

Fault Injection Attacks

Mohammad Tehranipoor

ECE6095: Hardware Security & Trust
University of Connecticut
ECE Department

1

What is Fault Injection?

Fault injection attacks intentionally cause errors in a system in order to compromise the security of the system

2

Organization of Presentation

- Section 1: Taxonomy of Attacks, Threats, and Security
- Section 2: Non-Invasive Attacks
- Section 3: Semi-Invasive Attacks
- Section 4: Invasive Attacks
- Section 5: Countermeasures
- Section 6: Practical Fault Injection Attacks

3

SECTION ONE

Taxonomy of Attacks, Threats, and Security

4

Taxonomy of Attack Classes

- Non-Invasive Attack
 - Lowest cost
 - No knowledge of inner workings of target
 - No physical tampering
- Semi-Invasive Attack
 - Intermediate cost
 - Some knowledge of inner workings of target
 - Minimal physical tampering required
- Invasive Attack
 - High cost
 - Full picture of inner workings of target
 - Best chance of compromising target

5

Classification of Threats

- Skilled Outsider
 - Exploit existing weaknesses
 - Minimal equipment sophistication
 - Black-box understanding of target system
- Knowledgeable Insider
 - Advanced education and technical expertise
 - Moderate equipment sophistication
 - Some functional knowledge of target system
- Funded Organization
 - Highest education and technical expertise available
 - High equipment sophistication
 - High-Complete functional knowledge of target

6

Levels of Security

- Level 1
 - Bare minimum required protection
 - Minimal defense against glitching and tampering
- Level 2
 - Some tamper proofing
 - Some defense against glitch attacks
- Level 3
 - Passive system lock-outs
 - Passive tamper proofing
- Level 4
 - Active system lock-outs
 - Active tamper detection

7

Cost of Breaking Protection

- None: \$N/A Open book to attacker
- Low: \$1,000 Security through Obscurity
- Med-Low: \$3,000 Regular Microcontroller
- Med: \$30,000 Secure Microcontroller
- Med-High: \$150,000 ASIC, Secure FPGA, Smartcard
- High: \$1,000,000 Secure ASIC

8

Design for Security

- Cost of a security breach
 - Loss of customers and reputation
 - Fines from government
 - Loss of bottom line
- Value of secured data to attacker
 - Commercial value
 - Strategic value
 - Profitability
- Cost of security implementations
 - Price increase
 - Area and complexity increase
 - Power consumption increase

9

SECTION TWO

Non-Invasive Attacks

10

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

11

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

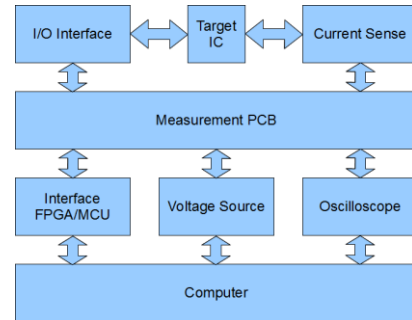
12

Side Channel Attacks

- Timing Attack
 - Number of cycles as a function of subroutine
 - Number of cycles as a function of subroutine's outcome
 - Number of cycles as a function of secret information
 - Can be used to reverse engineer the system
 - Can be used to reduce guesses for brute force
- Power Analysis Attack
 - Current consumption as a function of subroutine
 - Current consumption as a function of subroutine's outcome
 - Current consumption as a function of secret information
 - Can be used to reverse engineer the system
 - Can be used to reverse engineer data flow
 - Can be used to reduce guesses for brute force

13

Side Channel Attack Setup



14

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
 - Skip instructions
- Voltage Glitching
 - Burst of high or low voltage – timing critical
 - Requires knowledge gained from side-channel attack
 - Force $V_{DD} < V_{TH}$
 - Prevent security fuses from setting properly
 - Change control logic outputs
 - Change memory amplifier outputs

15

SECTION THREE

Semi-Invasive Attacks

16

Overview of Semi-Invasive Attacks

- Backside Decapsulation
 - Backside Imaging
 - Laser Scanning
 - Reverse Engineering
- Fault Injection Attacks
 - Local Heating
 - Flash Glitching
 - Laser Glitching

17

Backside Decapsulation



Source: http://freudlabs.com/sample_preparation

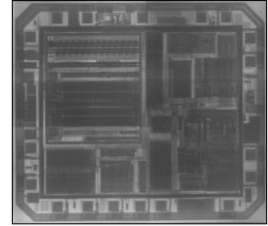
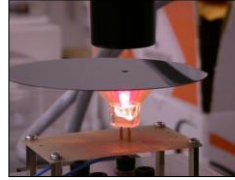
18

Backside Decapsulation

- Backside Imaging
 - Substrate penetrated by infrared light
 - Transistor layout is visible through substrate
 - Reverse engineering of block-level functionality
- Laser Scanning
 - Optical Beam Induced Current (Unpowered IC)
 - Light-Induced Voltage Alteration (Powered IC)
 - See memory structures and read stored values
- Reverse Engineering
 - Determine size of data bus
 - Determine location of control logic
 - Determine location of security logic

