Karlsruhe Institute of Technology

Practical Introduction to Hardware Security

Lecture 6: Fault Injection Attacks

Instructors: Mehdi Tahoori, Dennis Gnad, Jonas Krautter

INSTITUTE OF COMPUTER ENGINEERING (ITEC) – CHAIR FOR DEPENDABLE NANO COMPUTING (CDNC)

What is Fault Injection?

Fault injection attacks intentionally

cause errors in a system in order to

compromise the security of the system

Overview of Non-Invasive Attacks

- **Black Box Attacks**
  - Brute Force Attack
  - Software Attack
  - Data Remanence
- **Side Channel Attacks**
  - Timing Attack
  - Power Analysis Attack
  - Used in conjunction with Fault Injection
- **Fault Injection Attacks**
  - Clock Glitching
  - Voltage Glitching
  - Used to speed up Black Box Attacks

Black Box Attacks

- **Brute Force**
  - Memory verify guessing
  - Cryptographic key guessing
  - Cyphertext-to-Plaintext Guessing

- **Software Exploits**
  - Undocumented functions
  - Security function flaws
  - Test interface flaws

- **Data Remanence**
  - Lower temperature to -20C or less
  - Volatile memory retains data
  - Read volatile memory contents

Fault Injection Attacks

- **Clock Glitching**
  - Burst of double clock speed – timing critical
  - Requires knowledge gained from side-channel attack
  - Prevent flip-flops from latching correct data
  - Prevent security fuses from setting properly
  - Could cause skipping instructions
- **Voltage Glitching**
  - Burst of high or low voltage – timing critical
  - Requires knowledge gained from side-channel attack
  - Force VDD < VTH
  - Prevent security fuses from setting properly
  - Change control logic outputs
  - Change memory amplifier outputs

Overview of Semi-Invasive Attacks

- **Backside Decapsulation**
  - Backside Imaging
  - Laser Scanning
  - Reverse Engineering

- **Fault Injection Attacks**
  - Local Heating
  - Flash Glitching
  - Laser Glitching

Fault Injection Attacks

- **Local Heating**
  - High power laser is used to selectively heat small areas
  - Hot enough to change VTH but not hot enough to damage
  - Trial and error with location is used to determine glitches
- **Flash Glitching**
  - Magnified camera flash can cause mass glitching
  - Tinfoil masks created to cause selective glitching
  - Trial and error with location and timing is used to determine glitches
- **Laser Glitching**
  - Infrared laser is used to selectively glitch small areas
  - Trial and error with location and timing is used to determine glitches
  - Process is more precise than Flash Glitching

Practical Fault Injection Attacks

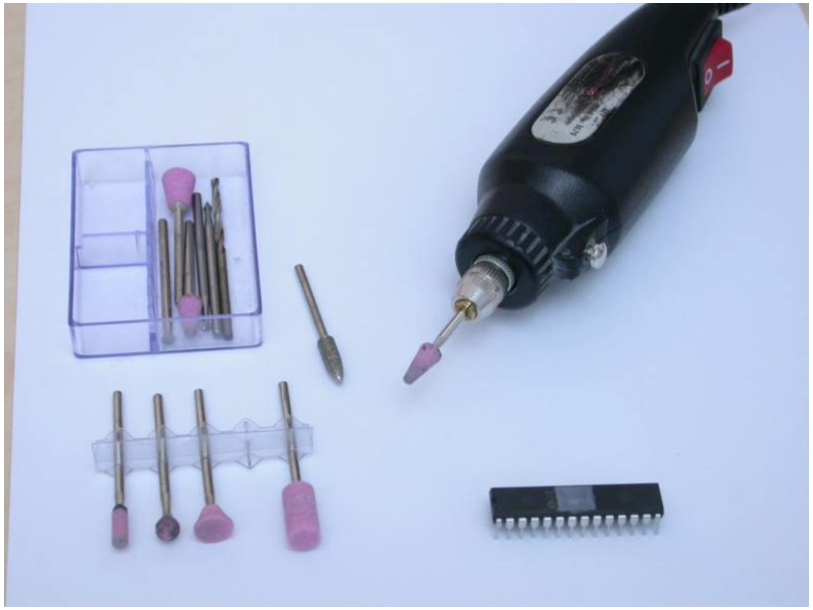
Overiew of Attacks

- **Bumping: Extract contents of protected memory with Verify**
  - Step 1: Backside Decapsulation
  - Step 2: Backside Imaging
  - Step 3: Side Channel Attack
  - Step 4: Laser Glitching Location
  - Step 5: Laser Glitching Timing
  - Step 6: Brute Force Attack

