Side-Channel Analysis

An introduction

Ulrich Kühne
ulrich.kuhne@telecom-paris.fr
2019–2020

Plan

Introduction

Side-Channel Attacks
- Introduction to SCA
- Power Consumption and EM Radiation
  - Introduction
  - DPA Example
  - Generalization
- Computation Time

Conclusion

What it's all about...

- Understanding the notion of side-channel analysis (SCA)
- Understanding classic side-channel attacks
- Understanding counter-measures against side-channel attacks

General Context

- Algorithm
- Implementation
  - Hardware (ASIC, FPGA...)
  - Software running on a processor (soft-core on an FPGA, micro-controller in an embedded system, general purpose CPU, specialized processor)
- With a specific security objective
  - Confidentiality (example: cipher algorithm)
  - Authentification (example: PIN code verification)
  - ...
- Handling a secret (can be the algorithm itself) that must not be accessible to the adversary

Plan

Introduction

Side-Channel Attacks

- Introduction to SCA
- Power Consumption and EM Radiation
  - Introduction
  - DPA Example
  - Generalization
- Computation Time

Conclusion

Plan

Introduction

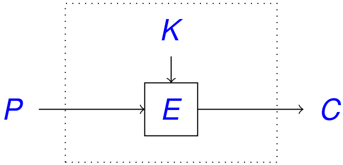
Side-Channel Attacks
- Introduction to SCA
- Power Consumption and EM Radiation
  - Introduction
  - DPA Example
  - Generalization
- Computation Time

Conclusion

Example

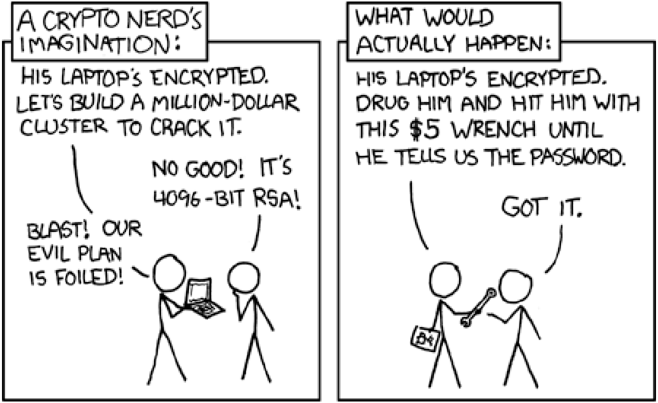
- Example: Cryptographic algorithm implemented on a smart card
- Input: plain text message
- Output: encrypted message
- By construction, the cryptographic key, which is embedded within the smart card, is not accessible via any operation on the input/output interface of the card.

Mathematical View



- KERCKHOFFS principle: P, C et E are public, security depends on K, which is unknown to the adversary
- There are numerous robust algorithms following this model

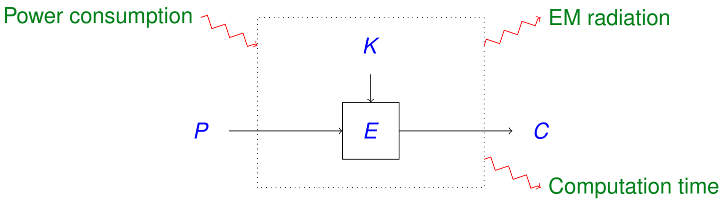
Cryptanalysis vs Reality. . .



[Source: https://www.xkcd.com/538/]

In real life…

…there's hardware



- Additional input/output channels: Side-channels
  - Electromagnetic radiation (EM)
  - Power consumption
  - Computation time
  - …

Side-channel Attacks

- Side-channels depend on the implementation of an algorithm:
  - In software
  - In hardware
- Side-channels cannot be observed on the algorithmic (mathematical, cryptanalytic) level.
- The implementation may leak sensitive information (secrets) via side-channels, even if those secrets never appear on the input/output interface.
- As a consequence, a passive observation can allow an attacker to get hold of the secret!

Concrete Example

Function verifying a PIN code

```
boolean verifyPIN(byte[] inputPIN)
{
  for (int i = 0; i < correctPIN.length; i++)
    if (inputPIN[i] != correctPIN[i])
      return false;

  return true;
}
```

- Suppose that the arrays `inputPIN` and `correctPIN` have size 4 and contain digits only (0–9)
- What is the complexity of an exhaustive search (try all the PINs)?
- Can the attacker be smarter than that?

Concrete Example

Function verifying a PIN code

- The attacker can measure the function's execution time
- Note that the function returns once it finds a wrong digit
- The attacker can try 0xxx, 1xxx, . . . , 9xxx
- One of those digits will result in a slightly longer execution, indicating the first correct digit
- Using this result, she can repeat the same test for the second (third, fourth) digit
- Complexity: We need a maximum of 40 tests (vs 9999 tests for an exhaustive search)
- The side-channel exploited by the attacker is the execution time ⇒ timing attack

Plan

Introduction

Side-Channel Attacks

- Introduction to SCA
- Power Consumption and EM Radiation
  - Introduction
  - DPA Example
  - Generalization
- Computation Time

Conclusion

Plan

Introduction

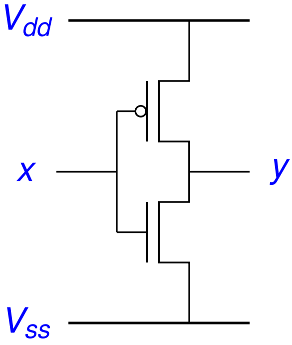
Side-Channel Attacks
- Introduction to SCA
- Power Consumption and EM Radiation
  - Introduction
  - DPA Example
  - Generalization
- Computation Time

