Introduction to implementation attacks and countermeasures

Lejla Batina

Digital Security Group
Institute for Computing and Information Sciences (ICIS)
Radboud University Nijmegen
The Netherlands

Training School on Trustworthy Manufacturing and Utilization of Secure Devices - TRUDEVICE

Lisbon, July 15, 2014
Crypto: theory vs physical reality

Algorithms are (supposed to be) theoretically secure

Implementations leak in physical world

- **timing**
- **fault injection**
- **Side-channels**
- **power**
- **sound**
Outline

• Intro
• Side-channel analysis basics
• Power analysis attacks
• Countermeasures
• Side-channel + theoretical cryptanalysis
• Recent work of fault analysis
• Challenges and conclusions
Introduction
Embedded cryptographic devices

What do attackers want to achieve?

Embedded security:
- resource limitation
- physical accessibility
The goals of attackers

• Secret keys/data
• Authorized access
• IP/piracy
• (Location) privacy
• (Theoretical) cryptanalysis
• Reverse engineering
• Finding backdoors in chips [SW12]
• …
“(In)security for Embedded Systems

“Researchers have extracted information from nothing more than the reflection of a computer monitor off an eyeball or the sounds emanating from a printer.” - Scientific American, May 2009.

Scientists Extract RSA Key From GnuPG Using Sound of CPU

kthreadd writes

"In their research paper titled RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis, Daniel Genkin, Adi Shamir and Eran Tromer et al. present a method for extracting decryption keys from the GnuPG security suite using an interesting side-channel attack. By analysing the acoustic sound made by the CPU they were able to extract a 4096-bit RSA key in about an hour (PDF). A modern mobile phone placed next to the computer is sufficient to carry out the attack, but up to four meters have been successfully tested using specially designed microphones."

Your Smartphone Spies On What You Type

Written by Mike James
Sunday, 29 September 2013 00:00

All you have to do is place your phone next to your keyboard to provide a direct channel for anyone to read what you are typing - and it’s all down to the vibration of the keys.

We all do it - place our phones down on the desk next to the keyboard. This might not be such a good idea if you want to keep your work to yourself.

EMBEDDED DEVICES LEAK AUTHENTICATION DATA VIA SNMP COMMUNITY STRING

Researchers have discovered previously unreported problems in SNMP on embedded devices where devices such as secondary market home routers and a popular enterprise-grade load balancer are leaking authentication details in plain text.
Scientists crack keyless entry security systems

Scientists from Ruhr University Bochum (RUB) in Germany have revealed a major flaw in vehicle keyless entry systems by demonstrating a new device that can 'learn and record' radio codes used to lock and unlock car doors. In the same way that a universal remote control can mimic commands from your TV and DVD controller, the new device could enable thieves to access buildings and cars after remote eavesdropping from a distance of up to 100 meters.

http://www.motorauthority.com/

Recent attacks on real products: contactless smartcards with Mifare Classic, DESFire, Atmel CryptoMemory,...
Physical security: before and today

- **Tempest** – known since early 1960s that computers generate EM radiation that leaks info about the data being processed
- In **1965**, MI5: microphone near the rotor-cipher machine used by the Egyptian Embassy the click-sound the machine produced was analyzed to deduce the core position of the machines rotors
- **1979**: effect of cosmic rays on memories (NASA & Boeing)
- First academic publications on SCA by Paul Kocher: 1996 (timing) and 1999 (power)
- Bellcore attack in 1997: Boneh, DeMillo and Lipton
- Many successful attacks published on various platforms and real products e.g. KeeLoq, CryptoMemory, (numerous) contactless cards
Concepts of side-channel leakage

- Side-channel leakage is based on (non-intentional) physical information
- Can enable new kind of attack
- Often, optimizations enable leakages
  - Cache: faster memory access
  - Fixed computation patterns
  - Square vs multiply (for PK)
- Power consumption is data dependent
Sources of side-channel information

- Timing (Kocher 1996), Power (KJJ 1999), EM (UCL & Gemplus 2001)
- Temperature (Naccache et al.)
  - information about the device's malfunction leaked-out via its temperature
- Light (Kuhn)
  - Reading CRT-displays at a distance
  - Observing high-frequency variations of the light emitted
- Sound (Shamir and Tromer) – since 2004
  - Distinguishing an idle from a busy CPU
  - Distinguish various patterns of CPU operations and memory access (RSA signatures)
  - New attack recovers 4096-bit RSA key in one hour
- Photonic emissions (TU Berlin)
Attack categories

• Side-channel attacks
  – use some physical (analog) characteristic and assume access to it

