Side Channel Attacks and Countermeasures

M. Tehranipoor

Introduction to Hardware Security & Trust
University of Florida

Acknowledgement: Several slides are obtained from Josep Balasch, KU Leuven ESAT / COSIC from his 5th International COSIC Course
Outline

- Introduction
- Side-Channel Emissions
- Attacks Using Side-Channel Information
  - Countermeasures
- Side-Channel Attacks on Microcontrollers
  - Countermeasures
Introduction

- Classic cryptography views the secure problems with mathematical abstractions
- The classic cryptanalysis has had a great success and promise
  - Analyzing and quantifying crypto algorithms’ resilience against attacks
- Recently, many of the security protocols have been attacked through physical attacks
  - Exploit weaknesses in the cryptographic system hardware implementation aimed to recover the secret parameters
Traditional Model (simplified view)

- Attack on channel between communicating parties
- Encryption and cryptographic operations in **black boxes**
- Protection by strong mathematic algorithms and protocols
- Computationally secure
A cryptographic device is an electronic device that implements a cryptographic algorithm and stores a cryptographic key. It is capable of performing cryptographic operations using that key.

**Embedded**: it is exposed to adversaries in a hostile environment; full physical access, no time constraints

- Remark: the adversary might be a legitimate user!
How is Embedded Security Affected?

• New Model (also simplified view):
  o Attack on channel and endpoints
  o Encryption and cryptographic operations in **gray boxes**
  o Protection by strong mathematic algorithms and protocols
  o **Protection by secure implementation**

• **Need secure implementations not only algorithms**
A system is as secure as its weakest link
A system is as secure as its weakest link

And this house is even more secure! The front door is four feet thick and made of solid titanium....

source: Paul Kocher
Side-Channel Leakage

Physical attacks ≠ Cryptanalysis
(gray box, physics) (black box, maths)

• Does not tackle the algorithm's math

• Observe physical quantities in the device's vicinity and use additional information during cryptanalysis
Some Side-Channels (not exhaustive)

- **Passive:**
  - Timing
    - Overall or "local" execution time
  - Power, Electromagnetic (EM) radiation
    - Predominant CMOS technology
    - Dynamic power consumption
    - Electric current induces an EM field
  - More exotic but shown to be practical
    - Sound, temperature, ...

- **Invasive:** Photonic emissions
Invasive Attacks

• **Passive:** micro-probing
  - Probe the bus with a very thin needle
  - Read out data from bus or individual cells directly
  - Several needles concurrently

• **Active:** circuit modification
  - Connect or disconnect security mechanism
    - Disconnect security sensors
    - RNG stuck at a fixed value
    - Reconstruct blown fuses
  - Cut or paste tracks with laser or focused ion beam
  - Add probe pads on buried layers

source: Helena Handschuh

[www.fa-mal.com]
Fault Injection Attacks (I)

- **Non-(semi)invasive**: apply combination of unaccounted environmental conditions
  - Vcc
  - Glitch
  - Clock
  - Temperature
  - UV
  - Light
  - X-Rays
  - ...

- And bypass security mechanisms or infer secrets

*slide source: Helena Handschuh*
Fault Injection Attacks (II)

• **Invasive**: exploit faulty behavior provoked by physical stress applied to the device

  o Laser fault injection allows to target a relatively small surface area of the target device
  o Laser pulse frequency ~ 50Hz
  o Fully automated scan of chip surface

  o Once you have a weak spot: perturbate and exploit

source: [www.new-wave.com](http://www.new-wave.com)
Side-Channel Emissions
In This Lecture

- **Power Consumption** -- Logic circuits typically consume differing amounts of power based on their input data.

- **Electro-Magnetic** -- EM emissions, particularly via near-field inductive and capacitive coupling, can also modulate other signals on the die.

- **Optical** -- The optical properties of silicon can be modulated by altering the voltage or current in the silicon.

- **Timing and Delay** -- Timing attacks exploit data-dependent differences in calculation time in cryptographic algorithms.

- **Acoustic** -- The acoustic emissions are the result of the piezoelectric properties of ceramic capacitors for power supply filtering and AC to DC conversion.
So What Really is Side-Channel Attack?

- Side-Channel attacks aim at side-channel inputs and outputs, bypassing the theoretical strength of cryptographic algorithms.

