

Secure sharing of FPGAs in the Cloud: New Challenges at the Technology Level Experimentation Part

Dennis Gnad, Jonas Krautter, Mehdi Tahoori

INSTITUT FÜR TECHNISCHE INFORMATIK – CHAIR OF DEPENDABLE NANO COMPUTING



KIT – University of the State of Baden-Wuerttemberg and National Research Center of the Helmholtz Association

Physical Fault and Side-Channel Attacks



Agenda of the Practical Part

Introduction

- Background
 - FPGAs, AES
- Side-Channel Attacks

Background and Presentation on Xilinx PCIe Board

Fault Attacks

Background and Presentation on Intel PCIe Board

Hands-on Part (You!)

CPA + DFA + DoS on Lattice FPGA Demo Boards



Introduction – What will you do here?



All types of electrical-level attacks in FPGAs

- Differential Fault Analysis (DFA)
- Correlation Power Analysis (CPA)
- Crash / Denial-of-Service based on Faults (DoS)
- (Category of non-invasive attacks)

No direct FPGA coding (but source code is available)

https://git.informatik.kit.edu/i83/security/nessy21

Introduction – GUI

=	MainWindow 📐 🗕 🗆 🗙	E MainWindow n
DFA SCA DoS		DFA SCA DoS
Serial device: /dev/ttyUSB1	Encryptions:	Serial device: //dev/ttyUSB1 Total correlation:
Baud rate: 19200 Close	Correct ciphertext: 4f:9b:60:db:b4:c9:f6:24:96:f0:4f:a3:cb:4a:2c:1c Faulty ciphertext: 4f:ef:60:db:e2:c9:f6:24:96:f8:4f:1f:cb:42:46:14 Fault injected: Correct ciphertext: cf:26:f5:34:49:0f:d6:3f:6f:85:ad:d6:26:f9:be:43	Baud rate: 500000 0.226
Reset Reprogram	Faulty ciphertext: cf:4a:5e:34:ea:62:de:3f:5a:85:ad:30:26:f9:3d:35 Fault injected: Correct ciphertext: 78:b6:bb:4d:09:2b:31:5a:ec:18:51:3e:59:de:63:34 Faulty ciphertext: 78:be:bb:4d:09:2b:31:5a:ec:18:51:3e:59:de:6b:34	Reset 0.149
Start Encryptions Show Encryptions	Fault injected: Correct ciphertext: 08:e7:eb:e8:b1:65:94:f5:72:76:59:87:a1:21:0b:81 Faulty ciphertext: 08:e7:eb:e8:b1:6d:9c:f5:72:7e:59:87:a1:29:0b:81 Faulty ciphertext: 08:e7:eb:e8:b1:6d:9c:f5:72:7e:59:87:a1:29:0b:81	Sensor:
RO-Mask:	Correct ciphertext: f0:11:8f:82:65:a6:6b:3f:c2:57:8c:27:31:8e:06:32 Faulty ciphertext: f0:11:8f:82:65:a6:63:3f:c2:57:8c:27:31:86:06:32 Fault injected:	Stop Encryptions
RO-Frequency (Hz):	Correct ciphertext: b3:92:c5:e1:21:2d:f5:49:53:2a:38:b2:50:fb:ca:41 Faulty ciphertext: b3:9a:c5:e1:21:25:fd:49:53:2a:38:b2:50:fb:ca:49	Progress step:
3000000	Fault injected: Correct ciphertext: f9:ea:fb:06:8e:3c:bb:62:4f:ee:e9:06:9c:49:b2:c0	50
RO-Duty-Cycle (%):	Faulty ciphertext: f9:ea:fb:06:8e:3c:bb:62:4f:ee:e9:06:9c:49:b2:c8 Fault injected:	Byte: -0.083
