



MAFTIA - Malicious and Accidental Fault Tolerance for Internet Applications

Paulo Esteves Veríssimo

University of Lisboa

Navigators Research Group, www.navigators.di.fc.ul.pt

**TF-CSIRT Workshop
September 2005, Lisboa**



MAFTIA - Malicious and Accidental Fault Tolerance for Internet Applications

Computer systems can fail for many reasons



MAFTIA investigated ways of making computer systems more dependable in the presence of both accidental and malicious faults



MAFTIA - Malicious and Accidental Fault Tolerance for Internet Applications

- The goal of MAFTIA was to systematically investigate the 'tolerance paradigm' for constructing large-scale dependable distributed applications
- Ideally, such systems should be constructed without vulnerabilities and faults, but this is far beyond current capabilities
- Thus, it is essential to build systems that can tolerate the consequences of residual faults and vulnerabilities
- An intrusion-tolerant system is one that can tolerate attacks, and continue to deliver a trustworthy service
- MAFTIA was the first project to explore the use of fault-tolerance techniques to build intrusion-tolerant Internet-based applications
- The project's major innovation was a comprehensive approach for tolerating both accidental faults and malicious attacks in such systems, including attacks by external hackers and by corrupt insiders.



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)** - Birgit Pftzmann
- **Universidade de Lisboa (P)** - Paulo Verissimo

- Project Coordinator - Newcastle



Project Objectives

- The objective of MAFTIA was to investigate the ‘tolerance’ paradigm for building secure, dependable, networked information systems
- Work was focused in three main areas:
 - the **conceptual model and 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
- The development of the MAFTIA conceptual model involved bringing together for the first time the basic ideas of the different research communities, and played a key role in unifying the project

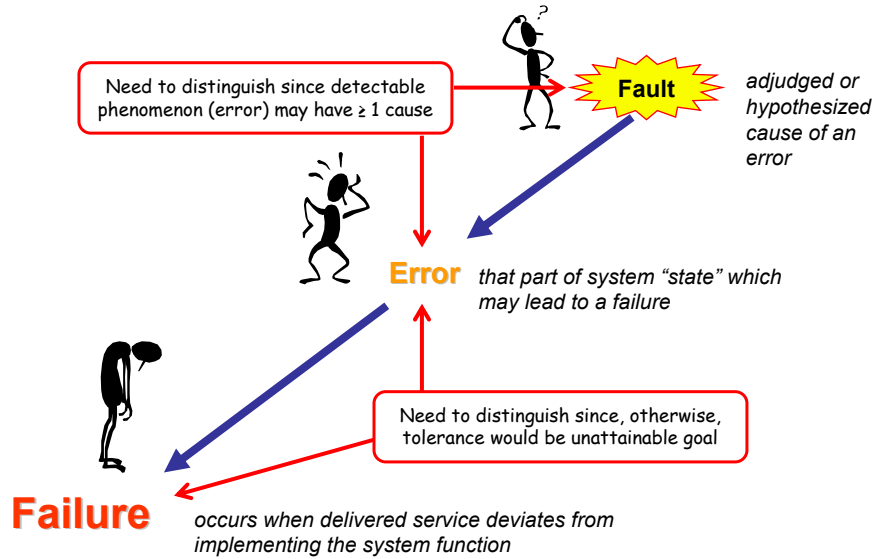


Principles of intrusion tolerance

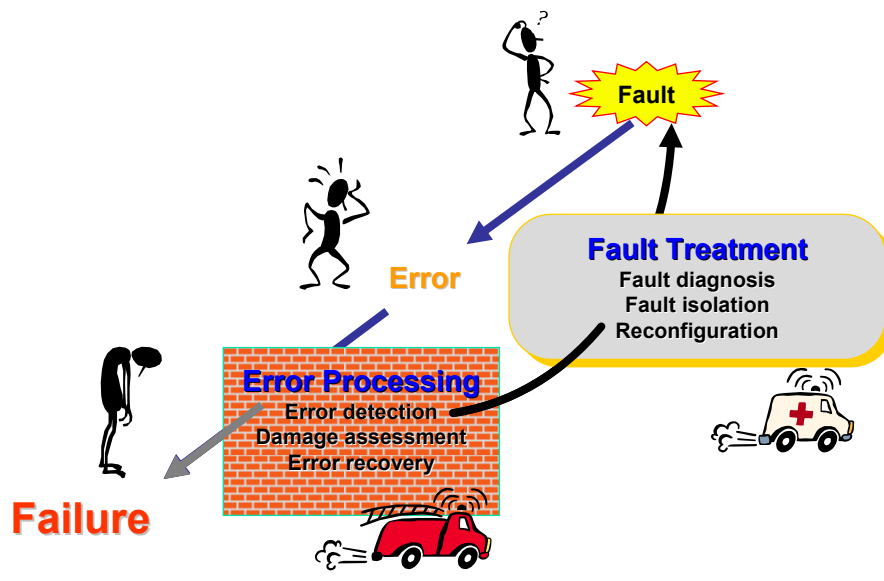
- An intrusion-tolerant system must be able to continue to deliver a secure service, despite the presence of intrusions
- So intrusions are allowed (instead of prevented), but this is not the end of the world