- Five commonly exploited side-channel emissions:
  - Power Consumption
  - Electro-Magnetic
  - Optical
  - Timing and Delay
  - Acoustic
Measuring Power Consumption

- Not average power over time, not peak power
- Instantaneous power over time
  - Trace or curve, many samples
Measuring Power Consumption

Typical (automated) measurement setup

SCOPE, POWER SUPPLY / FUNCTION GENERATOR

CENTRAL PC

CRYPTOGRAPHIC DEVICES
Measuring Power Consumption

• **Logic**: constant supply voltage, supply current varies

• **Predominant technology**: CMOS
  - Low static power consumption
  - Relatively high dynamic power consumption
  - Power consumption depends on input

• **CMOS inverter**:

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
<th>Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 → 0</td>
<td>1 → 1</td>
<td>Low</td>
</tr>
<tr>
<td>0 → 1</td>
<td>1 → 0</td>
<td>Discharge</td>
</tr>
<tr>
<td>1 → 0</td>
<td>0 → 1</td>
<td>Charge</td>
</tr>
<tr>
<td>1 → 1</td>
<td>0 → 0</td>
<td>Low</td>
</tr>
</tbody>
</table>
Two common victims of hardware cryptanalysis are **smart cards** and **FPGAs**

- Attacks on smart cards are applicable to any general purpose processor with a fixed bus architecture.
- Attacks on FPGAs are also reported. FPGAs represent application specific devices with parallel computing opportunities.
Smart Cards

- Smart cards have a small processor (8bit in general) with ROM, EEPROM and a small RAM
- **Eight wires** connect the processor to the outside world
- **Power supply**: There is no internal battery
- **Clock**: There is no internal clock
- Typically equipped with a **shield** that destroys the chip if a tampering happens
FPGAs allow parallel computing

Multiple programmable configuration bits
Consider a device capable of implementing the cryptographic function

The key is usually stored in the device and protected

Modern cryptography is based on Kerckhoffs's assumption → all of the data required to operate a chip is entirely hidden in the key

Attacker only needs to extract the key
Such attacks are usually composed of two phases:

- **Interaction phase**: interact with the hardware system under attack and obtain the physical characteristics of the device
- **Analysis phase**: analyze the gathered information to recover the key
Principle of divide-and-conquer attack

- The divide-and-conquer (D&C) attack attempts at recovering the key by parts.

- The idea is that an observed characteristic can be correlated with a partial key.
  - The partial key should be small enough to enable exhaustive search.

- Once a partial key is validated, the process is repeated for finding the remaining keys.

- D&C attacks may be iterative or independent.
Attack Classification

- **Invasive vs. noninvasive** attacks

- **Active vs. passive** attacks
  - Active attacks exploit side-channel inputs
  - Passive attacks exploit side-channel outputs
Simple vs. differential attacks

- Simple side-channel attacks directly map the results from a small number of traces of the side-channel to the operation of device under attack.
- Differential side-channel attacks exploit the correlation between the data values being processed and the side-channel leakage.
Power Attacks

- Measure the circuit's processing time and current consumption to infer what is going on inside it.
Measuring Phase

- The task is usually straightforward
  - Easy for smart cards: the energy is provided by the terminal and the current can be read
- Relatively inexpensive (<$1000) equipment can digitally sample voltage differences at high rates (1GHz++) with less than 1% error
- Device’s power consumption depends on many things, including its structure and data being processed
Simple Power Analysis (SPA)

- Originally proposed by Paul Kocher, 1996
- Monitor the device’s power consumption to deduce information about data and operation
- Example: SPA on DES – smart cards
  - The internal structure is shown on the next slide
- Summary of DES – a block cipher
  - a product cipher
  - 16 rounds iterations
    - substitutions (for confusion)
    - permutations (for diffusion)
  - Each round has a *round key*
    - Generated from the user-supplied key
DES Basic Structure

- **Input**: 64 bits (a block)
- **Li/Ri** – left/right half (32 bits) of the input block for iteration i – subject to substitution S and permutation P
- **K** - user-supplied key
- **Ki** - round key:
  - 56 bits used +8 unused
    (unused for encryption but often used for error checking)
- **Output**: 64 bits (a block)
- **Note**: Ri becomes L(i+1)
- All basic op’s are simple logical ops
  - Left shift / XOR

April 17, 2018
The upper trace – entire encryption, including the initial phase, 16 DES rounds, and the final permutation

The lower trace – detailed view of the second and third rounds

The power trace can reveal the instruction sequence
SPA can be used to break cryptographic implementations (execution path, instruction, key change, etc.)