Conclusion

Power Consumption of a CMOS Circuit

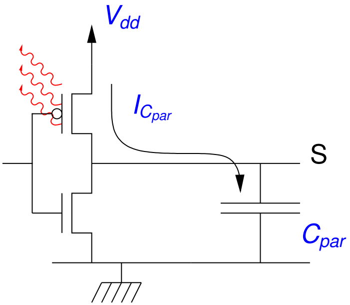
The inverter

- Given input $x = 0$
  - $\rightarrow$ $V_x = 0$
    - $\rightarrow$ nMOS is blocking
    - $\rightarrow$ pMOS is open
  - $\rightarrow$ $V_y = V_{dd}$
  - $\rightarrow$ Logic output is $y = 1$
- Given input $x = 1$
  - $\rightarrow$ $V_x = V_{dd}$
    - $\rightarrow$ nMOS is open
    - $\rightarrow$ pMOS is blocking
  - $\rightarrow$ $V_y = 0$
  - $\rightarrow$ Logic output is $y = 0$



Power Consumption of a CMOS Circuit

Energy dissipation



Rising edge

Falling edge

Power Consumption of a CMOS Circuit

Information leakage

- Except for static leakage current, a CMOS circuit only consumes power during state changes of its gates (dynamic power consumption)
- By observing the power consumption of a circuit, we can deduce its activity
- Note that the number of gates changing their output depends on both the operations and the manipulated data
- Thus, the power consumption can reveal information on the executed operations and the involved data, including secrets

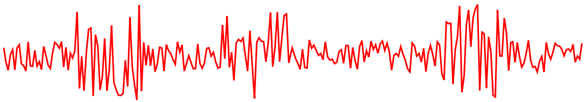
Simple Power Analysis (SPA)

Example: RSA

- Modular exponentiation algorithm

Inputs : M , K
$R = 1$;
for $i = |K| - 1$; $i \geq 0$; $i --$ **do**
    $R = R^2$;
    if $K_i == 1$ **then**
        $R = R \times M$;
    end if
end for
Return $R = M^K$;

- Power consumption profile



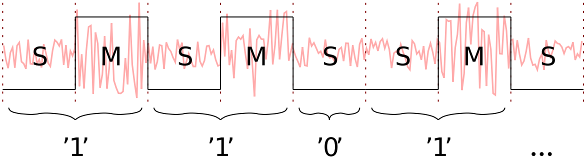
Simple Power Analysis (SPA)

Example: RSA

- Modular exponentiation algorithm

Inputs : M , K
$R = 1$;
for $i = |K| - 1$; $i \geq 0$; $i - -$ **do**
  $R = R^2$;
  if $K_i == 1$ **then**
    $R = R \times M$;
  end if
end for
Return $R = M^K$;

- Power consumption profile



Simple Power Analysis (SPA)

Example: RSA

- Recovery of the full secret (i.e. the key in case of RSA) with a single measurement
- Information is leaked due to different operations depending on the secret (multiply vs square) with a different power consumption profile.
- This type of attack using a single measure is called *Simple Power Analysis*
- Note that the computation time also leaks some information (difficult to exploit in this case)

Plan

Introduction

Side-Channel Attacks

- Introduction to SCA
- Power Consumption and EM Radiation
  - Introduction
  - DPA Example
  - Generalization
- Computation Time

Conclusion

Differential Power Analysis

- Often, the leakage is not as obvious
- Need to use a large number of measures
- Need to use statistical tools
- This type of attack is called DPA (*Differential Power Analysis*)
- There are several variants (CPA, . . .)

DPA: The Ingredients

Leakage Model $\mathcal{M}$ A model (function) predicting the behavior of the observed side-channel of the system, depending on a hypothesis on the system state

Distinguisher $\mathcal{D}$ Statistical tool that allows to detect a correlation between the real system's behavior and our prediction

DPA: The Ingredients

Leakage Model $\mathcal{M}$ A model (function) predicting the behavior of the observed side-channel of the system, depending on a hypothesis on the system state

Distinguisher $\mathcal{D}$ Statistical tool that allows to detect a correlation between the real system's behavior and our prediction

- Since the internal state of the system – in particular the secret – is unknown to the attacker, we need to make a hypothesis
- This hypothesis can be correct or wrong
- The distinguisher allows us to tell the good hypothesis (correct key) from the wrong ones (wrong keys)

DPA Manual 1/2

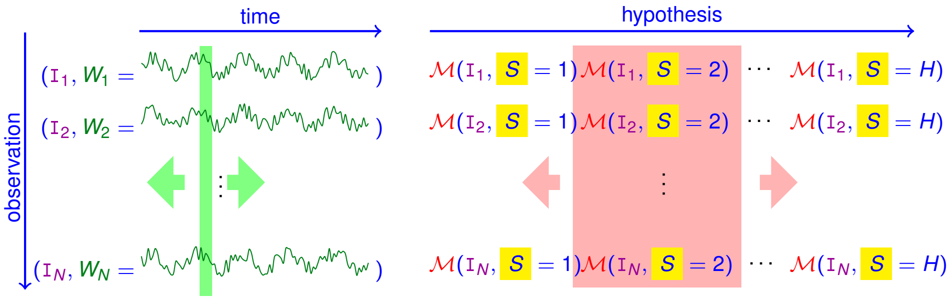
1. Determine a sensitive variable S depending on a part of the secret and on known inputs or outputs.
2. Establish a leakage model $\mathcal{M}(S)$ depending on S.
3. Perform observations (measurements) of the circuit's behavior on the considered side-channel, varying the known inputs or outputs.

DPA Manual 2/2

4. Analyze the data: For each possible value of S
   - For each known input/output P used during the observations, calculate $\mathcal{M}(S, P)$
   - Use the distinguisher $\mathcal{D}$ to check if there is a correlation between the behavior predicted by the leakage model (depending on the hypothesis) and the real world observations

- For the correct value of S, the leakage model predicts correctly the circuit's behavior. As a consequence, the observations will be correlated to the model, and the distinguisher will detect this correlation.
- For all other (wrong) values of S, the model does not predict correctly the behavior, and there will be no correlation between the model and the observations.

