NOC-centric Security of Reconfigurable SoCs

Jean-Philippe Diguet

LESTER lab.
CNRS / Université Européenne de Bretagne
Lorient, France
Outline

- Attacks on embedded systems
  - Classification
  - RSoC perspective
  - NoC perspective
  - Model of threats
  - Scenario Example

- Our approach
  - Strategy
  - Centralized decision & distributed execution
  - 4-steps access control strategy
  - Secure Network Interface and separate channels
  - Secure protocol for (re)configuration

- Implementation case studies

- Conclusion
Classification

- Embedded system security
  - Sensitive data
  - Personal devices

A cryptosystem should be secure even if everything about the system, except the key, is public knowledge.

A. Kerckhoff, J. Sc. Militaires, 01/1883
Classifications

- **Hardware vs Software attacks**
  - **Hardware attacks**
    - Chip cutting
    - Chemical attack of the chip
    - Glitch attack (power, clock)
    - Variation of Vdd or T°
    - Fault Injection
  - **Proximity-based**
    - Timing analysis
    - Power consumption Analysis
    - Electromagnetic emission analysis
    - Bus Eavesdropping
  - **Physical**
    - Irreversible (invasive)
  - **Physical**
    - Reversible (non-invasive)
  - **Active attacks**
  - **Passive attacks**
  - **Side-channel**
    - (non-invasive)

- **Software attacks**
  - Remote attacks
    - Trojan horse
    - Logic bomb
    - Virus
    - Worm
Classification

- **Security Objectives**: Protect Data / Programs / Design / System Against:
  - Extraction of secret information
  - Modification its the behavior
    - Hijacking
  - Denial of service
    - Overloading computing / communications resources

- **Solutions**
  1. **Ciphering**: Readable Data
  2. **Integrity checking**: before using Data or running Programs
  3. **Access Control**: to Data / Program / Configuration (bitstream)
    - Entity authentication
  4. **Monitoring and Countermeasures**
    - Detection of abnormal behaviors
Reconfiguration becomes an industry concern

- Time to market
  - Start design before standard full specification
  - Hardware required for Performances
- Hardware debug
- HW/SW firmware updates (Multimedia, Telecom Stds)
- New Opportunity for Attack (fake HW, hijacking) but also for countermeasures

What’s specific?

- Hardware is no more fully trustable
  - Confidentiality + Authentication solutions
- Configuration memories: new sensitive data
- Secured / unsecured area map can change
- Access control scheme can evolve
NoC perspective

- Means multiplication of IPs and complex communications
  - Non Centralized Management
  - NI = existing smart interfaces = opportunity for distributed access control
  - Traditional network security (IDS) not applicable in embedded SoC
- New threat: Denial of communication service
  - life-lock, dead-lock, incorrect paths
- Paths and Emitter @ = new features for identification
  - Separation between global and local access control
- New opportunities for HW-based monitoring security
  1. SW control access, OS supervision: important overhead in embedded SoC
  2. HW to alleviate security cost
     - Secure core
     - Encryption coprocessor
     - HW integrity / authentication: \textit{a posteriori} solution (board perspective)
  3. HW control access: few work, bus based solution (SECA, [Coburn05])
  4. NoC can provide HW efficient and scalable solutions
Model of threats

- Trusted and untrusted IP / Sensitive & non-sensitive memories
- NoC is a secured area but the payload may vehicle attacks
  - Secured packeting
  - Secured routing
- Security based on access control and monitoring
- Software attacks:
  - Hijacking: *Write control*
  - Data extraction: *Read Control*
  - Denial of service: *NoC use monitoring*
Scenario

- **Example of Attack Strategy with a unsecured NoC:**
  1. Execution of Fake Application: Trojan T installed, modifies NI path tables. GPP2 has now access to Mem.2 and GPP1 to Mem.3 containing secure data.
  2. GPP2 runs an infected application, a Worm W that can copy itself in Mem.2.
  3. GPP1 download malicious multimedia data (D.jpg) in M1, a buffer overflow launches W that copies secure data from Mem.3 to share Mem.4.
  4. W finally implements a logic Bomb for hiding the first attack, while producing later a denial of NoC services with infinite access loop to system memories.

![Diagram showing the attack scenario](image-url)
Outline

- Attacks on embedded systems
  - Classification
  - RSoC perspective
  - NoC perspective
  - Model of threats
  - Scenario Example

- Our approach
  - Strategy
    - Centralized decision & distributed execution
    - 4-steps access control strategy
    - Secure Network Interface and separate channels
    - Secure protocol for (re)configuration

- Implementation case studies

- Conclusion
Security objectives:
1. Detection of abnormal communication behaviors.
   - Control Global and Local R/W Accesses (Hijacking, Extraction)
   - Supervise Traffic, detection of overload (Denial Of Service)
2. Implement counter-attacks (close infected ports, reboot, …)

Methodology for security implementation based on 4 ideas:
- Separation high priority channel security-related traffic and low priority channel for application traffic
- Hierarchical access control strategy
- Secure Network Interface
- Secure protocol for (re)configuration
Centralized decision & distributed execution

- I-“Centralized Security Decision with Distributed Security Policy Execution through Secured NI”
  - One single IP for Security Management (SCM)
  - First mission stored in system boot memory.
  - Configures NI, i.e. control access rules
  - Pends on attack detection from NI
4-Steps access control strategy

