Memory Bus Security

Lifeng SU
STMicroelectronics, Rousset
Email: Lifeng.Su@st.com

Renaud PACALET
Télécom Paris, Sophia Antipolis
Email: Renaud.Pacalet@enst.fr
Agenda

• Introduction
• Confidentiality
• Integrity
• SecBus
  – hardware
  – software level issues
• Wrap-up
• Reference
Traditional computing platform

- One hypothesis: on-chip bus is secure
  - side-channel attacks
  - fault injection attacks
  - software attacks (OS)

- Our attention is paid to on-board bus
Data/instruction confidentiality

- Countermeasure: encryption primitive
Program flow confidentiality

- Countermeasure: memory permutation (HIDE project)
**Integrity**

- **Threat 1: spoofing attack**
  - countermeasure: MAC (message authentication code)

- **Threat 2: splicing attack**
  - countermeasure: MAC + address

- **Threat 3: replay attack**
  - countermeasure: hash tree
Hash tree

• Every time one data is read, integrity checking should be processed until the tree root

• Every time one data is written, hash tree should be updated until the tree root
Our project: SecBus

- Introduction
- Confidentiality
- Integrity
- SecBus (hardware)
- SecBus (software level issues)
- Reference
- Wrap-up

• Platform
  Altera Stratix
  S40
  40K logic cells
  @50 MHz

• CPU (Sparc)
  Gaisler Leon:
  http://www.gaisler.com/

• objective: protect the execution of essential applications in linux
OS in SecBus

- the basic protection unit: *process*

- some impacts on Linux kernel:
  - stack (heap)
  - dynamic linking/loading (shared library)
  - signal handling
  - system call (pointer passing)
  - swapping
  - shared memory zones
  - I/O (DMA)
  - IPC
  - others?

![Diagram showing Linux system, Leon processor, SecBus, and processes with keys]
Application loading

1. **encipher offline ELF but header**
   - symmetric cryptography (Ke)

2. **Modify offline ELF header**
   - indicate whether this application is secure or not

3. **Encipher offline ELF header with CPU public key (Kp)**
   - Every CPU has one asymmetric key pair (Kp, Ks)

4. **Load ELF**
   - CPU deciphers ELF header with its private-key (Ks)
   - ELF loader gets the private key (Ke) from ELF header
   - SecBus loads it into memory
## Related projects

<table>
<thead>
<tr>
<th></th>
<th>Data/instruction confidentiality</th>
<th>Program flow confidentiality</th>
<th>Integrity (splicing attack &amp; spoofing attack)</th>
<th>Integrity (replay attack)</th>
<th>OS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>XOM (Stanford)</strong></td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>n</td>
<td>y</td>
</tr>
<tr>
<td><strong>AEGIS (MIT)</strong></td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td>n</td>
</tr>
<tr>
<td><strong>HIDE (GIT)</strong></td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>CryptoPage/HIDE (ENST Bretagne)</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>n</td>
</tr>
<tr>
<td><strong>PE-ICE (ST)</strong></td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td>n</td>
</tr>
<tr>
<td><strong>SecBus</strong></td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td>y</td>
</tr>
</tbody>
</table>
Prospects

• Integration of hardware and OS

• Further improvement of the performance

• Multiprocessor
Reference

- XOM - *Architectural Support for Copy and Tamper-Resistant Software*,
  D. Lie, PhD thesis, Stanford University

- AEGIS - *AEGIS: A Single-chip Secure Processor*,
  G.E. Suh, PhD thesis, MIT

- HIDE - *HIDE: An Infrastructure for Efficiently Protecting Information Leakages on The Address Bus*,
  X. Zhuang, GIT

- CryptoPage/HIDE - *CRYPTOPAGE/HIDE : une architecture efficace combinant chiffrement, intégrité mémoire et protection contre les fuites d’informations*,
  Ronan Keryell, Guillaume Duc, ENST Bretagne

- PE-ICE - *A Parallelized Way to Provide Data Encryption and Integrity Checking on a Processor-Memory Bus*,
  Reouven Elbaz, STMicroelectronics