Physical Security of Smart Cards

Michael Tunstall

University College Cork, Ireland.

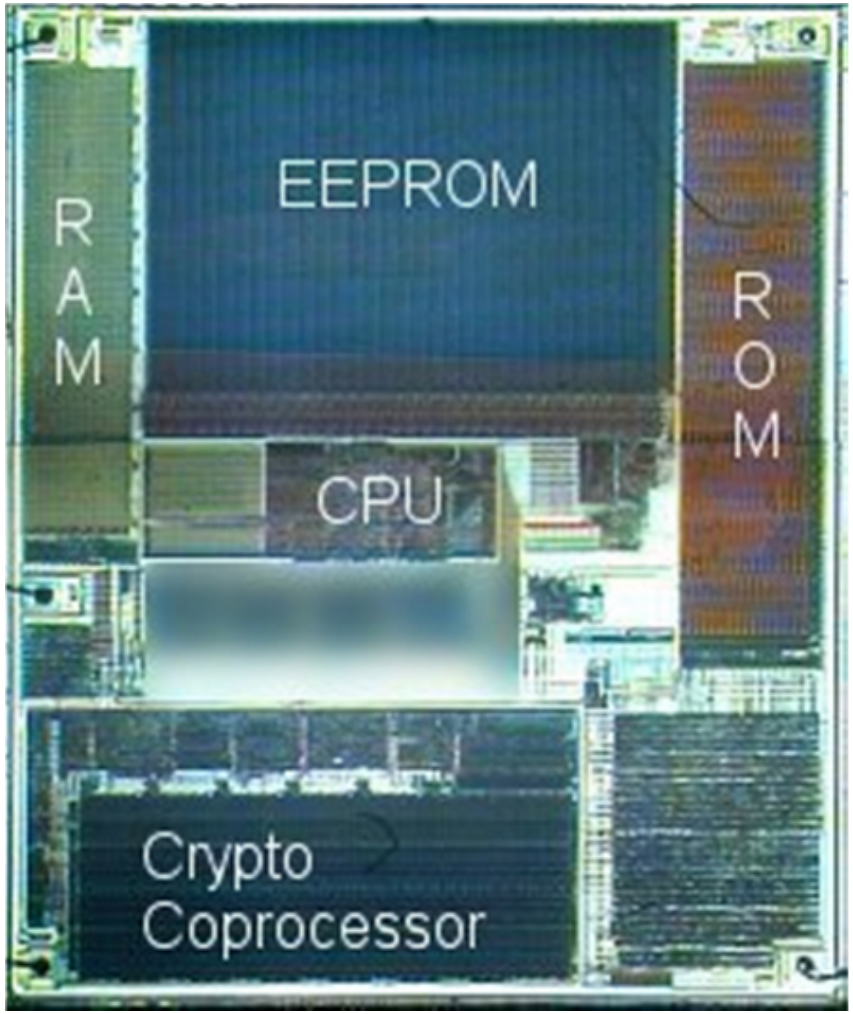
Limerick — March 5, 2008

Outline

1. Introduction
   - What is a Smart Card?
   - Why use Smart Cards?
2. Measuring the Power Consumption
   - The Experimental Setup
3. Simple Power Analysis
   - Attacking an Algorithm
   - Attacking an Algorithm
   - Reverse Engineering
4. Differential Power Analysis
   - Correlation Power Analysis
   - Using the Partial Correlation
   - Case Study: The DES block cipher
5. Fault Analysis
   - Case Study: The DES block cipher
6. Countermeasures
7. Other Problems
8. Conclusion

What is a Smart Card?

- Essentially a small computer.



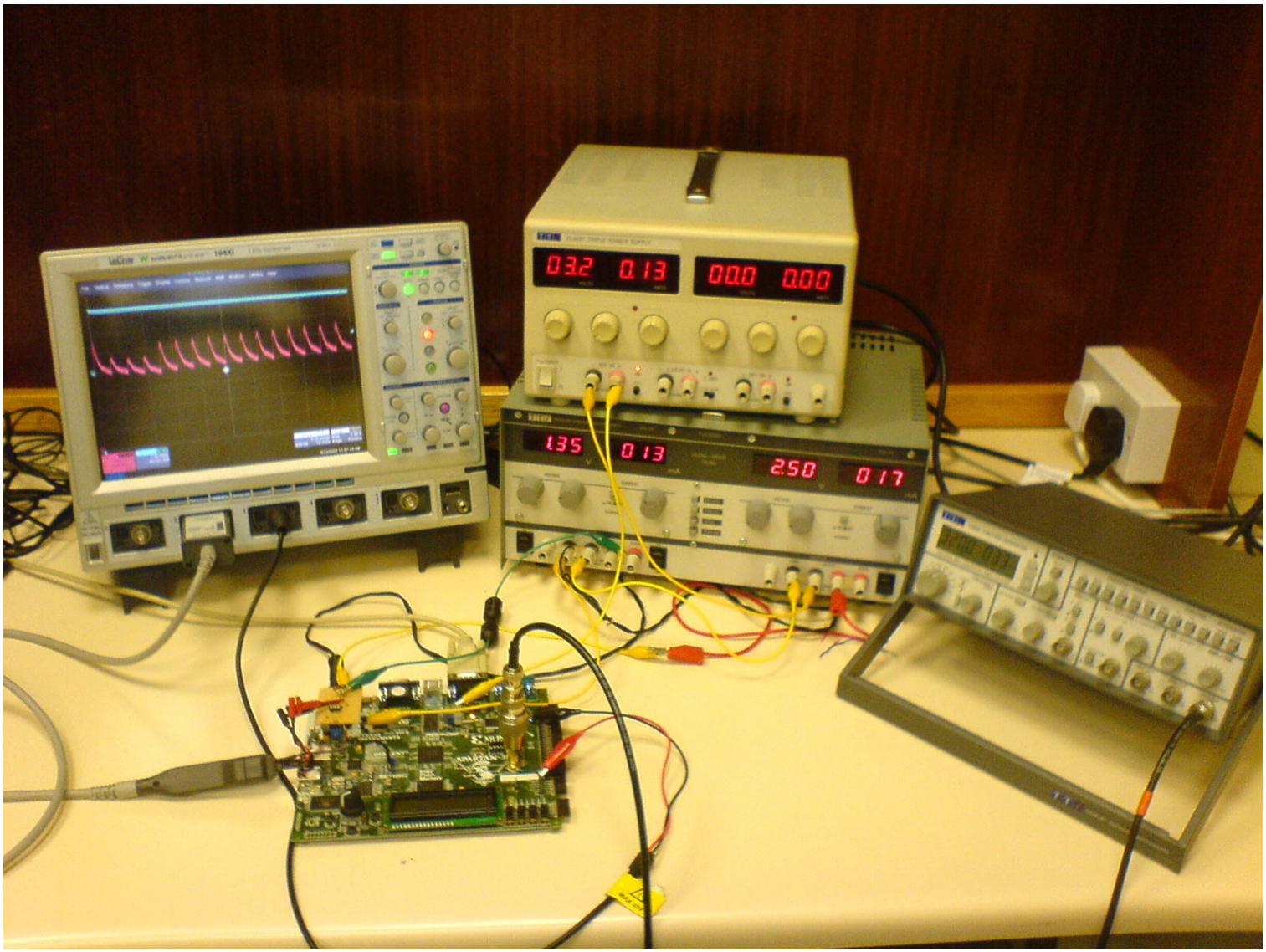
Why use Smart Cards?

