MAFTIA's Privacy Preserving Authorization Scheme

Yves Deswarte
Yves.Deswarte@laas.fr

LAAS-CNRS
Toulouse, France

MAFTIA: Malicious- and Accidental-Fault Tolerance for Internet Applications

- Systematic investigation of the 'tolerance paradigm' for constructing large-scale dependable distributed applications.
- Comprehensive approach for tolerating both accidental faults and malicious intrusions in such systems, including intrusions by external hackers and by corrupt insiders.

Contract Details

- Project Start Date: 1st Jan 2000
- Duration: 3 years
- Requested Funding: 2.5 M€
- No. of person years: 55

Partners

- Newcastle University (GB) (coordinator)
  Brian Randell, Robert Stroud
- DERA, Malvern (GB)
  Sadie Crees (QinetiQ), Tom McCutcheon (DSTL)
- IBM, Zurich (CH)
  Marc Dacier, Michael Waidner
- LAAS-CNRS, Toulouse (F)
  Yves Deswarte, David Powell
- Universität des Saarlandes (D)
  André Adeslbach, Birgit Pfitzmann (now Michael Steiner)
- Universidade de Lisboa (P)
  Nuno Neves, Paulo Veríssimo

Industrial Advisory Board

- Andrew Izon (North Durham NHS Trust, GB)
- Jean-Claude Lebraud (Rockwell-Collins, F)
- Derek Long (CISA Ltd., GB)
- Joachim Posegga (SAP Systems, D)
- Carlos Quintas (Easyphone, P)
- Gilles Trouessin (Ernst & Young Audit, F)
- Gritta Wolf (Credit Suisse, CH)

Objectives

- Architectural framework and conceptual model (WP1)
- Mechanisms and protocols:
  - dependable middleware (WP2)
  - large scale intrusion detection systems (WP3)
  - dependable trusted third parties (WP4)
  - distributed authorization (WP5)
- Validation and assessment techniques (WP6)

Objectives

- Architectural framework and conceptual model (WP1)
- Mechanisms and protocols:
  - dependable middleware (WP2)
  - large scale intrusion detection systems (WP3)
  - dependable trusted third parties (WP4)
  - distributed authorization (WP5)
- Validation and assessment techniques (WP6)

Authorization

- Contributes to protection:
  - Error detection/confinement
  - Intrusion prevention/confinement
- For Internet applications:
  - More flexible than "client-server" paradigm
  - Contributes to privacy:
    - personal information is disclosed only on a "need-to-know" basis
    - Operations are granted/denied according to proofs of authorization (capabilities, vouchers) rather than to client's identity

State-of-the art: client-server

- Server grants or denies privileges to client, according to client's claimed identity
- Personal data must be recorded:
  -> evidence in case of dispute:
  "log everything"
  … but they may be abused

This paradigm is obsolete

- Internet transactions involve more than 2 parties (e.g., customer, merchant, credit card company, banks, delivery company, …)
- The parties have different, competing interests
  => mutually suspicious

Need-to-know principle

- A merchant does not need to know the real identity of a customer, only the validity of the money order
- The customer's bank does not need to know the identity of the merchant, only the reference of his bank account
- Etc.

… of course

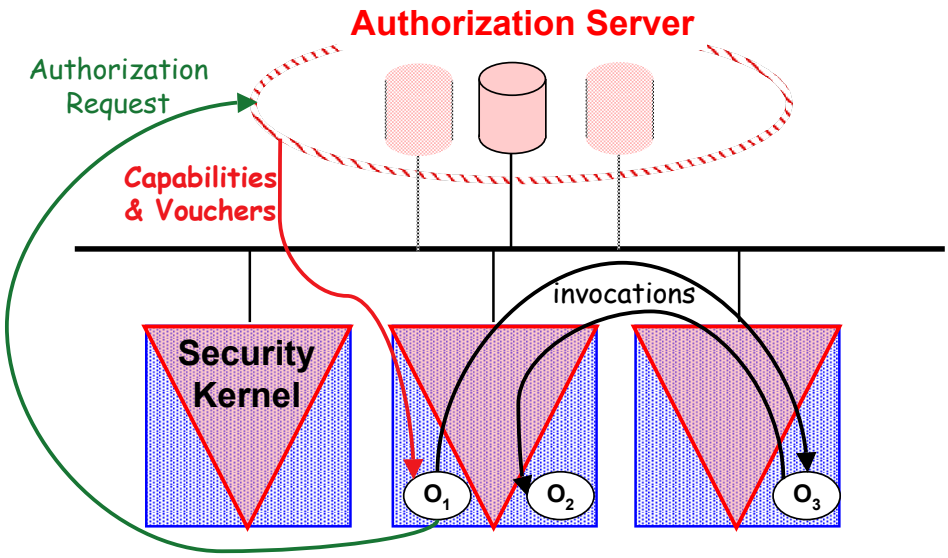
- Real identities would be disclosed to a judge in case of dispute, or on request by judicial authorities (to prevent money laundering, for instance)