- **Attacks on Cryptographic Algorithms**
  - Attack RSA Repeated Squaring – Retrieve Secret Key
  - Bellcore Attack – Find Prime Factor
  - Sign Change Fault – Elliptic Curve System Attack
  - Directly attack cryptoprocessor

Step 1: Backside Decapsulation

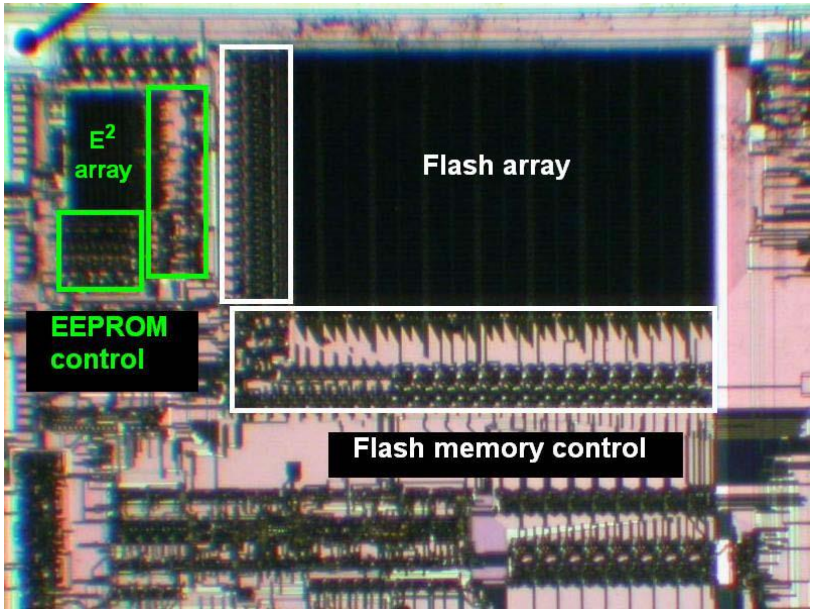
- Use dremel tool to remove backside of outer casing
- Clean surface of exposed substrate material
- Install the IC upside-down to a test interface board



Source: Skorobogatov. Semi-Invasive Attacks. Page 75

Step 2: Backside Imaging

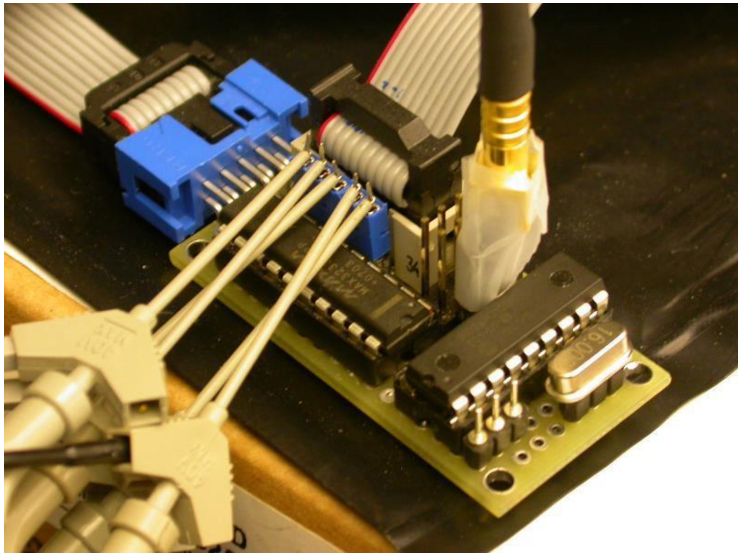
- Use 1000nm infrared light and an optical microscope
- Identify the location of the EEPROM/FLASH memory
- Identify the locations of the memory control logic
- Determine memory bus width