19

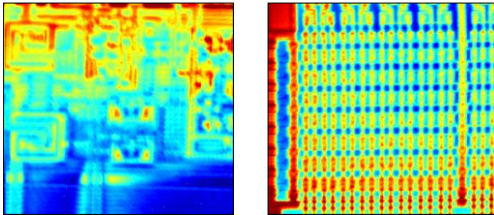
Backside Imaging



Source: Skorobogatov. Semi-Invasive Attacks. Page 94

20

Example of Laser Scan



Source: Skorobogatov. Semi-Invasive Attacks. Page 98

21

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

22

SECTION FOUR

Invasive Attacks

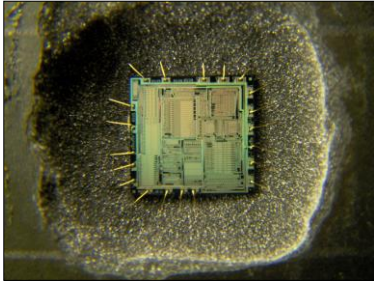
23

Overview of Invasive Attacks

- Reverse Engineering
 - Decapsulation
 - Layout Reconstruction
 - Memory Extraction
- IC Modification
 - Laser Cutting
 - Test Point Creation
 - Wire Bonding
- Micro Probing
 - Eavesdropping
 - Signal Injection
 - Fault Injection

24

Decapsulated IC



Source:

http://en.wikipedia.org/wiki/File:Yamaha_YM3812_audio_IC_decapsulated.jpg

25

Reverse Engineering

- Decapsulation
 - Use of acids to remove layers one by one
 - Provide physical access to all sections of IC
 - Provide knowledge about design of IC
- Layout Reconstruction
 - Image each layer before removing it
 - Build netlist from all images
 - Reverse engineer all functions of IC
- Memory Extraction
 - Read contents of ROM with reconstruction
 - Scan contents of SRAM
 - Scan contents of EEPROM/FLASH

26

IC Modification

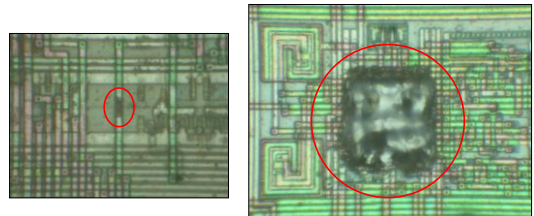
- Laser Cutting
 - Not completely destructive
 - Selective exposure of lower layers
 - Selectively disconnect nets
- Test Point Creation
 - Cut test points in to IC
 - More spots for micro probing below top layer
 - See more signals on more nets
- Wire Bonding
 - Use laser cutting to expose net
 - Cut test point for bonding target
 - Modify circuit paths as needed



Source: Skorobogatov.
Semi-Invasive Attacks.
Page 85

27

Example of Laser Cutting



Source: Skorobogatov. Semi-Invasive Attacks. Page 88

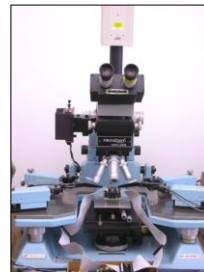
28

Micro Probing

- Eavesdropping
 - Listen to control lines
 - Listen to data bus
 - Full bypass of all protections
- Signal Injection
 - Insert control signals
 - Modify memory contents
 - Forcefully bypass security controls
- Fault Injection
 - High voltage between two probes
 - Destroy transistors
 - Destroy traces

29

Sample Micro Probing Station



Source: Skorobogatov. Semi-Invasive Attacks. Page 84

30

SECTION FIVE

Countermeasures

31

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

32

Brute Force Attacks

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

33

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

34

Data Remanence

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

35

Timing Attacks

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

36

Power Analysis Attacks

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

37

Clock Glitching

- Internal oscillator for bootloader code
- Internal oscillator for secure functions
- Make security fuses faster than control logic
- Asynchronous logic

38

Voltage Glitching

- Internal brownout reset
- Different voltage threshold for security fuses

39

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

40

IC Modification

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

41

Micro Probing

- Tamper sensors in metal layers
- Small transistor size
- Security through obscurity
- Glue Logic

42

Memory Attacks

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

43

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

44

SECTION SIX

Practical Fault Injection Attacks

45

Overview 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

46

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

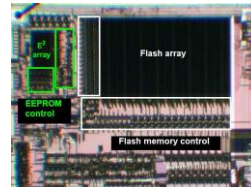


Source: Skorobogatov. Semi-Invasive Attacks. Page 75

47

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

48

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

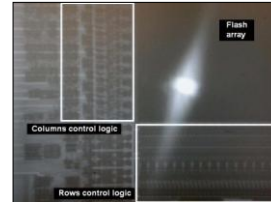


Source: Skorobogatov. Flash Memory Bump Attacks. Page 7

49

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

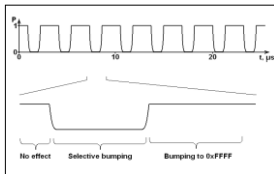


Source: Skorobogatov. Flash Memory Bump Attacks. Page 5

50

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

51

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

52

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

53

Questions

- Questions?

54

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-invasive Attacks – A new Approach to Hardware Security Analysis.
- Giraud and Thiebauld. 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).