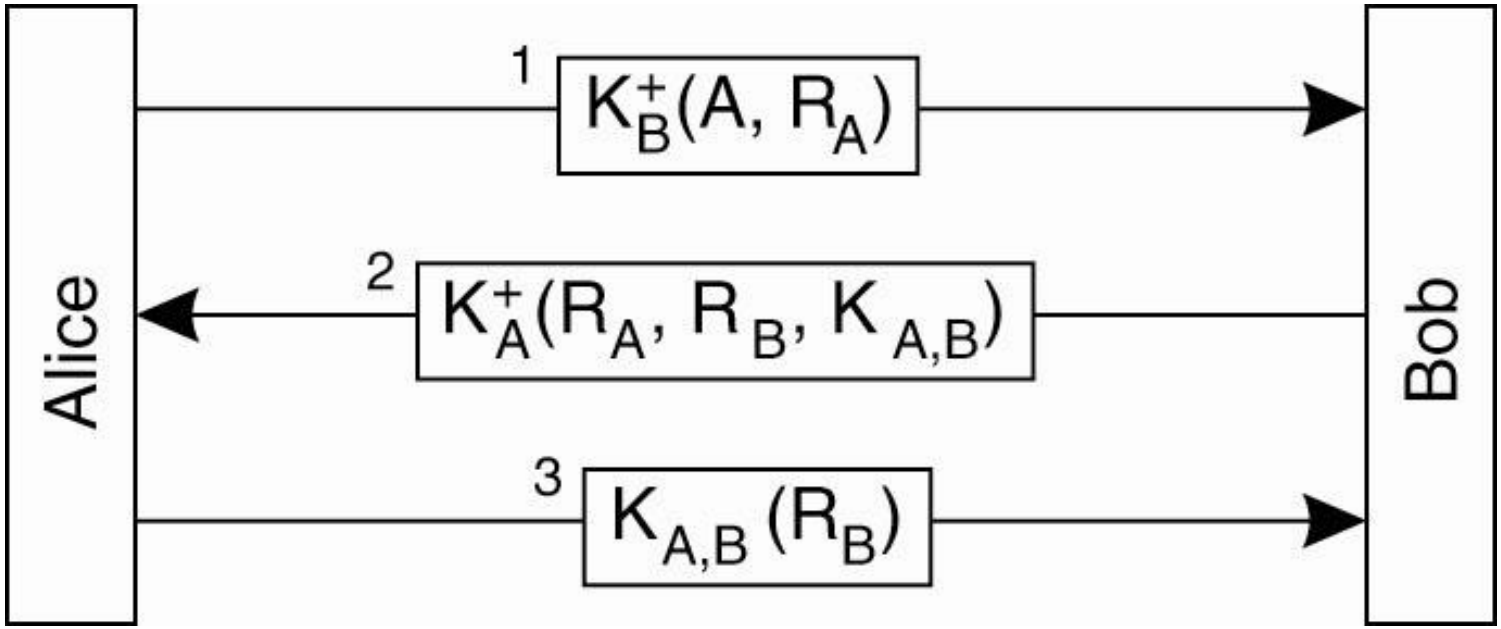


Figure 9-19. Mutual authentication in a public-key cryptosystem.

Digital Signatures (1)

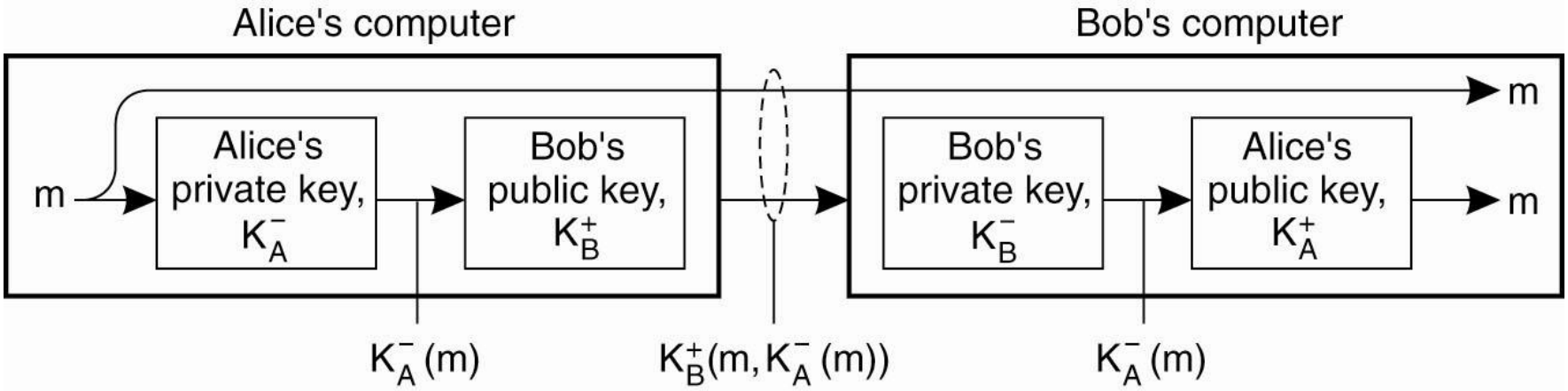


Figure 9-20. Digital signing a message using public-key cryptography.

Digital Signatures (2)

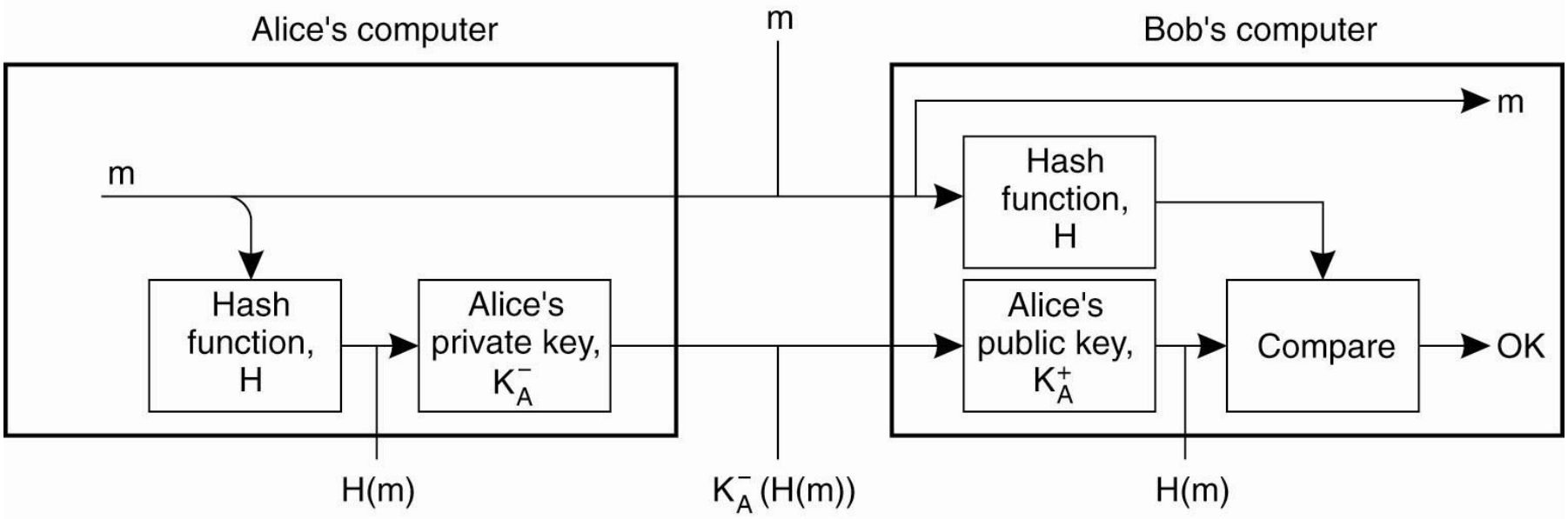


Figure 9-21. Digitally signing a message using a message digest.