DPA Overview



- $\mathtt{I}_i$: Plain text message (or other known inputs/outputs)
- W_i: Measured power consumption (power trace)
- $\mathcal{M}$: Leakage model, depending on secret S (and possibly known inputs/outputs)

$\Rightarrow$ Find a correlation between [pink box] and [green box]

Performing a DPA Attack

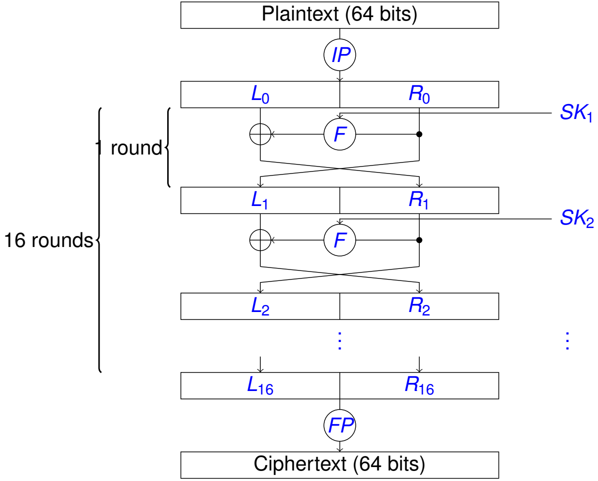
1. Which leakage model to choose?
2. Which distinguisher to choose?
3. How to perform the measurements?

Example

- Context: Hardware implementation of DES *(Data Encryption Standard)* in ECB mode
- What we are looking for: key (56 bits)
- The adversary can send plain text messages to the circuit
- She can read the cipher text and measure the power consumption during the encryption
- Used attack: DPA (*Differential Power Analysis*)

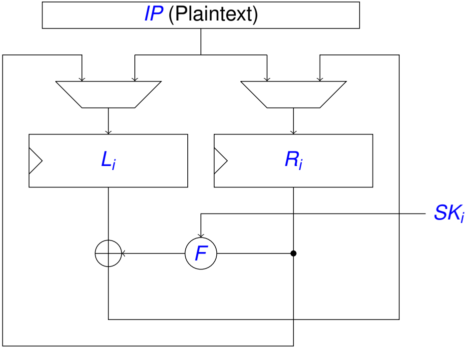
Example: DPA vs DES

DES: algorithmic view



Exemple: DPA vs DES

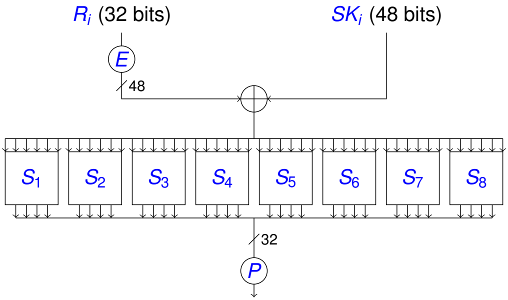
DES: iterative hardware implementation



IP	Initial permutation
F	Feistel function
SK_i	Sub-key (round key)

Exemple: DPA vs DES

DES: Feistel function



- E Extension (32 to 48 bits)
- P Permutation (bit shuffling)
- S_i Substitution

Example: DPA vs DES

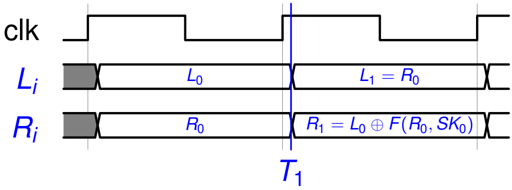
Power consumption model

- How to construct $\mathcal{M}$?
- Power consumption during encryption operation
- Problems
  - DES is not alone on the chip (I/O...)
  - Power consumption of DES heavily depends on the key (56 bits), but we cannot test all 2^{56} hypotheses (that's just brute force...)
- We need to concentrate on the power consumption of a part of the circuit, depending on a part of the key
- We consider the power consumption of the remaining circuit elements as noise

Example: DPA vs DES

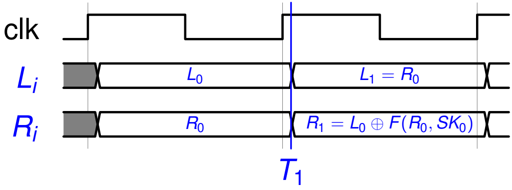
State register on DES data path

- Value change of the state registers (L_i et R_i) during an encryption operation (first round)



Example: DPA vs DES

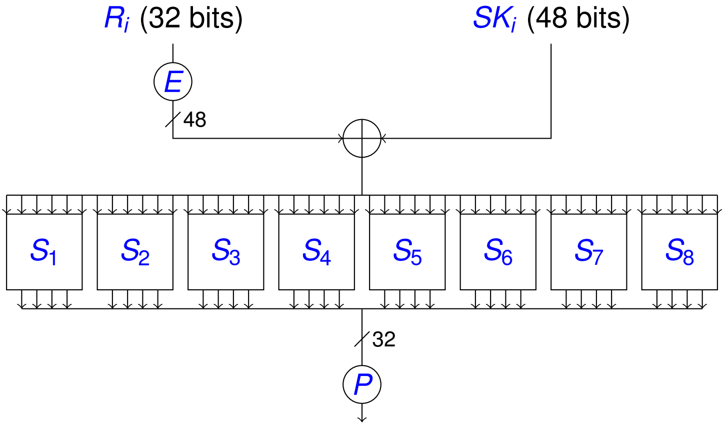
Power consumption of the state registers



- Power consumption of register R_i at time T_1:
  $P_{R_i}(T_1) = \delta \times \mathrm{HD}(R_0, L_0 \oplus F(R_0, SK_0))$
- Known variables: R_0 et L_0 (depending directly on plain text)
- Unknown variables: SK_0 (48 bits of the key K), T_1, and δ
- Still too many hypotheses: 2^{48}