II-“Hierarchy of simple tests”

1. I/O Rules loaded by the SCM
   - Global Inter-IP access checking:
     - R/W communication rules based on packet header (Path)
   - Local Access R/W checking
     - R/W communication based on local @ from the Payload
   - Overflow checking:
     - Comparison between announced (in Payload) and Real message sizes (Credit based)
   - Traffic Monitoring
     - Credits counting, comparison with bounds
     - Rules violation => Alerts transmitted to the SCM
Secured Network Interface and Separate Channels

- III-“Enhanced Secured NI applying security rules connected to SCM through a secure Virtual Channel”

- 2 Virtual Channels
  - No physical links between IP and Security Management
  - NI / SCM communications: Priority Best Effort
    - Configuration / Alerts
  - IP / IP: Best Effort

- NI overhead:
  - VC FIFOS
  - Counters
  - Security Table Memory

Diagram:
- SNI, IP configuration
- FIFO PBEOutChannel
- FIFO BEOutChannel
- FIFO ctrl
- Messages
- Statistics
- AGU
- NIconroller
- Security Controller
- FIFO PBEInChannel
- FIFO BEInChannel
- FIFO ctrl
- Packet
- Routing
- Local Credits
- Monitoring Tables
- Offset
- Message Size
- Access Config
- Received Credits
- Depacket
- Port
- Slave / Master Wrapper
- R/W Ctrl Signals
- Data
Secure protocol for (re)configuration

- IV-“Avoid security weakness due to SCM access to both VCs”
- 4 states FSM
- Specific configuration for SCM
  - Exclusive access to Secured VC (Priority BE) and Un-Secured VC (BE)
  - Access to Secure VC for NI configuration and monitoring in RUN Status
  - Access to Unsecure VC (BE) during (re)configuration
  - Switch to SNI or DPR
Outline

- Attacks on embedded systems
  - Classification
  - RSoC perspective
  - NoC perspective
  - Model of threats
  - Scenario Example

- Our approach
  - Strategy
  - Centralized decision & distributed execution
  - 4-steps access control strategy
  - Secure Network Interface and separate channels
  - Secure protocol for (re)configuration

- Implementation case studies
  - Synthetic Set-Top Box
  - SECA case study (DRM)
  - NoC generation

- Conclusion
Synthetic Set-Top Box

- Sensitive Data: Crypto Proc., Pgm, Private data, Network Accesses
- 1st step, boot/reset: SCM/ IP-SNI communications instanciated over PBE VC. SCM starts transferring security rules in SNI tables
Synthetic Set-Top Box

- 2nd step, as a result of 1st configuration process BE communications are instantiated between SCM and IP SNI for security rules configuration.
- Security rules may be reduced to sensitive access.

Diagram:

- Crypto Processor
- Crypto Proc. PGM Memory
- Private Data Memory
- Public Data Memory
- Network Processor
- Mac
- Ciphered Extra Bitstream memory
- Data Memory 1 (Clear)
- Data Memory 2 (Ciphered)
- GPP
- Video Processor
- GPPs
- DMA
- GPPs Program Memory
- SCM
- Graphics Engine
- Security/Ciphered Boot ROM
- External Reset ROM
- Global ciphered Bitstream

Legend:
- BE: Sensitive BE
- PBE not indicated
At run time, a reconfiguration for Firmware Update => new communication scheme
DRM case-study from SECA

- DRM architecture for Portable playback of MM content
- Different memory access rights for CPU A (ARM) and CPU B (crypto)

Access rules for proc. X:
- X-N: Not accessible,
- X-R: Read Only,
- X-RW: Read Write,
- XW: Write only

NoC-Centric security of RSoCs
µSpider NOC CAD Tool: flexible framework for NoC generation
- Wormhole Packet switching
- Topology, minimum routing instruction size
- Routing / Arbiter Policies
- Number of Virtual Channels
- TDM / BE / BE with priority

New Network Interfaces
- Security Tables
- Counters
- Rule Checkers
- No Time Overhead
- Limited Area Increase compared to routers
NoC Generation

- Test Topology for synthetic Set Top Box example:
  - 2D MESH : 4 X 3
  - 22 SNI
  - Bitwidth: 32; Buffer depth: 8 words for BE, 4 words for PBE
  - 7 Master IP, 13 Slave Memories,
  - Same architecture applicable to SECA example

- μSpider
  - VHDL generation
  - Specific Output for EDK Xilinx NOC IP generation

- Preliminary Results:
  - Without Security : 23818 slices
  - With Security : 34568
  - Overhead: 45%
  - Mainly due to Routers for implementing PBE VC
Conclusion

- A complete architecture and methodology is proposed for NoC-centric security applicable to RSoC.
- Security has a cost ... (2.2 % of main US company turnover in 07)
  - How much for personal security in the future?
  - Separate channels are necessary
- The Secured NoC overhead can be reduced:
  - By using a reduced number of routers with 2 Virtual Channels
  - Low bandwidth requirements
  - Bitwidth may be reduced
  - By improving synthesis, FPGA => overestimation
- Anyway, a systematic methodology is required to address complex access control schemes in future multi-processor RSoC
- Future work: Implementation of countermeasures strategy
  - What’s the reaction against attacks alerts?
Conclusion

Thank You