- Tamper resistance.
  - Storage.
  - Processing (e.g. authentication/ciphering algorithms).
- Portability.
  - Ease of use.
  - Onboard key generation.
  - Cost.

Outline

1. Introduction
   - What is a Smart Card?
   - Why use Smart Cards?
2. Measuring the Power Consumption
   - The Experimental Setup
3. Simple Power Analysis
   - Attacking an Algorithm
   - Attacking an Algorithm
   - Reverse Engineering
4. Differential Power Analysis
   - Correlation Power Analysis
   - Using the Partial Correlation
   - Case Study: The DES block cipher
5. Fault Analysis
   - Case Study: The DES block cipher
6. Countermeasures
7. Other Problems
8. Conclusion

The Experimental Setup



Outline

1 Introduction
- What is a Smart Card?
- Why use Smart Cards?

2 Measuring the Power Consumption
- The Experimental Setup

3 Simple Power Analysis
- Attacking an Algorithm
- Attacking an Algorithm
- Reverse Engineering

4 Differential Power Analysis
- Correlation Power Analysis
- Using the Partial Correlation
- Case Study: The DES block cipher

5 Fault Analysis
- Case Study: The DES block cipher

6 Countermeasures

7 Other Problems

8 Conclusion

Simple Power Analysis (SPA)

- Simple Power Analysis is the analysis of one, or several, power consumption traces to determine what is occurring within a device.
- SPA will always be specific to one implementation, i.e. a given algorithm on a given device (electrical properties).
- SPA can be used to:
  - Determine information on secret/private keys in some instances.
  - Reverse engineering of algorithms.
    - ★ Attacking an implementation of a cryptographic algorithm will involve the reverse engineering of the algorithm used and the key being manipulated.

Simple Power Analysis (SPA)

- If we consider the square and multiply algorithm.

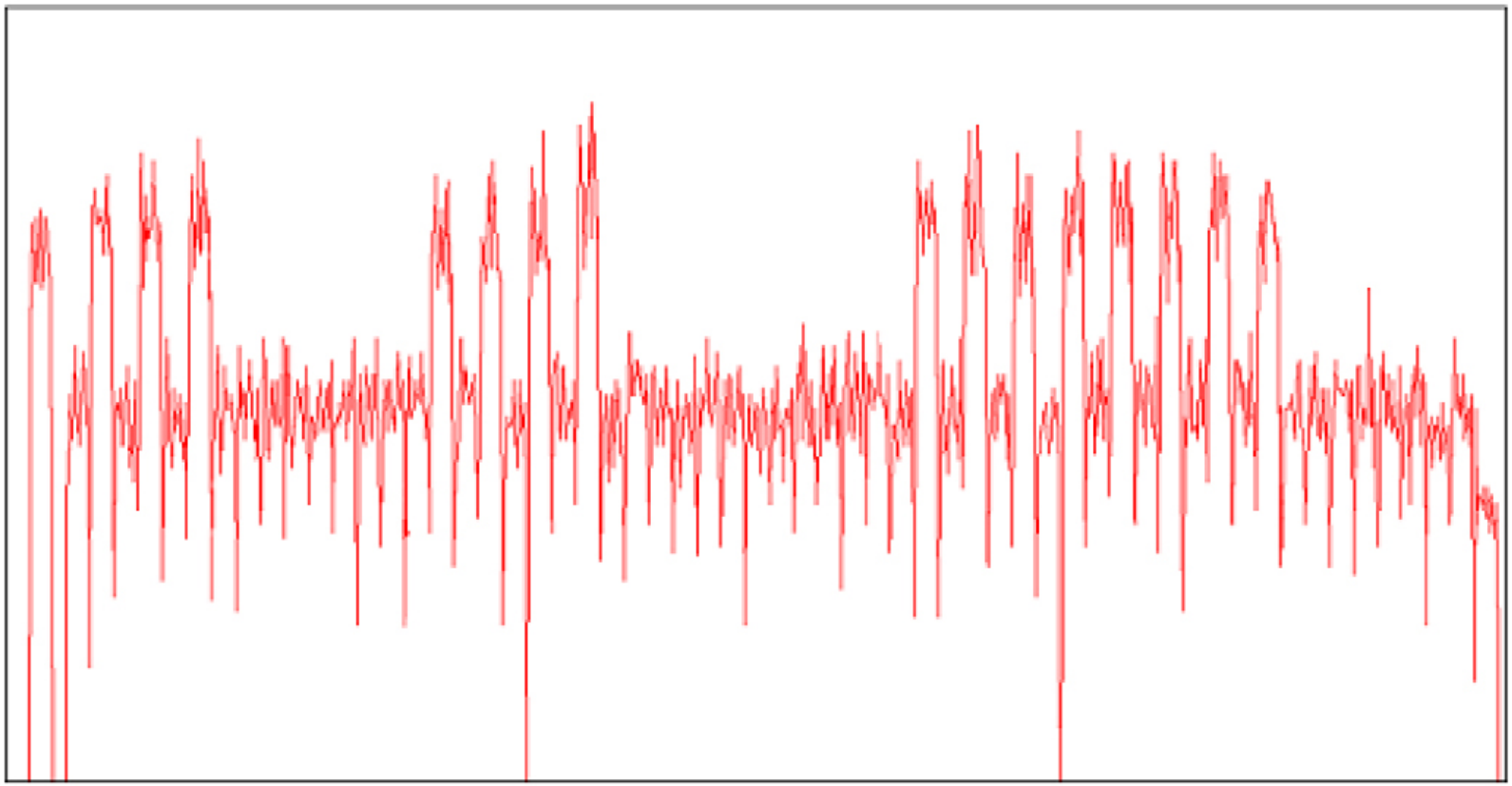
Algorithm 1: The Square and Multiply Algorithm

