A Trusted Open Platform

-Microsoft Corp., 2003

Craig Bergstrom
Overview

♦ Motivation
♦ Goals
♦ Authenticated Operations
♦ Design
♦ Conclusions
Motivation

♦ More and more personal data is electronically stored every day.

♦ A closed system is not likely to replace personal computers anytime soon.

♦ Open operating systems are extremely complex and their integrity cannot be ensured.
Goals

♦ Microsoft wants to create a system that is open & extensible, but yet provides a high degree of assurance.

High Degree of Assurance: Owner has a high degree of confidence in correct behavior.

Extensibility:
- Ability to add arbitrary hardware peripherals
- Ability to execute arbitrary software
- No central authority
Authenticated Operation

♦ Access controls are based (in part) on the identity of the program making a request.

♦ An executable program is fed through a hash function to create a Code-ID.

♦ The Code-ID acts as the identity of the program.

♦ Some programs’ input is as important as program code to the meaning of the program and so we can incorporate that into the hash function as well.
Authenticated Operation

Sealed Storage

Trusted Operating System

Sealing Program Code-ID\textsubscript{1}  
secret

Sealing Program Code-ID\textsubscript{2}
Authenticated Operation

Sealed Storage

![Diagram]

- Sealing Program Code-ID$_1$
- Sealing Program Code-ID$_2$
- Code-ID$_1$
- Code-ID$_2$
- secret

Trusted Operating System
Authenticated Operation

Sealed Storage

Trusted Operating System

Sealing Program Code-ID₁

Sealing Program Code-ID₂

secret
Authenticated Operation

**Attestation**

- Attestation lets programs authenticate their Code ID to remote parties.
- A platform has a public/private key pair certified by an **Identity Service Provider**.
Authenticated Operation

Attestation

♦ A recipient of data is sent a certificate containing the data, the code id of the sender, and is signed by the sender platform’s private key.

♦ This allows a distributed component to identify not only the program that is sending/receiving data, but the system on which that program is executing.
Secure I/O can be provided by the trusted part of the kernel so that input and output is only provided to authorized applications. This can prevent unauthorized code from eavesdropping on users entering password or providing false authorization to running programs.
Authenticated Operation of Kernel

- Authenticated operation of the kernel is provided by a security coprocessor.
- Multiple kernels running on the same hardware are able to keep and share secrets as well as authenticate themselves using a virtual machine monitor.
- The ability of kernels to keep secrets allows them to provide the same functionality to applications.
Proposed Design

Diagram showing the proposed design with four quadrants:
- Normal mode
- Trusted mode
- User mode
- Kernel mode

- Application
- Main operating system
- Drivers
- Machine monitor
- Agent
- Nexus
Applications

- **Soft Smart Card**, and **Network Logon** applications can benefit from **secure I/O** protection of passwords from malicious code.
- **Transaction Authorization** can use secure output to ensure that users see the desired output and secure input to ensure that a user is authorizing transactions.
- **Rights-management** can use attestation to authenticate the platform and software to which it is revealing data. This can provide a high degree of assurance that digital rights are not violated.
- **Document Signing** can be improved by using sealed storage to secure keys used for signing.
NGSCB offers….

♦ protection of secrets stored on computers regardless of if those computers are compromised by viruses or trojan horses.
♦ a way for content distributors to ensure that their content is received only by applications under their control.
♦ these benefits while still allowing the operating system to be extensible by adding arbitrary device drivers and software.