Example: DPA vs DES

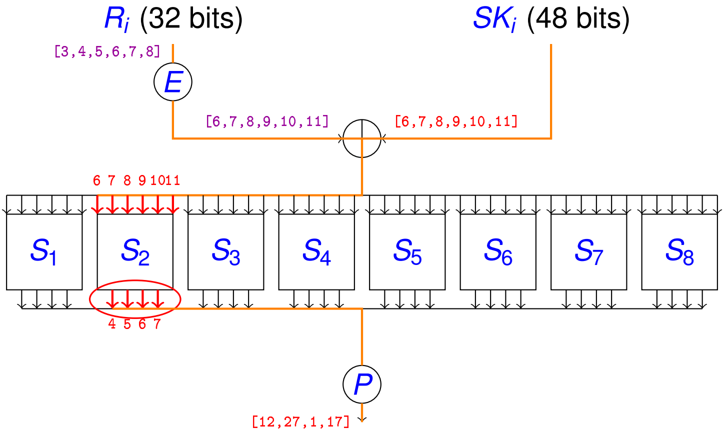
Zoom on the Feistel function



- How to construct a power consumption model depending on fewer bits of the secret key?

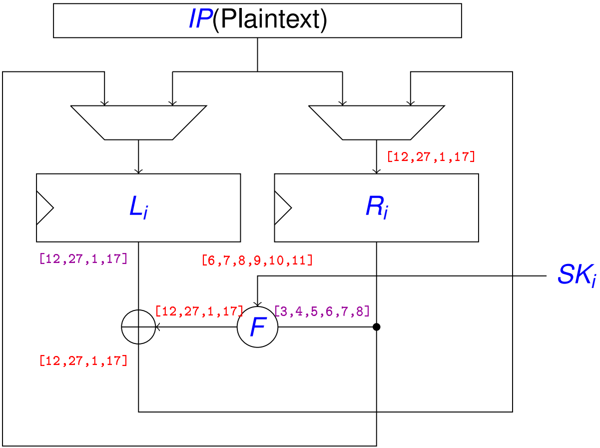
Example: DPA vs DES

Zoom on the Feistel function



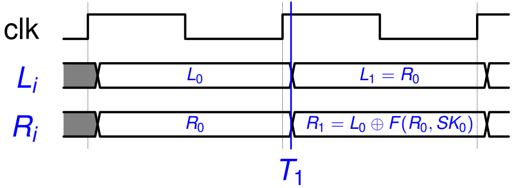
Example: DPA vs DES

Impact of the SBox 2 (first round)



Example: DPA vs DES

Power consumption of state registers (impact SBox 2)



- Considering bits `[12,27,1,17]` of register R_i
- Before T_1, their value depends on R_0 and thus directly on the (known) plain text
- After T_1, their value depends on
  - Bits `[12,27,1,17]` of L_0 (known)
  - Bits `[3,4,5,6,7,8]` of R_0 (known)
  - Bits `[6,7,8,9,10,11]` of SK_0 (unknown)

Example: DPA vs DES

Power consumption model HD on 4 bits

- Power consumption model: $P_{R_i[12,27,1,17]}(T_1) = \delta \times \mathrm{HD}(R_0[12,27,1,17], L_0[12,27,1,17] \oplus F(R_0[3,4,5,6,7,8], SK_0[6,7,8,9,10,11])$
- Depends on a hypothesis on 6 bits of the first round key ($2^6 = 64$ possible hypotheses)
- This model is only valid at instant T_1
- 5 possible output values (Hamming distance on 4 bits): $\{0, \delta, 2\delta, 3\delta, 4\delta\}$
- In the following, we suppose $\delta = 1$
- Finally: $P_4(\mathtt{I}, S) = P_{R_i[12,27,1,17]}(T_1)$, where
  - $\mathtt{I}$ is the plain text
  - S is the hypothesis on $SK_0[6,7,8,9,10,11]$

Example: DPA vs DES

Power consumption model vs actual power consumption

- Our model only predicts the power consumption of a small part of the circuit (4 flip flops) and only at one precise moment (T_1)
- Actual power consumption at T_1:

$$P_{real}(\mathtt{I}, K, T_1) = P_4(\mathtt{I}, S_{good}) + P_{rest}(\mathtt{I}, K, T_1),$$

  where S_{good} corresponds to the good hypothesis (correct value of $SK_0[6, 7, 8, 9, 10, 11]$ depending on K)
- We suppose that $P_{rest}(\mathtt{I}, K, T_1)$ is statistically independent of $P_4(\mathtt{I}, S_{good})$

Example: DPA vs DES

Measurements

- For the good hypothesis on S (S_{good}), at instant T_1, the actual power consumption depends partially on our model $P_4(\mathtt{I}, S)$)
- This dependency is weak, so we need a lot of measurements in order to detect it using the distinguisher
- Perform N measurements (with constant key) for varying plain text messages $\mathtt{I}_1, \ldots, \mathtt{I}_N$

Example: DPA vs DES

Measurements

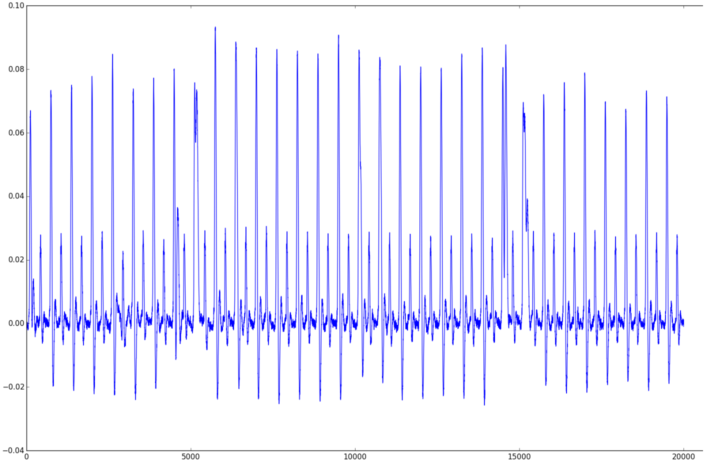
- Power measurement during one encryption operation = power trace
- Trace = vector of samples: $W(\mathtt{I}_i, K, t)$ for $t = 0, \ldots, T-1$ (with T the number of samples per trace)

$$W(\mathtt{I}_i, K, t) = P_{real}(\mathtt{I}_i, K, t) + Noise_{measure}$$

- In the following, we assume that the traces are aligned, i.e. that the index of the sample corresponding to instant T_1 is the same for all traces