Input: $M, d = (d_x, d_{x-1}, \ldots, d_0)_2, N$
Output: $C = M^d \bmod N$

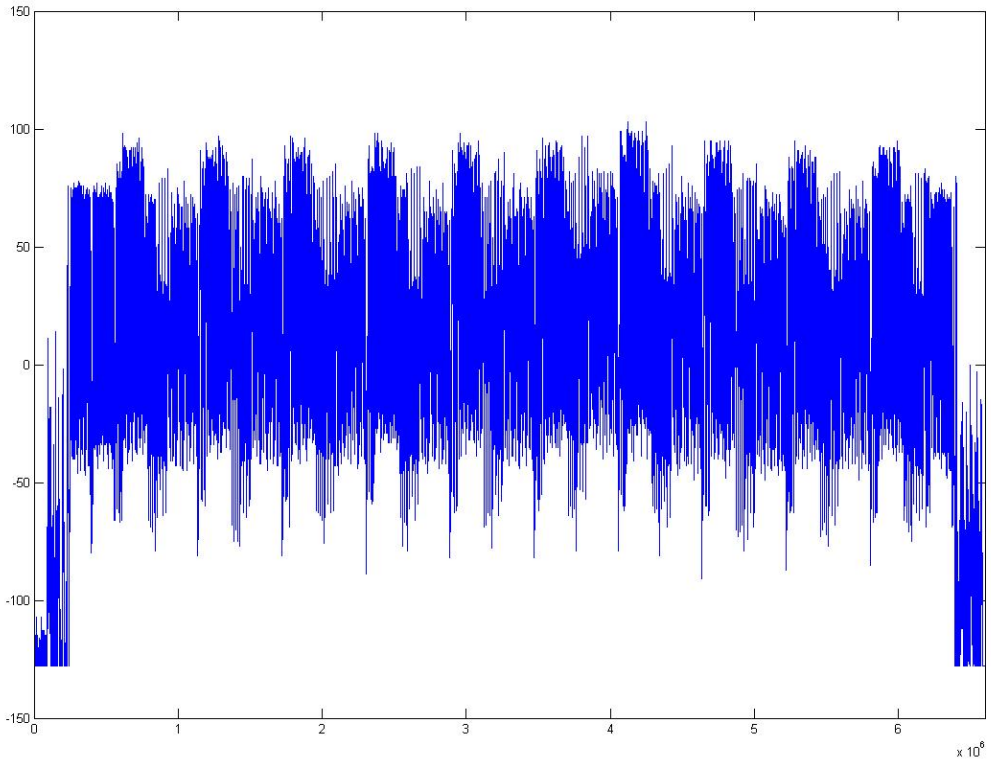
$R_0 \leftarrow 1$
$R_1 \leftarrow M$
for $i \leftarrow x$ **to** 0 **do**
    $R_0 \leftarrow {R_0}^2 \bmod N$
    if $(d_i = 1)$ **then**
        $R_0 \leftarrow R_0 \cdot R_1 \bmod N$
    end
end
return R_0

Simple Power Analysis (SPA)

- Individual operations can potentially be identified.

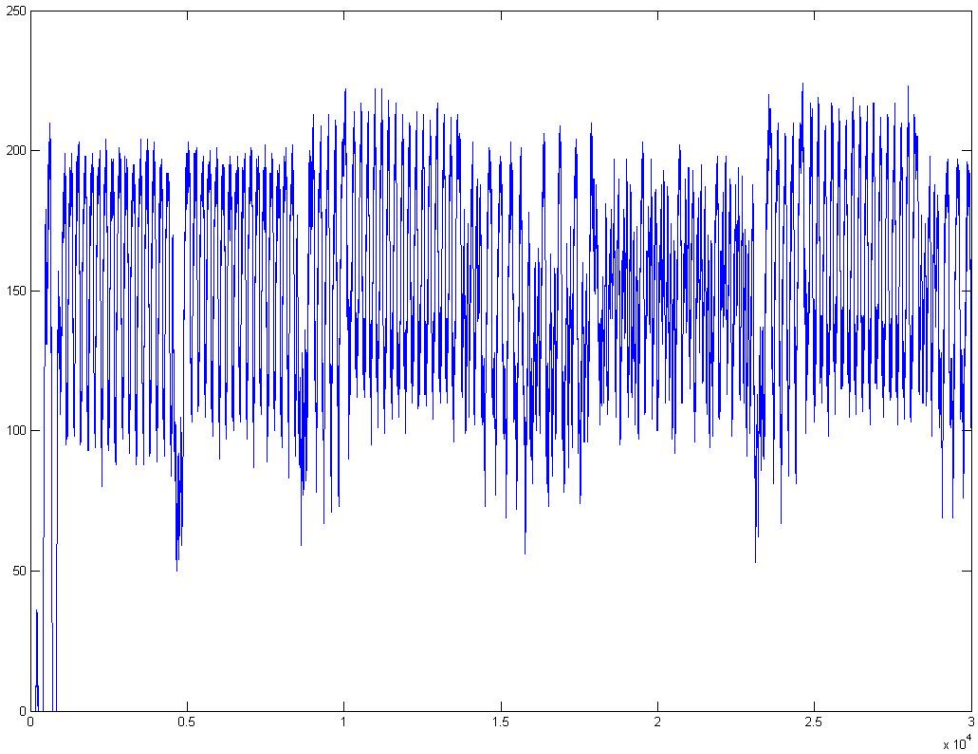


Reverse Engineering with SPA



- For example, cryptographic algorithms can be located in a power consumption trace because of the repeating rounds.
- In this case an implementation of AES on an ARM microprocessor — Nine identical rounds and a shorter tenth round.

Reverse Engineering with SPA



- A closer analysis can determine the functions within a round, e.g.:
  - Two initial permutations to reformat the message and key into a format convenient for calculating.
  - **ByteSub** function (a bytewise substitution), **MixColumn** and key schedule.

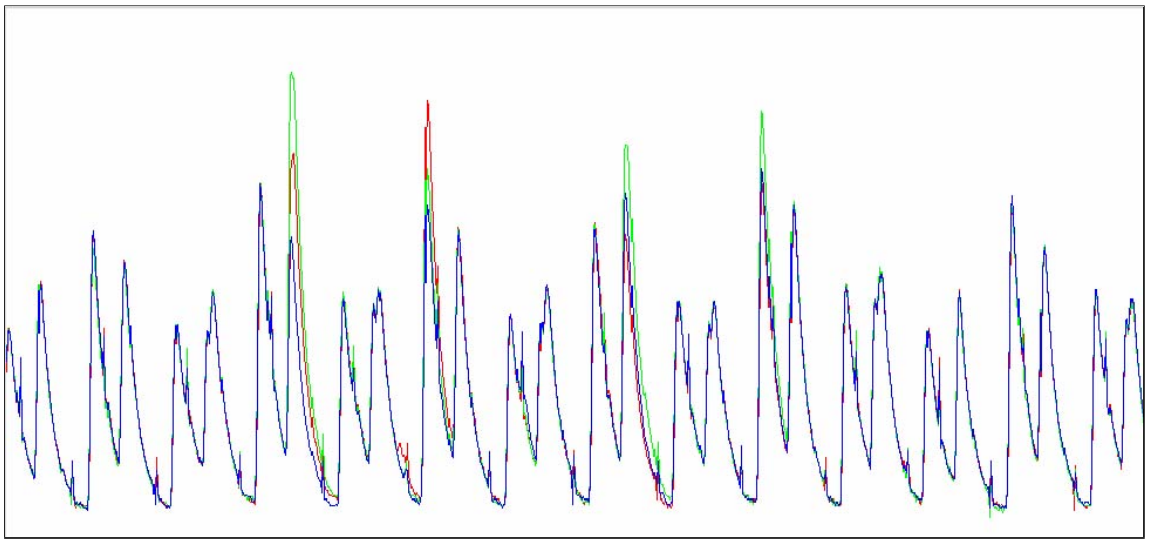
Outline

1. Introduction
   - What is a Smart Card?
   - Why use Smart Cards?
2. Measuring the Power Consumption
   - The Experimental Setup
3. Simple Power Analysis
   - Attacking an Algorithm
   - Attacking an Algorithm
   - Reverse Engineering
4. Differential Power Analysis
   - Correlation Power Analysis
   - Using the Partial Correlation
   - Case Study: The DES block cipher
5. Fault Analysis
   - Case Study: The DES block cipher
6. Countermeasures
7. Other Problems
8. Conclusion