- **DES key schedule**: Involves rotating 28-bit key registers
- **DES permutation**: involves conditional branching
- **Comparison**: Involves string and memory comparison operations performing a conditional branch when a mismatch is found
- **Multipliers**: Involves modular multiplication – The leakage function depends on the multiplier design but strongly correlated to operand values and Hamming weights
- **Exponentiators**: Involves squaring operation and multiplication

SPA Countermeasure:

- Avoid procedures that use secret intermediates or keys for conditional branching operation
The DES structure and 16 rounds are known
Instruction flow depends on data → power signature
Example: Modular exponentiation in DES is often implemented by square and multiply algorithm
Typically the square operation is implemented differently compared with the multiply (for speed purposes)
Then, the power trace of the exponentiation can directly yields the corresponding value
All programs involving conditional branching based on the key values are at risk!

```
exp1(M, e, N)
{
    R = M
    for (i = n-2 down to 0)
    {
        R = R^2 mod N
        if (ith bit of e is a 1)
            R = R \cdot M mod N
    }
    return R
}
```
Differential Power Analysis (DPA)

- SPA targets variable instruction flow
- DPA targets data-dependence
  - Different operands present different power
- Difference between smart cards and FPGAs
  - In smart cards, one operation running at a time
    - Simple power tracing is possible
  - In FPGAs, typically parallel computations prevent visual SPA inspection → DPA
DPA

- DPA can be performed on any algorithm that has the operation $\beta = S(\alpha \oplus K)$,
  - $\alpha$ is known and $K$ is the segment key

The waveforms are captured by a scope and sent to a computer for analysis
What is available after acquisition?

- After data collection, what is available?
  - N plain and/or cipher random texts
    - 00 B688EE57BB63E03E
    - 01 185D04D77509F36F
    - 02 C031A0392DC881E6 ...
  - N corresponding power consumption waveforms
DPA (cont’d)

- Assume the data are processed by a known deterministic function $f$ (transfer, permutation...)
- Knowing the data, one can re-compute off line its image through $f$
  \[ M_i \rightarrow f \rightarrow M'_i = f[M_i] \]
- Now select a single bit among $M'$ bits (in $M'$ buffer)
- One can predict the true story of its variations

<table>
<thead>
<tr>
<th>$i$</th>
<th>Message</th>
<th>$\text{bit}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>B688EE57BB63E03E</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>185D04D77509F36F</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>C031A0392DC881E6</td>
<td>1</td>
</tr>
</tbody>
</table>

The bit will classify the wave $w_i$
- Hypothesis 1: bit is zero
- Hypothesis 2: bit is one
- A differential trace will be calculated for each bit!
• Partition the data and related curves into two packs, according to the selection bit value...

\[ M_i \xrightarrow{f} \begin{cases} \text{bit } (M_i') = 0 \\ \text{bit } (M_i') = 1 \end{cases} \]

• Sum the signed consumption curves and normalise

\[ DPA = \frac{\sum W_1}{N_1} - \frac{\sum W_0}{N_0} \]

\[ 0 \quad \text{B688EE57BB63E03E} \quad 1 \\
1 \quad 185D04D77509F36F \quad 0 \\
2 \quad \text{C031A0392DC881E6} \quad 1 \quad \ldots \]
DPA (cont’d)

\[ \Delta_n = \frac{\sum_{w_i \in S_0} w_i}{|S_0|} - \frac{\sum_{w_i \in S_1} w_i}{|S_1|} \]
DPA -- testing

- The right guess provides the highest spikes!
DPA -- testing

Right guess

Exact prediction of the selection bit

0 B688EE57BB63E03E 1 1
1 185D04D77509F36F 0 0
2 C031A0392DC881E6 1 1
...

Real

Predicted

Average 1 - Average 0

DPA peak
DPA – the wrong guess

Wrong prediction of the selection bit

Real

Predicted

Average 1 - Average 0

No DPA peak
The DPA waveform with the highest peak will validate the hypothesis.
Example: DPA on DES

- Assumption: Attacker presumes detailed knowledge of the DES
- Divide-and-conquer strategy, comparing powers for different inputs
  - Record large number of inputs and record the corresponding power consumption
  - Start with round 15 -- We have access to $R_{15}$, that entered the last round operation, since it is equal to $L_{16}$
  - Take this output bit (called $M'_i$) at the last round and classify the curves based on the bit
    - 6 specific bits of $R_{15}$ will be XOR’d with 6 bits of the key, before entering the S-box
    - By guessing the 6-bit key value, we can predict the bit $b$, or an arbitrary output bit of an arbitrary S-box output
  - Thus, with 16 partitions, one for each possible key, we can break the cipher much faster

A closer look at HW Implementation of DES

(a) DES round

(b) f function
Attacking a secret key algorithm

- DPA works thanks to the perfect prediction of the selection bit
- How to break a key?