Secure Replicated Servers

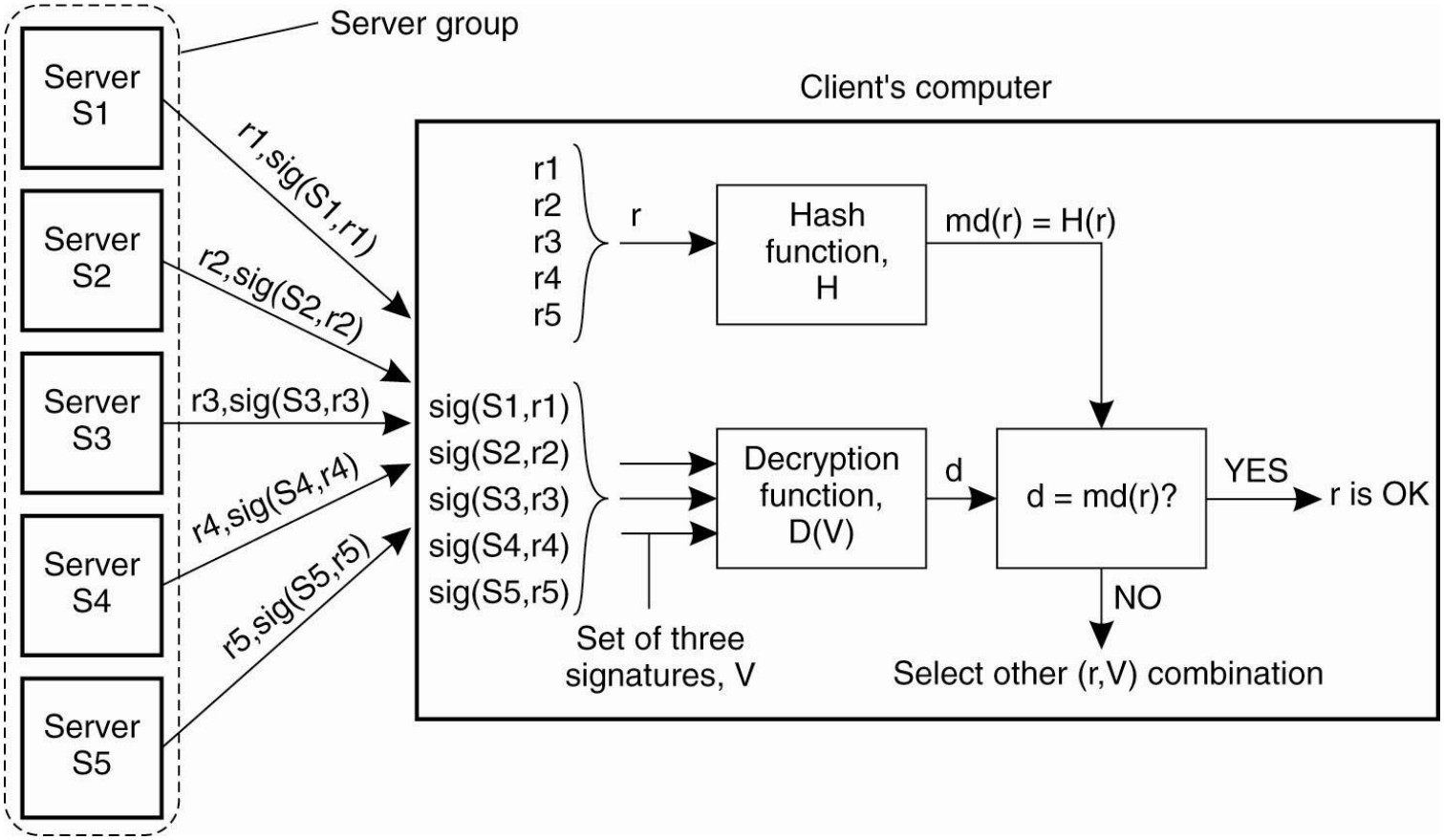


Figure 9-22. Sharing a secret signature in a group of replicated servers.

Example: Kerberos (1)

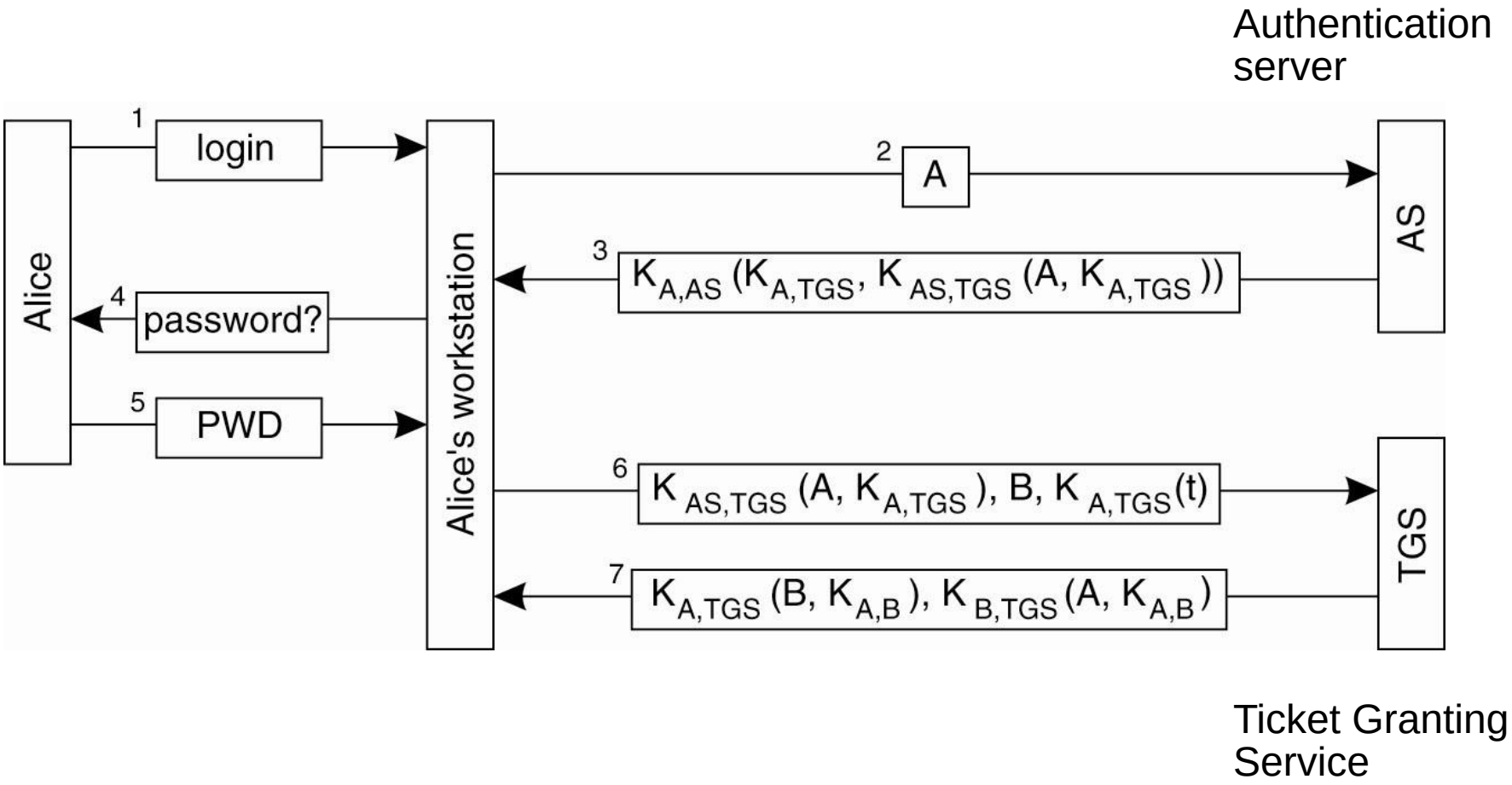


Figure 9-23. Authentication in Kerberos.