Example: DPA vs DES

Example power trace



- Arbitrary units (*x*: time, *y*: power consumption)

Example: DPA vs DES

Analysis algorithm

1. Make a hypothesis on $S = S_H$ (64 possible values, including the good one: S_{good})
2. Partition the set of traces depending on the prediction of the power consumption model: for each trace $W(\mathtt{I}_i, K, t)$ ($i = 1, \ldots, N$)
   - Compute the power consumption model: $P_4(\mathtt{I}_i, S_H)$ (5 possible values)
   - Classify the trace in one of 5 sets $E_{P_4=0}, \ldots, E_{P_4=4}$:

$$E_{P_4=j} = \{W(\mathtt{I}_i, K, t) \mid P_4(\mathtt{I}_i, S_H) = j\}$$

Example: DPA vs DES

Analysis algorithm

3. For each of the 5 sets, compute a mean trace (each sample i of the mean trace is the arithmetic mean of the i-th sample of all the traces in this set):

$$\overline{W}_{P_4=j}(t) = \frac{1}{n} \sum_{W \in E_{P_4=j}} W(\mathtt{I}_i, K, t)$$

for $t = 0, \ldots, T-1$ and with $n = |E_{P_4=j}|$ the number of traces in $E_{P_4=j}$

Example: DPA vs DES

Analysis algorithm

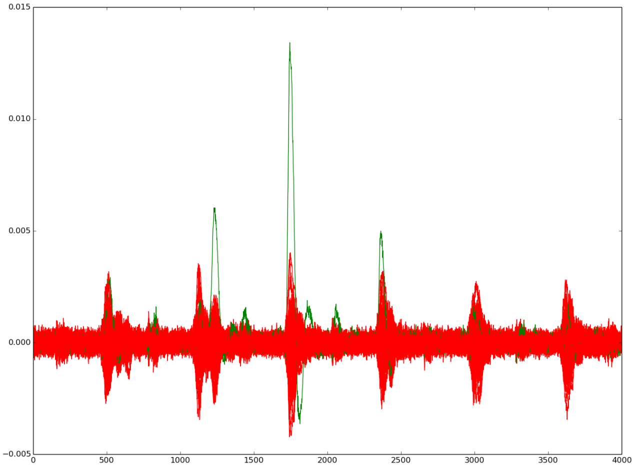
4. Compute a differential trace (for each hypothesis):

   $$W_\Delta(t) = -2 \times \overline{W}_{P_4=0}(t) - \overline{W}_{P_4=1}(t) + \overline{W}_{P_4=3}(t) + 2 \times \overline{W}_{P_4=4}(t)$$

   for $t = 0, \ldots, T-1$
5. Then find the maximum sample in the differential trace: $\mathcal{D}(S_H) = \max_t W_\Delta(t)$
6. Finally, we need to find out for which hypothesis on S, $\mathcal{D}(S_H)$ is maximal. This should be the good hypothesis: $S_{good} = \arg\max \mathcal{D}$

Example: DPA vs DES

Example of a differential trace



- 64 differential traces superposed for SBox 2

Example: DPA vs DES

Why does it work?

- We have:

$$W(\mathtt{I}_i, K, t) = P_{real}(\mathtt{I}_i, K, t) + \mathit{Noise}_{measure}$$

- At time instant T_1:

$$P_{real}(\mathtt{I}, K, T_1) = P_4(\mathtt{I}, S_{good}) + P_{rest}(\mathtt{I}, K, T_1)$$

- It follows:

$$W(\mathtt{I}_i, K, T_1) = P_4(\mathtt{I}_i, S_{good}) + P_{rest}(\mathtt{I}_i, K, T_1) + \mathit{Noise}_{measure}$$

- We consider the measurement noise and the power consumption of the rest of the circuit globally as noise:

$$W(\mathtt{I}_i, K, T_1) = P_4(\mathtt{I}_i, S_{good}) + \mathit{Noise}$$

Example: DPA vs DES

Why does it work? (good hypothesis)

- Let's suppose we make the correct hypothesis on S (i.e. $S_H = S_{good}$)
- If we apply the power consumption model, it correctly predicts, for each observation, the behavior of 4 bits of the state register
- Therefore, the partitioning of the whole set of traces is consistent with the real behavior of these 4 bits:
  For $j \in \{0, \ldots, 4\}, \ \forall W \in E_{P_4=j}$, we have:

$$W(\mathtt{I}_i, K, T_1) = j + Noise$$

Example: DPA vs DES

Why does it work? (good hypothesis)

- When we compute the mean traces, this consistency is preserved:

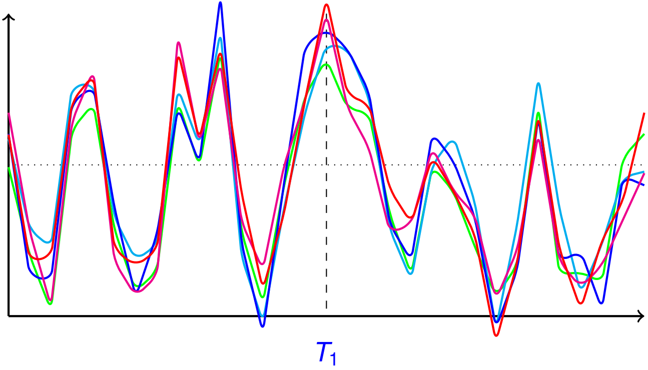
$$\overline{W}_{P_4=j}(T_1) = j + \overline{Noise}$$

