Applying MILS to multicore avionics systems

Eur Ing Paul Parkinson FIET
Principal Systems Architect, A&D

EuroMILS Workshop, Prague, 19th January 2016
Agenda

- A Brief History of MILS
- Implementation Considerations
- MILS & Multicore Convergence
- MILS Multicore Implementation on P4080
- MILS Multicore Use Cases
- Conclusions
Unintended information disclosure?
A Brief History of MILS

- **1981**: John Rushby paper
- **2002**: W. Mark Vanfleet Open Group
- **~2007**: Hardware virtualisation
- **2007**: SKPP v1.03
- **2009**: VxWorks MILS 2.0 (Single-core)
- **2007**: VxWorks MILS EAL6+ Package
- **2013**: VxWorks MILS 3.0 Multicore Edition
## Separation Kernel Architecture: Four Fundamental Security Policies

<table>
<thead>
<tr>
<th>Policy</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Information Flow</strong></td>
<td>Defines permitted information flows between partitions</td>
</tr>
<tr>
<td><strong>Data Isolation</strong></td>
<td>Ensures that a partition cannot access resources in other partitions</td>
</tr>
<tr>
<td><strong>Periods Processing</strong></td>
<td>Ensures that applications within partitions execute for the specified duration in the system schedule</td>
</tr>
<tr>
<td><strong>Fault Isolation</strong></td>
<td>Defines that a failure in one partition does not impact any other partition within the system</td>
</tr>
</tbody>
</table>
Implementation Considerations
Covert Channels of Communication

- Covert channels provide unintended channels of communication between isolated applications, enabling implicit communication of binary values.

- Types of covert channel:
  - **Covert Timing Channel** can occur when there is variation in duration of partition’s execution (partition jitter).
  - **Covert Storage Channel** includes “all vehicles that would allow the direct or indirect writing of a storage location by one process and the direct or indirect reading of it by another” [TCSEC].
Implementation Considerations

Virtualization Approaches

- Processor hardware virtualization support enables:
  - Hypervisor-based SK to run Guest OS in virtualized environment
  - Better enforcement of data isolation & data flows
  - Reduced potential of covert channels

- Two virtualization approaches often used:
  - Paravirtualization where guest OS is modified for common resource execution
  - Full virtualization where guest OS runs unmodified due to hardware virtualization support

Hypervisor must be provably secure to withstand threats of high attack potential
MILS Hypervisor Virtualisation Model

Processor Privilege Levels

- User Non-Root
- Supervisor Non-Root
- Root

Partition 1
- User Application 1
- Guest OS 1

Partition n
- User Application n
- Guest OS n

Separation Kernel (Secure Type 1 Hypervisor Base)

Trusted Hardware
MILS Cross-Domain System use case
Notional MILS-based Cross-Domain System network gateway

GOS: Guest Operating System  
HAE: High Assurance Environment
MILS & Multicore Convergence

- Need to exploit multicore performance in MILS systems
  - Enables consolidation of ‘air-gap’ security systems

- Primary multicore architectural challenges:
  - Application isolation
  - Core separation
  - Resource sharing

- FAA multicore safety research
  - MPC8572 multicore architecture
  - Found shared resource contention - determinism, denial of service
  - CAST-32 paper provides additional objectives for multicore safety
  - ARINC 653 Specification updated to support development of multicore IMA systems

- High-assurance security systems overriding concern:
  - Covert channels of communication with increased bandwidth
MILS Multicore Implementation on P4080

- **Multicore Software Architecture**
  - Many permutations possible on 8 core processor
  - AMP may provide better security characteristics than SMP
  - CoreNet 36bit (64GByte) address space per core, 32bit execution space per core

- **Core Virtualisation**
  - 3 privilege levels on P4080, enables use of Guest OS with memory protection
  - Full hardware virtualisation support reduces guest OS implementation effort

- **Secure Boot**
  - Initialisation of each core to known state and defined boot order
  - Automated tool support to provide continuous validation of system configuration to ensure security issues not introduced through boot of multiple cores in wrong order
MILS Multicore Implementation on P4080 (2)

- **Core Schedule Synchronisation**
  - MILS unicore systems handle different security domains *sequentially*
  - MILS multicore has potential to handle different security domains *concurrently*
  - Potential for increased bandwidth of covert channels
    - Mitigation via P4080 core separation (memory ranges, L2 cache, 2 memory controllers)
    - Use of Synchronised Time-Sliced Scheduling model (example to follow)

- **Inter-Core Communications**
  - Needs to be transparent to application
  - Implementation using Secure IPC with ARINC 653 APEX port interface

- **Independent Core Configuration**
  - Need to support incremental composition of a system
  - Use of XML-based configuration for individual cores and platform with automated tool support and continuous validation
Notional MILS-based Cross-Domain System
Multicore Implementation Considerations

GOS: Guest Operating System
HAE: High Assurance Environment
Notional MILS-based Cross-Domain System

Multicore Implementation Considerations

- Minimise potential for covert channels
- Allocation of applications to different cores
- Time-sliced synchronised scheduling
  - Limit number of security levels or domains processed at same time
MILS Integrated Modular Avionics use cases
Multi-Level Secure IMA

- IMA systems deployed using ARINC 653 architecture
- Growing requirements to support multi-level secure (MLS) in IMA environment
- Can be addressed using MILS:
  - Enforce information flows
  - Use of guards
MILS Integrated Modular Avionics use cases
Multi-Level Secure IMA
MILS Integrated Modular Avionics use cases
IMA Domain Consolidation

- **Aircraft Control Domain**
  - Flight Control Applications
  - Cabin Core Functions

- **Airline Information Services Domain**
  - Admin Functions
  - Flight Support Functions

- **Passenger Info & Entertainment Services Domain**
  - Embedded IFE
  - Passenger Web Portal
  - Passenger Device Interface

- **Passenger-Owned Devices Domain**
  - Passenger devices

Aircraft Data Networks
Conclusions

- MILS has enabled development of high-assurance multi-level secure systems
  - At affordable cost compared to monolithic approaches

- Multicore is a disruptive technology
  - Impacts software programming models
  - Provides additional security challenges

- Multicore provides potential for MILS systems with increased performance throughput, including:
  - Cross-Domain Systems
  - Multi-Level Secure IMA
  - IMA Domain Consolidation