• Faults
  – use abnormal conditions causing malfunctions in the system

• Microprobing
  – accessing the chip surface directly in order to observe, learn and manipulate the device
Taxonomy of Implementation Attacks

• Active versus passive
  – Active
    • The key is recovered by exploiting some abnormal behavior e.g. power glitches or laser pulses
    • Insertion of signals
  – Passive
    • The device operates within its specification
    • Reading hidden signals

• Invasive versus non-invasive
  – Invasive aka expensive: the strongest type e.g. bus probing
  – Semi-invasive: the device is de-packaged but no contact to the chip e.g. optical attacks that read out memory cells
  – Non-invasive aka low-cost: power/EM measurements

• Side-channel attacks: passive and non-invasive
Analysis capabilities

• “Simple” attacks: one or a few measurements - visual inspection
• Differential attacks: multiple measurements
  – Use of statistics, signal processing, etc.
• Higher order attacks: $n$-th order is using $n$ different samples
• Combining two or more side-channels
• Combining side-channel attack with theoretical cryptanalysis
Implementation attacks - equipment
Power Analysis: Measurement setup@RU
Simple Power Analysis (SPA)
Simple Power Analysis (SPA)

- Based on one or a few measurements
- Mostly discovery of data-(in)dependent but instruction-dependent properties e.g.
  - Symmetric:
    - Number of rounds (resp. key length)
    - Memory accesses (usually higher power consumption)
  - Asymmetric:
    - The key (if badly implemented, e.g. RSA / ECC)
    - Key length
    - Implementation details: for example RSA w/wo CRT
    - Search for repetitive patterns
Simple Power Analysis
Using SPA to find a good place to attack
**Insecure RSA implementation**

RSA modular exponentiation

In: message $m$, key $e$ (l bits)
Output: $m^e \mod n$

$A = 1$
for $j = l - 1$ to 0
  $A = A^2 \mod n$ /* square */
  if (bit $j$ of $k$) is 1 then
    $A = A \times m \mod n$ /* multiply */
Return $A$

Side-Channel
Simple Power Analysis (RSA)

- What is the private RSA exponent?

[courtesy: C. Clavier]
Simple Power Analysis (RSA)

[courtesy: C. Clavier]
Differential Power Analysis (DPA)

- Real key
  - Real side-channel
    - Real output
  - Model of side-channel
    - Hypothetical output
- Statistical analysis
  - Hypothesis correct?
- Key hypothesis
Power Analysis

- Direct attacks
  - Simple Power Analysis (1999)
  - Differential Power Analysis (1999)
  - Correlation Power Analysis (2004)
  - Collision Attacks (2003)
- Two-stage attacks
  - Template Attacks (2002)
  - Stochastic Models (2005)
  - Linear Regression Analysis (LRA)
- Advanced attacks: Mutual Information Analysis - MIA (2008), Diff. cluster analysis (2009), PCA (2011), ...
Attacking AES 1\textsuperscript{st} round

ShiftRows

SubBytes

MixColumns

Power Measurements

Difference of means

Correlation
Noisy Traces

Raw Traces

Correlation Trace
Preprocessing

Pre-Processed Traces

Correlation Trace
Intro to Static CMOS

• Most popular circuit style!
• Power consumed when an output signal switches is much higher (than when no switch happens)

0->0: static (low)
0->1: static + dynamic (high)
1->0: static + dynamic (high)
1->1: static (low)

“We don’t understand electricity. We use it.”
- Maya Angelou

=> Dynamic power consumption is the dominant factor in the total power consumption and it is data dependent!
Leakage models

• Transition = Hamming distance model
  – Counts number of 0->1 and 1->0 transitions
  – Assuming same power consumed for both, ignores static power consumption
  – Typically for register outputs in ASICs
  – $\text{HD}(v_0, v_1) = \text{HW}(v_0 \text{ xor } v_1)$
  – Requires knowledge of preceding or succeeding $v_i$

• Hamming weight model
  – Typical for pre-charged busses

• Weighted Hamming weight/distance model

• Signed Hamming distance (0->1 neq 1->0)

• Dedicated models for combinational circuits
Power consumption of smartcard μC
Practical attacks on all platforms

• In the beginning mainly in-house made set-ups

• **Attacks on actual products:**
  - 2008: products employing KeeLoq: Remote Keyless Entry (RKE) systems (chip embedded in RFID transponders) [EK+08]
  - 2009-2011: real-world contactless payment applications based on MIFARE Classic cards, MIFARE DESFire cards (public transport etc.) [KK+09, KS+10]
  - 2012: Atmel CryptoMemory devices (used for printers, gaming, laundromats, parkings etc.) [BG+09]
  - 2013: system 3060 manufactured and marketed by SimonsVoss (wireless door openers)
Combining side-channel with theoretical cryptanalysis
Classical vs side-channel cryptanalysis

