Building fault models for microcontrollers

Albert Spruyt aspruyt@os3.nl

University of Amsterdam

July 5, 2012
Goal:
*Create a method to model the effects of voltage glitches on microcontrollers.*

Voltage glitching:
*Introduction of faults by controlling voltages.*

Talk will focus on results instead of methodology.
Applications

Control over running code:

• Bypassing PIN/password protection
• Key retrieval
• Extraction of firmware
• Retrieval of user data for evidence
Investigation process

Figure: Investigation process

1 Source: Dr. M. Worring
Setup

Figure: Setup schematic
Atmel XMEGA64A3

- 8-bit data path
- RISC architecture
- Harvard architecture
- Two stage pipeline
- Clock speed of up to 32 Mhz

Figure: XMEGA A3

\(^a\)Source: mcuzone.com
Timing profile

Figure: Independent glitch profile. (Red: glitch signal Blue: Vcc)
Instrumentation

- Initialize peripherals/variables
- Set trigger
- Critical section/test
- Clear trigger
- Send state:
  - General purpose registers
  - Status register
  - Stack pointer
  - Memory
Instruction/glitch timing

Figure: Glitch timing and instruction execution
Instructions

- ALU operations
- Flow control
- Load and store
Results: ALU Operations

Not executed
Corrupted registers
  • Different registers
  • Lower registers
Registers initialized to zero
High chance of a zero result
Results: Flow control

Not executed

Unexpected branches

To different location

• Jump is smaller

• Always forwards
Results: Load and store

Not executed

Incorrect address
  • Lower address
  • Sometimes not from SRAM

Memory initialized to zero
Fault model

Glitches are more likely to:

- Affect the fetch stage
- Jump forward
- Use a lower register
- Use lower memory address
- Transition 1 bits to 0

Figure: Multiply instruction encoding
Attack model

- Do not execute instructions
- Jump to a different location
- Corrupt calculations
- Load/store incorrect values

Example:

```c
hash = sha1Hash(password);
if(memcmp(hash, correct, 20) == 0)
    sendFirmware();
else
    error("incorrect password");
```
Conclusion

• Create a method for building fault models
• Method is described in paper
• XMEGA fault model
Questions?
References