Example: Kerberos (2)

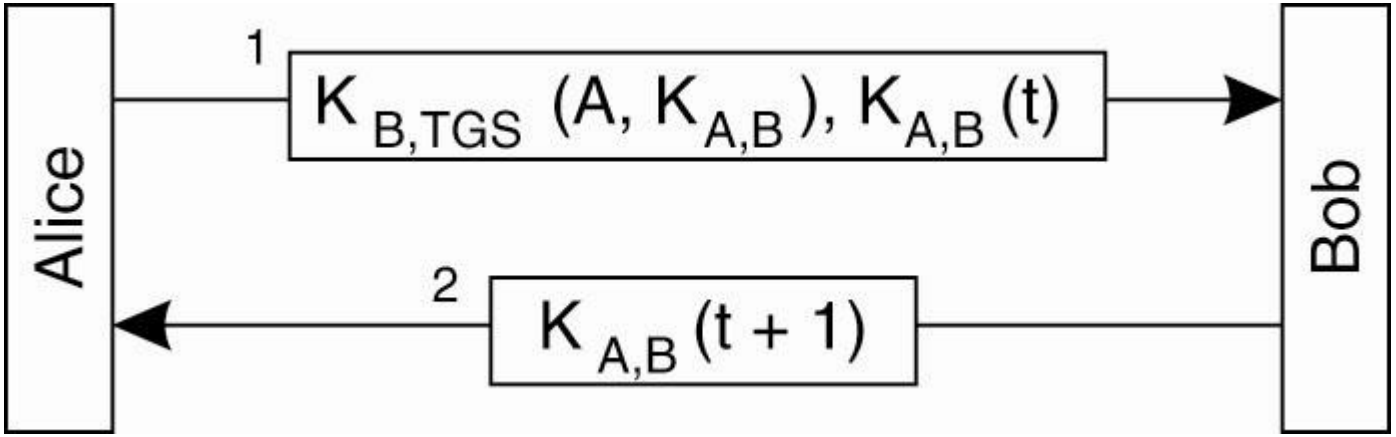


Figure 9-24. Setting up a secure channel in Kerberos.

General Issues in Access Control

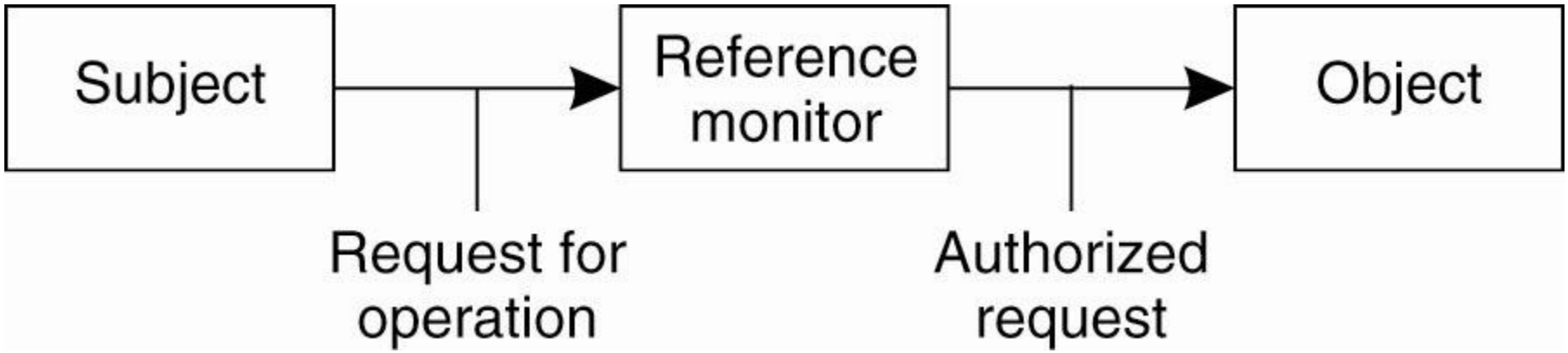


Figure 9-25. General model of controlling access to objects.

Access Control Matrix (1)

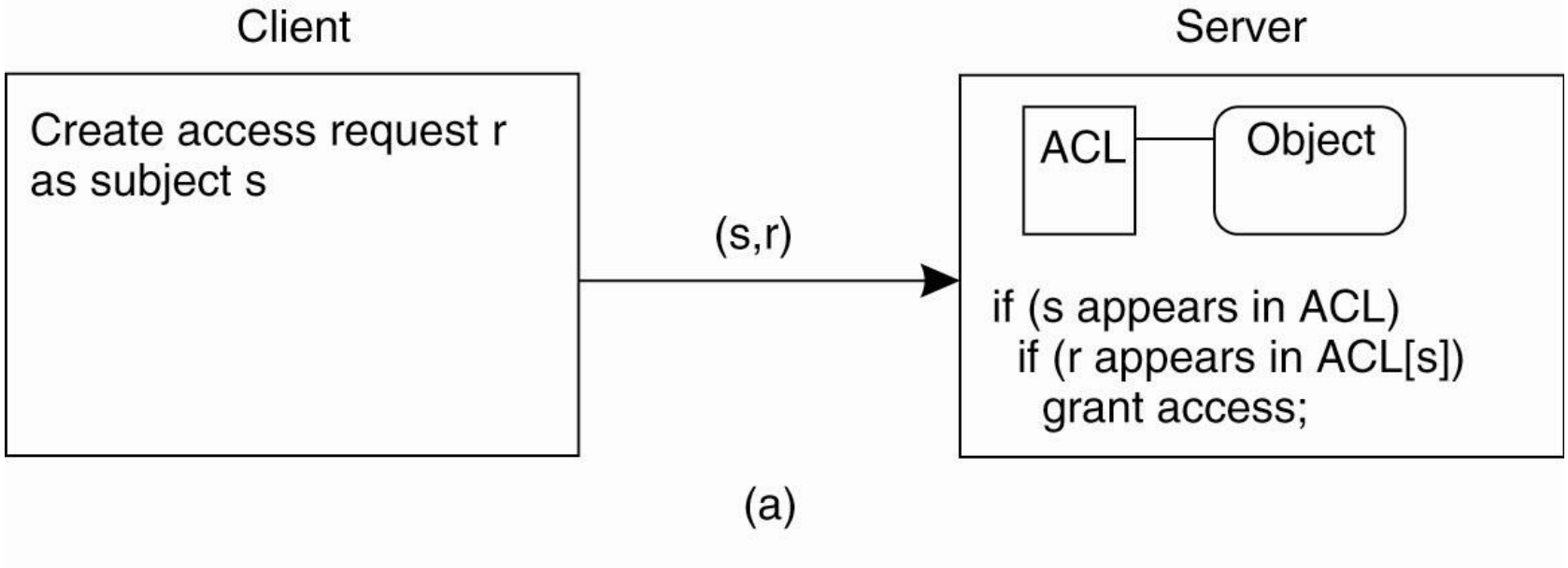


Figure 9-26. Comparison between ACLs and capabilities for protecting objects. (a) Using an ACL.