Differential Power Analysis

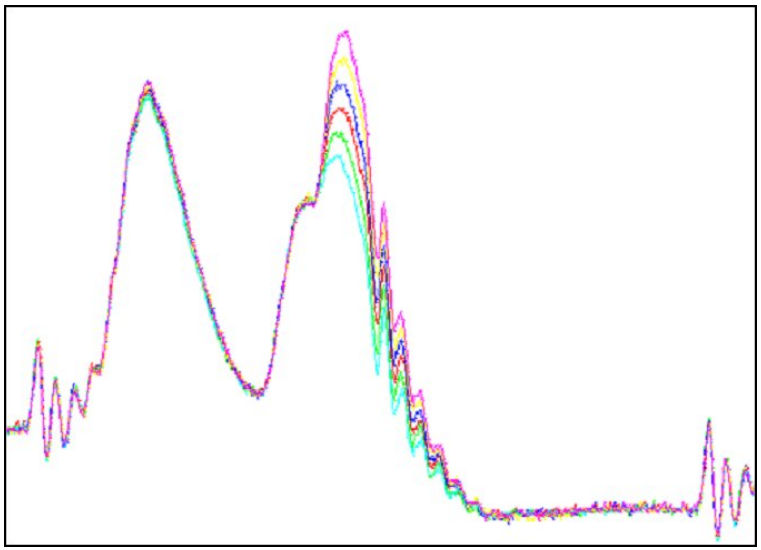
- A statistical analysis of power consumption traces can be conducted with a series of acquisitions.
  - Differential Power Analysis is often used as a generic term for any treatment involving more than one trace.
- A series of acquisitions will result in a series of traces and corresponding messages and ciphertexts.

01	B688EE57BB63E03EC031A0392DC881E6
02	185C881E64D7751A0392DC887509F36F
03	EE2DC88750957B673B63185C881E64E0
⋮	⋮



Differential Power Analysis

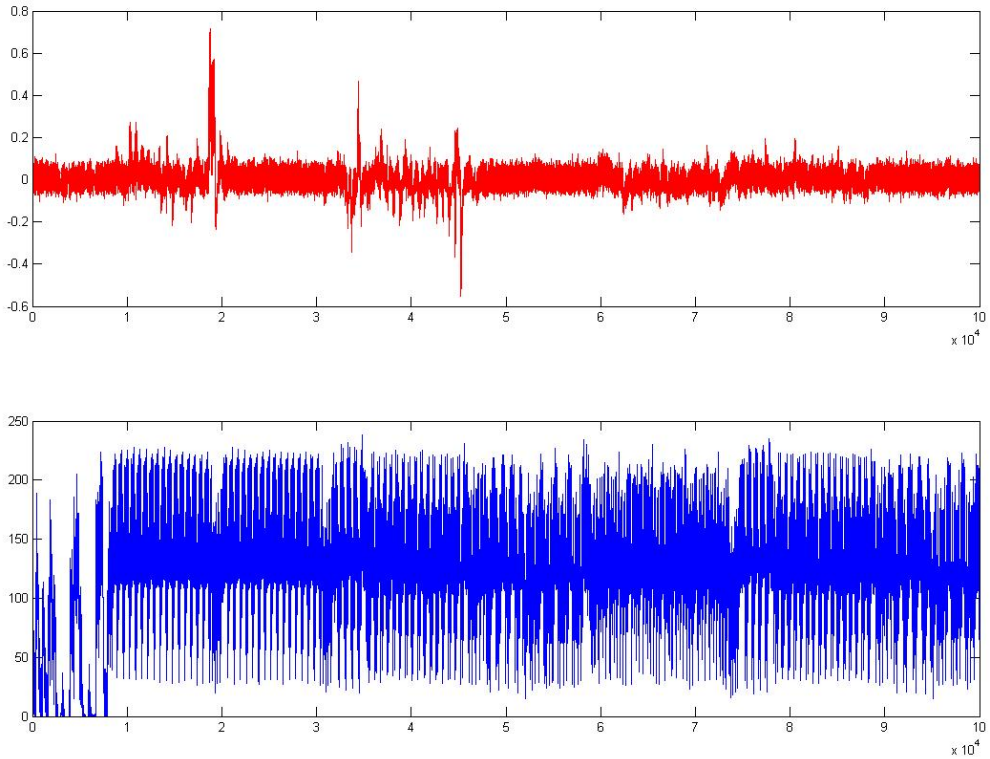
- Looking closely at superposed traces, small differences can be observed.



- Where the difference is either:
  - Proportional to the Hamming weight of the data being manipulated (Hamming weight model).
  - Proportional to the Hamming weight of the data being manipulated XORed with some unknown constant previous state (Hamming distance model).