• Knowledge:
  – Input/output pairs
  – Input/output pairs + some leakage

• Applicability
  – Generally applicable
  – Limited to certain implementation

Combining both could be beneficial when access to side-channel info is restricted!
Algebraic side-channel attacks [RS09]

• Phase 1 – on-line:
  – Adversary selects as many intermediate computations in the target algorithm as possible and measures their physical leakage represented e.g. by Hamming weight

• Phase 2 – off-line:
  – Adversary writes the algorithm as a system of equations and adds the previously defined functions with known outputs to the system and uses e.g. SAT solvers to find the solutions
Algorithmic side-channel attacks on block ciphers

- Applied on PRESENT and AES
  - Single encrypted plaintext was enough
  - Experiments on an 8-bit microcontroller
  - AES example: 18,000 equations in 10,000 variables
    - Exploring SubBytes implemented as a 256-byte table lookup
    - Exploring MixColumn implemented as four 256-byte table lookups and 9 XOR operations (giving 13 potential leakage points)
- Extended beyond the Hamming weight model
- Generalization of collision-based attacks
Side-channel attacks: Countermeasures
Countermeasures

Purpose: destroy the link between intermediate values and power consumption

– Masking
  • A random mask concealing every intermediate value
  • Can be on all levels (arithmetic -> gate level)

– Hiding
  • Making power consumption independent of the intermediate values and of the operations
  • Special logic styles, randomizing in time domain, lowering SNR ratio
Software countermeasures

• Time randomization: the operations are randomly shifted in time
  – use of NOP operations
  – add random delays
  – use of dummy variables and instructions (sequence scrambling)
  – data balancing (a data element is represented redundantly to make H.w. constant)

• Permutated execution
  – rearranged instructions e.g. S-boxes

• Masking techniques
Hardware countermeasures

• Noise generation
  – hw noise generator would include the use of RNG
  – total power is increased (problem for handheld devices)

• Power signal filtering
  – ex.: RLC filter (R-resistor, C-capacitor, L-inductor)
    smoothing the pow. cons. signal by removing high frequency components
  – one should use active comp. (transistors) in order to keep pow. cons. relatively constant - problem for mob. phones
  – detached power supplies - Shamir

• Novel circuit designs
  – special logic styles
Masking

- Random masks used to hide the correlation between the power consumption and the secret data
- Two types of masking
  - Boolean masking: use $\oplus$, $x' = x \oplus r_x$
  - Arithmetic masking: use addition and subtraction modulo $2^w$ (where $w$ is the digit size), e.g. $x' = (x - r_x) \mod 2^w$
- The conversion from one type to another
- Costs for an example platform
  - Software e.g. 32-bit ARM processor: cycle count - factor 1.96; RAM - 6.27, ROM - 1.36 [Mes00]
- Hardware, ASIC: overhead for masking triples the size of the S-box, from 234 gates (NAND equivalents) to 700 gates [CB08]
Masking AES

• A masking function: \( f(x, m) = x \times m \)
  – * additive or multiplicative masking
• AES includes all linear transformations except S-boxes

\[
S(x + m) = S(x) + m' \neq S(x) + S(m)
\]

• several solutions:
  – Re-computation of masked S-box s.t.
    \( \text{Masked } S(x + m) = S(x) + m \)
  – Multiplicative masking \( S(x) = A \times x^{-1} + b \)
  – Masking in tower fields: in GF(2^2) inversion is linear
Hardware countermeasures

• Dynamic and differential logic (pre-charged dual rail)
  • Duplicate logic
  • Bits are encoded as pairs, e.g. $0 = (1,0)$ and $1 = (0,1)$
  • Circuit is pre-charged, e.g. to all zero $(0,0)$
  • Each DRP gate toggles exactly once per evaluation
    – The number of bit flips is constant and data independent
CMOS vs. WDDL (Tiri, Verbauwhede 2004)

Doesn’t work for small devices!
Fault Analysis: Recent challenges
History

• 1978: one of the first examples fault injection was unintentional, discovered by May and Woods (radioactive particles)
• 1979: effect of cosmic rays on memories (NASA & Boeing)
• 1992: use of laser beam to charge particles on microprocessors, discovered by Habing
• 1997: 1st academic pub. by Boneh, DeMillo, and Lipton showing what’s possible with a single fault  [BDL97]
• 1997: differential fault analysis on secret-key cryptosystems by Biham and Shamir [BS97]
• 2002: 1st pub. implementing Bellcore attack [AB+12]
• 2004: 1st FDTC workshop
Methods

