SIDE-CHANNEL ATTACKS ON HARDWARE IMPLEMENTATIONS OF CRYPTOGRAPHIC ALGORITHMS

Sıddıka Berna Örs Yalçın

Istanbul Technical University
Department of Electronics and Communication Engineering

Introduction

A side-channel analysis attack takes advantage of implementation specific characteristics

Divided into two groups as

- active (tamper attacks): the attacker has to reach the internal circuitry of the cryptographic device
  - probing attack [38]: inserting sensors into the device
  - fault induction attack [6, 32]: disturbing the device's behavior
- passive [34]: The physical and/or electrical effects of the functionality of the device are used for the attack

Passive Attacks

If physical and/or electrical effects unintentionally deliver information about the key, then they deliver side-channel information and are called side-channels.

Four groups according to the side-channel information that they exploit:

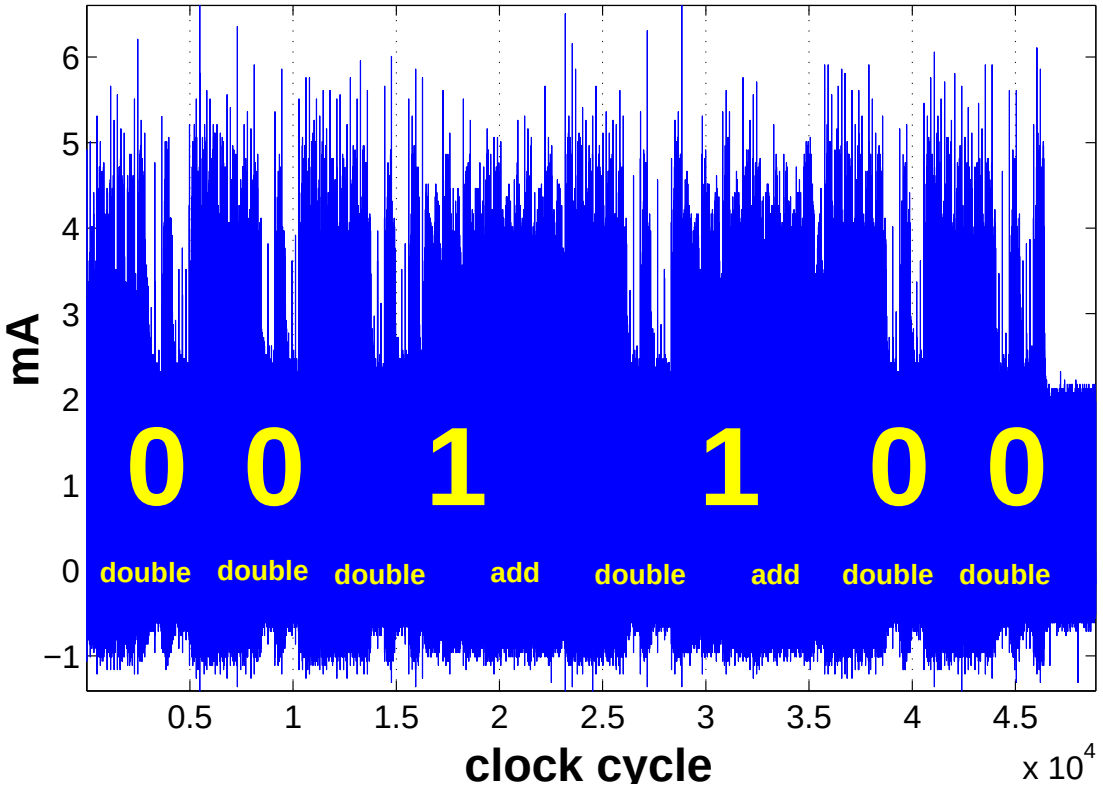
- Timing attacks (TA) [34, 18, 24, 26, 22, 37, 58, 55, 56, 60, 4, 8, 10, 42, 31]
- Power attacks (PA) [7, 35, 5, 21, 36, 45, 46, 1, 12, 14, 25, 48, 3, 28, 39, 41, 44, 47, 52, 19, 40, 42, 51, 49, 63, 50, 61, 62, 31]
- Electromagnetic attacks (EMA) [20, 54, 9, 15, 16, 27]
- Acoustic (sound) analysis [59]

All the groups of the passive attacks have two types:

- simple
- differential analysis

Simple Attacks

An attacker uses the side-channel information from one measurement directly to determine (parts of) the secret key. A simple analysis attack exploits the relationship between the executed operations and the side-channel information.



Differential Attacks

Many measurements are used in order to filter out noise. A differential analysis attack exploits the relationship between the processed data and the side-channel information.

- hypothetical model of the attacked device: The model is used to predict several values for the side-channel information of a device.
- These predictions are compared to the real, measured side-channel information of the device. Comparisons are performed by applying statistical methods on the data.

Distance of Mean Test

1. Run the cryptographic algorithm for N random values of input.
2. For each of the N inputs, I_i, a discrete time side-channel signal, $S_i[j]$, is collected and the corresponding output, O_i, may also be collected.
3. The $S_i[j]$ are split into two sets using a partitioning function, $D(\cdot)$:
   $S_0 = \{S_i[j] \,|D(\cdot) = 0\}$
   $S_1 = \{S_i[j] \,|D(\cdot) = 1\}$
4. Compute the average side-channel signal for each set:
   $A_0[j] = \frac{1}{|S_0|} \sum_{S_i[j] \in S_0} S_i[j]$
   $A_1[j] = \frac{1}{|S_1|} \sum_{S_i[j] \in S_1} S_i[j]$
   where $|S_0| + |S_1| = N$.
5. subtracting the two averages, a discrete time differential side-channel bias signal, $T[j]$, is obtained: $T[j] = A_0[j] - A_1[j]$.

Correlation Analysis

1. The model predicts the amount of side-channel information for a certain moment of the execution.

2. These predictions are correlated to the real side-channel information.

This correlation can be measured with the Pearson correlation coefficient [11].

$C(T,P) = \frac{E(T \cdot P) - E(T) \cdot E(P)}{\sqrt{Var(T) \cdot Var(P)}} \quad -1 \leq C(T,P) \leq 1.$

T and P are said to be uncorrelated, if $C(T,P)$ equals zero. Otherwise, they are said to be correlated.

If their correlation is high, *i.e.*, if $C(T,P)$ is close to $+1$ or -1, it is usually assumed that the prediction of the model, and thus the key hypothesis, is correct.

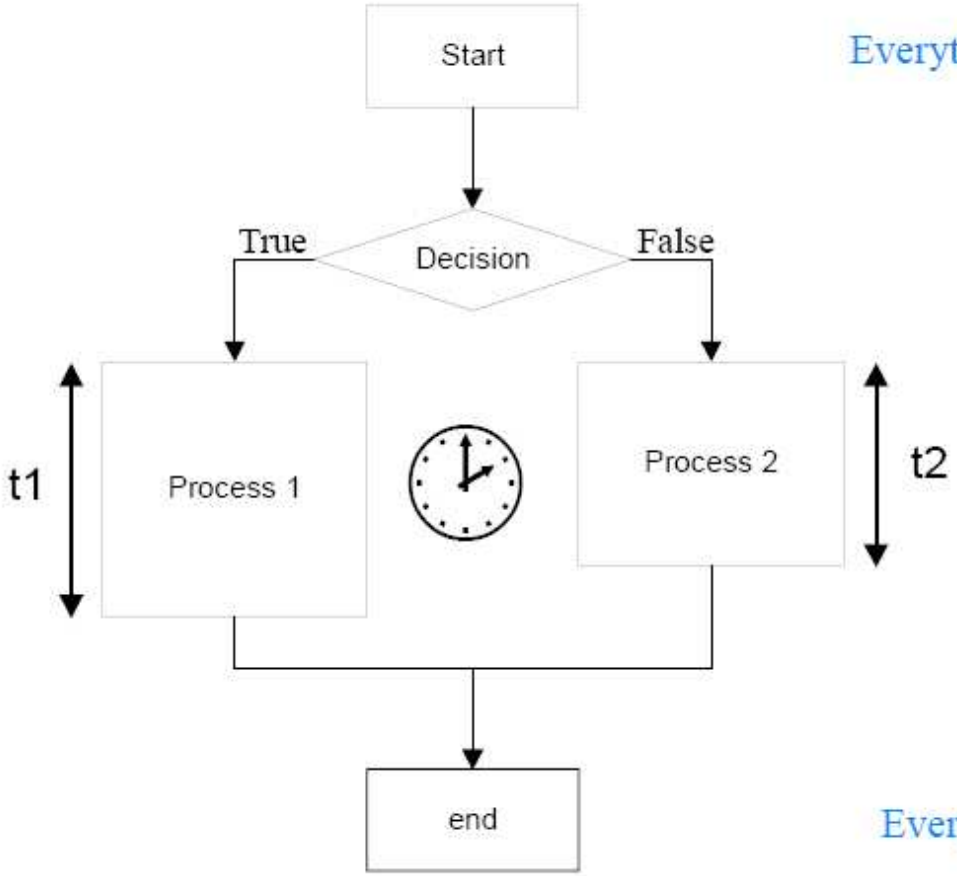
Timing Attacks

- The term "Timing Attack" was first introduced at CRYPTO'96 in Paul Kocher's paper
- Few other theoretical approaches without practical experiments up to the end of '97
- GEMPLUS put theory into practice in early'98

What are Timing Attacks? (1/2)

- Principle of Timing Attacks:
  - Secret data are processed in the card
  - Processing time
    - depends on the value of the secret data
    - leaks information about the secret data
    - can be measured (or at least their differences)
- Practical attack conditions
  - Possibility to monitor the processing of the secret data
  - Have a way to record processing duration
  - Have basic computational & statistical tool
  - Have some knowledge of the implementation

What are Timing Attacks? (2/2)



Everything performed unconditionally before the test

A test based on secret data is performed
that leads to a boolean decision

Depending on the boolean condition,
the process may be long (t1) or short (t2)

Everything performed unconditionally after the test

Simple Timing Attack on an FPGA implementation of an Elliptic Curve Cryptosystem (1/3)

The basic operation for ECC algorithms is point multiplication: $Q = [k]P$.