- The equation of the differential trace distinguishes this coherence for the sample correspoding to T_1:

$$\begin{aligned} W_{\Delta}(T_1) &= -2 \times \overline{W}_{P_4=0}(T_1) - \overline{W}_{P_4=1}(T_1) + \overline{W}_{P_4=3}(T_1) + 2 \times \overline{W}_{P_4=4}(T_1) \\ &= -2 \times (0 + \overline{Noise}) - (1 + \overline{Noise}) + (3 + \overline{Noise}) + 2 \times (4 + \overline{Noise}) \\ &\approx 10 \end{aligned}$$

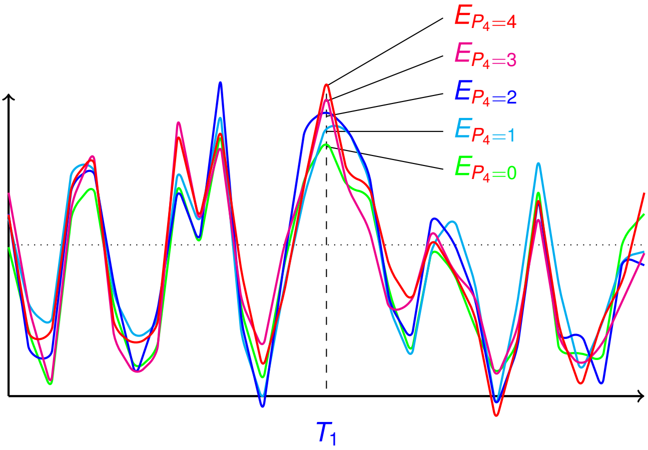
Example: DPA vs DES

Why does it work? (good hypothesis)



Example: DPA vs DES

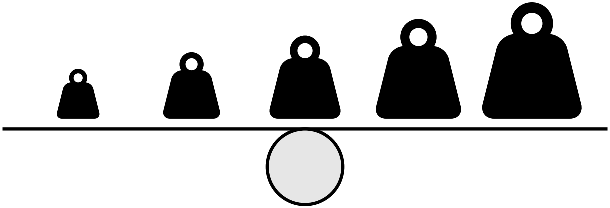
Why does it work? (good hypothesis)



Example: DPA vs DES

Why does it work? (good hypothesis)

$\overline{W}_{P_4=0}$ $\overline{W}_{P_4=1}$ $\overline{W}_{P_4=2}$ $\overline{W}_{P_4=3}$ $\overline{W}_{P_4=4}$

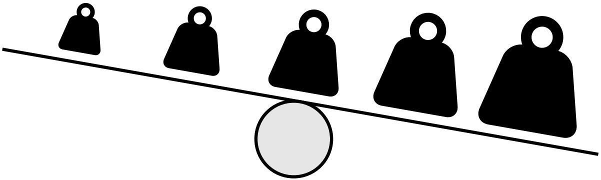


$$W_{\Delta}(t) = -2 \times \overline{W}_{P_4=0}(t) - \overline{W}_{P_4=1}(t) + \overline{W}_{P_4=3}(t) + 2 \times \overline{W}_{P_4=4}(t)$$

Example: DPA vs DES

Why does it work? (good hypothesis)

$\overline{W}_{P_4=0}$ $\overline{W}_{P_4=1}$ $\overline{W}_{P_4=2}$ $\overline{W}_{P_4=3}$ $\overline{W}_{P_4=4}$



$$W_\Delta(t) = -2 \times \overline{W}_{P_4=0}(t) - \overline{W}_{P_4=1}(t) + \overline{W}_{P_4=3}(t) + 2 \times \overline{W}_{P_4=4}(t)$$

Example: DPA vs DES

Why does it work? (bad hypothesis)

- Now suppose we have made a wrong hypothesis on S ($S_H \neq S_{good}$)
- When applying the power consumption model, it does not predict correctly the power consumption of the state register
- Therefore, the partitioning of the traces is inconsistent with the real behavior of the state register:
  For $j \in \{0, \ldots, 4\}, \ \forall W(\mathtt{I}_i, K, t) \in E_{P_4 = j}$, we have:

$$W(\mathtt{I}_i, K, T_1) = k_i + Noise$$

  for some $k_i \in \{0, \ldots, 4\}$

Example: DPA vs DES

Why does it work? (bad hypothesis)

- As a consequence of the inconsistent (more or less random) partitioning, the mean traces of the different partitions are identical:

$$\overline{W}_{P_4=j}(T_1) = 2 + \overline{Noise}$$

- The equation for the differential trace results in a value around 0:

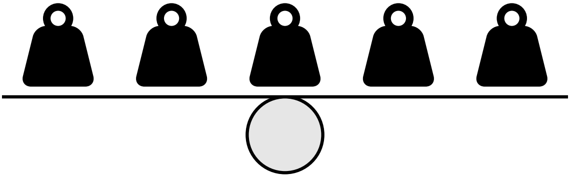
$$\begin{aligned} W_\Delta(T_1) &= -2 \times \overline{W}_{P_4=0}(T_1) - \overline{W}_{P_4=1}(T_1) + \overline{W}_{P_4=3}(T_1) + 2 \times \overline{W}_{P_4=4}(T_1) \\ &= -2 \times (2 + \overline{Noise}) - (2 + \overline{Noise}) + (2 + \overline{Noise}) + 2 \times (2 + \overline{Noise}) \\ &\approx 0 \end{aligned}$$

- This is also the case for all other samples which do not correspond to T_1, for good and bad hypotheses

Example: DPA vs DES

Why does it work? (bad hypothesis)

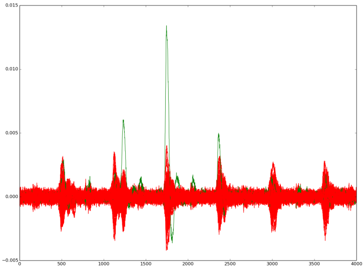
$\overline{W}_{P_4=0}$ $\overline{W}_{P_4=1}$ $\overline{W}_{P_4=2}$ $\overline{W}_{P_4=3}$ $\overline{W}_{P_4=4}$



Example: DPA vs DES

Why does it work?