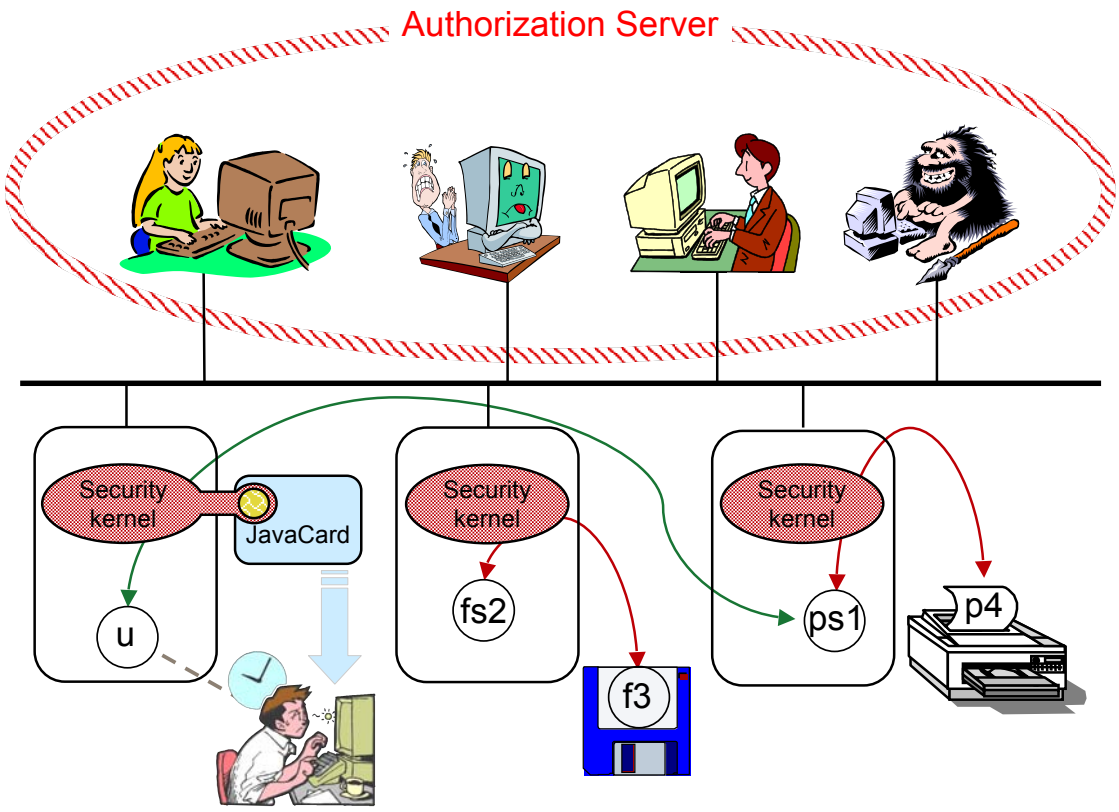
Authorization Scheme

- Distributed, multi-party transactions: O-O
- Elementary operations are granted or denied according to proofs of authorization (~capabilities)
  - Need-to-know principle
- Groups of capabilities are delivered for each high-level operation by an authorization server
  - Capabilities and "vouchers"
  - Least privilege principle

Authorization Scheme for DOOS

[Nicomette & Deswarte, 1997]





http://www.research.ec.org/maftia/