Require: EC point $P = (x, y)$, integer k, $0 < k < M$, $k = (k_{\ell-1}, k_{\ell-2}, \cdots, k_0)_2$, $k_{\ell-1} = 1$ and M
Ensure: $Q = [k]P = (x', y')$
1: $Q \leftarrow P$
2: **for** i from $\ell - 2$ downto 0 **do**
3: $\quad Q \leftarrow 2Q$
4: $\quad$ **if** $k_i = 1$ **then**
5: $\quad\quad Q \leftarrow Q + P$
6: $\quad$ **end if**
7: **end for**

STA on an FPGA implementation of an ECC (2/3)

Elliptic curve point addition over $GF(p)$

Input: $P_1 = (x, y, 1, a)$, $P_2 = (X_2, Y_2, Z_2, V_2)$

Output: $P_1 + P_2 = P_3 = (X_3, Y_3, Z_3, V_3)$

1.	$T_1 \leftarrow Z_2 * Z_2$	
2.	$T_2 \leftarrow x * T_1$	
3.	$T_1 \leftarrow T_1 * Z_2$	$T_3 \leftarrow X_2 - T_2$
4.	$T_1 \leftarrow y * T_1$	
5.	$T_4 \leftarrow T_3 * T_3$	$T_5 \leftarrow Y_2 - T_1$
6.	$T_2 \leftarrow T_2 * T_4$	
7.	$T_4 \leftarrow T_4 * T_3$	$T_6 \leftarrow T_2 + T_2$
8.	$Z_3 \leftarrow Z_2 * T_3$	$T_6 \leftarrow T_4 + T_6$
9.	$T_3 \leftarrow T_5 * T_5$	
10.	$T_1 \leftarrow T_1 * T_4$	$X_3 \leftarrow T_3 - T_6$
11.	$V_3 \leftarrow Z_3 * Z_3$	$T_2 \leftarrow T_2 - X_3$
12.	$T_3 \leftarrow T_5 * T_2$	
13.	$V_3 \leftarrow V_3 * V_3$	$Y_3 \leftarrow T_3 - T_1$
14.	$V_3 \leftarrow a * V_3$	

Elliptic curve point doubling over $GF(p)$

Input: $P_1 = (X_1, Y_1, Z_1, V_1)$

Output: $2P_1 = P_3 = (X_3, Y_3, Z_3, V_3)$

1.	$T_1 \leftarrow Y_1 * Y_1$	$T_2 \leftarrow X_1 + X_1$
2.	$T_3 \leftarrow T_1 * T_1$	$T_2 \leftarrow T_2 + T_2$
3.	$T_1 \leftarrow T_2 * T_1$	$T_3 \leftarrow T_3 + T_3$
4.	$T_2 \leftarrow X_1 * X_1$	$T_3 \leftarrow T_3 + T_3$
5.	$T_4 \leftarrow Y_1 * Z_1$	$T_3 \leftarrow T_3 + T_3$
6.	$T_5 \leftarrow T_3 * V_1$	$T_6 \leftarrow T_2 + T_2$
7.		$T_2 \leftarrow T_6 + T_2$
8.		$T_2 \leftarrow T_2 + V_1$
9.	$T_6 \leftarrow T_2 * T_2$	$Z_3 \leftarrow T_4 + T_4$
10.		$T_4 \leftarrow T_1 + T_1$
11.		$X_3 \leftarrow T_6 - T_4$
12.		$T_1 \leftarrow T_1 - X_3$
13.	$T_2 \leftarrow T_2 * T_1$	$V_3 \leftarrow T_5 + T_5$
14.		$Y_3 \leftarrow T_2 - T_3$

STA on an FPGA implementation of an ECC (3/3)

- The total execution time of an EC point addition is $14T_*$.
- The total execution time of an EC point doubling is $8T_* + 6T_\pm$.
- The latency of one point multiplication:
  $T_{PMUL} = (\ell-1)T_{PDB} + (w-1)T_{PAD} = (8\ell + 14w - 22)\,T_* + 6(\ell-1)T_\pm$

Somebody who knows the execution time of one '*' and '$\pm$' and can measure the execution time of one 160-bit elliptic curve point multiplication will learn the Hamming weight of the key.

Countermeasure for STA

Require: EC point $P = (x, y)$, integer k, $0 < k < M$, $k = (k_{\ell-1}, k_{\ell-2}, \cdots, k_0)_2$, $k_{\ell-1} = 1$ and M
Ensure: $Q = [k]P = (x', y')$

1: $Q \leftarrow P$
2: **for** i from $\ell - 2$ downto 0 **do**
3: $\quad Q_1 \leftarrow 2Q$
4: $\quad Q_2 \leftarrow Q_1 + P$
5: $\quad$ **if** $k_i = 0$ **then**
6: $\quad\quad Q \leftarrow Q_1$
7: $\quad$ **else**
8: $\quad\quad Q \leftarrow Q_2$
9: $\quad$ **end if**
10: **end for**

The latency of one point multiplication:
$T_{PMUL} = (\ell - 1)\,(T_{PDB} + T_{PAD}) = (\ell - 1)\,(22T_* + 6T_\pm)$.

Differential Timing Attack on a Hardware Implementation of AES (1/2)

S-Box Operation in AES

Require: $in = \{in_1 in_0\}$
Ensure: $out = \text{S-Box}\,(in) = \{out_1 out_0\}$
1: **if** $in = \{00\}$ **then**
2: $\quad out = \{00\}$
3: **else**
4: $\quad out = \text{MultInv}\,(in)$
5: **end if**
6: $out = \text{AffTrans}\,(out)$

The input of the first S-Box operation in the first round is the first byte of the output of the
AddRoundKey(Plaintext,Key)=Plaintext$\oplus$Key.

DTA on a Hardware Implementation of AES (2/2)

Step 2 is executed in shorter time than Step 4. The attacker's steps:

1. Feed the hardware with N plaintexts
2. Measure the time which takes for encrypting each of them and form a $N \times 1$ matrix M_1 with these timing data.
3. Calculate $Plaintext_1 \oplus Key_1$ for N plaintexts for each possible 256 values of the first byte of the key and for each plaintext.
4. Form a $N \times 256$ matrix M_2 with the expected time of S-box $(Plaintext_1 \oplus Key_1)$ operation.

Now the attacker should choose a statistical analysis method described for finding the first byte of the key.

If he chooses the correlation analysis, then he should find the correlation between M_1 and each column of M_2. The highest correlation will give the right first byte of the key.

Countermeasures

Executing the operations in constant time independent form the processed data.

- Dhem in [17], Walter in [64, 65] and Hachez and Quisquater in [23] propose several countermeasures that typically consist of removing the time variation in Montgomery Multiplication.
- Kocher suggests a countermeasure consist of randomizing the exponent in RSA by adding a random multiple of $\varphi(n)$, a modification that does not effect the final result in [34].
- Using double and add always algorithm proposed by Coron in [13] during the elliptic curve point multiplication allows to hide the Hamming weight of the keys.
- Izu and Takagi propose the binary right to left point multiplication algorithm by executing point addition and doubling in parallel in [29, 30] for ECC.

Is There a Future for Timing Attacks?

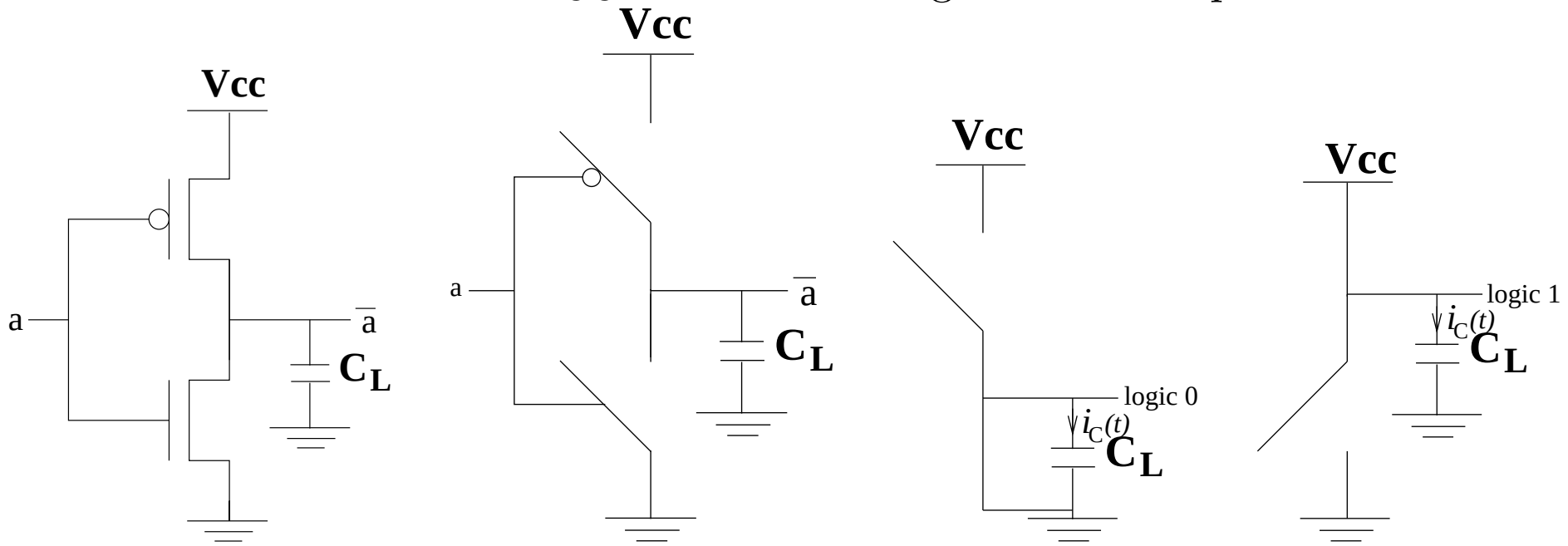
- Associated with other side-channels, it becomes far more efficient
  - Global measurements are replaced by local ones
- Timing attacks are still an important threat
  - Against existing devices applied to secret management
  - Not only a smart cards issue
  - Designers have to think about it
- Solutions exist

Power Attacks

The dominating factor for the power consumption of a CMOS gate is the dynamic power consumption [33]:

$P_D = C_L V_{DD}^2 P_{0\to 1} f$

The current absorbed from V_{CC} is used to charge the load capacitor C_L.