- As a conclusion, all samples of all differential traces are approximately zero except for the one corresponding to time instant T_1 for the good hypothesis on S



DPA in a Nutshell

Inputs: Model $\mathcal{M}$, traces W_i, inputs $\mathtt{I}_i$ for $1 \leq i \leq N$

for each hypothesis S_H on secret S **do**

  for $i \in \{1, \ldots, N\}$ **do**

    $j \leftarrow \mathcal{M}(\mathtt{I}_i, S_H)$

    $E_{\mathcal{M}=j} \leftarrow E_{\mathcal{M}=j} \cup \{W_i\}$

  end for

  for $j \in \mathsf{range}\,\mathcal{M}$ **do**

    compute mean trace $\overline{W}_{\mathcal{M}=j}$

  end for

  compute differential trace W_Δ

  $\mathcal{D}(S_H) \leftarrow \mathsf{max}_t\, W_\Delta(t)$

end for

$S_{good} \leftarrow \mathsf{arg\,max}\,\mathcal{D}$

Return S_{good}

Example: DPA vs DES

Final observations

- We have recovered 6 bits of SK_0, which gives us directly 6 bits of K
- By repeating the attack on the other S-boxes, we can recover all 48 bits of SK_0, and therefore 48 bits of K
- For the remaining 8 bits, we can attack the second round (the first round is now entirely known), or just do an exhaustive search
- Total complexity of the attack: 64 hypotheses for each of the 8 S-boxes plus exhaustive search: $64 \times 8 + 256$ operations[1]

[1] What is the complexity of one operation?

Plan

Introduction

Side-Channel Attacks

Introduction to SCA

Power Consumption and EM Radiation

Introduction

DPA Example

Generalization

Computation Time

Conclusion

Leakage Models

- Hamming weight: $\mathcal{M}(S) = \mathrm{HW}(S)$
  - Suitable for buses which are reset to zero (or high impedance) after transmission
- Hamming distance [2]:
  $\mathcal{M}(S) = \mathrm{HD}(S, S_{-1}) = \mathrm{HW}(S \oplus S_{-1})$
  - Suitable for hardware implementations (CMOS power consumption)
- Switching distance [8]: $\mathcal{M}(S) = 1$ for transition $0 \rightarrow 1$, and $(1 - \delta)$ for transition $1 \rightarrow 0$, else 0
  - Suitable for near field EM

Statistical Distinguishers

Classification by [9]

- Partitioning
  - Difference of means [7]: DPA
  - Covariance [1]
  - Mutual information [5]: MIA
- Comparison
  - Correlation [2]: CPA

Correlation Power Analysis (CPA)

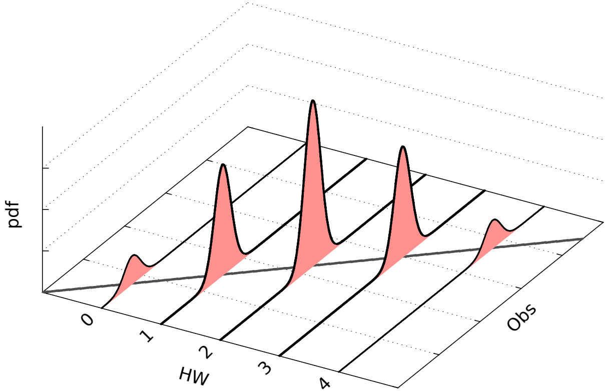
PEARSON correlation coefficient

$$\rho_{X,Y} = \frac{\mathrm{cov}(X, Y)}{\sigma_X \sigma_Y},$$

where $\mathrm{cov}(X, Y) = E[(X - E[X])(Y - E[Y])]$.

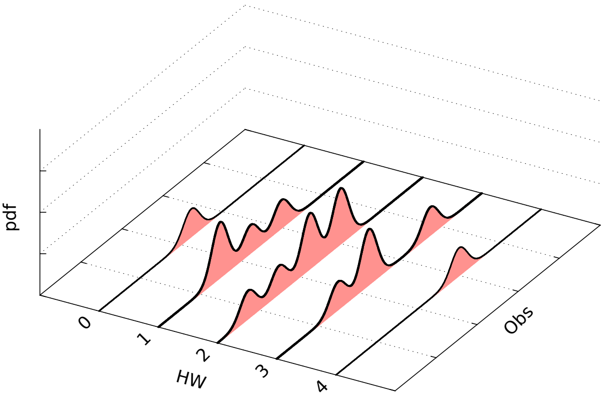
- If there is a linear dependence between the prediction of the leakage model and the real behavior of the circuit, the linear correlation coefficient can be used to test the hypothesis

Correlation Power Analysis (CPA)



Good key hypothesis $\Rightarrow$ correlation $\neq 0$

Correlation Power Analysis (CPA)



Bad key hypothesis ⇒ correlation ≈ 0

Template Attack [4]

Principle

- If we dispose of a second circuit, which is identical to the target circuit, and which we are able to control, we can perform a template attack
- The idea is to learn (profile) how the circuit leaks before using this knowledge on the target circuit for an attack with few traces
- There is two phases
  1. The profiling phase on the test circuit
  2. The attack phase on the target circuit