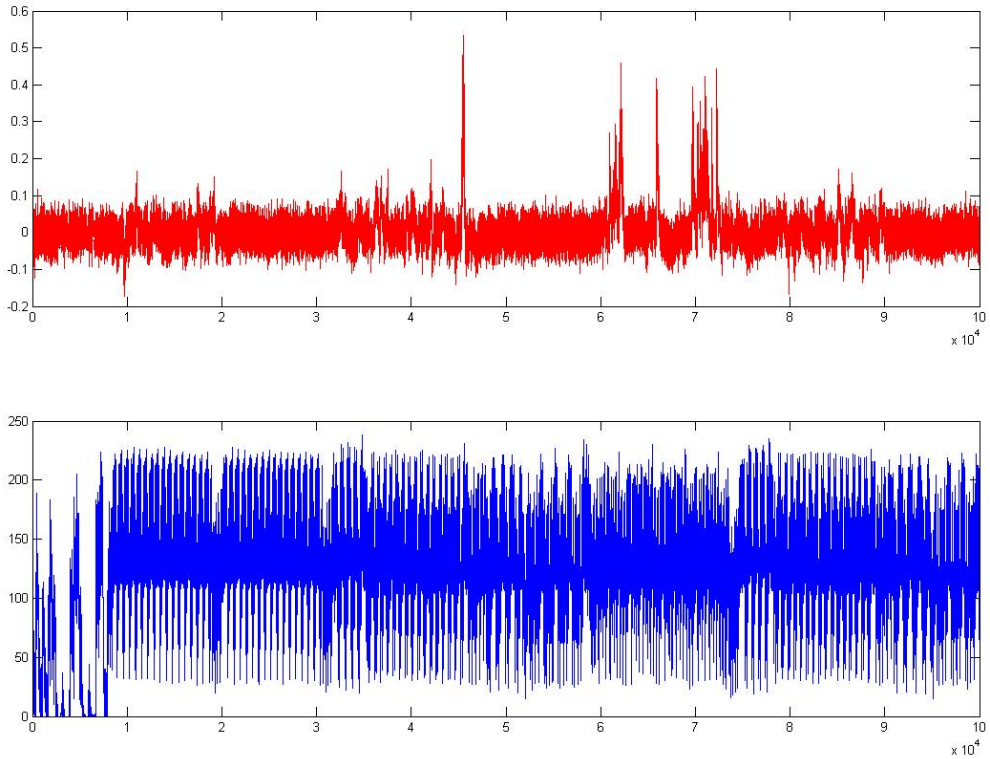
Correlation Power Analysis

- Where a given byte of a message is manipulated can be determined by calculating the correlation between that byte and the instantaneous power consumption.
- For example correlating the first byte of 1000 random plaintexts enciphered using AES:



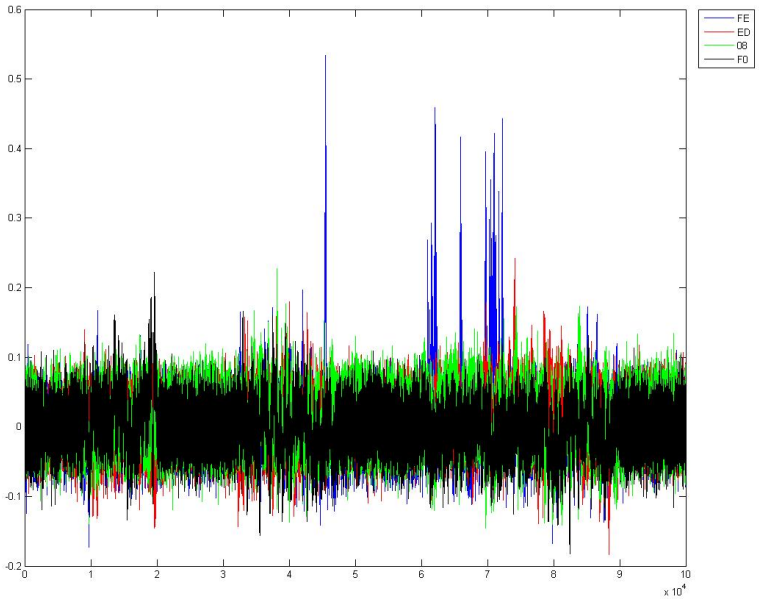
Correlation Power Analysis

- To attack an implementation of a cryptographic algorithm using the correlation, one needs to predict the data being manipulated by the device during the computation of the algorithm.
- Knowing the secret key the output of one byte of the **ByteSub** can be computed and a correlation trace generated.



Correlation Power Analysis

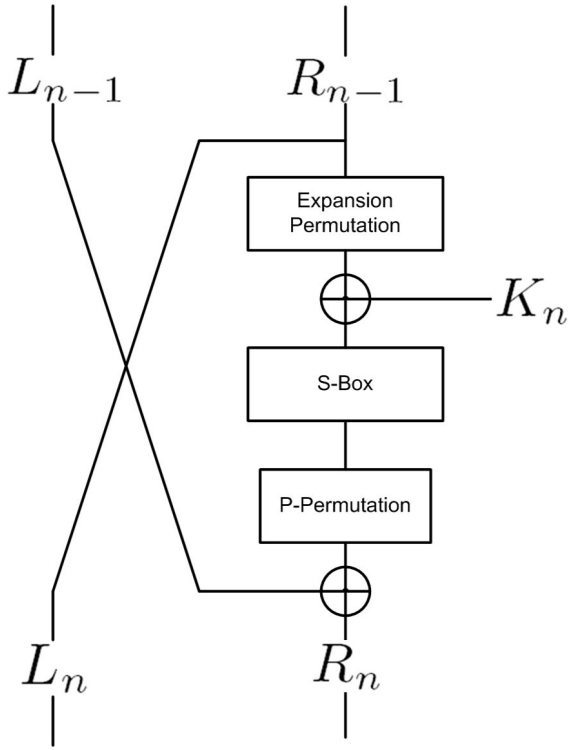
- If the key is unknown, all possible key values that affect the first byte **ByteSub** need to be considered, i.e. one key byte.
- A correlation trace can be generated for the each possible value of the key byte.
  - A trace will also be necessary for each previous state if the device conforms to the Hamming distance model.
- The correct hypothesis should give the largest correlation.



Using the Partial Correlation

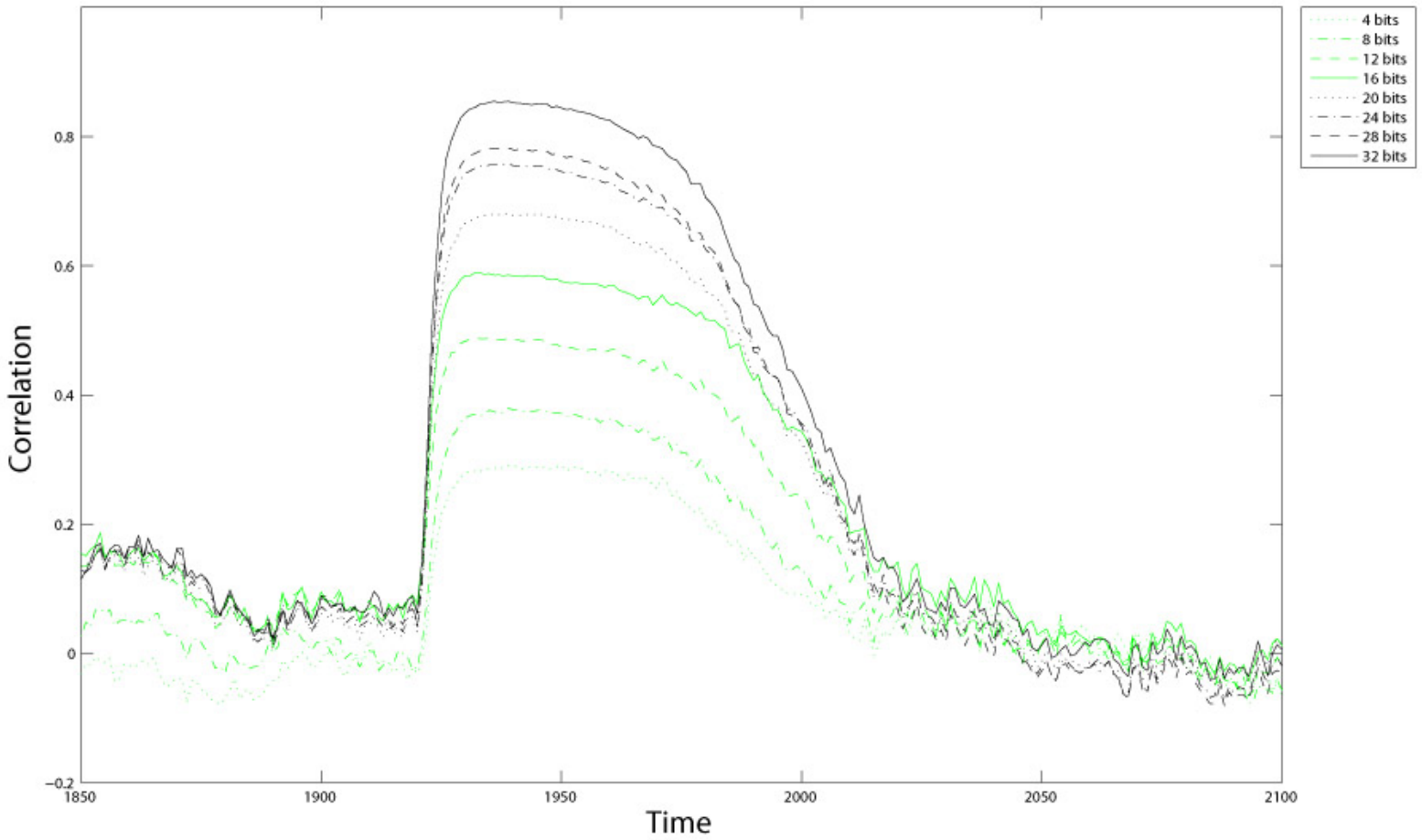
- An attacker is obliged to predict a machine word that is being manipulated to be sure of their results.
- This is prohibitively time consuming for platforms with large word sizes, e.g. 32-bit platforms, FPGA implementations.
- The partial correlation can be used to determine portions of the data being manipulated to eliminate certain hypotheses.
- For a hardware DES implementation on a smart card the first 48-bit subkey can be determined by correlating with the 32-bit word produced by the output of the S-boxes (traces donated by Gemalto).