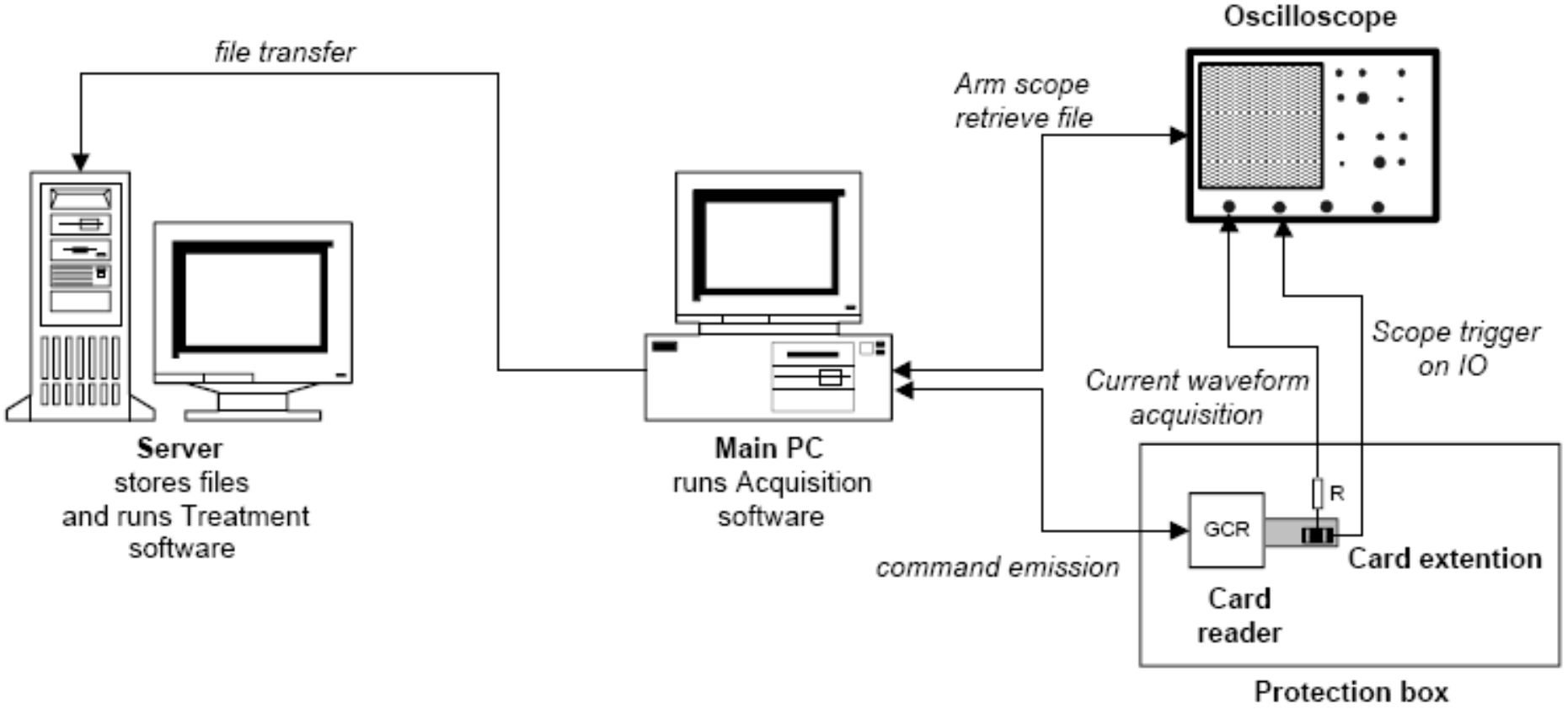
The voltage on the load capacitor is the output level of the inverter either logic 0 (V_{CC} V) or 1 (0 V).

Power Attacks

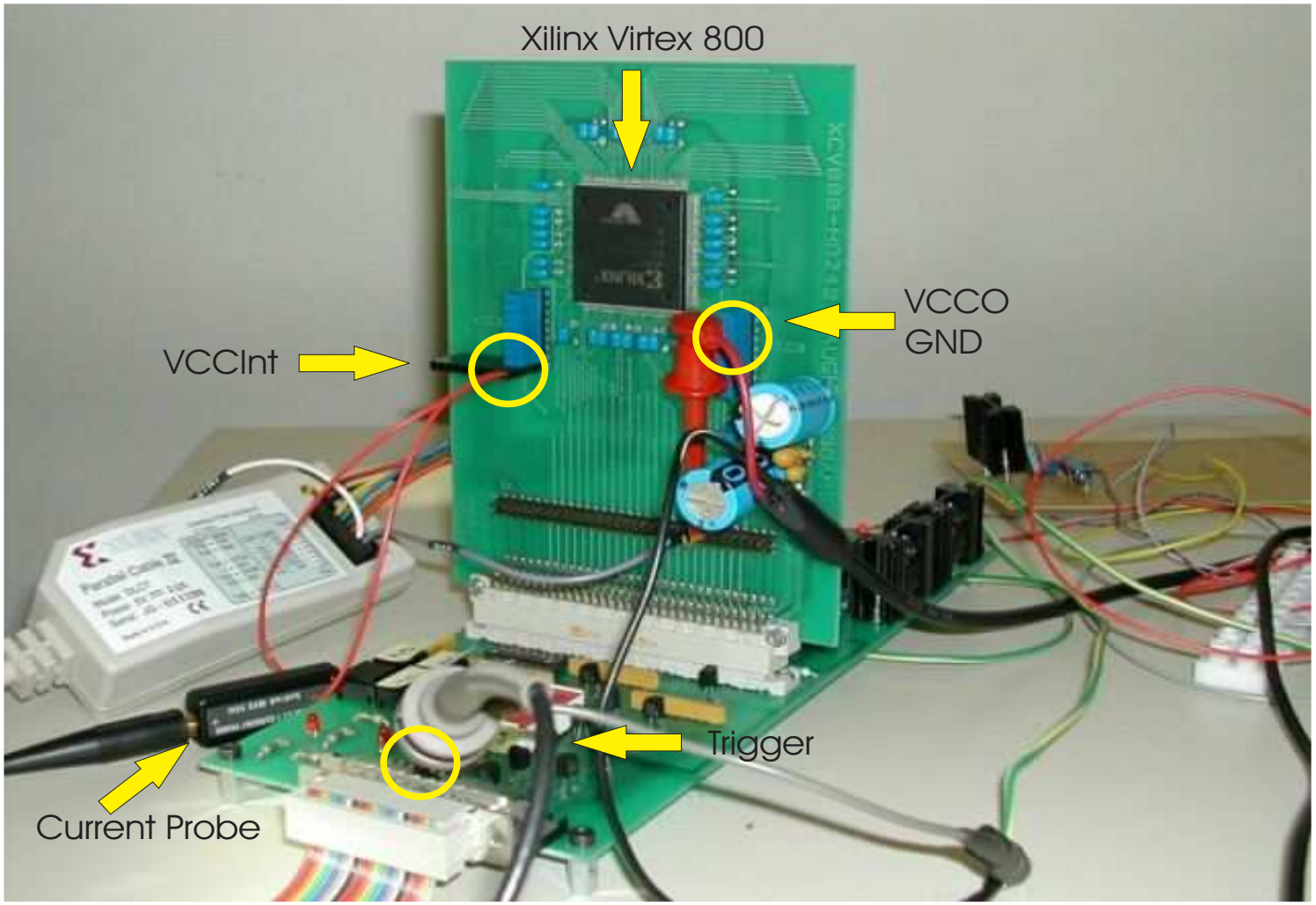
The current-voltage relation of a capacitor is defined as:
$i_C(t) = C\frac{d}{dt}v(t)$

- We will observe a current only during the $0 \rightarrow 1$ transition at the output of the inverter.
- This transition depends on the input of the inverter, so the processed data in the gate.
- By observing the current consumption of a gate we can learn some information about the processed data.
- If this data has some relation with the secret information than we gain some information about the secret.

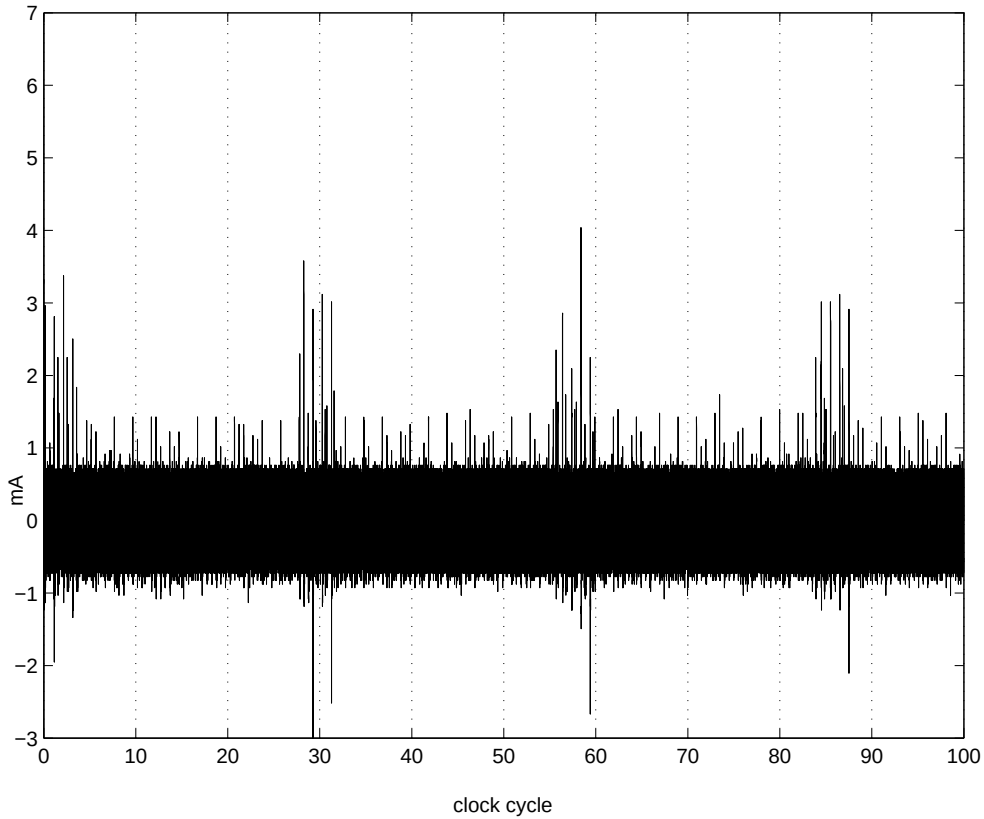
Measurement Setup (1/2)