Causal Chain of Impairments

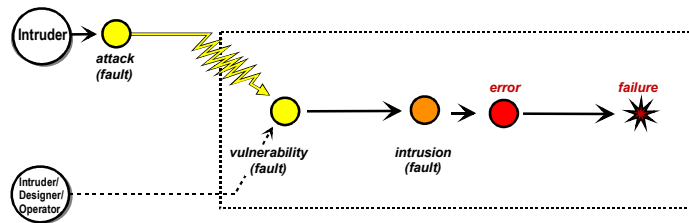


Fault Tolerance





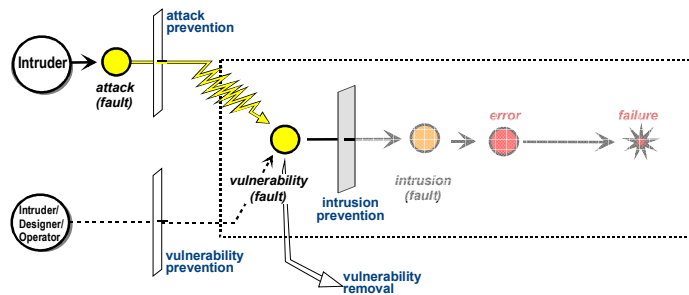
Attack, Vulnerability, Intrusion –AVI Composite fault model



➤ sequence : *attack + vulnerability* → *intrusion* → *failure*



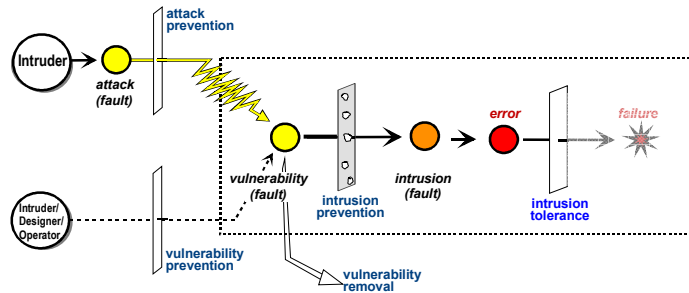
Attack, Vulnerability, Intrusion –AVI Composite fault model