• Variation in supply voltage i.e. glitching
  – Can cause a processor skip instruction
  – Actively investigated by smartcard industry
• Variation in the external clock
  – May cause data misread or an instruction miss
• Change in temperature
  – Change in RAM content
  – Write operations work better
• White light – photons induce faults
• X-rays and ion beams
Goals

• Insert computational fault
  – Null key
  – Wrong crypto result (Differential Fault Analysis - DFA)

• Change software decision
  – Force approval of false PIN
  – Reverse life cycle state
  – Enforce access rights

• ...

...
Low-cost fault injection techniques

• Attacker collects a large number of faulty computations and selects exploitable faults

• Examples:
  – under-powering of a computing device (can cause a single-bit error and no knowledge of the implementation details of the platform is needed!)
  – injection of well-timed power spikes on the supply line of a circuit (possible to skip the execution of a single instruction of microprocessor code)
  – tampering with the clock (shorten the length of a single cycle or overclocking the device)
  – Increasing temperature
  – EM pulses (Eddy current)
Fault injection by light

• UV lamp or a camera flash can be used
• Can cause the erasure of EEPROM and Flash memory cells (usually constants are kept there!)
• It is possible also to selectively wipe out only a part of the stored data
• Optical attacks by S. Skorobogatov [SA02]
Typical problems

- Inaccurate timing of fault injection
- Card breaks down after fault injection test
- Too many parameters that have to be fixed
Parameters search for fault injection

joint work with
Rafael Boix Carpi, Stjepan Picek
and Domagoj Jakobovic
Context of the problem
The question

Q: Can we automatically find good values for parameters using few measurements?
Basic notation

• FL - fault injection
• VCC - voltage at a common connector
• TOE - target of evaluation
• GA - genetic algorithm
What do we know about VCC Fl and a generic TOE?

- A glitch:

  - Gl. Voltage (amplitude)
  - Gl. Length

- Doing this separation reduces problem complexity

1st

Shape

Parameter sets

2nd

Timing
What do we know about VCC FI and a generic TOE?

- Physical behavior of a generic TOE w.r.t.
- Example: Training card #6
  - Glitch voltage [-0.05, -5] V, glitch length [2, 150] ns
  - Timing properties: random values

Successful glitches IN THIS REGION!
Possible outcomes

When we inject the fault in the card, it can have several possible responses:

1. Normal (green)
2. Mute (blue)
3. Reset (blue)
4. Changing = Inconclusive (yellow)
5. Successful = useful for cryptanalysis (red)

In 1st experimental setup, we concentrate only on glitch length and glitch voltage parameters
All investigated targets so far...

JCOP, TC6, GD, JCOP10: ...have this behavior w.r.t.
Results: A real-life smartcard

– Plot of Adaptive Zoom & Bound for the Glitch Shape

~600 measurements

different response types in different instants
Results with the new algorithm [BP+13]

Results of the complete search with the new algorithm

– Total number of measurements: 1812
– Successful glitches: 34
– And the card survived
Preliminary results using GA

– Plot of GA for the Glitch Shape

8 successes out of ~ 2500 measurements
Recent work on an extension of this problem

• Using a memetic algorithm that combines the strengths of the following 3 algos: genetic algorithm, tabu search and local search

• Considered 3 parameters: glitch length, glitch voltage and glitch offset

• Good results for locating faults and characterizing search space using only 300 measurements
Setup
Random search is inefficient
Exhaustive search

![Diagram showing a scatter plot with two axes: Voltage on the x-axis and length on the y-axis. The plot contains data points that form two distinct regions, one in blue and the other in green, with a gradient in between.]
New algorithm

250 measurements: 21 glitches = 8.5% of total measured points
SCA and FA: Recent developments

• Theory
  – Framework for side-channel analysis
  – Leakage resilient crypto
• Theory and Practice
  – More advances in attacks: algorithm specific (combined with cryptanalysis)
  – Machine learning methods
    • Similar techniques apply to traffic analysis
  – New countermeasures
  – New models (going sub-micron)
Conclusions and open problems

• Physical access allows many attack paths
• Passive channels are commonly used for active attacks, for the timing, as additional information etc.
• Trade-offs between assumptions and computational complexity
• Requires knowledge in many different areas
• Combining SCA with theoretical cryptanalysis
• Fault attacks: how to find right choices for so many parameters?
References and further reading (1/3)


References and further reading (2/3)


References and further reading (3/3)


Questions?