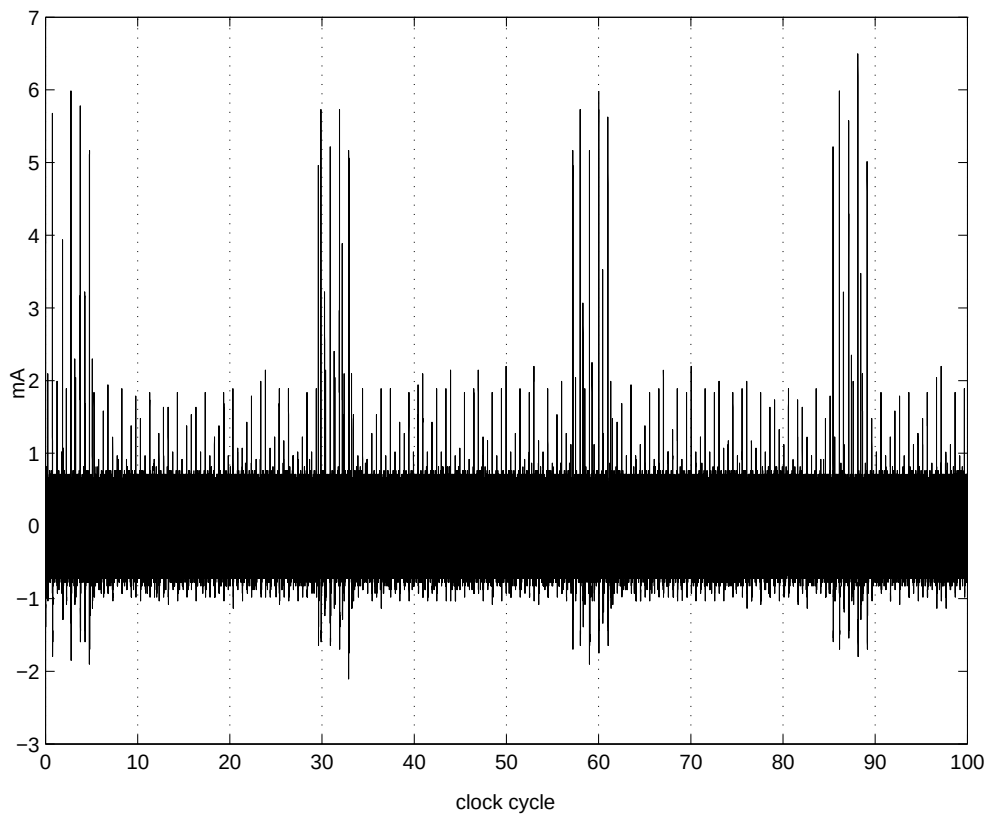
Measurement Setup (2/2)



Information Leakage



Power consumption trace of a 3000-bit register.

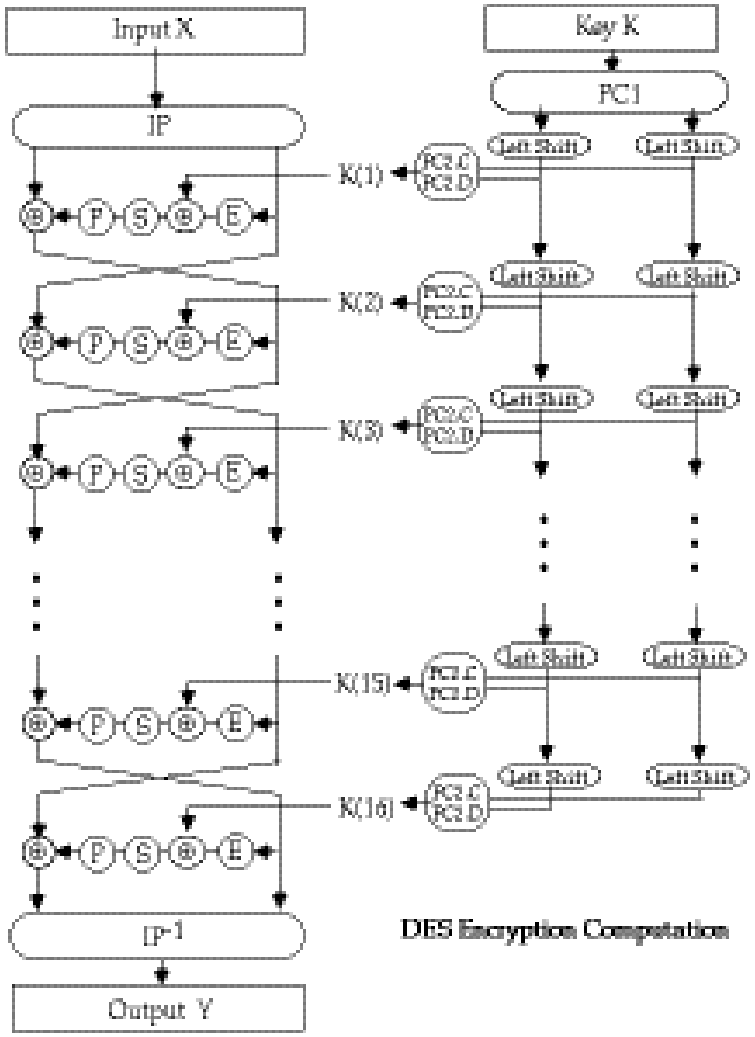


Power consumption trace of a 6000-bit register.

Simple Power Analysis Attack

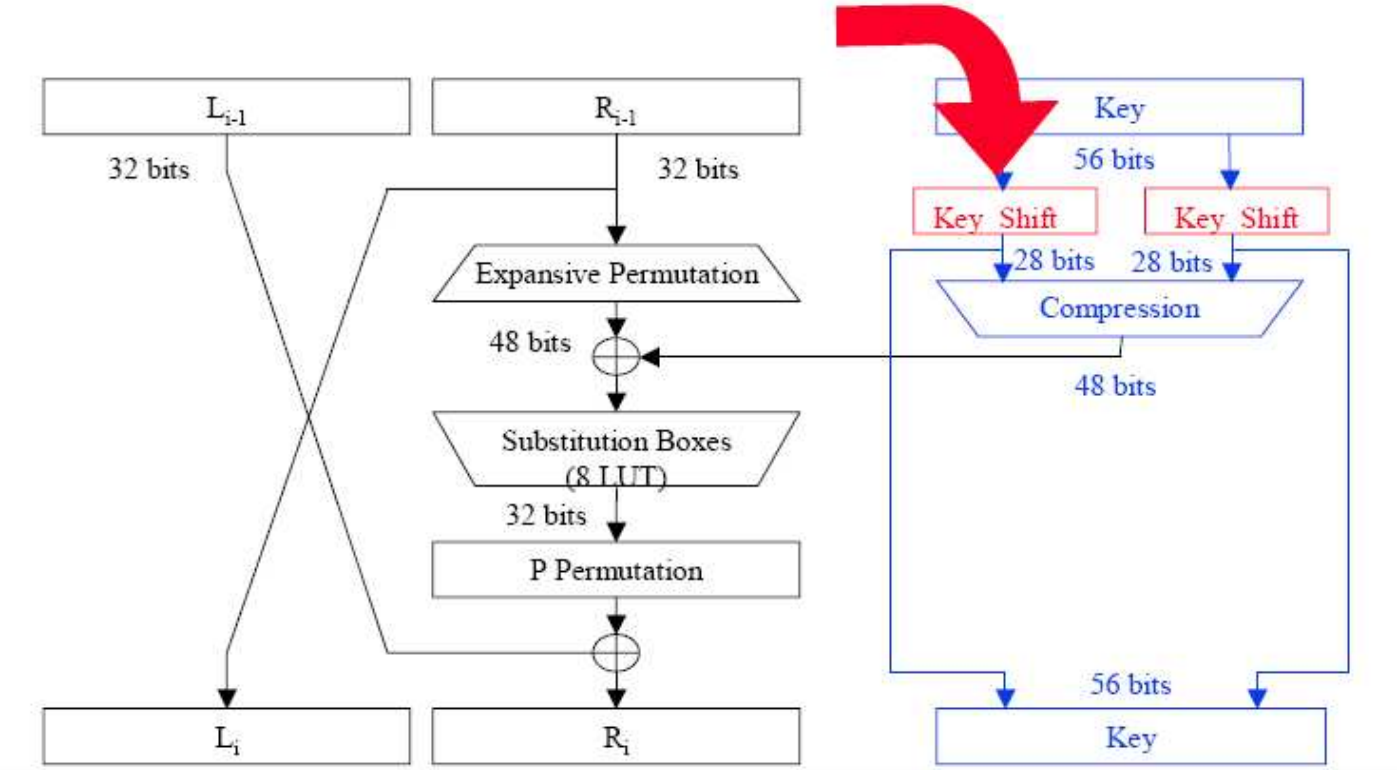
- Context
  - Find out a secret or private key
  - Known algorithm
  - Unknown implementation (background culture is recommended)
- Conditions
  - 1 cryptographic device available
  - Reverse engineering phase is required (power signature location)
  - Possibly known plain or ciphertext
- 2 target examples
  - DES key schedule
  - ECC private key multiplication