50	Correct ciphertext: 03:2a:31:73:83:ac:a1:29:7c:17:79:38:c4:e4:47:00 Faulty ciphertext: 03:2a:31:73:83:a4:a1:29:7c:17:79:38:c4:ec:47:00	
Key recovery: Faults collected (usable/total):	Fault injected: Correct ciphertext: f7:ad:27:e1:4c:22:46:36:57:44:1c:9c:bc:06:a0:19	Bit: Correlation progress:
4/55	Faulty ciphertext:f7:ad:27:e1:4c:22:46:36:57:44:1c:9c:bc:0e:a0:19Fault injected:	Progress point (-1 for auto):
Faults collected (bytes 0,7,10,13):	Correct ciphertext: 20:3c:c0:12:42:46:e0:e2:16:66:4d:60:ee:0b:a7:f4 Faulty ciphertext: 20:34:c0:12:42:46:e0:e2:16:66:4d:60:ee:0b:a7:f4	-1
2	Fault injected: Correct cinhertext: cc:10:5c:ed:a5:3a:7d:02:77:76:72:h1:11:9c:02:62	Number of traces:
Faults collected (bytes 1,4,11,14):	Faulty ciphertext: cc:18:5c:ed:a5:3a:75:02:77:7e:72:b1:19:9c:02:62	0.31
	Correct ciphertext: 74:5e:c9:5e:82:4d:05:4d:28:e4:27:90:2b:a0:77:aa Faulty ciphertext: 74:5e:c9:5e:82:4d:05:4d:20:e4:27:90:2b:a0:77:aa	Correct key byte:
Faults collected (bytes 2,5,8,18):	Fault injected:	a8 Uithet and the har
Faults collected (bytes 3,6,9,12):	Faulty ciphertext: c6:c4:75:50:a8:cf:bc:03:33:12:04:45:03:e2:bf:30 Faulty ciphertext: c6:c4:75:50:a8:cf:bc:03:33:12:04:45:03:e2:bf:38 Fault injected: Fault injected:	a8
1	Correct ciphertext: 6c:00:98:e3:7d:32:bf:01:44:a7:f2:29:e4:db:e0:b0 Faulty ciphertext: 6c:00:98:eb:7d:32:bf:01:44:a7:f2:29:e4:db:e0:b0	-0.18
Key candidates remaining:	Fault injected: Correct ciphertext: 07:d6:b8:c6:21:43:f2:84:a1:7d:5b:7c:36:62:d8:84 Faulty ciphertext: 07:d6:b8:ce:21:4b:f2:84:a1:7d:5b:7c:36:6a:d8:84	
Reset	Fault injected: Correct ciphertext: bc:a3:05:6b:64:ce:ad:62:d7:fc:c5:61:22:c1:a6:78 Faulty ciphertext: b4:a3:05:6b:64:ce:ad:62:d7:f4:c5:61:22:c1:a6:70	-0.43 0 322 644 966 1288 1611 1933 2255 2577 2900

E
DFA SCA DoS
Serial device: [dev/ttyUSB1
Baud rate: 19200
Close
Reset
Reprogram
Crash



[src: Wikipedia]

Board that you will use

Lattice HX8K Breakout Board
 Tiny compared to PCIe Accelerators!

 8k vs. 1M programmable logic elements

 Yet, very similar in technology



Background – Advanced Encryption Standard (AES) 1/2

- Victim of our DFA and CPA attacks
- Symmetric Block Cipher with <u>128</u>/192/256-bit
- Round-based operation in 10/12/14 rounds, 4 basic operations:
 - SubBytes Substitution (non-linear)
 - ShiftRows Permutation/Transposition
 - MixColumns Permutation/Diffusion (not in the last round)
 - AddRoundKey XOR Round Subkey (computed from 128/192/256-bit key)
- For more: The NIST Standard Document is very helpful!

