Security Threats, Policies, and Mechanisms

Types of security threats to consider:

- Interception
- Interruption
- Modification
- Fabrication

Example: The Globus Security Architecture (1)

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

Example: The Globus Security Architecture (2)

4. Operations between entities in different domains require mutual authentication.
5. Global authentication replaces local authentication.
6. Controlling access to resources is subject to local security only.
7. Users can delegate rights to processes.
8. A group of processes in the same domain can share credentials.
Example: The Globus Security Architecture (2)

Figure 9-1. The Globus security architecture.

Focus of Control (1)

Data is protected against wrong or invalid operations

(a) Invocation

(b) Method

Object

State

Figure 9-2. Three approaches for protection against security threats. (c) Protection against unauthorized users.
Layering of Security Mechanisms (1)

High-level protocols

<table>
<thead>
<tr>
<th>Application</th>
<th>Middleware</th>
<th>OS Services</th>
</tr>
</thead>
</table>

OS kernel

Transport

Network

Data link

Physical

Low-level protocols

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

Layering of Security Mechanisms (2)

Encryption device

SMDS

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

Distribution of Security Mechanisms

Servers running secured services

No direct access from other machines

Access control device

Figure 9-5. The principle of RISSC as applied to secure distributed systems.

Cryptography (1)

Passive intruder only listens to C

Active intruder can alter messages

Active intruder can insert messages

Plaintext, P

Encryption method

Encryption key, E_k

Cipher text, C = E_k(P)

Decryption method

Decryption key, D_k

Figure 9-6. Intruders and eavesdroppers in communication.
Cryptography (2)

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_{a,b}$</td>
<td>Secret key shared by A and B</td>
</tr>
<tr>
<td>$K_A$</td>
<td>Public key of A</td>
</tr>
<tr>
<td>$K_A^-$</td>
<td>Private key of A</td>
</tr>
</tbody>
</table>

Figure 9-7. Notation used in this chapter.

Symmetric Cryptosystems: DES (1)

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

Symmetric Cryptosystems: DES (2)

Figure 9-9. Details of per-round key generation in DES.
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 \( e \times d = 1 \mod z \).

Hash Functions: MD5 (1)

Figure 9-10. The structure of MD5.

Hash Functions: MD5 (2)

Figure 9-11. The 16 iterations during the first round in a phase in MD5.

Authentication Based on a Shared Secret Key (1)

Figure 9-12. Authentication based on a shared secret key.
Authentication Based on a Shared Secret Key (2)

Figure 9-13. 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.

Authentication Using a Key Distribution Center (1)

Figure 9-15. The principle of using a KDC.

Authentication Using a Key Distribution Center (2)

Figure 9-16. Using a ticket and letting Alice set up a connection to Bob.
Figure 9-17. The Needham-Schroeder authentication protocol.

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

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

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. (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.

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)

Stack frame 02
disable_privilege
Call enable_privilege
Stack frame 01
Check access rights
disable_privilege
Stack frame first method call
disable_privilege

Figure 9-32. The principle of stack introspection.

Key Establishment

Alice picks x

Bob computes

\((g^x \mod n)^y = g^{xy} \mod n\)

Bob picks y

Alice computes

\((g^y \mod n)^x = g^{xy} \mod n\)

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

Key Distribution (1)

 Plaintext, P → Encryption method → Decryption method → Plaintext

 Encryption key, K

 Symmetric key generator

 Secure channels with confidentiality and authentication

 Plaintext, P → Encryption method → Decryption method → Plaintext

(aba)

(aba)

Figure 9-34. (b) Public-key distribution [see also Menezes et al. (1996)].
Secure Group Management

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

Capabilities and Attribute Certificates (1)

Figure 9-36. A capability in Amoeba.

Capabilities and Attribute Certificates (2)

Figure 9-37. Generation of a restricted capability from an owner capability.

Delegation (1)

Figure 9-38. The general structure of a proxy as used for delegation.
Delegation (2)

Figure 9.39. Using a proxy to delegate and prove ownership of access rights.