Data Encryption Standard

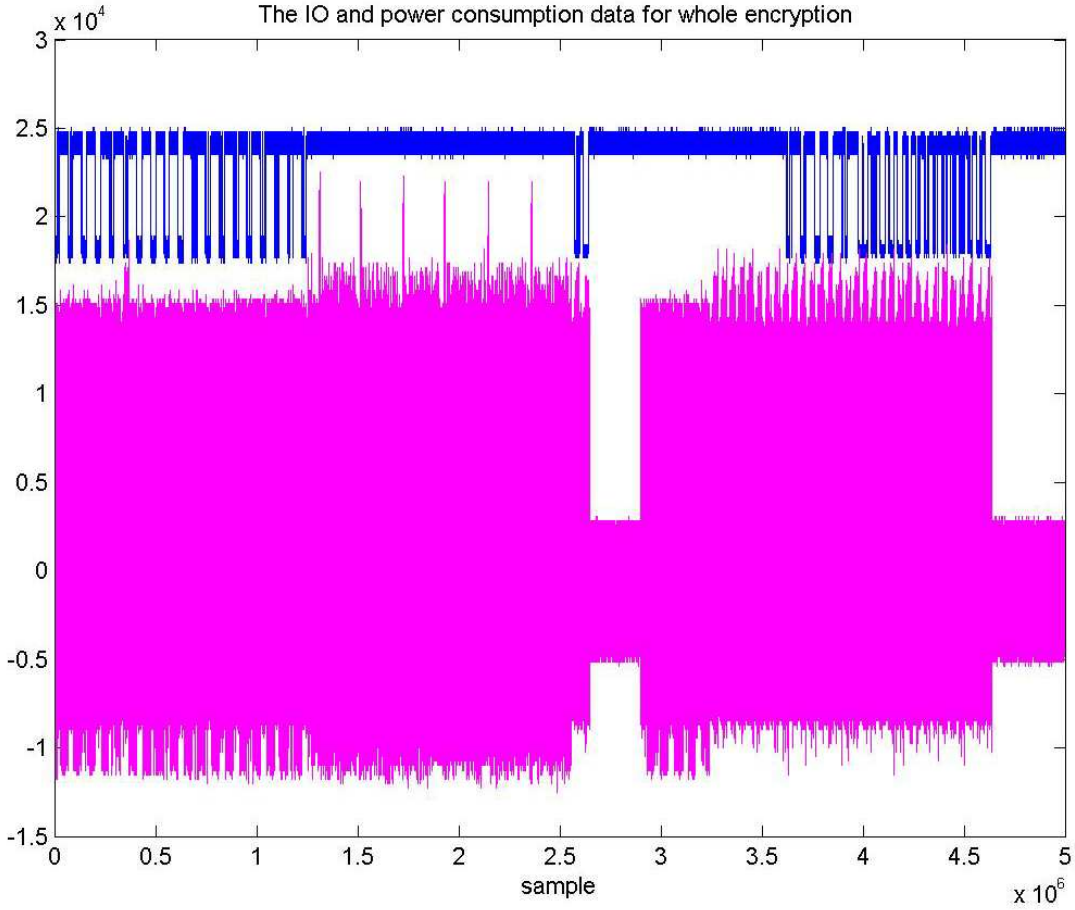


SPA on DES

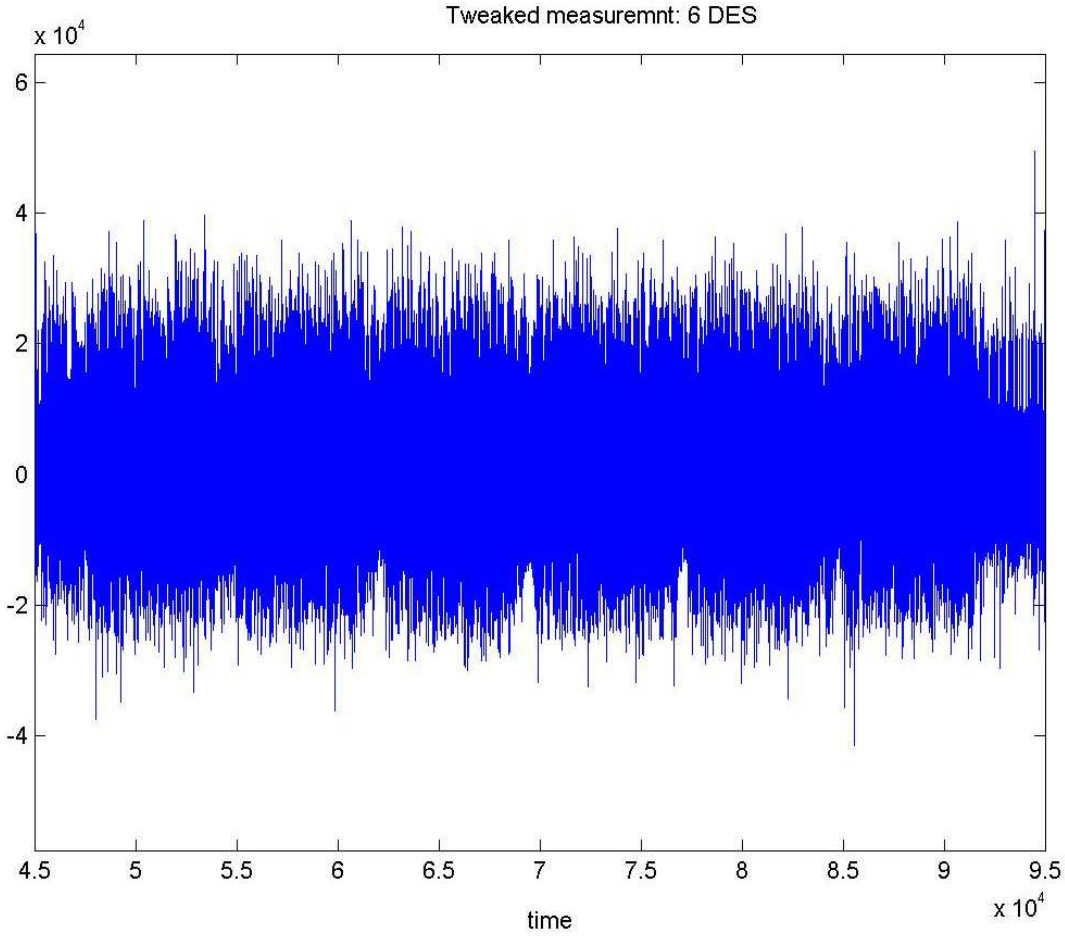
- Goal of the attack: find the DES secret key (5 bits)
- Knowledge on the implementation
- Target of the attack: key schedule



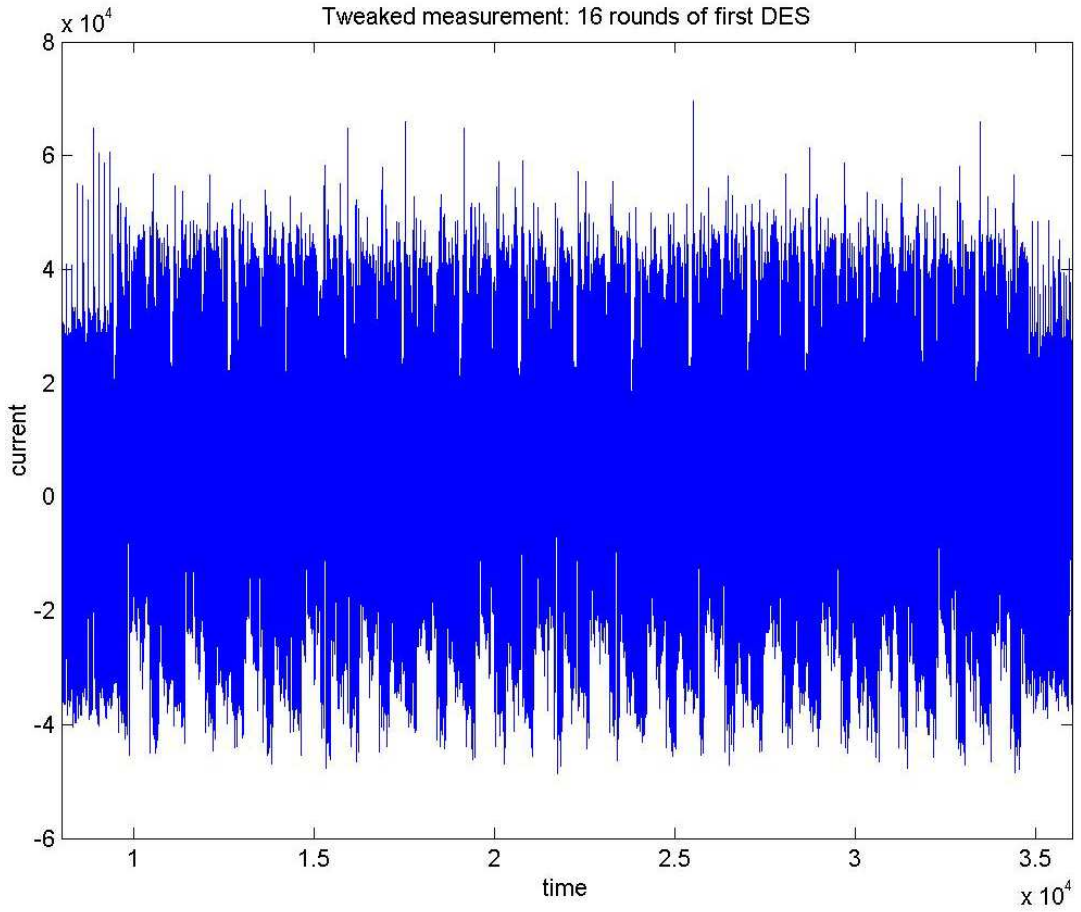
SPA on a Smartcard Implementation of DES (1/5)



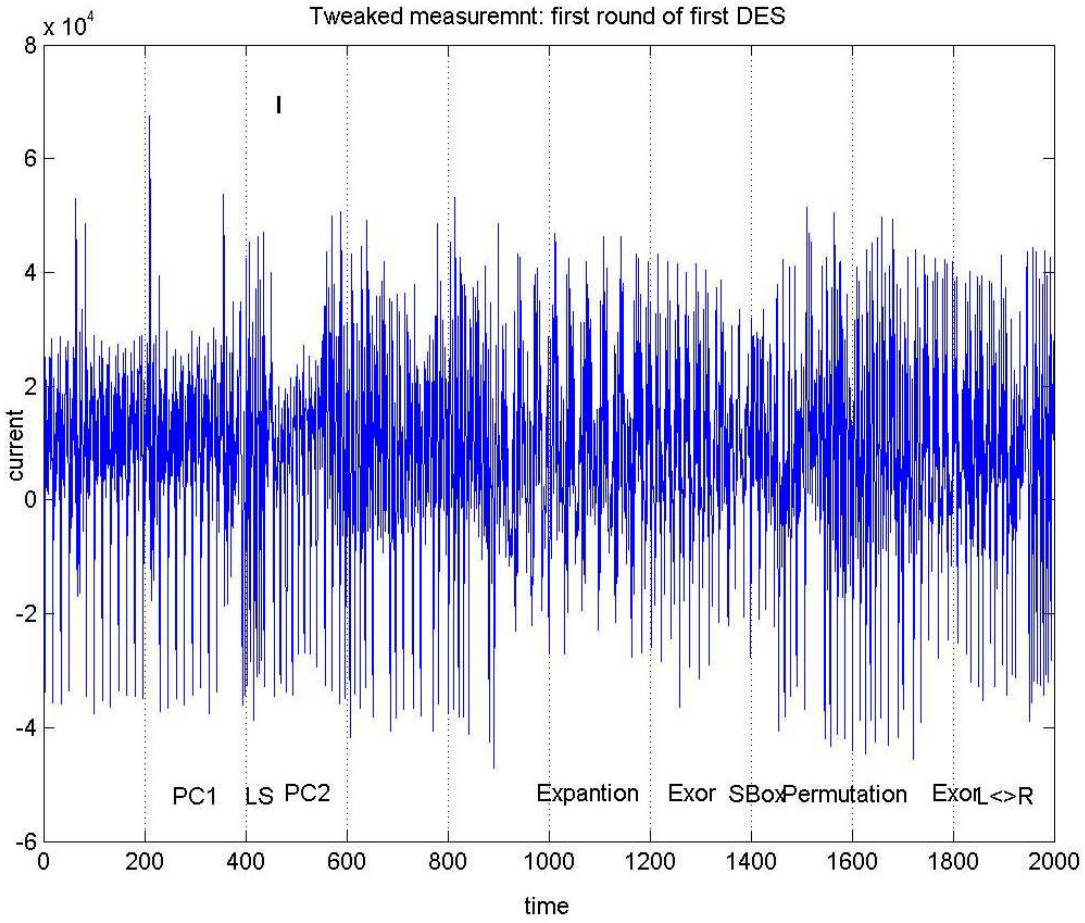
SPA on a Smartcard Implementation of DES (2/5)