Background – Advanced Encryption Standard (AES) 2/2











[src: Wikipedia]

Our AES Implementation

4x Parallel SubBytes Operations, 4 Clock Cycles

- = 4 SBoxes, implemented in logic
- ShiftRows on the entire Matrix, 1 Clock Cycle
- MixColumns, 1 Clock Cycle
- AddRoundKey, 1 Clock Cycle
- Executed after each other, some transition cycles between

Power Analysis Side-Channel Attack with Correlation Power Analysis (CPA)

Demonstrated on Xilinx Kintex KC705 PCIe Board

You: Experimentation on Lattice HX8K Breakout Board

Side-Channel Attacks with Correlation Power Analysis

The attack we will perform here: Correlation Power Analysis (CPA)
 We measure power/voltage/current but just call it "Power" here

Approach – acquire sets and correlate them:

Set of Measured power values ("traces")

- same key, different plaintext messages
- Multiple Sets of Modeled power values
 - based on: guessed secret key byte and ciphertext messages
- Correlate measured set with each hypothesized set

12



256x 1000x $P_{modeled}(k_{hyp},c)$

Side-Channel Attacks with Correlation Power Analysis

- Correlations separately done per time sample
- Set of Measured power values ("traces")
 - same key, different plaintext messages

• Multiple Sets of Modeled power values

- based on: guessed secret key byte and ciphertext message
- dependent on operations of the algorithm:
- $P_{\text{modeled}}(k_{\text{hvp}},c) = \text{Sbox}^{-1}(k_{\text{hvp}} \oplus c_{i}) \land (2^{\text{bitpos}})$

 k_{hvp} – Hypothesized/Guessed secret key byte, i.e. 0...255

c_i – Ciphertext Byte

13



1.51

> 1000x (Ciphertexts) $P_{modeled}(k_{hvp},c)$

20

30

1000x (Traces)

Detailed Approach

- Acquire Ciphertext + Power Traces over whole AES execution time
- We have multiple measured power values per plaintext (over time) "horizontally"
- We have multiple key hypothesis per plaintext/ciphertext "vertically"
- Correlate all points in time with all hypothesized key-powers
 - For instance, resulting in 256 correlation plots x time samples
 - Leakage occurs at specific time points of the operation we attack
- (If we know the exact time, only "vertical" correlation needed)

Detailed Approach

For instance, resulting in 256 correlation plots x time samples
 Leakage occurs at specific time points of the operation we attack



Detailed Approach



- Note! Most publications use only a single point in time for the progress, based on all power traces' result
- Obviously not possible here when plotting interim results

← This is the result of correlating
 5000 power traces
 Plotting the most correlating time
 sample, depending on the amount
 of traces used ↓



Measured Power Traces used in Correlation \rightarrow

Side-Channel Attacks with Correlation Power Analysis

Some information on the used leakage model:

$$P_{\text{modeled}}(k_{\text{hyp}}) = \text{Sbox}^{-1} (k_{\text{hyp}} \oplus c_{i}) \land (2^{\text{bitpos}})$$

 k_{hyp} – Hypothesized/Guessed secret key byte, i.e. 0...255

c_i – Ciphertext Byte

You see 2^{bitpos} which means we correlate single bits

However, one bit's leakage shows the respective byte leaks

Correlating only a bit can be faster than the whole byte

17

Live Demo

Xilinx Kintex KC705 PCIe Board



Experimental Part

Lattice HX8K Breakout Board

Side-Channel Attack Design



Fault Attack Design



Denial-of-Service (DoS) Design



A few Questions and Things to Look At

SCA:
 Collect 1000+ traces before taking a detailed look into bytes/bits leakage
 After switching sensors you might want to reset CPA (or they mix)
 Q: Do you see a pattern in which bytes leak, and why could that be?



Find best parameter combination maximizing usable faults injection
 Different depending on the board!

Look at the LEDs: Two LEDs should be blinking for correct operation

DoS: Try different boards – not all crash / in same way
 Try to reset after the crash, and see if it can recover or not

Git for very interested participants ;-) https://git.informatik.kit.edu/i83/security/nessy21