![Diagram of cryptographic process](image)

- Messages $M_i$
- Cryptographic Algorithm
- Ciphertexts $C_i$
- Power Consumption Curves $W_i$
Typical DPA Target

- Basic mechanism in Secret Key algorithms (AES, DES...)

Message byte $M$

Key byte $K$

$SBox$

$S$

Exclusive OR

Non-linear substitution by table look-up:

$S = SB(K \oplus M)$
Example – DPA on AES

- Example: AES 128 bits key = 16 bytes $K_i$ (i = 1 to 16)
  - Test 256 guesses per $K_i$ with 256 DPA
  - 128 key bits disclosed with $16 \times 256 = 4096$ DPA ($<< 2^{128}$!)

![Diagram of DPA on AES](image-url)
Example – hypothesis testing

DPA on AES: 1\textsuperscript{st} round and 1\textsuperscript{st} byte (right guess = 1)
General Countermeasures

- **Hiding** -- reduce the SNR by either increasing the noise or reducing the signal
  - Noise Generators, Balanced Logic Styles, Asynchronous Logic, Low Power Design and Shielding

- **Masking/Blinding** -- remove the correlation between the input data and the side-channel emissions from intermediate nodes in the functional block

- **Design Partitioning** -- separate regions of the chip that operate on plaintext from regions that operate on ciphertext

- **Physical Security and Anti-Tamper** -- denial of proximity, access, and possession
Anti-DPA countermeasures

- Applicative counter-measures: make message free randomization impossible!
  - Fix some message bytes
  - Constrain the variable bytes (ex: transaction counter)
- Decorrelate power curves from data
  - by hardware: current scramblers (additive noise)
  - by software: data whitening
- Desynchronise the N traces (curves misalignment)
  - software random delays
  - software random orders (ex: SBoxes in random order)
  - hardware wait states (dummy cycles randomly added by the CPU)
  - hardware unstable internal clock (phase shift)
- DPA is powerful, generic (to many algorithms) and robust (to model errors)...
- ... but there are counter-measures!
Anti-DPA

- Internal clock phase shift
Timing attacks

- Running time of a crypto processor can be used as an information channel
- The idea was proposed by Kocher, Crypto’96

• You put $28 in one of the pots and $10 in the other:

• Question: Compute
  ▪ Blue * 10 + Red *7
  ▪ Tell me if the result is odd or even.

• Is your answer enough to reveal what’s in each pot?
Timing attacks (cont’d)

- Well, normally not:
  \[ 28 \times 7 + 10 \times 10 = 296 \] is an even number
  and
  \[ 10 \times 7 + 28 \times 10 = 350 \] is also even...

- However, just by monitoring the time it takes to give the answer one can tell where each amount is!
RSA Cryptosystem

Key generation:
- Generate large (say, 2048-bit) primes $p$, $q$
- Compute $n=pq$ and $\varphi(n)=(p-1)(q-1)$
- Choose small $e$, relatively prime to $\varphi(n)$
  - Typically, $e=3$ (may be vulnerable) or $e=2^{16}+1=65537$ (why?)
- Compute unique $d$ such that $ed = 1 \pmod{\varphi(n)}$
- Public key = $(e, n)$; private key = $(d, n)$
  - Security relies on the assumption that it is difficult to factor $n$ into $p$ and $q$

Encryption of $m$: $c = m^e \pmod{n}$

Decryption of $c$: $c^d \pmod{n} = (m^e)^d \pmod{n} = m$
How Does RSA Decryption Work?

- RSA decryption: compute $y^x \mod n$
  - This is a **modular exponentiation** operation
- Naive algorithm: **square and multiply**

Let $s_0 = 1$.

For $k = 0$ upto $w - 1$:

- If (bit $k$ of $x$) is 1 then
  - Let $R_k = (s_k \cdot y) \mod n$.
- Else
  - Let $R_k = s_k$.

Let $s_{k+1} = R_k^2 \mod n$.

EndFor.

Return $(R_{w-1})$. 

April 17, 2018
Kocher’s Observation

Let \( s_0 = 1 \).
For \( k = 0 \) upto \( w - 1 \):
    If (bit \( k \) of \( x \)) is 1 then
        Let \( R_k = (s_k \cdot y) \mod n \).
    Else
        Let \( R_k = s_k \).
    Let \( s_{k+1} = R_k^2 \mod n \).
EndFor.
Return \( (R_{w-1}) \).

Whether iteration takes a long time depends on the \( k^{th} \) bit of secret exponent.

This takes a while to compute.