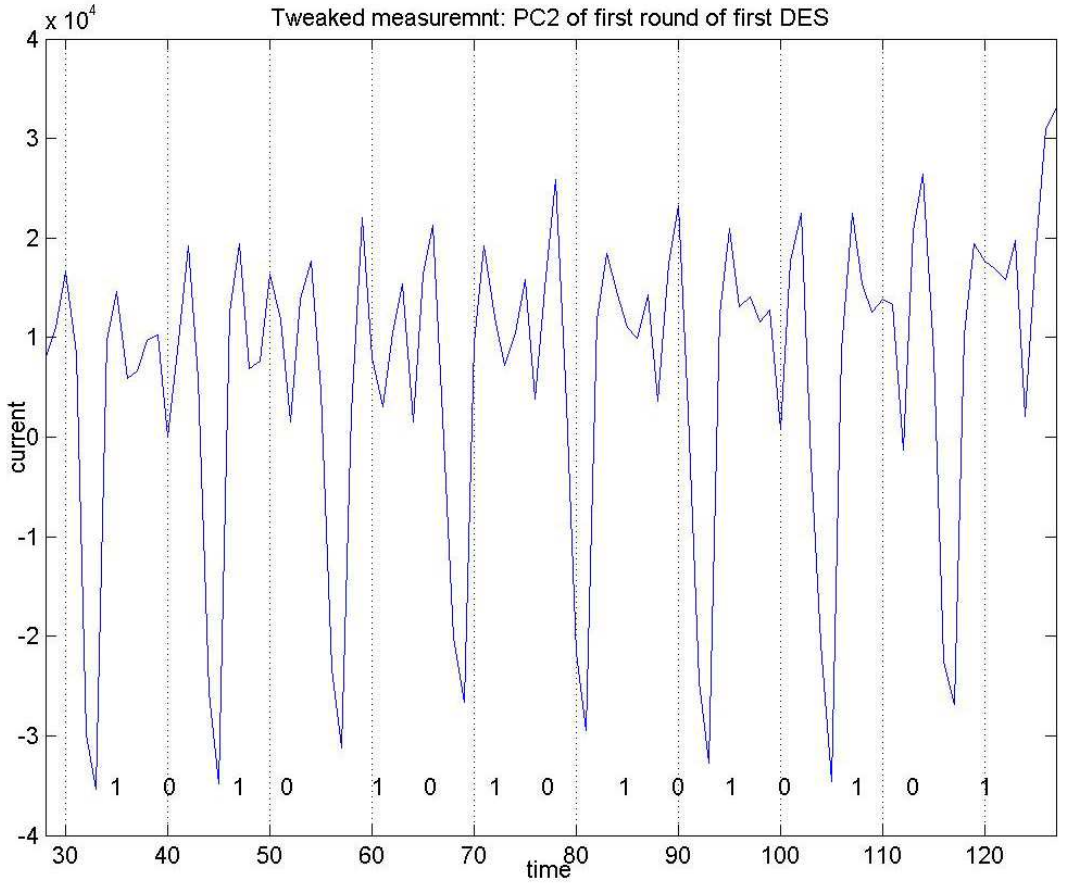
SPA on a Smartcard Implementation of DES (3/5)



SPA on a Smartcard Implementation of DES (4/5)



SPA on a Smartcard Implementation of DES (5/5)

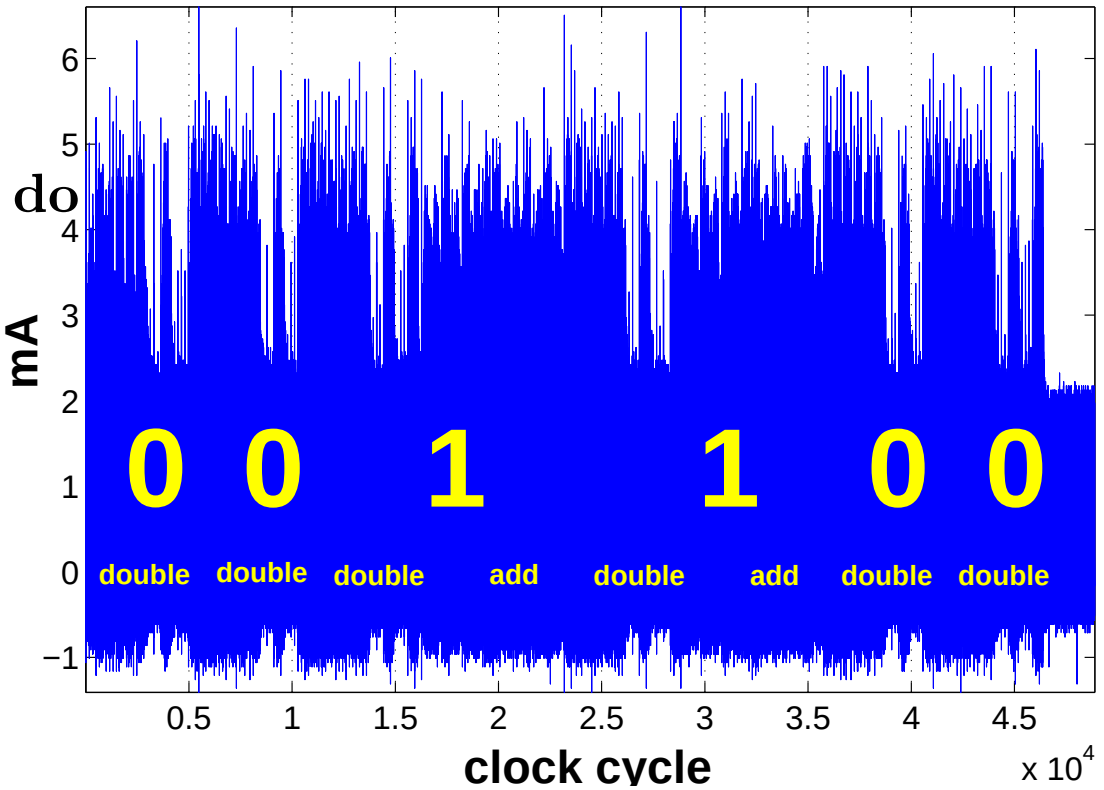


SPA on an FPGA Implementation of ECC

Require: EC point $P = (x, y)$, integer k, $0 < k < M$, $k = (k_{\ell-1}, k_{\ell-2}, \cdots, k_0)_2$, $k_{\ell-1} = 1$ and M

Ensure: $Q = [k]P = (x', y')$

1: $Q \leftarrow P$
2: **for** i from $\ell - 2$ downto 0 **do**
3: $\quad Q \leftarrow 2Q$
4: $\quad$ **if** $k_i = 1$ **then**
5: $\quad\quad Q \leftarrow Q + P$
6: $\quad$ **end if**
7: **end for**

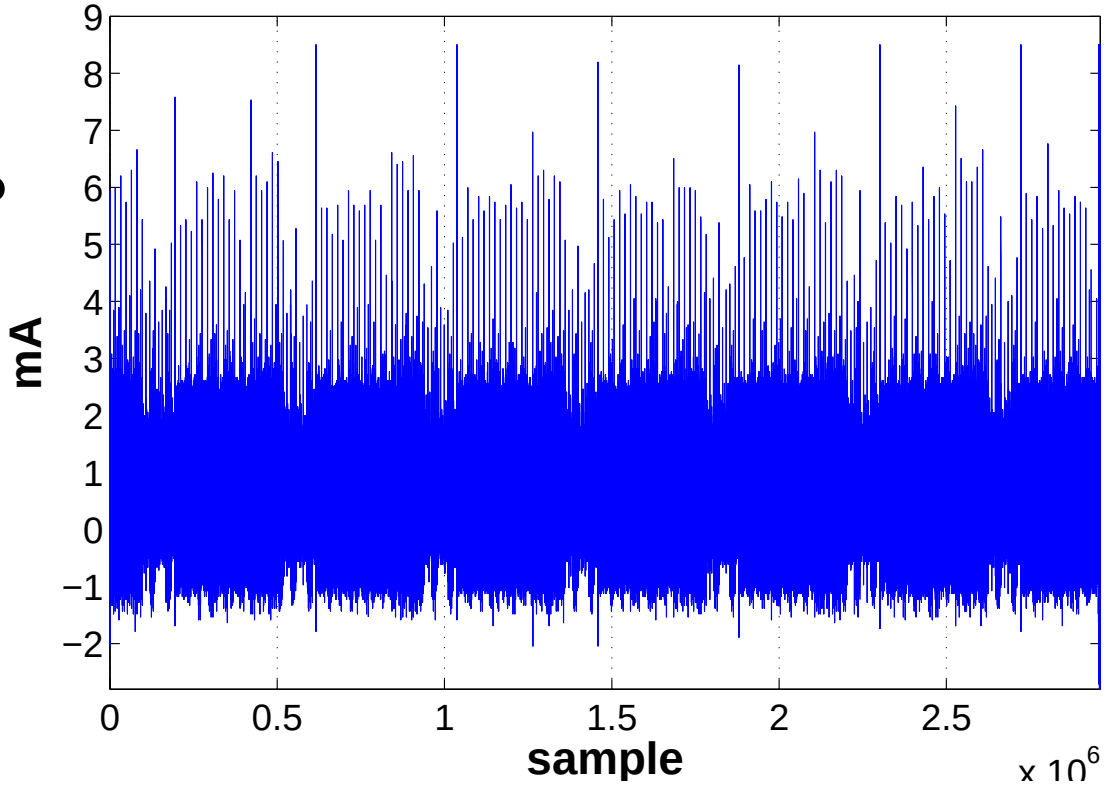


Countermeasure for SPA on an FPGA Implementation of ECC

Require: EC point $P = (x, y)$, integer k, $0 < k < M$, $k = (k_{\ell-1}, k_{\ell-2}, \cdots, k_0)_2$, $k_{\ell-1} = 1$ and M

Ensure: $Q = [k]P = (x', y')$

1: $Q \leftarrow P$
2: **for** i from $\ell - 2$ downto 0 **do**
3: $\quad Q_1 \leftarrow 2Q$
4: $\quad Q_2 \leftarrow Q_1 + P$
5: $\quad$ **if** $k_i = 0$ **then**
6: $\quad\quad Q \leftarrow Q_1$
7: $\quad$ **else**
8: $\quad\quad Q \leftarrow Q_2$
9: $\quad$ **end if**
10: **end for**



Conclusion

- SPA uses implementation related patterns
- SPA strategy
  - algorithm knowledge
  - reverse engineering phase
  - representation tuning (height of view, zoom, visualisation)
  - then play with implementation assumptions...
- SPA is always specific due to
  - the algorithm implementation
  - the application constraints
  - the chip's technology (electrical properties)
  - possible countermeasures

Countermeasures

- Anything that foils the attack
- Trivial countermeasure
  - prohibit code branches conditioned by the secret bits
- Advanced countermeasures
  - algorithm specification refinement
    - code structure
    - data whitening (blinding)
  - implementation design based on the chip's resources
    - play with instruction set
    - hardware electrical behaviour (current scrambler, desynchronisation, cryptoprocessor, ...)

