Chapter 9
Security
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

Invocation

Method

State

Object

(a)
Focus of Control (2)

Data is protected against unauthorized invocations.

(b)
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 as applied to secure distributed systems.
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.

Tannenbaum & Van Storm, Distributed Systems: Principles and Paradigms, 2e, © 2007 Prentice-Hall, Inc. All rights reserved. 0-13-238227-5
Slide 15 (Notes)

(a) 

- $x_1$ 
  - Round 1 
  - $\vdots$ 
  - Round 16 
  - Final permutation 

(b) 

- $L_{i-1}$ 
  - $R_{i-1}$ 
  - $L_{i-1} \oplus f(R_{i-1}, K_i)$ 

- $L_i$ 
  - $R_i$ 

Tanebaum & Van Steen, Distributed Systems: Principles and Paradigms, 2e, (c) 2007 Prentice-Hall, Inc. All rights reserved. 0-13-22327-5
Symmetric Cryptosystems: DES (3)

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)

<table>
<thead>
<tr>
<th>Iterations 1–8</th>
<th>Iterations 9–16</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p \leftarrow (p + F(q,r,s) + b_0 + C_1) \ll 7 )</td>
<td>( p \leftarrow (p + F(q,r,s) + b_8 + C_9) \ll 7 )</td>
</tr>
<tr>
<td>( s \leftarrow (s + F(p,q,r) + b_1 + C_2) \ll 12 )</td>
<td>( s \leftarrow (s + F(p,q,r) + b_9 + C_{10}) \ll 12 )</td>
</tr>
<tr>
<td>( r \leftarrow (r + F(s,p,q) + b_2 + C_3) \ll 17 )</td>
<td>( r \leftarrow (r + F(s,p,q) + b_{10} + C_{11}) \ll 17 )</td>
</tr>
<tr>
<td>( q \leftarrow (q + F(r,s,p) + b_3 + C_4) \ll 22 )</td>
<td>( q \leftarrow (q + F(r,s,p) + b_{11} + C_{12}) \ll 22 )</td>
</tr>
<tr>
<td>( p \leftarrow (p + F(q,r,s) + b_4 + C_5) \ll 7 )</td>
<td>( p \leftarrow (p + F(q,r,s) + b_{12} + C_{13}) \ll 7 )</td>
</tr>
<tr>
<td>( s \leftarrow (s + F(p,q,r) + b_5 + C_6) \ll 12 )</td>
<td>( s \leftarrow (s + F(p,q,r) + b_{13} + C_{14}) \ll 12 )</td>
</tr>
<tr>
<td>( r \leftarrow (r + F(s,p,q) + b_6 + C_7) \ll 17 )</td>
<td>( r \leftarrow (r + F(s,p,q) + b_{14} + C_{15}) \ll 17 )</td>
</tr>
<tr>
<td>( q \leftarrow (q + F(r,s,p) + b_7 + C_8) \ll 22 )</td>
<td>( q \leftarrow (q + F(r,s,p) + b_{15} + C_{16}) \ll 22 )</td>
</tr>
</tbody>
</table>

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.

Tanebaum & Van Steen, Distributed Systems Principles and Paradigms, 2e, (c) 2007 Prentice-Hall, Inc. All rights reserved. 0-13-238227-5
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.
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.

Tannenbaum & Van Steen, Distributed Systems: Principles and Paradigms, 2e, © 2007 Prentice-Hall, Inc. All rights reserved. 0-13-238227-5
Authentication Using a Key Distribution Center (3)

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

Tanenbaum & Van Steen, Distributed Systems: Principles and Paradigms, 2e, (c) 2007 Prentice-Hall, Inc. All rights reserved. 0-13-235227-5
Authentication Using a Key Distribution Center (5)

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

27

Tannenbaum & Van Steen, Distributed Systems: Principles and Paradigms, 2e, (c) 2007 Prentice-Hall, Inc. All rights reserved. 0-13-28527-5
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.
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)

Client

Create access request \( r \) as subject \( s \)

\[ (s, r) \]

Server

ACL → Object

if \( s \) appears in ACL
if \( r \) appears in ACL[\( s \)]
grant access;

Client

Create access request \( r \) for object \( o \). Pass capability \( C \)

\[ (o, r) \rightarrow C \]

Server

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

![Diagram showing a hierarchical organization of protection domains as groups of users.](image)

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

(a) A sandbox. (b) A playground.

Figure 9-30. (a) A sandbox. (b) A playground.
Protecting the Target (3)

Local resources accessible through objects

Protected area

Unprotected area

Reference handed out at loading time

Downloaded program

Figure 9-31. The principle of using Java object references as capabilities.
Protecting the Target (4)

Stack frame 02

disable_privilege

Stack frame 01

disable_privilege

\vdots

Stack frame first method call

disable_privilege

Call enable_privilege

Check access rights

Figure 9-32. The principle of stack introspection.
Key Establishment

![Diagram showing key establishment process]

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

Tannenbaum & Van Steen, Distributed Systems: Principles and Paradigms, 2e, (c) 2007 Prentice-Hall, Inc. All rights reserved. 0-13-235227-5
Key Distribution (1)

Plaintext, P → Encryption method → Ciphertext → Decryption method → Plaintext

Encryption key, K

Secure channels with confidentiality and authentication

(a)

Plaintext, P → Encryption method → Ciphertext → Decryption method → Plaintext

Encryption method

Symmetric key generator

Decryption key, K

(a)
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)

<table>
<thead>
<tr>
<th></th>
<th>Server port</th>
<th>Object</th>
<th>Rights</th>
<th>Check</th>
</tr>
</thead>
<tbody>
<tr>
<td>48 bits</td>
<td>24 bits</td>
<td>8 bits</td>
<td>48 bits</td>
<td></td>
</tr>
</tbody>
</table>

Figure 9-36. A capability in Amoeba.
Capabilities and Attribute Certificates (2)

```
<table>
<thead>
<tr>
<th>Capability</th>
<th>Proposed new rights</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Object 1111111</td>
</tr>
</tbody>
</table>
```

```
Exclusive or

<table>
<thead>
<tr>
<th>One-way function</th>
</tr>
</thead>
</table>
```

```
<table>
<thead>
<tr>
<th>Restricted capability</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Object 00000001</td>
</tr>
</tbody>
</table>
```

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

Tannenbaum & Van Steen, Distributed Systems: Principles and Paradigms, 2e, © 2007 Prentice-Hall, Inc. All rights reserved. 0-13-238527-5
Delegation (1)

Certificate

<table>
<thead>
<tr>
<th>R</th>
<th>$S^\text{proxy}$</th>
<th>$\text{sig}(A, {R, S^\text{proxy}})$</th>
<th>$S^\text{proxy}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>access rights</td>
<td>public part of secret</td>
<td>signature</td>
<td>private part of secret</td>
</tr>
</tbody>
</table>

Figure 9-38. The general structure of a proxy as used for delegation.
Figure 9-39. Using a proxy to delegate and prove ownership of access rights.