➤ sequence : *attack + vulnerability* → *intrusion* → *failure*



Attack, Vulnerability, Intrusion –AVI Composite fault model



➤ sequence : *attack + vulnerability* → *intrusion* → *failure*

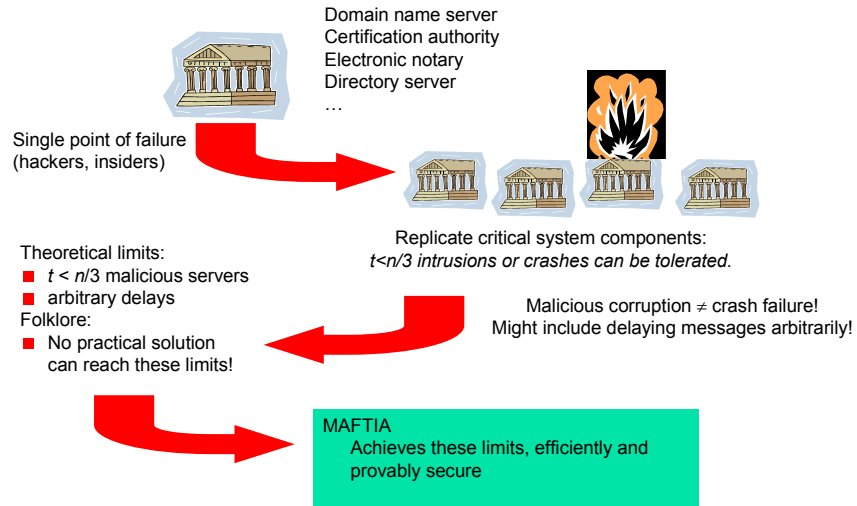


MAFTIA's intrusion-tolerance capabilities

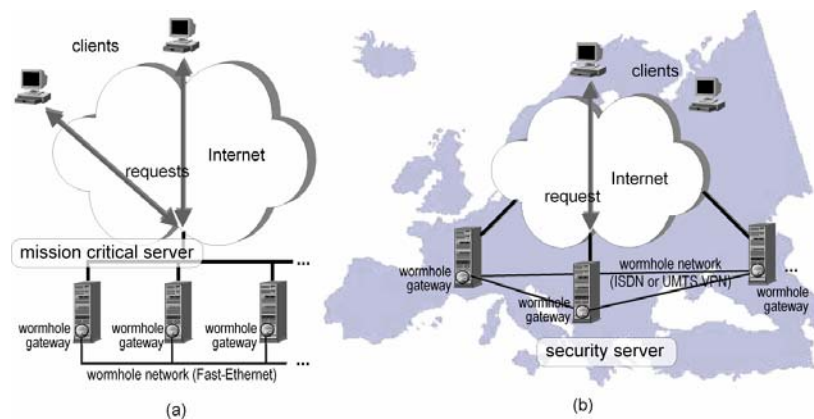
- Using two architectural approaches, MAFTIA developed a variety of intrusion-tolerant capabilities:
 - **Secure group communication**
 - **Transactional support**
 - **Distributed authorisation service**
 - **Intrusion detection system**
- This involved integrating components and services developed by different partners
- In addition, other partners used formal validation techniques to prove that selected MAFTIA components were secure and intrusion tolerant



Secure replication of trusted services



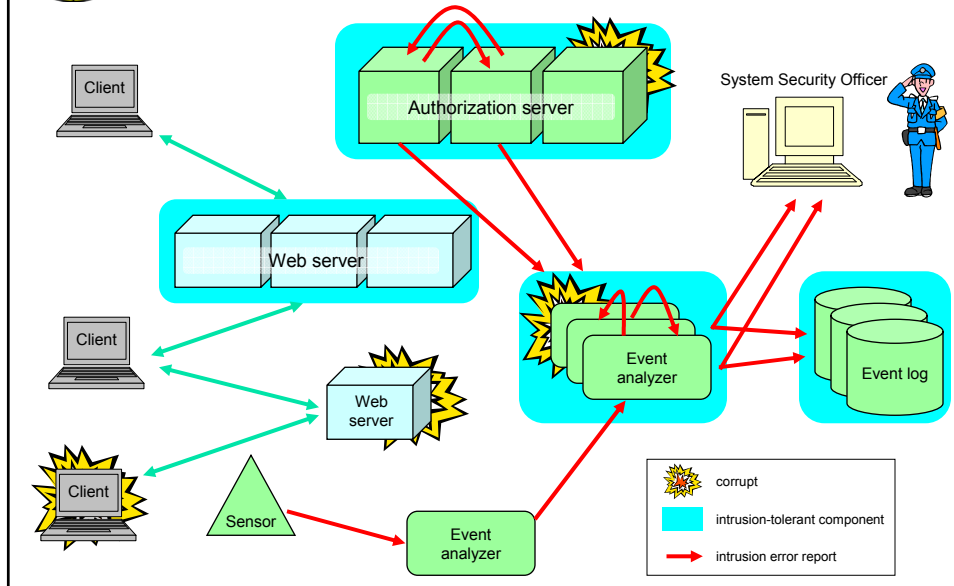
Using wormholes to build secure replicated servers



Wormholes provide a basic trustworthy infrastructure that simplifies the construction of intrusion-tolerant applications in a hostile environment

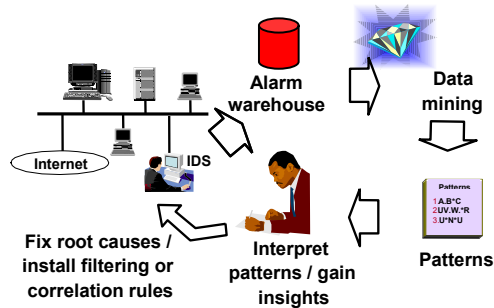


Putting it all together...