Case Study: The DES block cipher



- The DES round function:
  - K_n is 48-bits.
  - Reduced to 32-bits after the S-box function.
  - In hardware the S-box function can be applied to the 48-bits at the same point in time.

Case Study: The DES block cipher

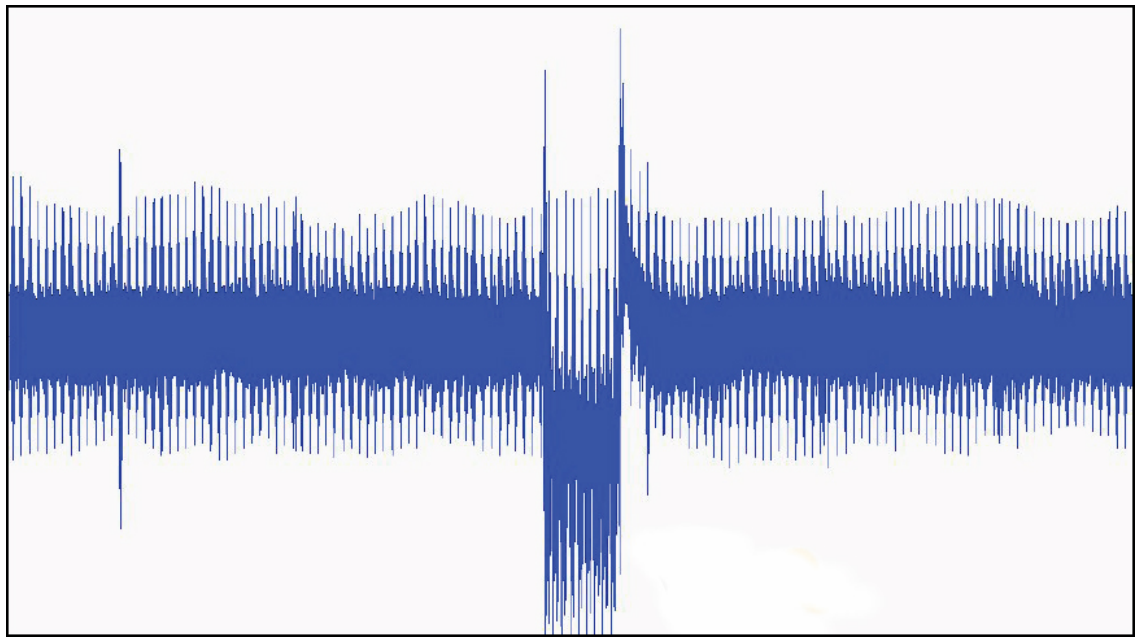


Outline

1. Introduction
   - What is a Smart Card?
   - Why use Smart Cards?
2. Measuring the Power Consumption
   - The Experimental Setup
3. Simple Power Analysis
   - Attacking an Algorithm
   - Attacking an Algorithm
   - Reverse Engineering
4. Differential Power Analysis
   - Correlation Power Analysis
   - Using the Partial Correlation
   - Case Study: The DES block cipher
5. Fault Analysis
   - Case Study: The DES block cipher
6. Countermeasures
7. Other Problems
8. Conclusion