Source: Skorobogatov. Optical Fault Masking Attacks. Page 4

Step 3: Side Channel Attack

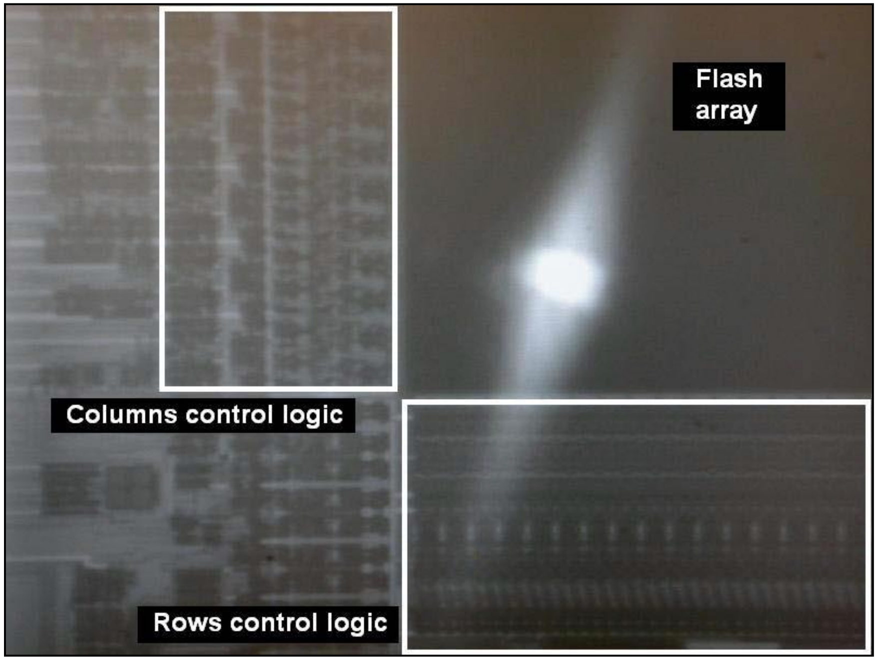
- Set up a power analysis attack using a 10ohm sense resistor
- Perform a Verify function on a dummy input
- Monitor transient current to reverse engineer the process
- Determine packet size of Verify function



Source: Skorobogatov. Flash Memory Bump Attacks. Page 7

Step 4: Laser Glitching Location

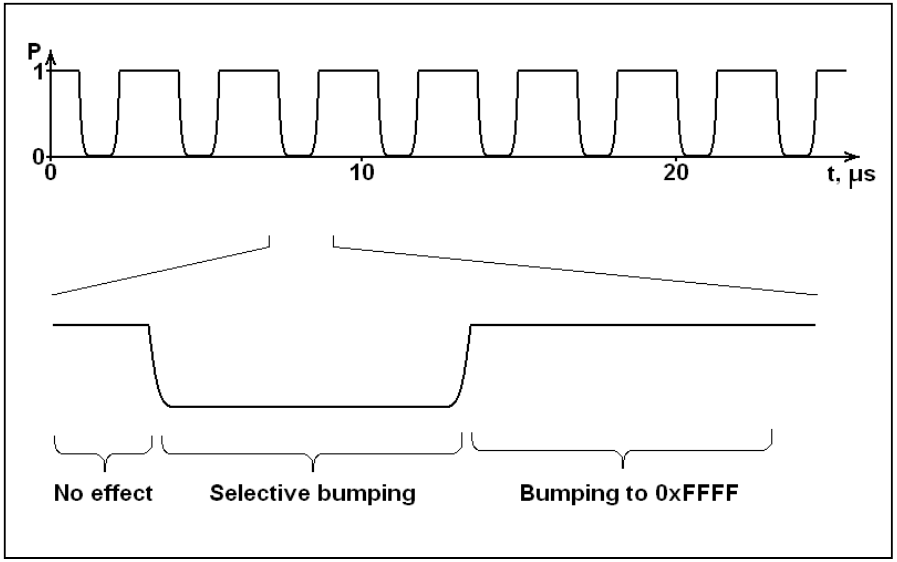
- Set Verify to a pattern of all ‘1’ or all ‘0’
- Find a location in the memory control logic to attack
- Keep trying until your verify pattern succeeds



