MAFTIA Project Overview

presented by

Robert Stroud,
University of Newcastle upon Tyne
MAFTIA - Malicious and Accidental Fault Tolerance for Internet Applications

- The MAFTIA project will systematically investigate the ‘tolerance paradigm’ for constructing large-scale dependable distributed applications.

- Its major innovation is a comprehensive approach for tolerating both accidental faults and malicious attacks in such systems, including attacks by external hackers and by corrupt insiders.
Contribution to EC Policies

- Large network infrastructures, such as the Internet, are vital for citizens to benefit from the services provided by the Information Society.

- Development depends on how much the users will ‘trust’ the services offered to them.

- Critical to make such services dependable, and in particular resilient to malicious attacks perpetrated by external hackers or by corrupt insiders.
Added Value

- Project consistent with the ‘European Dependability Initiative’.
- Expertise gathered at International level in the fields of fault-tolerance, distributed computing, computer security, intrusion detection and cryptography.
Partners

- QinetiQ, Malvern (UK) - Sadie Creese
- IBM, Zurich (CH) - Andreas Wespi / Michael Waidner
- LAAS-CNRS, Toulouse (F) - Yves Deswarte / David Powell
- Newcastle University (UK) - Robert Stroud / Brian Randell
- Universität des Saarlandes (D) - Andre Adelsbach
- Universidade de Lisboa (P) - Paulo Veríssimo

- Project Coordinator - Newcastle
Industrial Advisory Board

- Representative of a wide range of industrial sectors with an interest in security and intrusion tolerance:
  - Jean-Claude Lebraud, Rockwell Collins
  - Derek Long, Cisa Ltd
  - Gritta Wolf, Credit Suisse Financial Services
  - Joachim Posegga, SAP
  - Andrew Izon, North Durham Healthcare NHS Trust
  - Gilles Trouessin, Ernst Young
  - Carlos Quintas, Altitude Software
  - Tom McCutcheon, DSTL
- Invited to all workshops, kept informed of progress, and asked to comment on future directions
Project Objectives

- The objective of MAFTIA is to investigate the ‘tolerance’ paradigm for security systematically.

- Work is focused in three main areas:
  - the **architecture** of MAFTIA: providing a framework that ensures the dependability of distributed applications in the face of a wide class of faults and attacks,
  - the design of **mechanisms and protocols**: providing the required building blocks to implement large scale dependable applications,
  - the **formal assessment** of our work: rigorously defining the basic concepts developed by MAFTIA and verifying the results of the work on dependable middleware.
Workpackages

- WP1 - Conceptual Model and Architecture
- WP2 - Dependable Middleware
- WP3 - Intrusion Detection
- WP4 - Dependable Trusted Third Parties
- WP5 - Distributed Authorization
- WP6 - Verification and Assessment
WP1 – Conceptual Model and Architecture

- A discussion of the relationship between security policies, goals, rules, and security failures
- An analysis of attacks, vulnerabilities and intrusions in terms of the basic dependability concepts of fault, error and failure
- A classification of ten security methods for dealing with attacks, vulnerabilities and intrusions
- The development of an integrated intrusion detection/tolerance framework for building intrusion tolerant systems
- The identification of various architectural strategies for building intrusion tolerant components based on different models of trust
- A fault tree analysis of MAFTIA’s intrusion tolerance capabilities based on a simplified but realistic use case
WP2 – Dependable Middleware

Guiding principles:
- Hybrid failure assumptions
- Recursive use of fault tolerance and fault prevention
- Components trusted to the extent of their trustworthiness

Within this framework, a number of different approaches to the construction of dependable middleware have been explored:
- Fail uncontrolled
- Fail controlled with local trusted components
- Fail controlled with distributed trusted components

Specification, design and implementation of APIs and protocols for various aspects of the MAFTIA middleware, namely:
- The Trusted Timely Computing Base (TTCB)
- Secure group communication protocols (two versions)
- An intrusion tolerant transactional support service built using these protocols
Architecture Overview
Host architecture

Payload channel (Internet)
Control channel

Hardware
Untrusted Hardware
Trusted Hardware

Local Support
Activity Support Services
Communication Support Services
Multipoint Network

Distributed Software

O.S.
TTCB
Security Kernels

AS - Authorisation Service, IDS - Intrusion Detection Service, TTP - Trusted Third Party Service

Trusted— vs. untrusted— hardware

most of MAFTIA’s hardware is untrusted, but small parts considered trusted in the sense of tamper-proof by construction
Architecture Overview

Host architecture

Security kernels materialising fail-controlled subsystems trusted to execute a few functions correctly, albeit immersed in an environment subjected to malicious faults

AS - Authorisation Service, IDS - Intrusion Detection Service, TTP - Trusted Third Party Service
WP3 – Intrusion Detection

- The development of a taxonomy and framework for analyzing the strengths and weaknesses of existing Intrusion Detection Systems (IDSs)
- The development of a novel algorithm for clustering together ID alerts with similar root causes
- Techniques for reducing the rate of false positives and false negatives
- A testbed for evaluating IDSs
- The specification of an architecture for a large scale intrusion tolerant distributed IDS
- Development of a prototype intrusion tolerant IDS using the MAFTIA middleware
WP4 – Dependable Trusted Third Parties

- A blueprint for building generic trusted third-party services using state-machine replication
- The full specification of a distributed certification authority (CA) and a trusted party for optimistic fair exchange, developed according to this blueprint
- A prototype implementation of the distributed CA using the protocol suite developed in WP2
Secure replication of trusted service

- Domain name server
- Certification authority
- Electronic notary
- Directory server
- ...