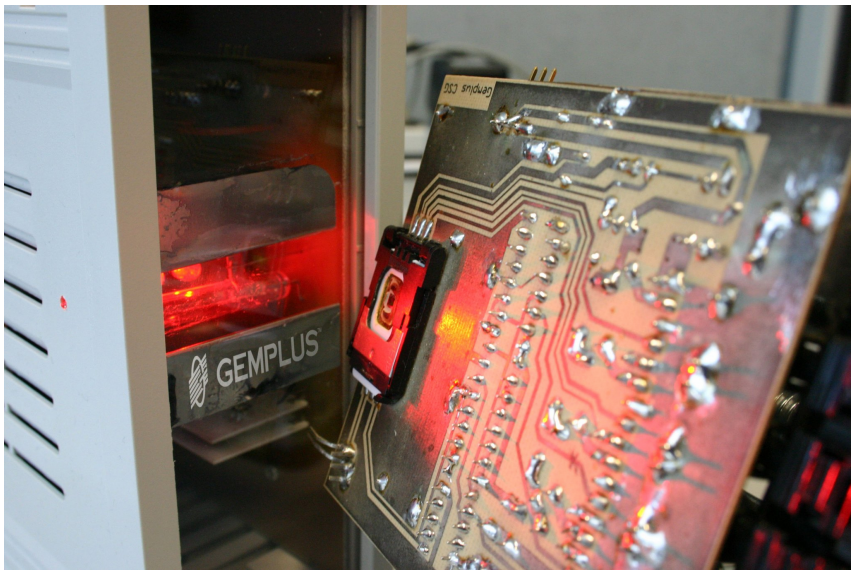
Fault Analysis

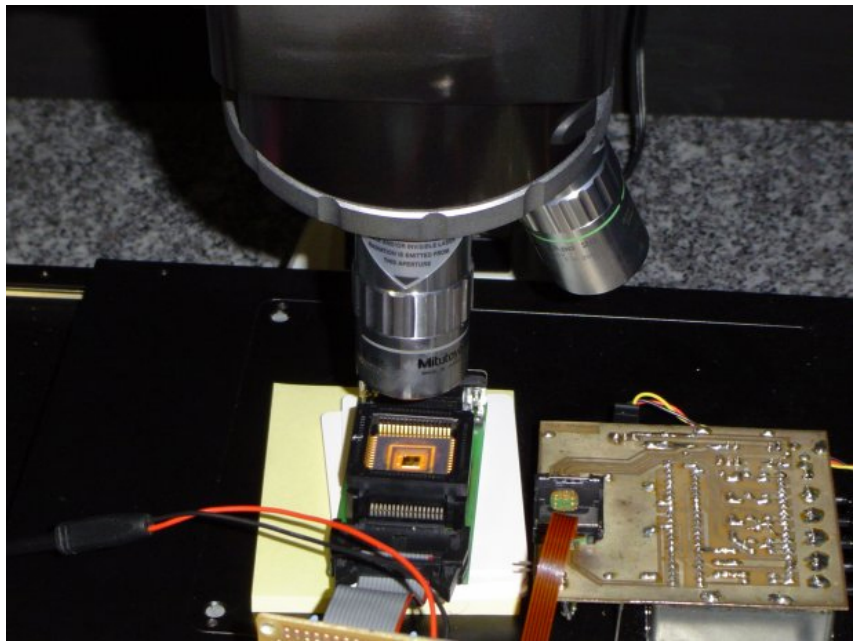
- "Jolt" the smart card off its normal processing.
- Exploit any information that might be revealed.
- E.g. glitches.



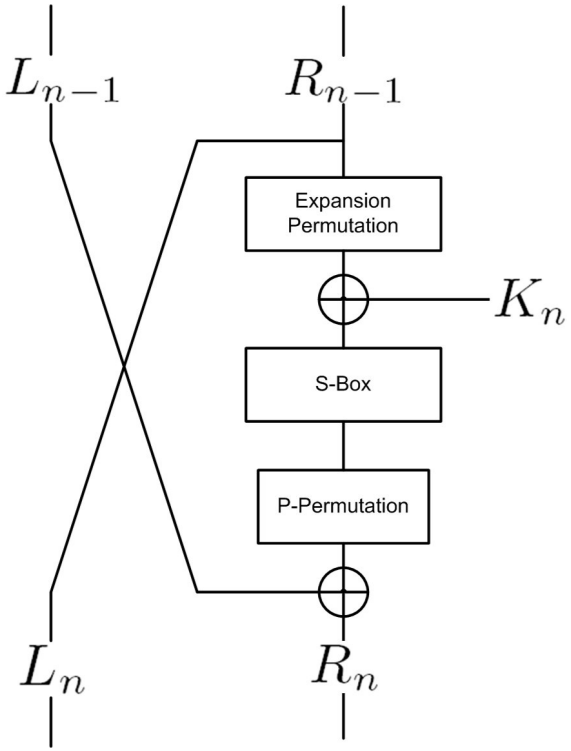
Fault Analysis

- or white light flashes or laser light.





Case Study: The DES block cipher



- If we consider the last round.

$$R_{16} = S(R_{15} \oplus K_{16}) \oplus L_{15} = S(L_{16} \oplus K_{16}) \oplus L_{15}$$

Case Study: The DES block cipher

- If a fault occurs in R_{15} to produce R'_{15}.

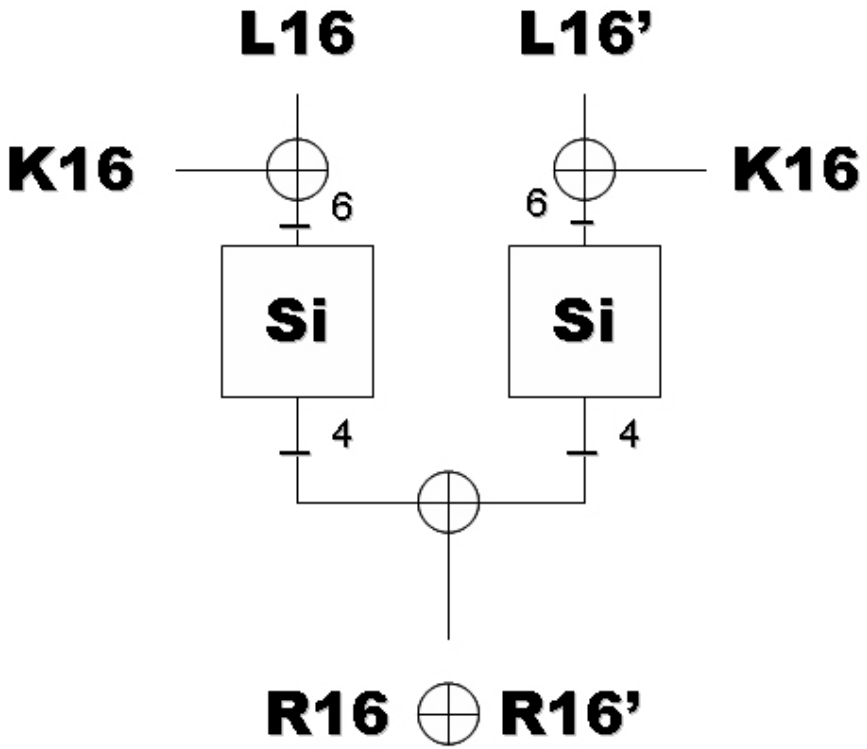
$$\begin{aligned} R_{16} &= S(R_{15} \oplus K_{16}) \oplus L_{15} \\ &= S(L_{16} \oplus K_{16}) \oplus L_{15} \end{aligned} \qquad \begin{aligned} R'_{16} &= S(R'_{15} \oplus K_{16}) \oplus L_{15} \\ &= S(L'_{16} \oplus K_{16}) \oplus L_{15} \end{aligned}$$

- The results can be compared by XORing R_{16} with R'_{16}.

$$\begin{aligned} R_{16} \oplus R'_{16} &= S(L_{16} \oplus K_{16}) \oplus L_{15} \oplus S(L'_{16} \oplus K_{16}) \oplus L_{15} \\ &= S(L_{16} \oplus K_{16}) \oplus S(L'_{16} \oplus K_{16}) \end{aligned}$$

- Where the only unknown is the key K_{16}.

Case Study: The DES block cipher



- Gives a list of possible key values 2^{27}.
- Leads to an exhaustive search.
- Number of hypotheses can be further reduced by injecting a fault to produce another R'_{16}.