Source: Skorobogatov. Flash Memory Bump Attacks. Page 5

Step 5: Laser Glitching Timing

- Configure Laser timing to attack all but one block
- Verify that your timing delivers repeatable results
- Maximum unmasked length is the data bus width
- The fewer bits you can unmask at a time the better



Source: Skorobogatov. Flash Memory Bump Attacks. Page 12

Step 6: Brute Force Attack

- Perform a brute force attack on the first unmasked segment
- Unmask the next segment and repeat
- Repeat until all segments are determined

- Example: Verification of a 1024 bit memory on an 8-bit bus
- Traditional Brute Force = 2^{1024} Combinations
- Bump Attack = $128 * 2^8$ = 2^{15} Combinations

- Example: Verification of a 16384 bit memory on a 16-bit bus
- Traditional Brute Force = 2^{16384} Combinations
- Bump Attack = $1024 * 2^{16}$ = 2^{26} Combinations

To the Victor go the Spoils:

- **Commercial IP theft**
- **Recovery of cryptographic keys**
- **Modify software to insert exploits**
- **See plaintext messages**
- **Use stolen keys to extract encrypted data**

Countermeasures

Overview of Exploits

- **Brute Force Attacks**
- **Software Exploits**
- **Data Remanence**
- **Timing Attacks**
- **Power Analysis Attacks**
- **Clock Glitching**
- **Voltage Glitching**
- **Reverse Engineering**
- **IC Modification**
- **Micro Probing**
- **Memory Attacks**
- **Optical Glitching**

Brute Force Attacks

- **Do not return piecemeal Verify results**
- **Large number of possible combinations**
- **Encryption**

Software Exploits

- **Software Quality Assurance**
- **Design for security**
- **Stay one step ahead of attackers**
- **Exception handling**
- **No readbacks on memory**
- **Destroy programming interface after use**

Data Remanence

- **Erase all volatile memory on power-up**
- **Temperature sensor monitoring**
- **Erase all memory on out-of-spec temperature**

Timing Attacks

- **Make all outcomes of subroutine same number of cycles**
- **Insert noops where needed**
- **Randomize response times**

Power Analysis Attacks

- **Intentionally noisy power signal**
- **Make operations consume similar power**
- **Increase the signal-to-noise ratio**

Clock Glitching

- **<u>Internal oscillator</u> for bootloader code**
- **<u>Internal oscillator</u> for secure functions**
- **Make security fuses faster than control logic**
- **Asynchronous logic**

Voltage Glitching

- **Internal brownout reset**
- **Different voltage threshold for security fuses**

Reverse Engineering

- **Security through Obscurity**
- **Additional metal layers to cover design**
- **Re-mark or un-mark all ICs on PCB**
- **Glue logic**
- **Small transistor size**
- **Use of ASICs to replace glue logic on PCB**

IC Modification

- **Metal protection layers on top**
- **Critical signals routed on top of important targets**
- **Tamper sensors in metal layers**

Micro Probing

- **Tamper sensors in metal layers**
- **Small transistor size**
- **Internal shielding**
- **Top level shielding**
- **Security through obscurity**
- **Glue Logic**

Memory Attacks

- **UV Protection**
- **Temperature lockout sensors**
- **Tamper sensors to detect decapsulation**
- **Close proximity between security fuses and memory**

Optical Glitching

- **Protective metal layers to block optical penetration**
- **Tamper sensors in metal layers**
- **UV Protection**
- **IR Protection**
- **Proximity of security fuses and control logic to memory**

Works Cited

- Otto, Martin. 2004. Dissertation, Fault Attacks and Countermeasures.
- Skorobogatov, Sergei. Flash Memory ‘Bumping’ Attacks.
- Skorobogatov, Sergei. 2009. Local Heating Attacks on Flash Memory Devices.
- Skorobogatov, Sergei. Optical Fault Masking Attacks.
- Skorobogatov, Sergei. 2005. Technical Report, Semi-invaseive Attacks – A new Approach to Hardware Security Analysis.
- Giraud and Thiebeauld. 2004. Basics of Fault Attacks.
- Skorobogatov, Sergei. 2011. Fault Attacks on Secure Chips: From Glitch to Flash. Design and Security of Cryptographic Algorithms and Devices (ECRYPT II).