Self-reconfigurable security-enhanced communications in FPGA-based MPSoCs

Pascal Cotret, Laboratoire Lab-STICC
Université De Bretagne-Sud
<pascal.cotret@univ-ubs.fr>

Château de Goutelas, Marcoux
June 21st, 2012
This FPGA-based SoC can be attacked!

- Trusted/Untrusted areas.
- Threat model: external bus and memory.
This FPGA-based SoC can be attacked!

- Trusted/Untrusted areas.
- Threat model: external bus and memory.
This FPGA-based SoC can be attacked!

- Trusted/Untrusted areas.
- Threat model: external bus and memory.
- Two kinds of transactions: `internal` and `external`. 
This FPGA-based SoC can be attacked!

- Trusted/Untrusted areas.
- Threat model: external bus and memory.
- Two kinds of transactions: **internal** and **external**.
This FPGA-based SoC can be attacked!

- Properties to be met in **internal** transactions:
  - No illegal accesses.

- Properties to be met in **external** transactions:
  - Protection of data and code.
  - No modification.
  - Illegible contents.
Threat model for this FPGA-based SoC

Remote software attacks
Virus, worms, Trojan horses

Proximity-based hardware attacks
Power/EM analysis

Reversible proximity-based attacks
Fault injection
Threat model is FPGA-based SoC

• Countermeasures for spoofing, replay and relocation.

• **Internal** transactions metrics:
  • Read/Write rights.
  • Memory mapping.
  • Transaction formats.

• **External** transactions metrics:
  • Confidentiality.
  • Integrity.
This FPGA-based SoC can be secured!

- Hardware secure-enhanced interfaces (« firewalls »).
- Goals:
  - Protection against defined threat model.
  - Low-latency feature.
  - Security updates in case of attack or new application settings.
Proposed solution in a few steps

- Static security

- Dynamic update of security rules
Static solution

- Two kinds of interfaces:
  - Local Firewall (LF) for plaintext sections.
  - Cryptographic Firewall (CF) for DDR protection.
Static solution – Local Firewall (overview)

- Security rules storage in a Block RAM.
- Controls:
  - Address domains (memory mapping).
  - Read/Write accesses.
  - Allowed formats.
Static solution – Cryptographic Firewall

- Behavior similar to a Local Firewall.
- Crypto Module: flexible cipher/integrity functions.
Static solution – Cryptographic Firewall

Ciphering of a 32-bit data block (# of cycles)

- AES + SHA: 74 cycles
- AES + MD5: 90 cycles
- AES-GCM: 22 cycles

Crypto. algorithm (AES-GCM)

Efficient key-dependent message authentication in reconfigurable hardware, FPT 2011.
Static solution requirements

- **Internal** transactions metrics:
  - Read/Write rights.
  - Memory mapping.
  - Transaction formats.

- **External** transactions metrics:
  - Confidentiality.
  - Integrity.
Static solution requirements

- **Internal** transactions metrics:
  - ✓ Read/Write rights.
  - Memory mapping.
  - Transaction formats.

- **External** transactions metrics:
  - Confidentiality.
  - Integrity.
Static solution requirements

- **Internal** transactions metrics:
  - ✓ Read/Write rights.
  - ✓ Memory mapping.
  - • Transaction formats.

- **External** transactions metrics:
  • Confidentiality.
  • Integrity.
Static solution requirements

- **Internal** transactions metrics:
  - ✓ Read/Write rights.
  - ✓ Memory mapping.
  - ✓ Transaction formats.

- **External** transactions metrics:
  - • Confidentiality.
  - • Integrity.
Static solution requirements

• **Internal** transactions metrics:
  ✓ Read/Write rights.
  ✓ Memory mapping.
  ✓ Transaction formats.

• **External** transactions metrics:
  ✓ Confidentiality.
  • Integrity.
Static solution requirements

- **Internal** transactions metrics:
  - ✔ Read/Write rights.
  - ✔ Memory mapping.
  - ✔ Transaction formats.

- **External** transactions metrics:
  - ✔ Confidentiality.
  - ✔ Integrity.
Summary of the static solution

- Flexible cryptography for the external memory.
- Protection against defined threat model even from plaintext sections.
Summary of the static solution

- Flexible cryptography for the external memory.
- Protection against defined threat model even from plaintext sections.

- What’s next?
  - How an attack is detected?
  - How can we update security policies associated with each firewall?
Adaptive solution

Properties to be met:

- Rights restriction.
- IP isolation.
- System reboot.
Adaptive solution

- Attacks monitoring.
- Block RAMs update (security policies, crypto parameters).
- Security-related events saved in a log file.
Monitoring IP

- Point-to-point between IP and firewall.
- Interruption launched when an attack is detected.
Runtime update

- Blocking incoming data.
- Uses AXI protocol features.
- Resuming when security update is finished.
Security policies evolution

- Modifications of read/write rights only.
- Depending on user requirements.
- Critical case: system reboot (downloading the initial bitstream).
Summary of the adaptive solution

- Attacks monitoring.
- Update of security policies without malicious data leakage.
- Low-latency configuration of security Block RAMs.
- Evolution of security policies.
- Existing efforts:

<table>
<thead>
<tr>
<th></th>
<th>SECA</th>
<th>NoC + DPU</th>
<th>Our work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comm. Topology</td>
<td>Bus</td>
<td>NoC</td>
<td>Bus</td>
</tr>
<tr>
<td>Crypto.</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Update</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Threat model</td>
<td>Wide range attacks</td>
<td>Buffer ov.</td>
<td>Wide range attacks</td>
</tr>
</tbody>
</table>
Implementation results

- Xilinx ML605 (Virtex-6).
- ISE Design Suite 13.4.

### Crypto. parameters

<table>
<thead>
<tr>
<th>Crypto. mode</th>
<th>Crypto. parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>C11/D11</td>
<td>Conf. + integrity</td>
</tr>
<tr>
<td>D12</td>
<td>Integrity only</td>
</tr>
<tr>
<td>C12</td>
<td>Plaintext</td>
</tr>
</tbody>
</table>

### Read/Write parameters

<table>
<thead>
<tr>
<th></th>
<th>BRAM</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>MB1</td>
<td>Read only</td>
<td>Read/Write</td>
</tr>
<tr>
<td>MB2</td>
<td>Read/Write</td>
<td>Write only</td>
</tr>
</tbody>
</table>
Implementation results - Area

- Overhead of the configurable version is negligible.
- Area due to crypto.
Implementation results - Latency

- Latency decrease: 33% compared to SECA approach (J. Coburn).

**picProc**: image processing on case study.

**picDrm**: DRM application.

**picDec**: software AES deciphering.
Conclusion on static/configurable solutions

• Static solution: low-latency solution based on hardware-only blocks.

• Runtime security update without data leakage.

• Acceptable area/latency trade-off.

• Compromise of existing works.
Self-reconfigurable security-enhanced communications in FPGA-based MPSoCs

Pascal Cotret, Laboratoire Lab-STICC
Université De Bretagne-Sud
<pascal.cotret@univ-ubs.fr>

Château de Goutelas, Marcoux
June 21st, 2012