Differential Power Analysis Attack

- DPA Statistical Principle
  - Acquisition procedure
  - Selection & prediction
  - Differential operator and curves
  - Reverse engineering using the DPA indicator
- Two Targets
  - A FPGA Implementation of Elliptic Curve Cryptosystem over $GF(p)$
  - An ASIC Implementation of AES
- Countermeasures

The target is k_{l-2}.
The points Q_1, Q_2 and Q are updated as:

Step 1: $Q \leftarrow P$

Step 3: $Q_1 \leftarrow 2Q = 2P$

Step 4: $Q_2 \leftarrow Q_1 + P = 3P$

Step 5: $Q \leftarrow \begin{cases} Q_1 = 2P & k_{l-2} = 0 & \text{Step 6} \\ Q_2 = 3P & k_{l-2} = 1 & \text{Step 8} \end{cases}$

Step 3: $Q_1 \leftarrow \begin{cases} 2Q = 4P & k_{l-2} = 0 \\ 2Q = 6P & k_{l-2} = 1 \end{cases}$

Step 4: $Q_2 \leftarrow \begin{cases} Q_1 + P = 5P & k_{l-2} = 0 \\ Q_1 + P = 7P & k_{l-2} = 1 \end{cases}$

Step 5: $Q \leftarrow \begin{cases} 2Q = 4P & k_{l-2} = 0 & k_{l-3} = 0 & \text{Step 6} \\ 2Q = 6P & k_{l-2} = 1 & k_{l-3} = 0 & \text{Step 6} \\ 2Q = 5P & k_{l-2} = 0 & k_{l-3} = 1 & \text{Step 8} \\ 2Q = 7P & k_{l-2} = 1 & k_{l-3} = 1 & \text{Step 8} \end{cases}$

DPA of a FPGA Implementation of ECC over $GF(p)$

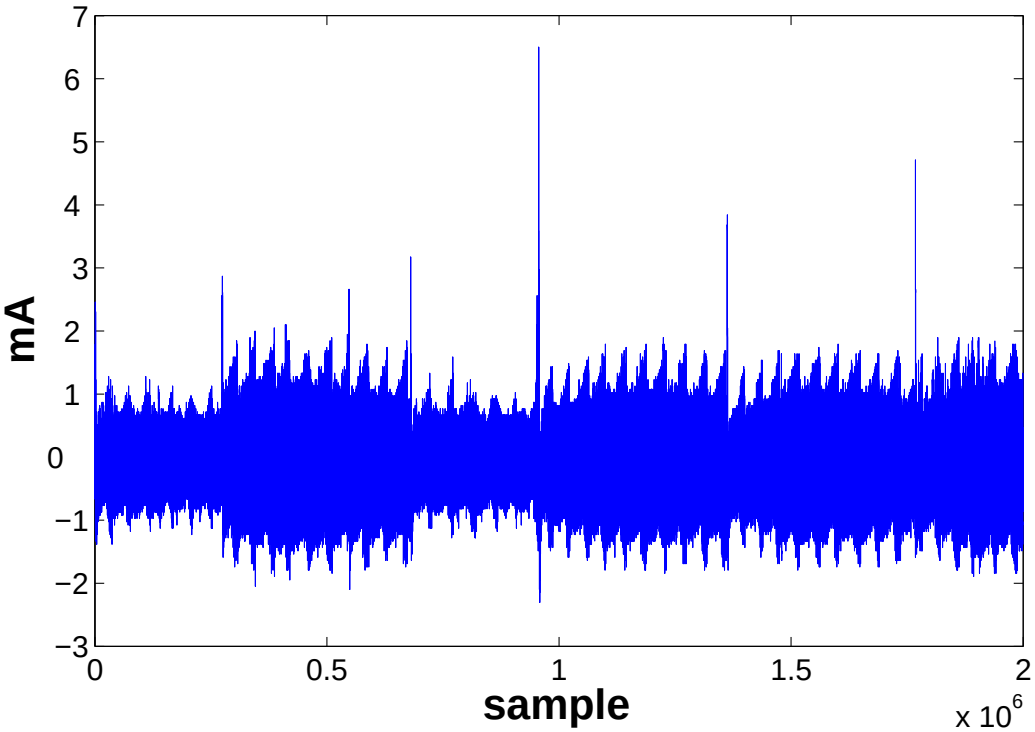
The first step of the DPA attack is to find the point of the measurements.

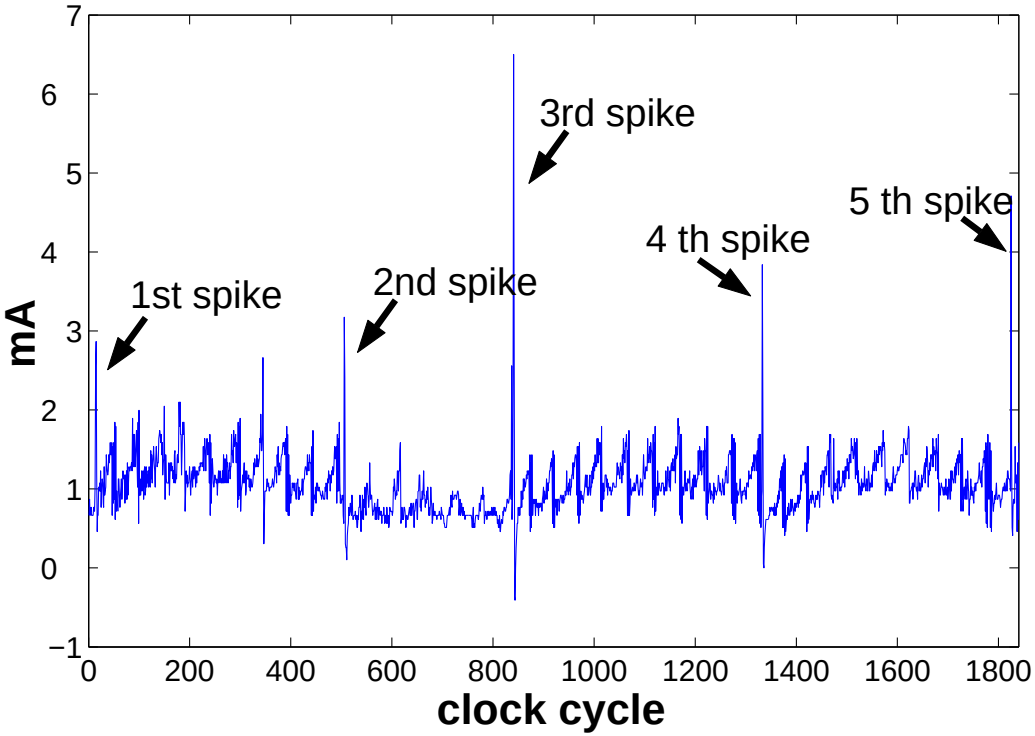
- The highest seven spikes show the end of seven EC point doubling operations.
- The first one corresponds to the end of the first EC doubling operation. This spike shows the ending of the second operation which is $Q_1 \leftarrow 2P$ and this step is executed independent from the key bits.
- The third, fourth and so an spikes need the knowledge of the k_{l-2}, k_{l-3} etc.
- Hence our choice for the measurement point is the second update of Q_1 after the second EC point doubling (Step 3).
- We use the transitions between the previous value of Q_1, $2P$, and the new value at our target point, $4P$ or $6P$ according to the value of k_{l-2} as the power consumption predictions.

Correlation Analysis (1/3)

1. produced a power consumption file
2. chosen N random points on the EC and one fixed, but random key, k
3. FPGA executes N point multiplications such that $Q_i = [k]P_i$ for $i = 1, 2, \cdots, N$
4. measured the power consumption of the FPGA during 2400 clock cycles around the second update of Q_1 (clock frequency: 300 kHz, sampling frequency: 250 MHz)
5. produced a $N \times 2\,000\,000$ matrix, M_1

Correlation Analysis (2/3)

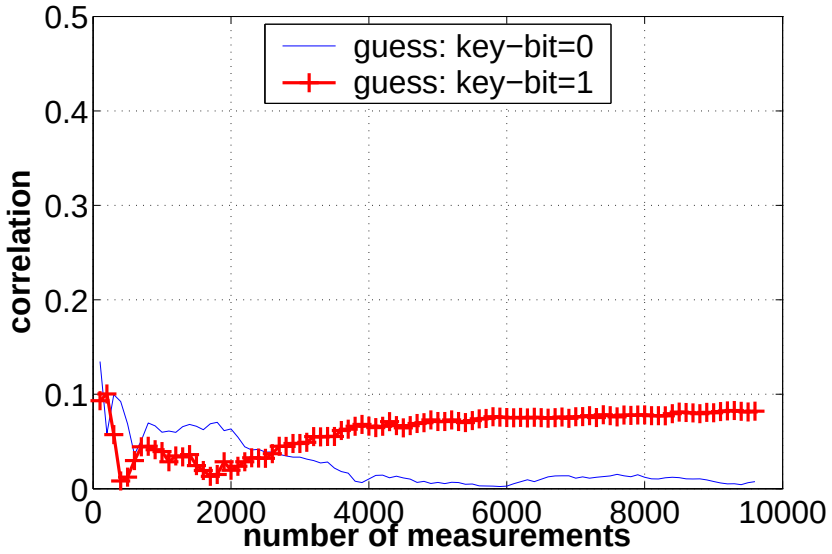




The maximum value of the measurement data in each clock cycle is found, M_2.

Correlation Analysis (3/3)

1. compute N EC point multiplications
2. compute the number of bit transitions from 0 to 1 in Q_1
   - $k_{l-2} = 0$: between $2P$ and $4P$ (M_3),
   - $k_{l-2} = 1$: between $2P$ and $6P$ (M_4)
3. $corr\,(M_3, M_2) > corr\,(M_4, M_2) \Rightarrow k_{l-2} = 0$



Distance of Mean Test (1/2)

1. We use the prediction matrices M_3 (for $k_{l-2} = 0$ guess) and M_4 (for $k_{l-2} = 1$ guess) in order to split the measurements into sets.