Access Control Matrix (2)

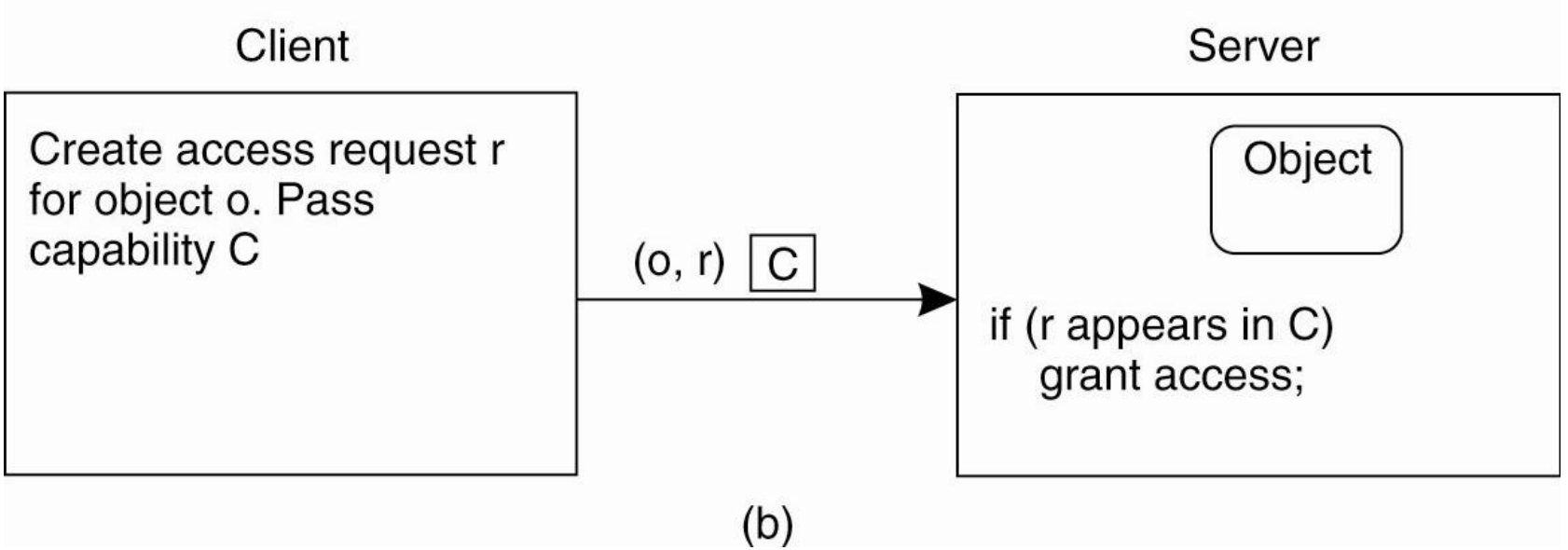


Figure 9-26. Comparison between ACLs and capabilities for protecting objects. (b) Using capabilities.

Protection Domains

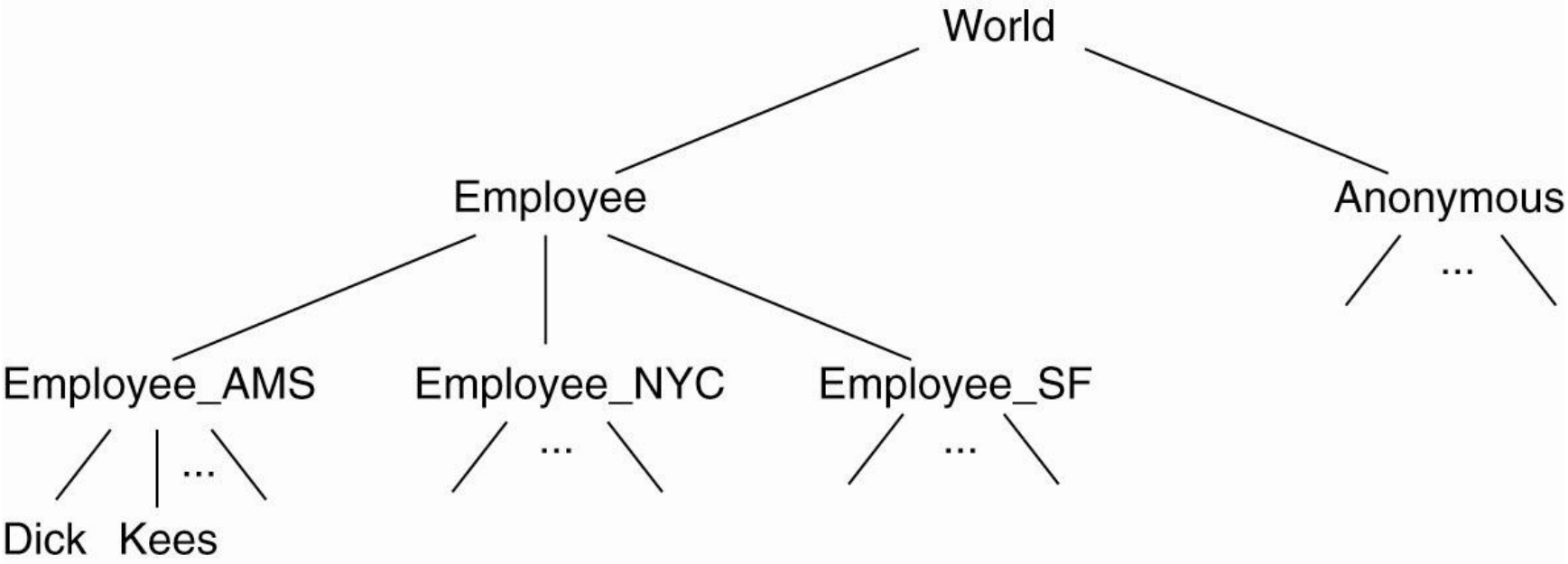


Figure 9-27. The hierarchical organization of protection domains as groups of users.