Single point of failure (hackers, insiders)

Theoretical limits:
- \( t < \frac{n}{3} \) malicious servers
- arbitrary delays

Folklore:
- No practical solution can reach these limits!

Replicate critical system components:
\( t < \frac{n}{3} \) intrusions or crashes can be tolerated.

Malicious corruption ≠ crash failure!
Might include delaying messages arbitrarily!

MAFTIA
Achieves these limits, efficiently and provably secure
Design and implementation of an intrusion tolerant distributed authorization service

Each MAFTIA host (user workstation or server) has a local reference monitor, partially implemented on a JavaCard

A distributed, intrusion-tolerant authorization server uses replication together with a threshold signature scheme to grant capabilities

Java Cards are assumed sufficiently tamper-proof by design

However, the overall security of the system does not depend on trusting individual hosts, so the effects of any corruption are limited

A prototype implementation has been built, using the protocols provided by the MAFTIA middleware
WP6 – Verification and Assessment

Goals

- Develop a rigorous model of selected malicious- and accidental-fault tolerance concepts
- Formalize some properties and protocols in the language CSP and verify them with a model checker
- Investigate how cryptography can be integrated into such formalizations in a faithful way

Results

- A cryptographic model for formalizing basic concepts of MAFTIA systems using a simulatability definition
- The formalisation and verification of selected components of the MAFTIA middleware, including the TTCB
- A composition theorem, which supports modular proofs that bridge the gap between these two models
Cryptography vs Formal Methods

**Cryptography:**
- ✓ Precise definitions and proofs
- ✗ Each definition long and error-prone
- ✗ Proofs long and error-prone
- ✗ No tool support

**Formal methods:**
- ✓ Well-defined protocol languages (e.g., CSP)
- ✓ Tool-support (e.g., FDR)
- ✗ No cryptographic semantics
- ✗ Need to abstract from reality
Composition theorem

\[ \text{CertMail-Sys.} \geq \text{CertMail Spec.} \]

\[ + \]

\[ \text{Crypto-Sys.} \geq \text{CryptoSpec.} \]

\[ \iff \]

\[ \text{CertMail-Sys.} \geq \text{CertMail Spec.} \]

\[ \text{Crypto-Sys.} \geq \text{CryptoSpec.} \]
Links between work packages

- On concepts and models
  - WP1, WP2, WP3, WP6

- On architecture
  - WP1, WP2, WP3, WP4, WP5

- On middleware
  - WP2, WP4

- On distributed services
  - WP2, WP3, WP4, WP5

- On formal validation and assessment
  - WP2, WP4, WP6
Year 3 Deliverables

- **Reports**
  - D9 - Complete specification of APIs and protocols for the MAFTIA middleware
  - D10 - Design of an intrusion tolerant IDS
  - D21 - Conceptual model and architecture of MAFTIA
  - D22 - Final report on verification and assessment
  - D99 - Architectural analysis of MAFTIA’s intrusion tolerance capabilities

- **Demonstrations**
  - D11 - Running prototype of MAFTIA middleware
  - D12 - Demonstration of dependable TTPs
  - D13 - Demonstration of an intrusion tolerant IDS
  - D14 - Demonstration of distributed authorization
Key deliverables by workpackage

- D21  Conceptual Model and Architecture of MAFTIA (WP1)
- D9  Complete Specification of APIs and Protocols for the MAFTIA Middleware (WP2)
- D3  Taxonomy of IDSs and Attacks (WP3)
- D10 Design of an Intrusion Tolerant IDS (WP3)
- D26 Specification of Dependable Third Party Services (WP4)
- D5  Full Design of Dependable Third Party Services (WP4)
- D27 Specification of Authorisation Services (WP5)
- D6  Design of the Local Authorisation Checker (WP5)
- D4  Formal Model of Basic Concepts (WP6)
- D22 Final report on Verification and Assessment (WP6)
Publications and Dissemination

- Publications at major conferences, including:
  - DSN, Oakland, Crypto, RAID, SRDS, EDCC, CCS, e-Smart, …

- Participation in workshops:
  - EU/US Dependability Initiative
  - Pan-dependability workshop
  - Road mapping and consensus building, planning for FP6
  - IFIP WG 10.4
  - DARPA, Survivability, …

- Links with other projects
  - DSoS, Matisse, …
  - DIRC
  - Cabernet