Intrusion Detection

- Real world Intrusion Detection Systems generate a high number of false alarms and overwhelm the operator
- Using data mining techniques, applied to real data, MAFTIA was able to reduce the number of false alarms by up to 90%
- This technique is now being used by IBM Managed Security Services



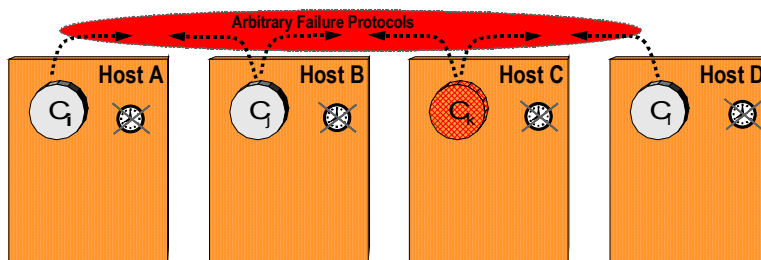


Brief Snapshots of the architecture



Fail-uncontrolled

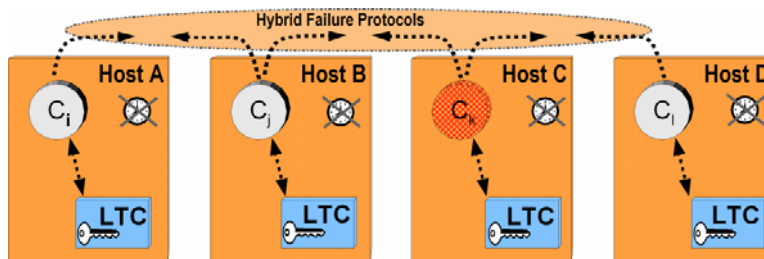
- Time-free
- Arbitrary failure environment
- Arbitrary failure protocols
- Used in: probabilistic Byzantine-agreement based set of protocols





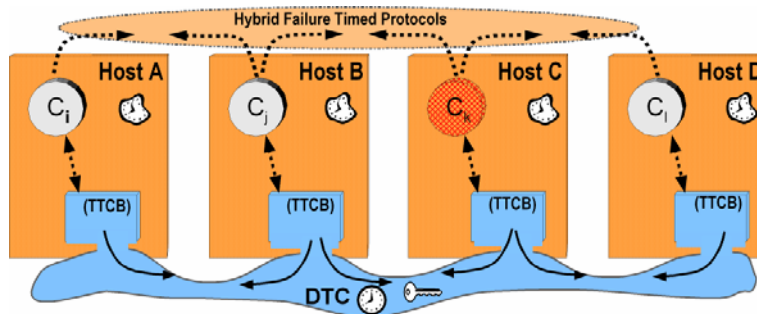
Fail-controlled with Local trusted components

- Time-free
- Arbitrary failure environment + LTC
- Hybrid failure protocols
- Used in: construction of the authorisation service
- Trusted to the extent of: presenting certain hardness to being broken, and of operating correctly until then



Fail-controlled with Distributed trusted components

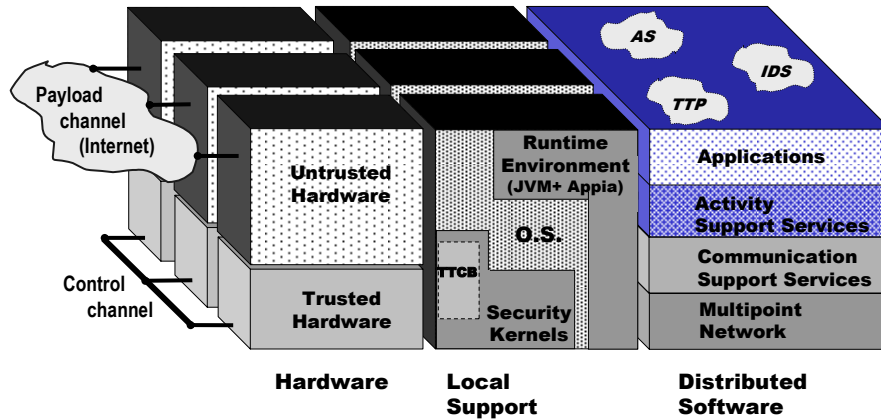
- Time-free or timed with uncertain synchrony
- Arbitrary failure environment + synchronous DTC
- Hybrid failure protocols
- Used in: construction of malicious-F-T comm's protocols
- Trusted to the extent of: not being feasible to subvert it





Architecture Overview

Host architecture



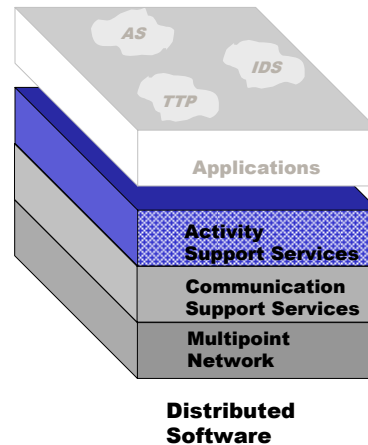
AS - Authorisation Service, IDS - Intrusion Detection Service, TTP - Trusted Third Party Service