Firewalls

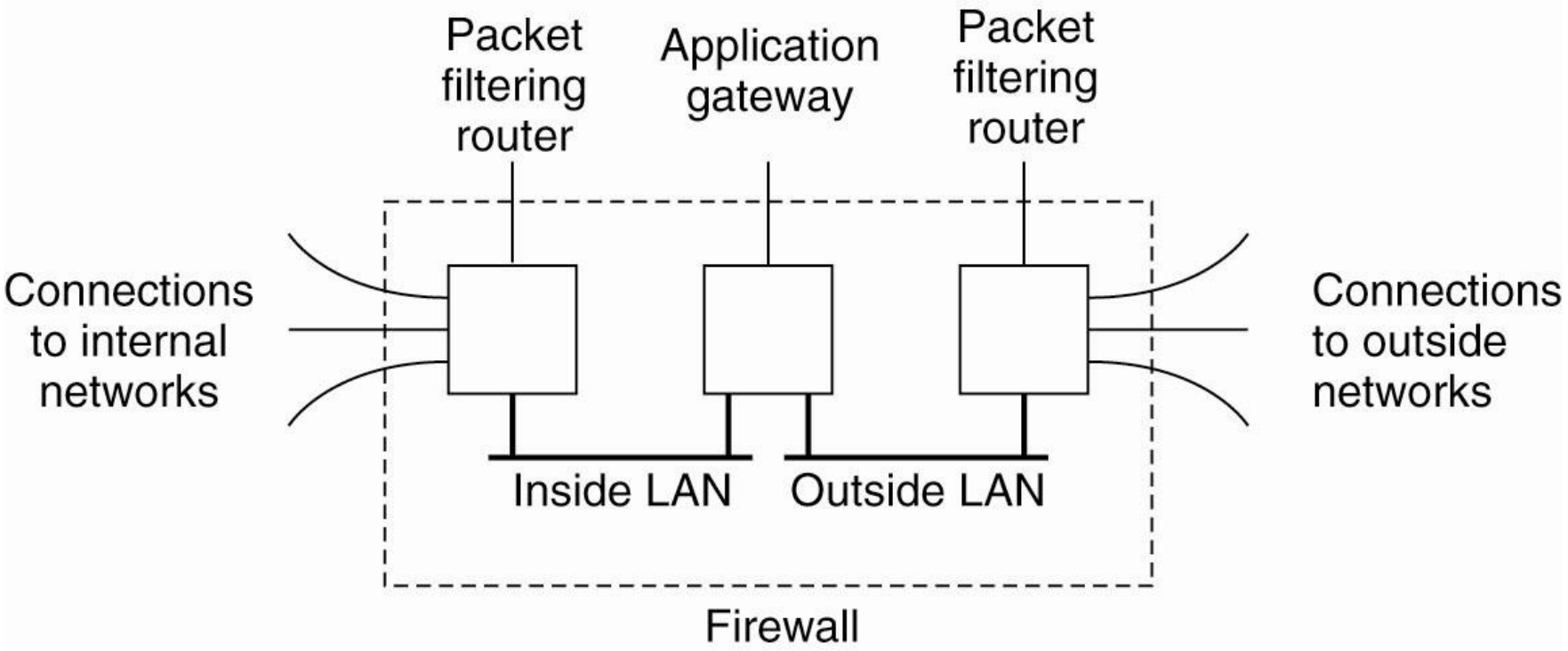


Figure 9-28. A common implementation of a firewall.

Types of Firewalls

- Packet filtering gateway
- Application-level gateway
- Proxy gateway

Protecting the Target (1)

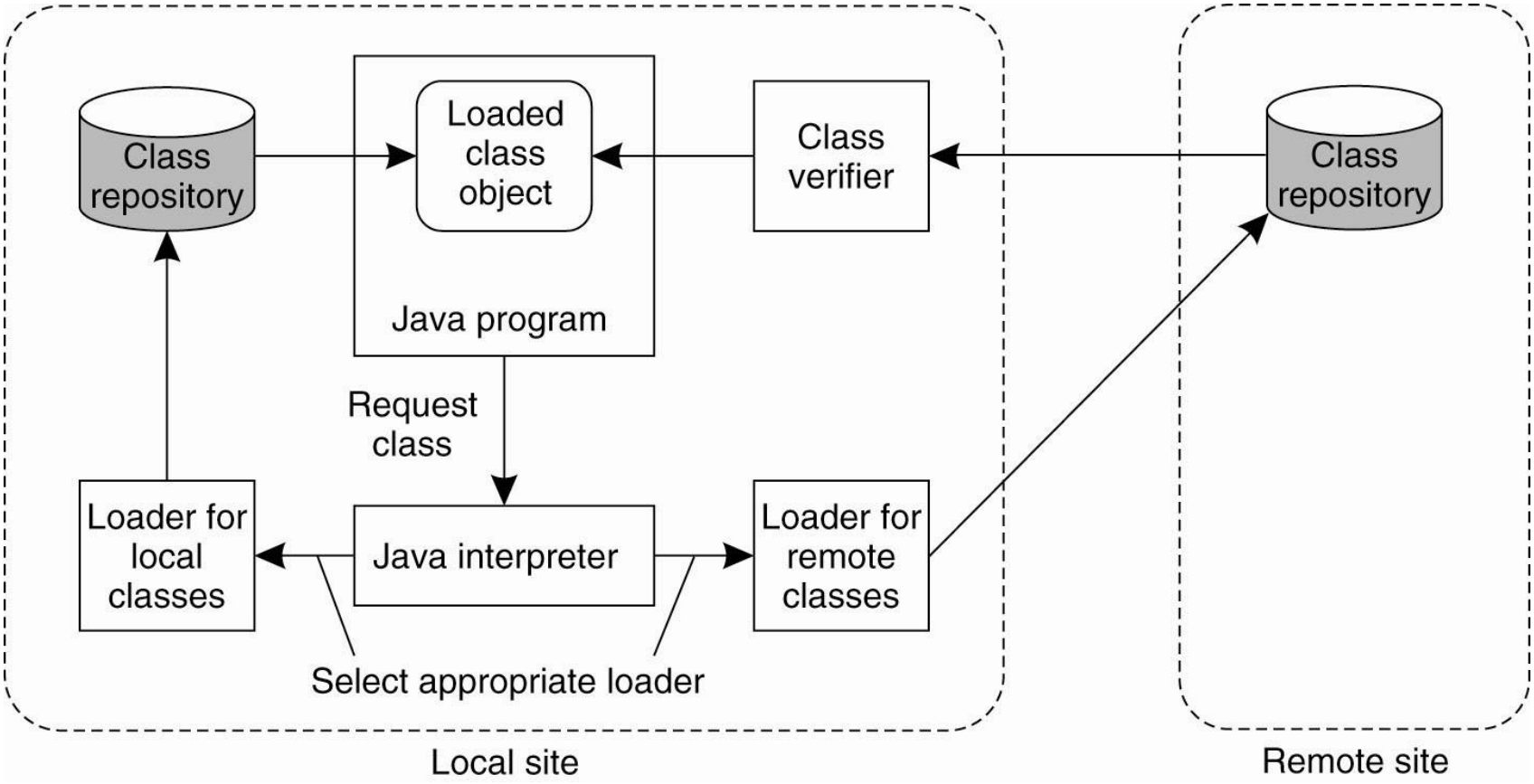


Figure 9-29. The organization of a Java sandbox.

Protecting the Target (2)