This is instantaneous.
Outline of Kocher’s Attack

- Idea: guess some bits of the exponent and predict how long decryption will take
- If guess is correct, we will observe correlation; if incorrect, then prediction will look random
  - This is a signal detection problem, where signal is timing variation due to guessed exponent bits
  - The more bits you already know, the stronger the signal, thus easier to detect (error-correction property)
- Start by guessing a few top bits, look at correlations for each guess, pick the most promising candidate and continue
Electromagnetic Power Analysis
• Hamming distance model for information leakage
  Correlated to the number of flipping bits (CMOS, VLSI)
• Electrical transitions disturb EM near field (and its flow \( \phi \))
• Captation by inductive probe

- Handmade solenoid \( V = -\frac{d\phi}{dt} \)
  (Diameter = 150 to 500 \( \mu \)m)
- Difficult to calibrate
  (Bandwidth > 100 MHz, low voltage, parasitic effects)
- Good acquisition chain required, but no Faraday cage
  (Sampling at 1GHz)
EMA signal

- Raw signals (TIA: transfer into accumulator instruction)
  - Power is less noisy
  - But EM signatures are sharper!
Spatial Positioning

- Horizontal cartography (XY plane)
  - to pinpoint instruction related areas
  - better if automated
Spectral density of the chip surface
Spatial Positioning

- EM signals versus XY probe position
  - Differential traces between (00h ⊕ 00h) and (FFh ⊕ 00h) picked up at different locations

![Graph showing EM and Power signals with Data signatures marked]
EMA (cont’d)

Advantage of EMA versus PA

- Local information more “data correlated”
- EMA bypasses current smoothers
- EMA goes through HW countermeasures: shields, randomized logic

Drawbacks

- Experimentally more complicated
- Geometrical scanning can be tedious
- Low level and noisy signals (decapsulation required)
Countermeasures

- **Software (crypto routines)**
  - Coding techniques
  - Same as anti DPA/SPA (data whitening…)

- **Hardware (chip designers)**
  - Confine the radiation (metal layer)
  - Blur the radiation (e.g. by an active emitting grid)
  - Reduce the radiation (technology trends to shrinking)
  - Cancel the radiation (dual logic)
Side-Channel Attacks and Countermeasures for Embedded Microcontrollers
Source of side-channel leakage in a microcontroller

- Memory-store instructions
- Memory-load instructions
- Arithmetic instructions
- Control-flow instructions
The leakage caused by $\nu$ is a function of the key value $k^*$, and it can be expressed as follows:

$$L(k^*) = f_{k^*}(p) + \varepsilon$$

The function $f_{k^*}$ is dependent on the crypto-algorithm as well as on the nature of the implementation in hardware and software. The error $\varepsilon$ is an independent noise variable.

Objective: retrieve the internal secret key $k^*$ of a crypto-algorithm.
The PC sends a sample plaintext to the PowerPC on the FPGA for encryption. During the encryption, the digital oscilloscope captures the power consumption from the board. After the encryption is completed, the PC downloads the resulting power trace from the oscilloscope, and proceeds with the next sample plaintext.
Two important aspects of a practical CPA:

- **The selection of the power model**
  
  The power model is chosen so that it has a dependency on a part of the secret key. A good candidate is the output of the substitution step.

- **The definition of the attack success metric**

  **Measurements to Disclosure (MTD):** the more measurements that are required to successfully attack a cryptographic design with side-channel analysis, the more secure that design is.
An example of 256 correlation coefficient traces. Around time 100 µs, the black trace which corresponds to the correct key byte emerges from all the other 255 traces.
Side Channel Countermeasures for Microcontrollers

- Two different kinds of countermeasures:
  - **Algorithm-Level Countermeasures**
    Transform the C program so that the generation of dangerous side-channel leakage is avoided.
  - **Architecture-Level Countermeasures**
    Create a better microcontroller, for example using special circuit techniques, so that no side-channel leakage is generated.
(a) A CMOS standard NAND has **data-dependent** power dissipation;
(b) A DRP NAND gate has a **data-independent** power dissipation

DRP requires the execution of the direct and complementary data paths in parallel.
(a) Concept of balanced processor and VSC programming;
(b) The balanced processor does not show side-channel leakage

The power dissipation from the direct operation always has a complementary counterpart from the complementary operation. The sum of these two is a constant.
References


Videos

- https://www.youtube.com/watch?v=OlX-p4AGhWs&t=3638s
Announcements

• Please submit your course evaluations
• Final Exam scheduled on April 25
  • A set of practice problems will be uploaded in Canvas by tomorrow
  • For EDGE students, an exam will be created and students will be able to select their time
  • Everything covered by Thursday are included
  • Only materials covered after midterm is included
• Grad only term paper (individual) is due on April 28. NO extension.
• Makeup class lecture will be posted later (by Thursday).