Middleware

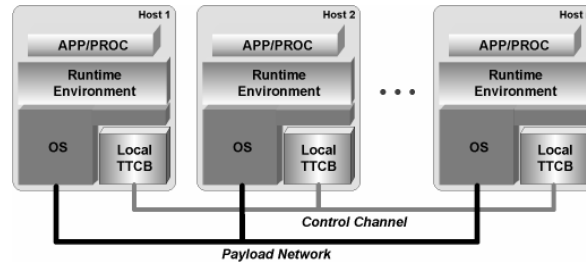
- composition of micro-protocols
- uniform APIs
- different pgm'ing profiles

- *Multipoint Network*
 - adapts physical infrastructure
- *Communication Support Services*
 - implement secure group comm's
- *Activity Support Services*
 - assist participant activity





Trusted Timely Computing Base

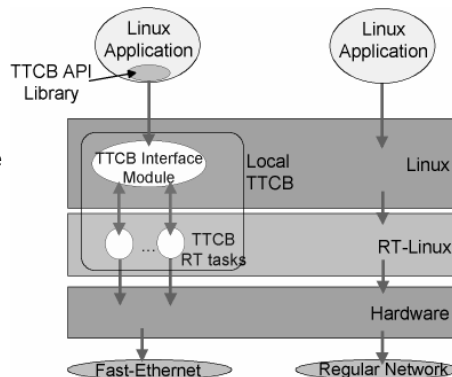


- TTCB is a distributed security kernel that provides a minimal set of trusted and timely services, such as
 - local authentication
 - agreement on a fixed sized block of data
 - globally meaningful timestamps



TTCB Implementation

- TTCB can be a
 - *special hardware module* (e.g. tamperproof device)
 - *secure real-time microkernel running on a workstation or PC underneath the OS*
- TTCB control channel has to be both timely and secure
 - *separate physical network*
 - *virtual network* with predictable characteristics coexisting with the payload channel





AS: Intrusion Tolerant Transactional Support

- Provides standard ACID properties
- Uses fault masking to tolerate intrusions
- Main components
 - clients
 - transactional manager
 - resource manager
 - resources
- Offers a CORBA-style transaction service interface



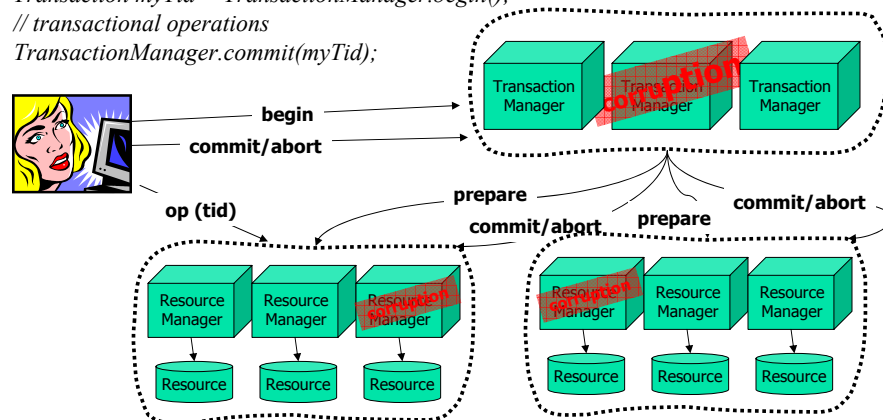
Intrusion Tolerant Transactional Support

Example:

```
Transaction myTid = TransactionManager.begin();
```

```
// transactional operations
```

```
TransactionManager.commit(myTid);
```





Key achievements

- MAFTIA pioneered the subject of intrusion tolerance, now being researched worldwide
- It brought together, for the first time, researchers from security and dependability to tackle this subject
- It created a new conceptual model, clarifying the relationships between the different fields
- It designed, implemented, and demonstrated the first coherent system architecture for intrusion tolerance
- It invented a number of ground-breaking software components, and used formal methods to validate their correctness
- It thus laid the foundations for an effective defence against the ever-growing threats against the global information infrastructure
- MAFTIA technology has already been incorporated into product and service offerings from IBM



The End

- www.navigators.di.fc.ul.pt
- www.maftia.org