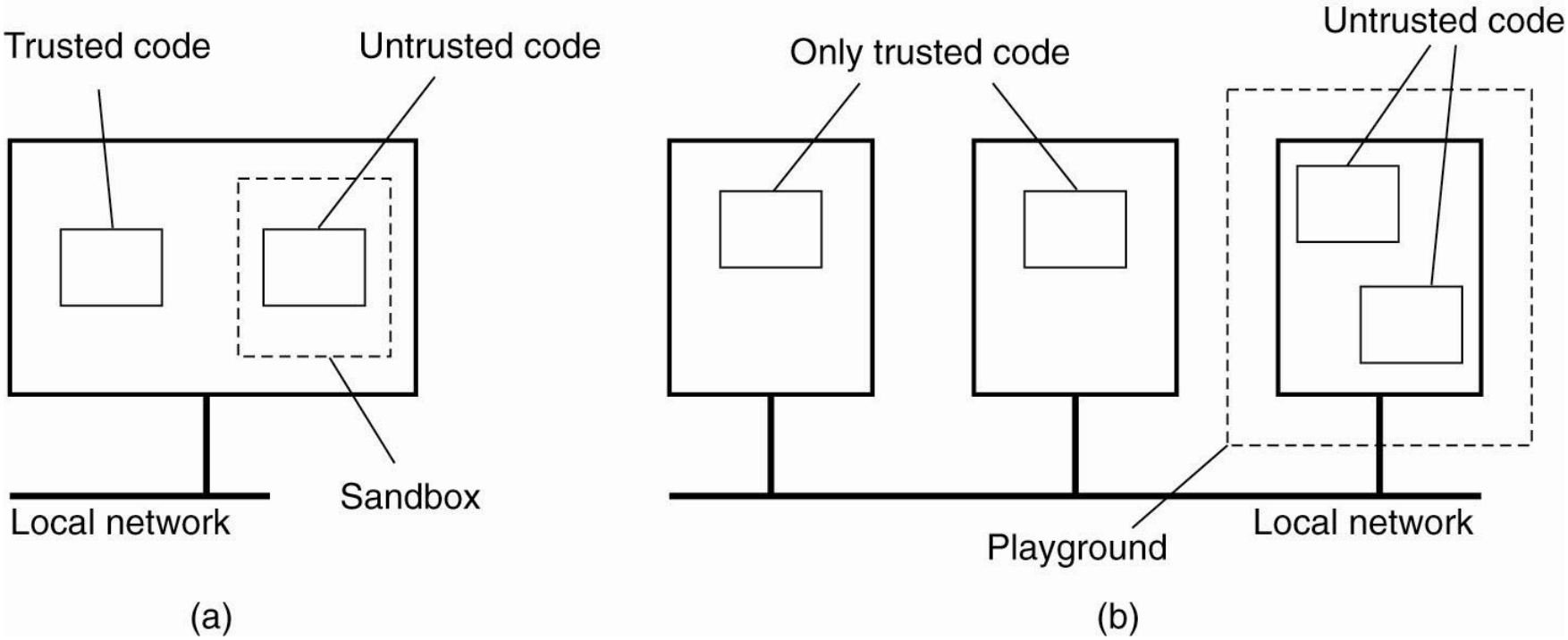


Figure 9-30. (a) A sandbox. (b) A playground.

Protecting the Target (3)

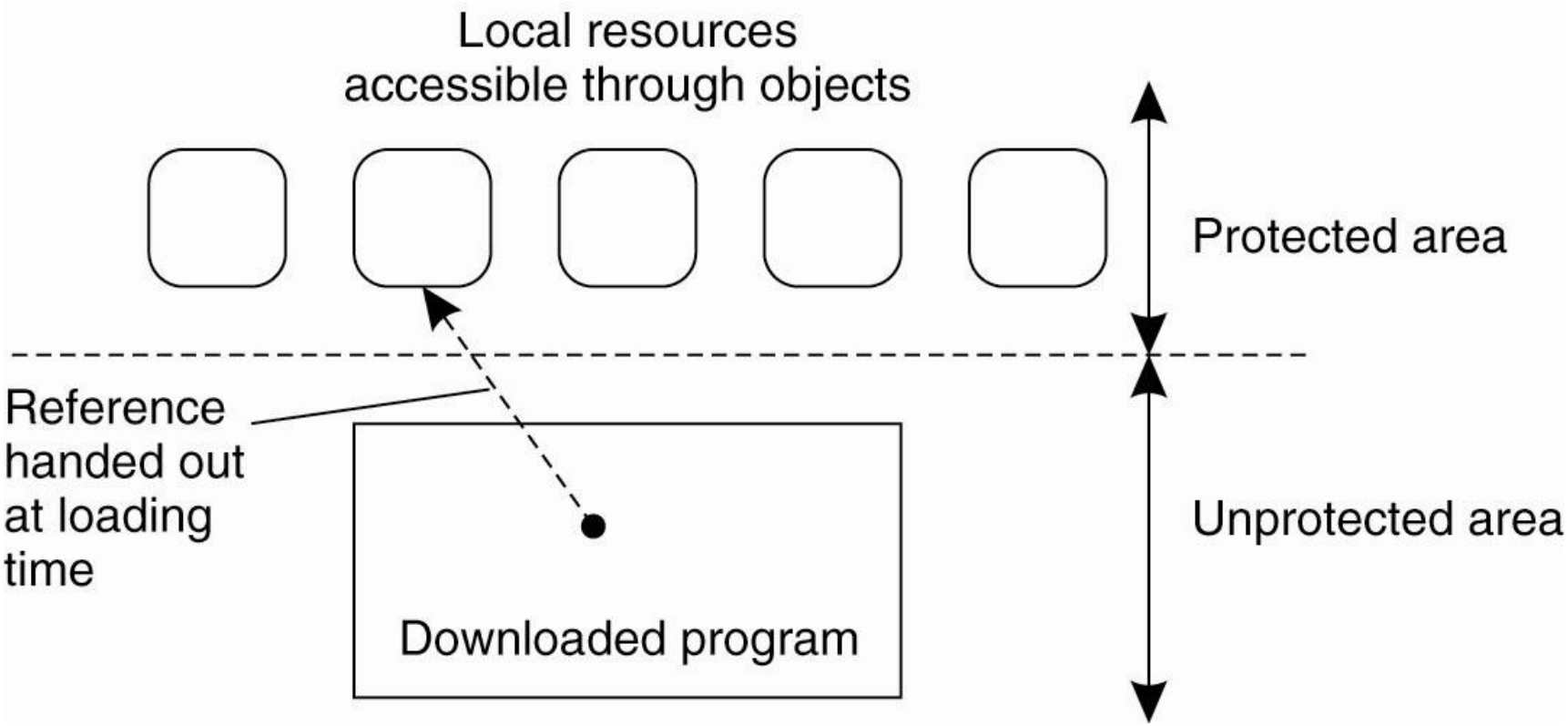


Figure 9-31. The principle of using Java object references as capabilities.

Protecting the Target (4)