Outline

1 Introduction
- What is a Smart Card?
- Why use Smart Cards?

2 Measuring the Power Consumption
- The Experimental Setup

3 Simple Power Analysis
- Attacking an Algorithm
- Attacking an Algorithm
- Reverse Engineering

4 Differential Power Analysis
- Correlation Power Analysis
- Using the Partial Correlation
- Case Study: The DES block cipher

5 Fault Analysis
- Case Study: The DES block cipher

6 Countermeasures

7 Other Problems

8 Conclusion

Countermeasures

- The principles of the countermeasures used to prevent the attacks presented are well understood.
- In the case of fault attacks the countermeasure consists of redundance, and sensors to detect attacks.
- The most basic side channel countermeasure is to implement everything such that an algorithm is computed in constant time, and uses the same code irrespective of the data entered, i.e. no data dependent branching.
- A common countermeasure for removing the correlation between known, or guessable, information and the power consumption is data masking.
  - I.e. the function,

$$y = f(x)$$

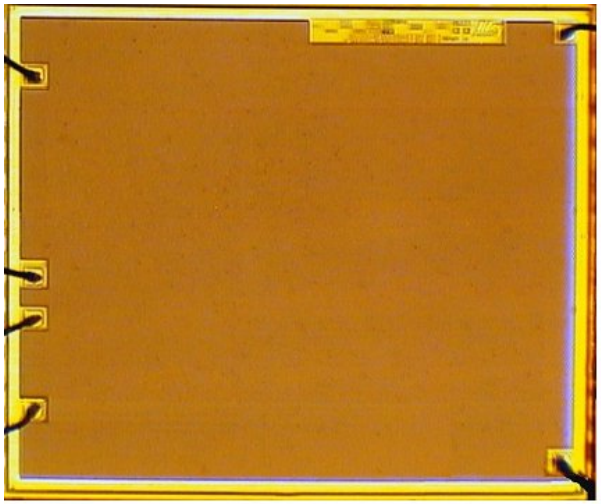
can be replaced with

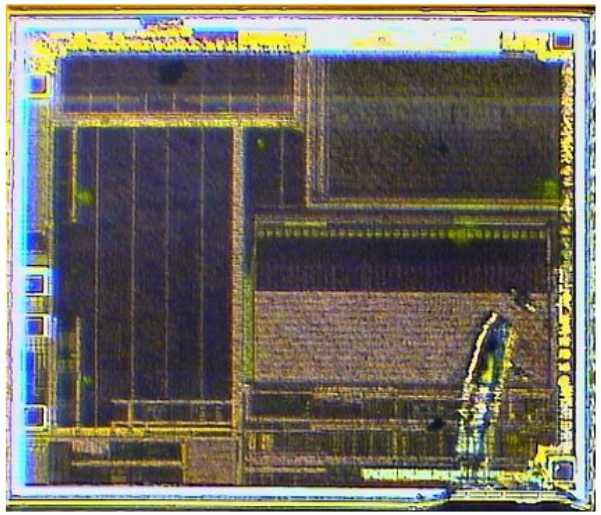
$$y \oplus r = f'(x \oplus r) \text{ or, alternatively}$$
$$y \oplus r = f'(x \oplus r, r)$$

where r is a random value generated each time f' is computed.

Countermeasures

- All the hardware countermeasures are beyond the scope of this presentation.





- A physical shield can be used to prevent inspection and reverse engineering.

Countermeasures

- In the case of RSA, this takes the form.

Algorithm 2: Randomised Exponentiation Algorithm

Input: $M, d, N,$ small random values r_1, r_2, r_3
Output: $C = M^d \bmod N$

$M' \leftarrow M + r_1 \cdot N$
$d' \leftarrow d + r_2 \cdot \lambda(N)$
$N' \leftarrow r_3 \cdot N$
$C' \leftarrow M'^{d'} \bmod N'$
$C \leftarrow C' \bmod N$

return C

where λ is Euler's Totient function.

Countermeasures

- Other countermeasures can also be included to make life as difficult for the attacker as possible.
- The order that a given function processes information can be randomised, e.g. randomising the order bytes are accessed in the **ByteSub** function of AES.
  - This also prevents some higher-order attacks that attempt to remove data masking by comparing different points in the same trace, but these attacks are largely theoretical.
- Random delays in either software (dummy loops) or hardware (dummy clock cycles) can be used to force an attacker to resynchronise their acquisitions *a posteriori*.
  - This is a relatively trivial task but can be time consuming.

Outline

1. Introduction
   - What is a Smart Card?
   - Why use Smart Cards?
2. Measuring the Power Consumption
   - The Experimental Setup
3. Simple Power Analysis
   - Attacking an Algorithm
   - Attacking an Algorithm
   - Reverse Engineering
4. Differential Power Analysis
   - Correlation Power Analysis
   - Using the Partial Correlation
   - Case Study: The DES block cipher
5. Fault Analysis
   - Case Study: The DES block cipher
6. Countermeasures
7. Other Problems
8. Conclusion

Wrong Cryptographic Design

- Short keys.
- Weak algorithms.
- Broken protocols
- Examples
  - French Carte Bleu
  - COMP128 used in GSM

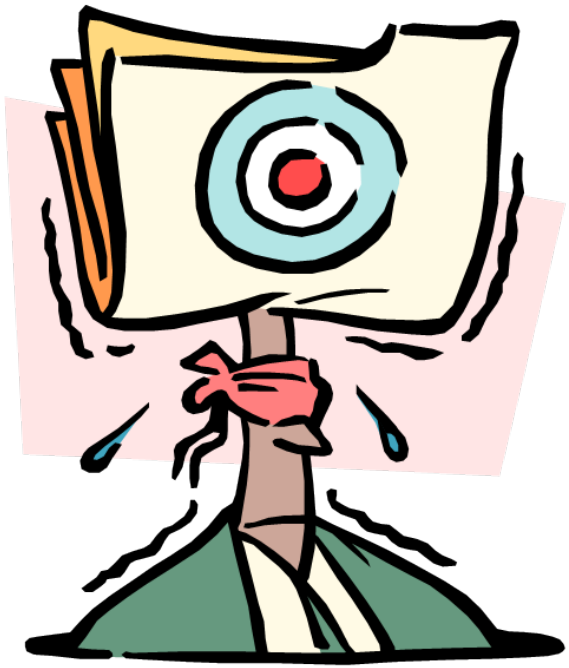
Outline

1. Introduction
   - What is a Smart Card?
   - Why use Smart Cards?
2. Measuring the Power Consumption
   - The Experimental Setup
3. Simple Power Analysis
   - Attacking an Algorithm
   - Attacking an Algorithm
   - Reverse Engineering
4. Differential Power Analysis
   - Correlation Power Analysis
   - Using the Partial Correlation
   - Case Study: The DES block cipher
5. Fault Analysis
   - Case Study: The DES block cipher
6. Countermeasures
7. Other Problems
8. Conclusion

Conclusion

- Smart cards are like any security devices: they have limitations.
- A system should be designed with these limitations in mind.
- A system should be upgradeable to deal with the (inevitable?) security breach or the aging of the technology

Comments/Questions?



http://www.geocities.com/mike.tunstall/