Template Attack

Profiling

We assume that the circuit executes one out of of K operations: $O_1, \ldots, O_K$ (example: manipulating a sensitive variable)

1. Collect multiple traces of the test circuit for each of the K operations $O_1, \ldots, O_K$
2. Compute the mean traces: $\overline{W}_1, \ldots, \overline{W}_K$
3. Optional: Compute the differences between mean traces in order to identify points of interest $P_1, \ldots, P_N$

Template Attack

Profiling

4. For each operation O_i:
   4.1 For each trace W of this operation O_i, the noise vector for W is given as

$$N_i(W) = (W[P_1] - \overline{W}_i[P_1], \ldots, W[P_N] - \overline{W}_i[P_N])$$

   4.2 Compute the noise covariance matrix: for any pair P_u and P_v of points of interest

$$\Sigma_i[u, v] = \mathrm{cov}(N_i[P_u], N_i[P_v])$$

   4.3 The template for operation O_i is $(\overline{W}_i, \Sigma_i)$

Template Attack

Attack phase

Given an observation S of the target circuit

1. For each possible operation O_i:
   1.1 Compute the observed noise vector

$$\mathbf{n} = N_i(S) = (S[P_1] - \overline{W}_i[P_1], \ldots, S[P_N] - \overline{W}_i[P_N])$$

   1.2 Compute the probability to observe $\mathbf{n}$ (multivariate normal distribution)

$$p_i(\mathbf{n}) = \frac{1}{\sqrt{(2\pi)^N |\Sigma_i|}} \exp(-\frac{1}{2}\mathbf{n}^T \Sigma_i^{-1} \mathbf{n}),$$

   where $|\Sigma_i|$ is the determinant of Σ_i

2. The most probable operation is the one for which the probability of observing the noise $\mathbf{n}$ is maximal

Template Attack

Improvements

- A Principal Component Analysis (PCA) can be used to reduce the size of the templates
- Template attacks are very powerful and can often recover the entire secret using a single or few traces

Plan

Introduction

Side-Channel Attacks

Introduction to SCA

Power Consumption and EM Radiation

Introduction

DPA Example

Generalization

Computation Time

Conclusion

Timing Attacks

- Attacks based on power consumption or EM radiation require physical access to the target device
- In contrast, timing attacks can be performed remotely, including over a network
- Examples:
  - Remote key recovery over the network [3]
  - Key recovery from another virtual machine running on the same host [6]
- Possible sources of timing variations:
  - Algorithmic
  - Hardware optimizations of the host processor: cache, pipeline, . . .

Timing Attacks

Example: Attacking RSA over the network [3]

- RSA in OpenSSL (version 0.9.7)
- Due to some optimizations (Chinese remainder theorem, Montgomery reduction, sliding window expoentiation, Karatsuba multiplication) the execution time slightly depends on the secret key
- The attack has been demonstrated locally and remotely over a network
- Taking the mean of many tries, the latency and jitter introduced by the network are not sufficient to mask the small timing variations
- More attacks in the μ-architecture chapter

Plan

Introduction

Side-Channel Attacks
- Introduction to SCA
- Power Consumption and EM Radiation
  - Introduction
  - DPA Example
  - Generalization
- Computation Time

Conclusion

Conclusion

- Physical implementations leak information on various side-channels
  - Power
  - EM radiation
  - Timing
  - ...
- If the leakage depends on sensitive data (such as a cryptographic key), it can be exploited by a side-channel attack
- These attack mostly require physical access to the target system
- Statistical side-channel attacks can be very effective

Bibliography I

[1] Régis Bevan and Erik Knudsen.
Ways to Enhance Differential Power Analysis.
In *ICISC*, volume 2587 of *Lecture Notes in Computer Science*, pages 327–342. Springer, 2002.

[2] Éric Brier, Christophe Clavier, and Francis Olivier.
Correlation Power Analysis with a Leakage Model.
In *CHES*, volume 3156 of *LNCS*, pages 16–29. Springer, August 11–13 2004.
Cambridge, MA, USA.

[3] David Brumley and Dan Boneh.
Remote timing attacks are practical.
In *Proceedings of the 12th Conference on USENIX Security Symposium*, pages 1–14, 2003.

[4] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi.
Template Attacks.
In *CHES*, volume 2523 of *LNCS*, pages 13–28. Springer, August 2002.
San Francisco Bay (Redwood City), USA.

[5] Benedikt Gierlichs, Lejla Batina, Pim Tuyls, and Bart Preneel.
Mutual information analysis.
In *CHES, 10th International Workshop*, volume 5154 of *Lecture Notes in Computer Science*, pages 426–442. Springer, August 10-13 2008.
Washington, D.C., USA.

[6] Gorka Irazoqui, Thomas Eisenbarth, and Berk Sunar.
S$a: A shared cache attack that works across cores and defies vm sandboxing — and its application to aes.
In *2015 IEEE Symposium on Security and Privacy (SP)*, pages 591–604, May 2015.

Bibliography II

[7] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun.
Differential Power Analysis.
In *Proceedings of CRYPTO'99*, volume 1666 of *LNCS*, pages 388–397. Springer-Verlag, 1999.
(PDF).

[8] Éric Peeters, François-Xavier Standaert, and Jean-Jacques Quisquater.
Power and electromagnetic analysis: Improved model, consequences and comparisons.
Integration, The VLSI Journal, special issue on "Embedded Cryptographic Hardware", 40:52–60, January 2007.
DOI: 10.1016/j.vlsi.2005.12.013.

[9] François-Xavier Standaert, Benedikt Gierlichs, and Ingrid Verbauwhede.
Partition vs. Comparison Side-Channel Distinguishers: An Empirical Evaluation of Statistical Tests for Univariate Side-Channel Attacks against Two Unprotected CMOS Devices.
In *ICISC*, volume 5461 of *LNCS*, pages 253–267. Springer, December 3-5 2008.
Seoul, Korea.