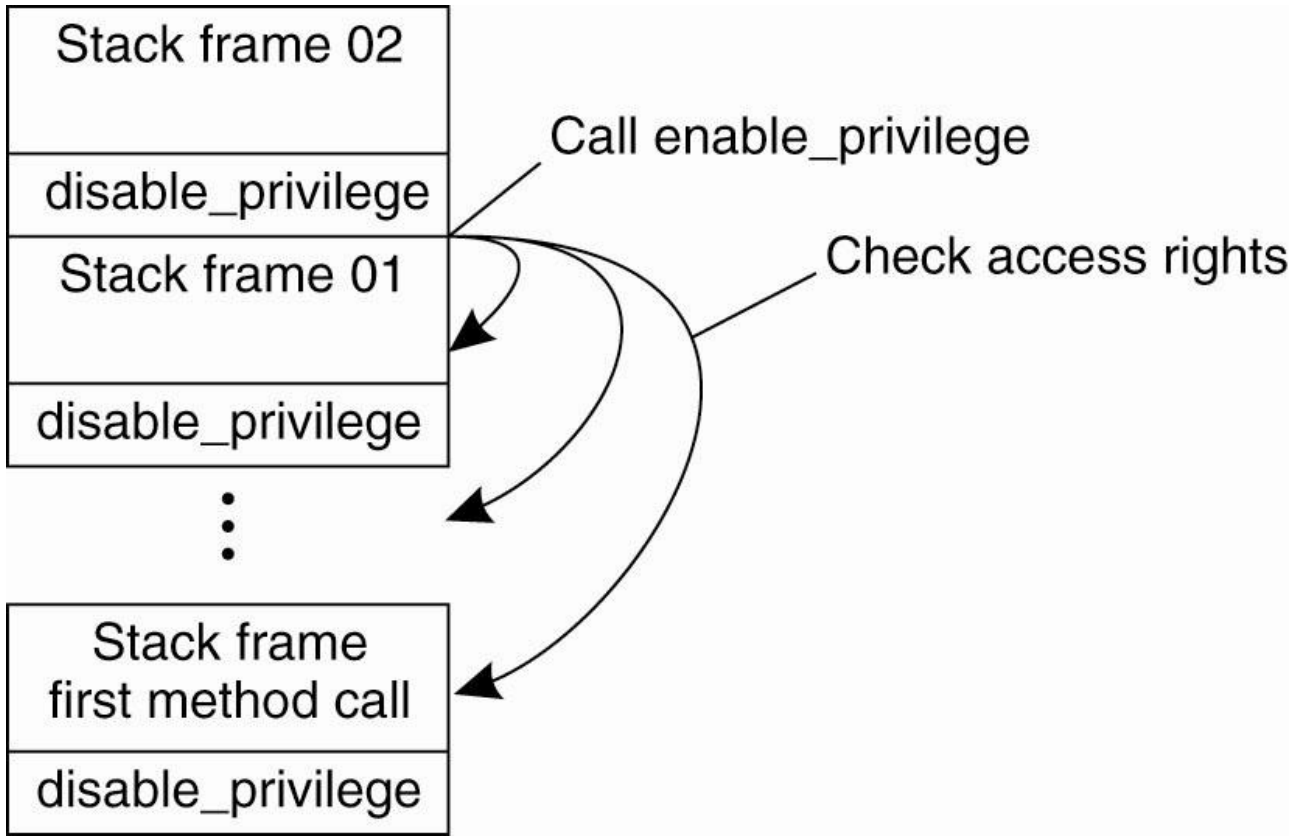


Figure 9-32. The principle of stack introspection.

Security Management

- General management of cryptographic keys
- Securely managing a group of servers
- Authorization management with capabilities and attribute certificates

Key Establishment

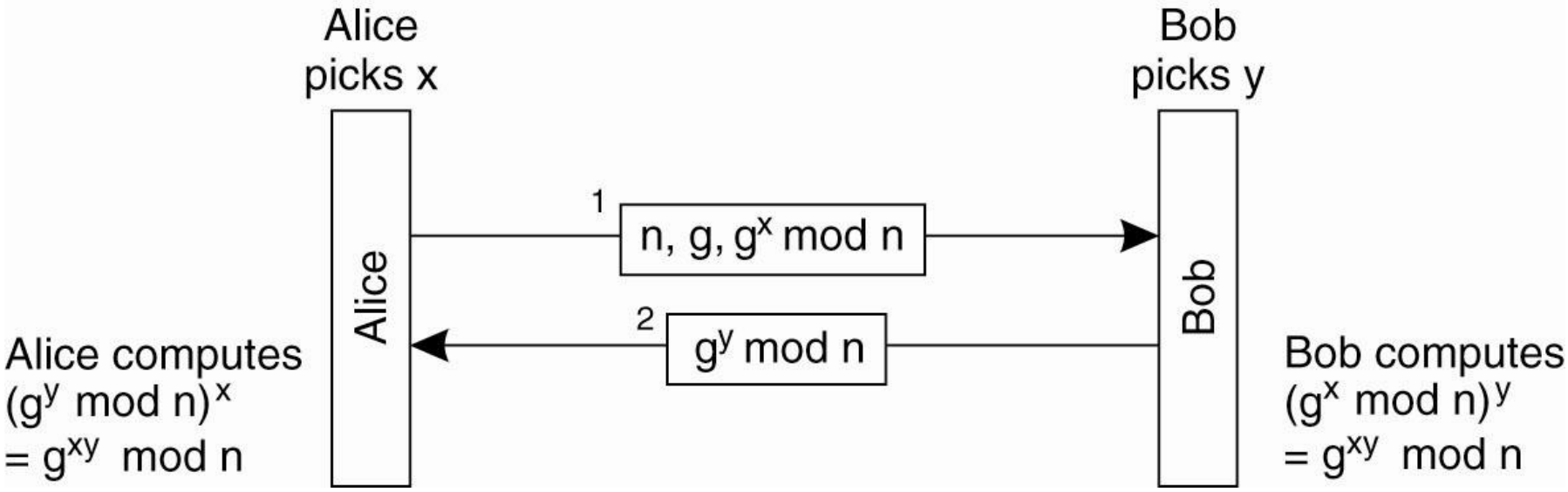


Figure 9-33. The principle of Diffie-Hellman key exchange.

Key Distribution (1)

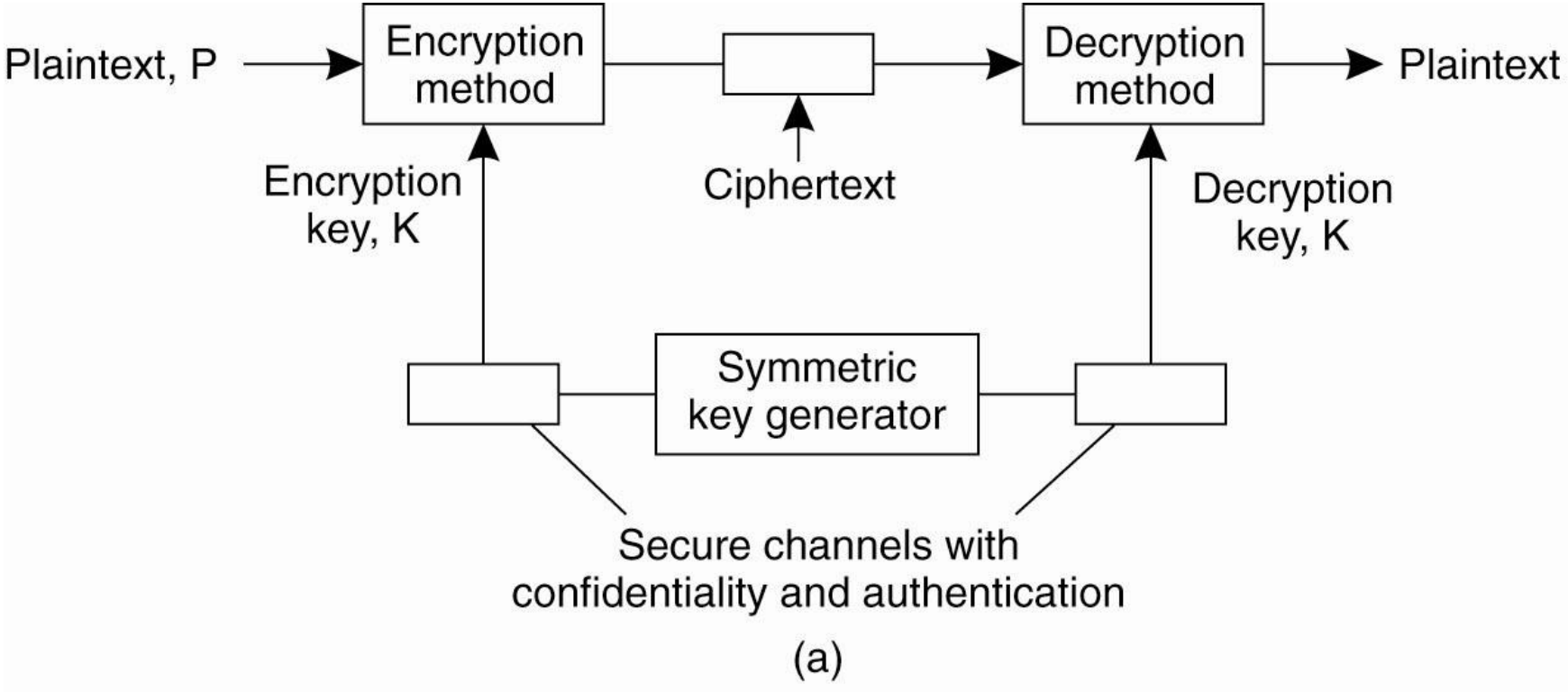


Figure 9-34. (a) Secret-key distribution.

