FPGA Implementation of A Secure Microprocessor

Takafumi Iwasa and Koji Inoue
Fukuoka University
Kyushu University/PREST
Problem!

Computer virus is a serious problem!

- **Q:** Who execute attack codes?
  - **A:** Microprocessor does!

- **Q:** Why the microprocessor execute the malicious code?
  - **A:** He is crazy about performance and power,
  - **A:** But not for Security!
Microprocessor MUST Consider Security!

- Trusted Program
- Malicious Program

- Security
- Branch Prediction
- Selective Activation
- Value Prediction
- Pipelining
- Signal Gating
- High Performance
- SuperScalar
- TLP
- OOO Exe.
- On-chip Cache

- Low Power/Energy
- Resizing
- Drowsy Operation
- Clock Gating
- DVS
Disadvantage of Current Approach

- **Virus Scan**
  - Find Malicious Codes

  - Doesn’t work for unknown viruses

- **Static Certification**
  - Allow to execute ONLY trusted programs

  - Software can be corrupted
  - No way to protect the computer if the trusted program is hijacked by a malicious code
Our Approach

- **Dynamic Program Certification**
  - Believe not software but **hardware**!
  - Regard **execution behavior** as the certification key!
  - Control the execution behavior by a **compiler**!
  - Monitor the behavior at run time by a **HW profiler**!
  - If the profiler does not see the promised behavior, current execution should be dangerous!

---

**Diagram:**
- User
  - Common Secret Key
  - Secure Profiler
  - Microprocessor
  - HW Synthesis
  - Exe. Behavior As the Key!

- Provider
  - Application Program
  - Secure Compiler
  - Object code
  - Exe. Behavior As the Key!
An Implementation Example

Key Information
Address $K$ is always accessed in every $N$ instructions!

Trusted Program

Profiler

N Instructions

Attack Code

Profiler
Evaluation and FPGA Prototyping

- StrongARM Processor w/ Profiler
- Designed by HDL for prototyping
- Simulation using an extended simplescalar

Graph:
- X-axis: Standardized Basic-Block Size
- Y-axis: Norm. Execution Time
- Bar chart showing execution time for different sizes. The sizes include 25, 20, 15, 10, and 5.

Notes:
- StrongARM Model
  - In-Order execution
  - Branch Pred. (not taken)