Authorisation in MAFTIA

Yves Deswarte

deswarte@laas.fr
LAAS-CNRS
Toulouse, France
MAFTIA Workplan

- WP1: Conceptual model and architecture
- WP2: Dependable middleware
- WP3: Intrusion detection
- WP4: Dependable trusted third parties
- WP5: Distributed authorisation
- WP6: Assessment
Who are the intruders?

1: “ Outsider”

2: User

3: Privileged User

because the “least privilege principle” is not implemented
Intrusion Tolerance

Intrusion into a part of the system should give access only to non-significant, non-sensitive information

FRS: Fragmentation-Redundancy-Scattering

- **Fragmentation**: split the data into fragments so that isolated fragments contain no significant information: *confidentiality*

- **Redundancy**: add redundancy so that fragment modification or destruction would not impede legitimate access: *integrity + availability*

- **Scattering**: isolate individual fragments
Different kinds of scattering

- **Space:** use different transmission links and different storage sites
- **Time:** mix fragments (from the same source, from different sources, with jamming)
- **Frequency:** use different carrier frequencies (spread-spectrum)
- **Privilege:** require the co-operation of differently privileged entities to realise an operation (separation of duty, secret sharing)
Prototype (1986-1996)

[Blain & Deswarte 1994]

Smartcard

User Sites

Application Windows

Storage Sites

Networks

File Fragment

Data Processing Sites

Application Fragment

Key Shadows

Security Sites

[Deswarte et al. 1991]

[Fraga & Powell 1985]

[Fray et al. 1986]

[Fabre et al. 1994]
FRSed Security Management

- No single trusted site or administrator
- Global trust in a majority of security sites (and administrators)
Authentication

1. Smartcard Activation
2. Local Authentication
3. Global Decision
4. Session key distribution
Authorisation

Security Server

1. Request to open a session
2. Global Decision
3. tickets
4. direct access

Secured Servers

<table>
<thead>
<tr>
<th>File name</th>
<th>ACL</th>
<th>Key shadow</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Authorisation: reference monitor

- Subjects
  - Subject (active) ≈ process

- Reference Monitor
  - tamperproof
  - always invoked
  - verifiable as correct

- Objects
  - Object (passive) ≈ information containers
Distributed Authorisation? (1)

Reference Monitor

ACM

S O

S O
Distributed Authorisation? (2)

Red Book (TNI)

😊 No bottleneck, no single-point-of-failure
😊 Mutual trust between TCBs, consistency?
Authorisation Scheme for DOOS

[Nicomette & Deswarte 1997]
## Authorisation Scheme

### Access Matrix:

- **Method rights:** corresponding to the authorisation for an object to call another object methods
- **Symbolic rights:** corresponding to the authorisation for an object to execute high level operations

<table>
<thead>
<tr>
<th></th>
<th>ps1</th>
<th>fs2</th>
<th>f3</th>
<th>p4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>u</strong></td>
<td></td>
<td>PF(this, PRINTER)</td>
<td>PF(FILE, this)</td>
<td></td>
</tr>
<tr>
<td>ps1</td>
<td></td>
<td></td>
<td></td>
<td>print</td>
</tr>
<tr>
<td>fs2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Symbolic right rules:** to check authorisation for high level operations
- **Capability creation rules:** to grant capabilities and *vouchers* to enable high level operations
Example: $u:: PF(f3,P4)$

- **Access under AS's control**
  - $R(printf, ps1)$
  - $V(ps1, Fread(f3))$
  - $R(readf, fs2)$
  - $V(fs2, read(f3))$
  - $R(read, f3)$
  - $R(print, p4)$
  - $R(write, tf)$

- **Access under SK's control**
  - $R(delete, tf)$
  - $R(write, tf)$
MAFTIA Authorisation

- Intrusion tolerant authorisation servers
  - Multi-party transactions
    (not simple client-server relations)
  - Local protection
IT Authorisation Servers

Same technique as for FRS Security servers:

- Non-confidential information is replicated
- Confidential information is fragmented (threshold crypto)
- Global consensus (majority voting or Byzantine agreement)
- Distribution of capabilities/vouchers (threshold crypto)
Multi-party transactions

Based on DOOS Authorisation Scheme, but...

- Implementing “separation of duty”
  (transactions initiated jointly by several users)
Local protection

- Internet applications => no modification of user workstations
- No security kernel, but JVM (?)
- Capabilities/vouchers => applets
- Possibly enforced by smartcards (e.g., JavaCards) ... which are already necessary for authentication
Use cases / Scenarios

- **Medical:**
  - French Healthcare Professional Network
    - Patient smartcard Vitale
    - Professional smartcard CPS
    - Electronic prescriptions
    - Network, PKI, anonymisation, delegation, ...

- **Electronic Commerce:**
  - Auction services, privacy enforced payment,...
References


Commercials

- MAFTIA: http://www.research.ec.org/maftia/
- SQUALE: http://www.research.ec.org/squale/
  http://www.dependability.org
  - Workshop on Dependability despite malicious faults
- Recent Advances in Intrusion Detection (RAID 2000): 2-4 October 2000, Toulouse
  http://www.raid-symposium.org
- European Symp. on Research in Computer Security (ESORICS 2000): 4-6 October 2000, Toulouse
  http://www.esorics.org