Key Distribution (2)

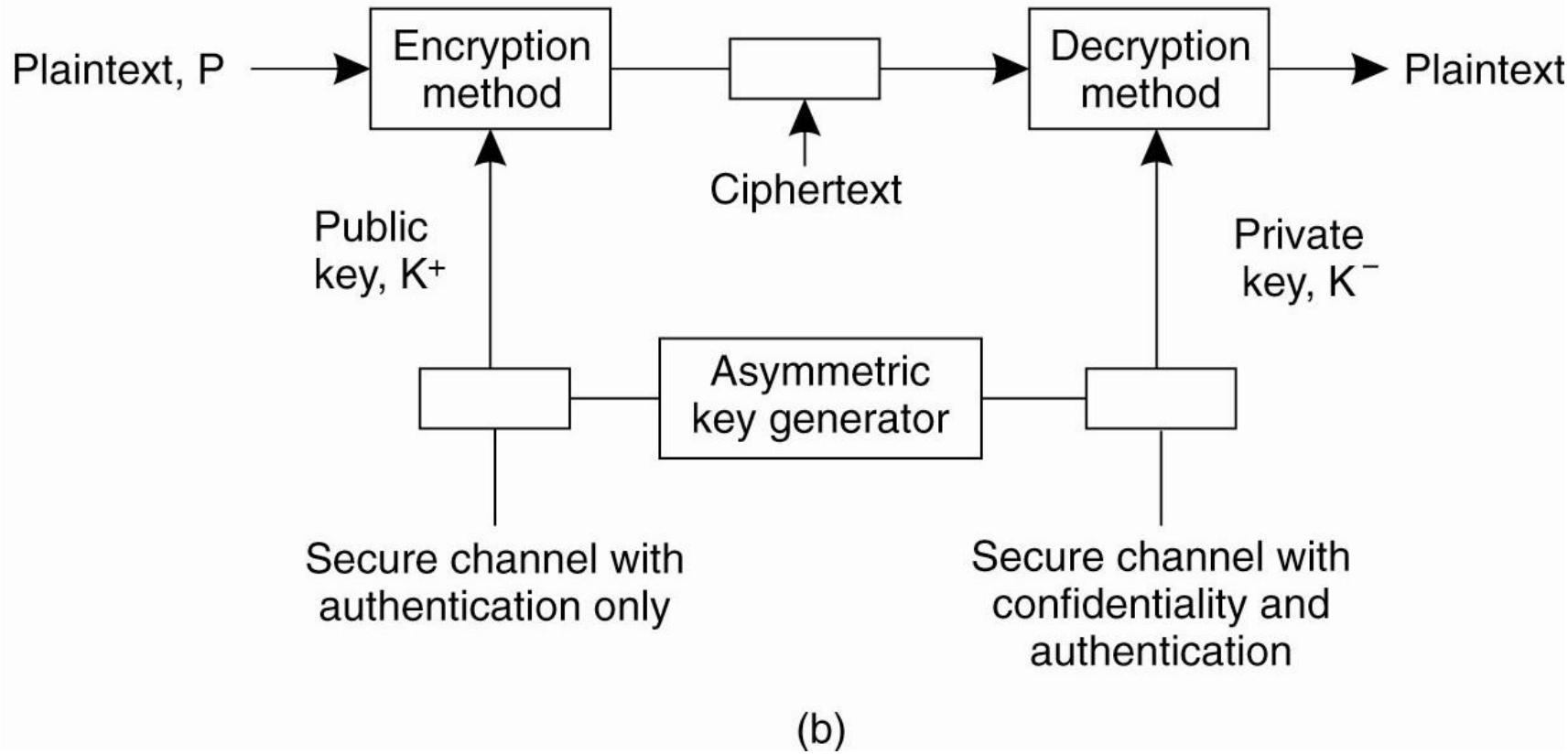


Figure 9-34. (b) Public-key distribution

Secure Group Management

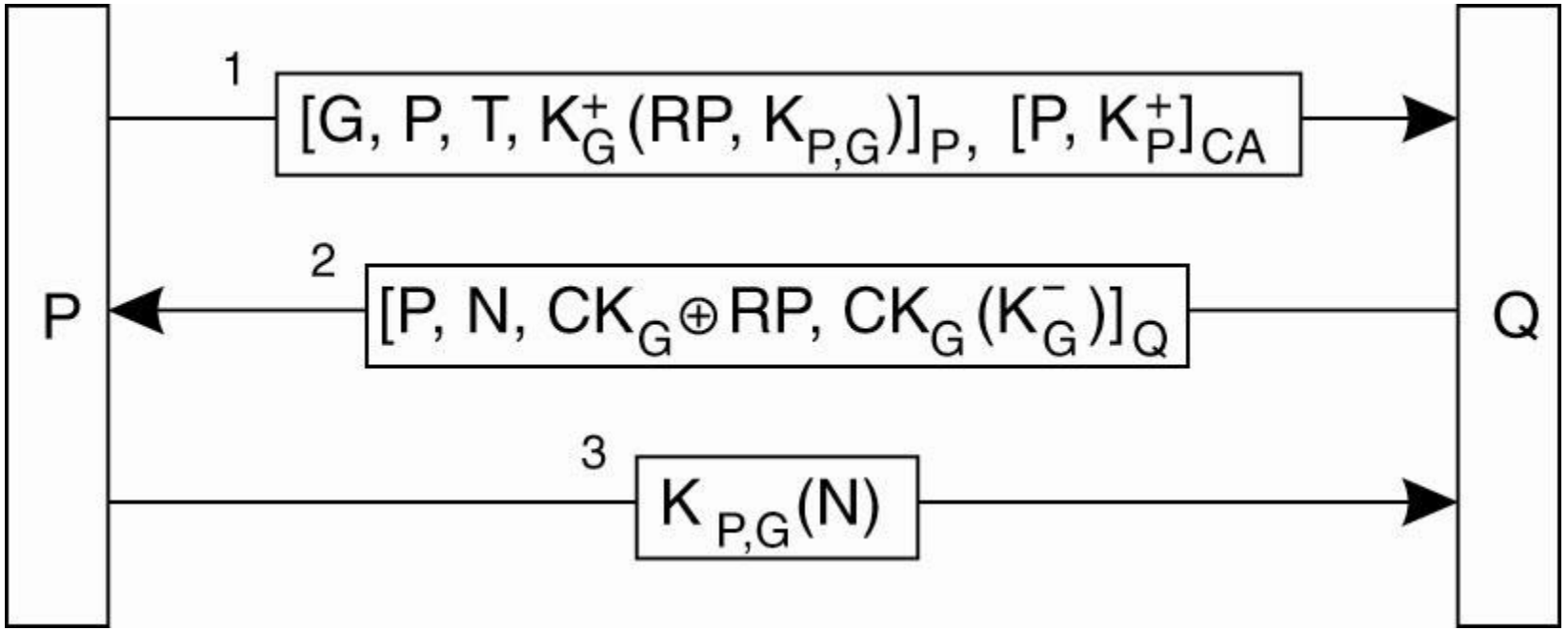


Figure 9-35. Securely admitting a new group member.