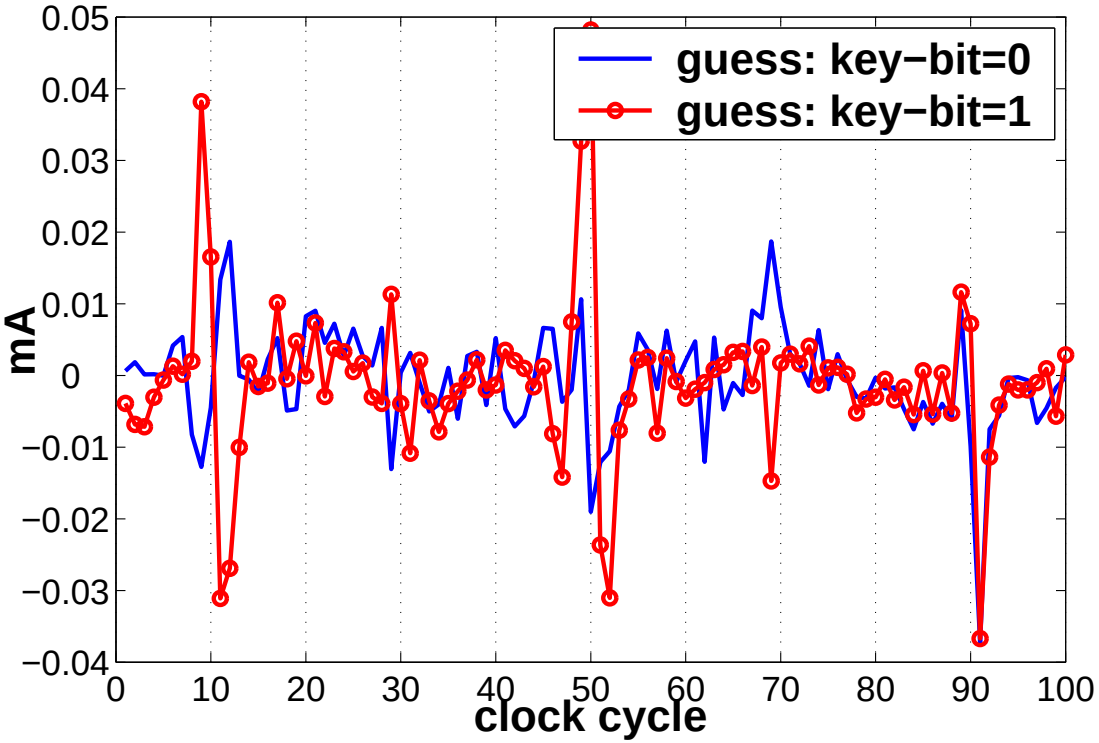
2. We calculate the mean value of M_3 and M_4, $E(M_3)$ and$E(M_4)$.

3. If $M_3(j) < E(M_3)$ then jth measurement is put in set $S_{1,1}$, otherwise in set $S_{1,2}$.

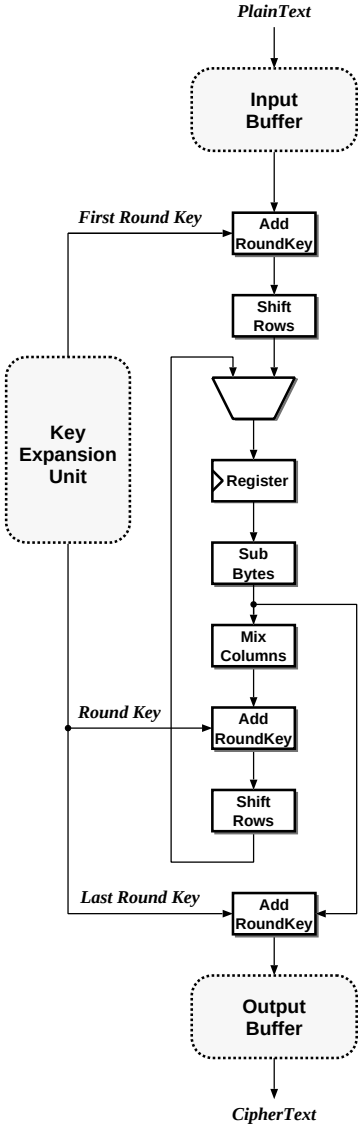
4. If $M_4(j) < E(M_4)$ then jth measurement is put in set $S_{2,1}$, otherwise in set $S_{2,2}$.

5. bias signals $T_1 = E(S_{1,2}) - E(S_{1,1})$ and $T_2 = E(S_{2,2}) - E(S_{2,1})$

Distance of Mean Test (2/2)



DPA on an ASIC Implementation of the AES



The target for our DPA attack were the 8 MSBs of the state after the initial key addition operation.

A DPA Using Simulated Data (1/3)

- Behavioral HDL simulations were used for the prediction of the dynamic power consumption.
- It allows to simulate attacks in an early stage of the design flow
- we did not reset the chip after each AES execution. At the beginning of an AES execution, the state still contained some value which is related to the previous AES execution.

A DPA Using Simulated Data (2/3)

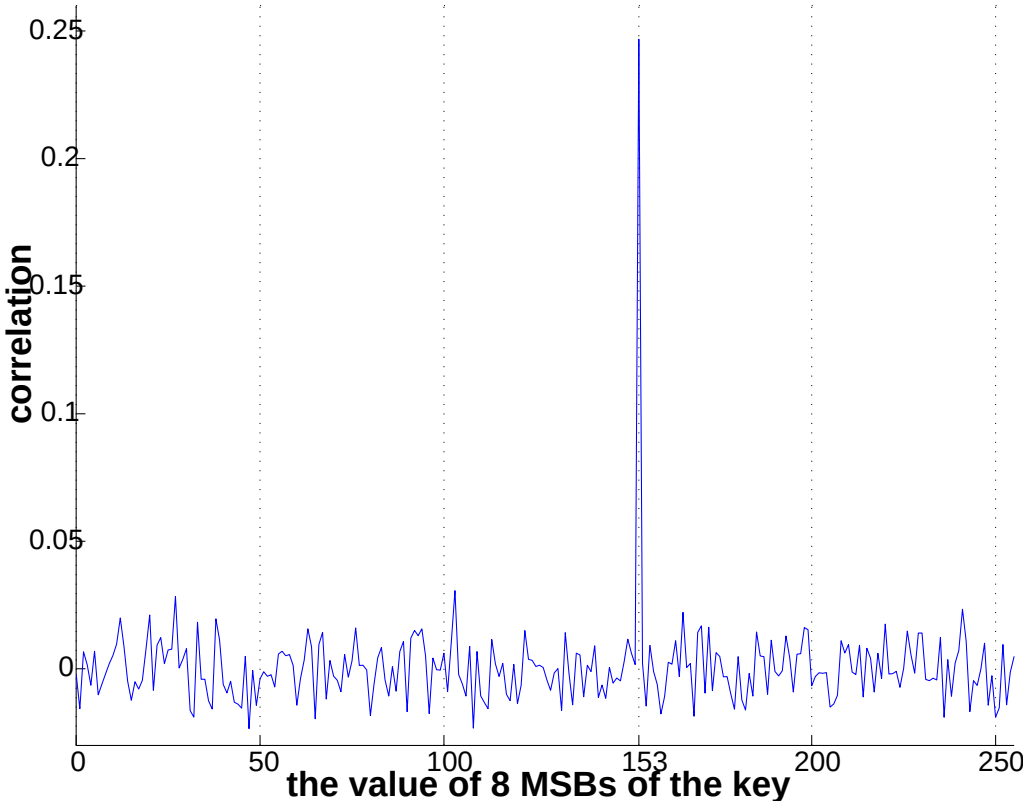
1. produced a simulated power consumption file. We have chosen N random plaintexts and one fixed, but random key. After each first clock cycle, the simulator has written the total number of bit changes between the previous and the current values of the state to this file. Hence, the simulator has produced a file which contains an $N \times 1$ matrix, M_1, with values between 0 and 128.

2. Then we calculated an $N \times 2^L$ matrix M_2. Each column of the matrix M_2 contains the prediction for the bit changes in the state for a particular guess of the L attacked key bits of the initial key addition.

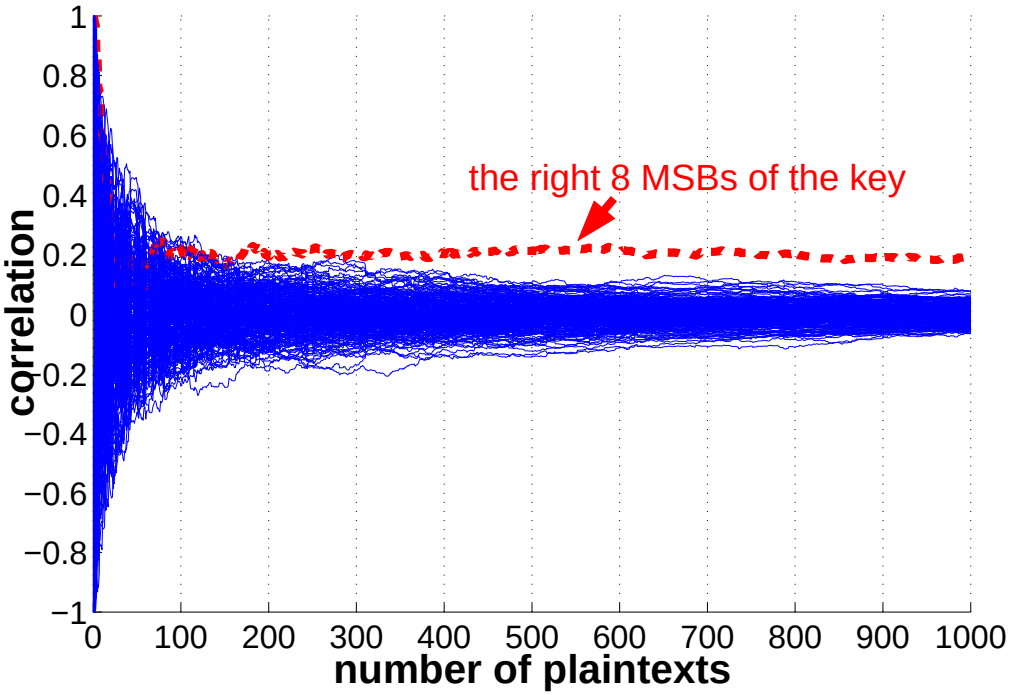
3. We calculate the correlation coefficients between the predictions of all the possible keys and M_1 as $c_i = C(M_1, M_2(1:N, i)) \quad i = 0, \ldots, 2^L - 1$.

We expect that only one value, corresponding to the correct L key bits, leads to a high correlation coefficient.

A DPA Using Simulated Data (3/3)



(a)

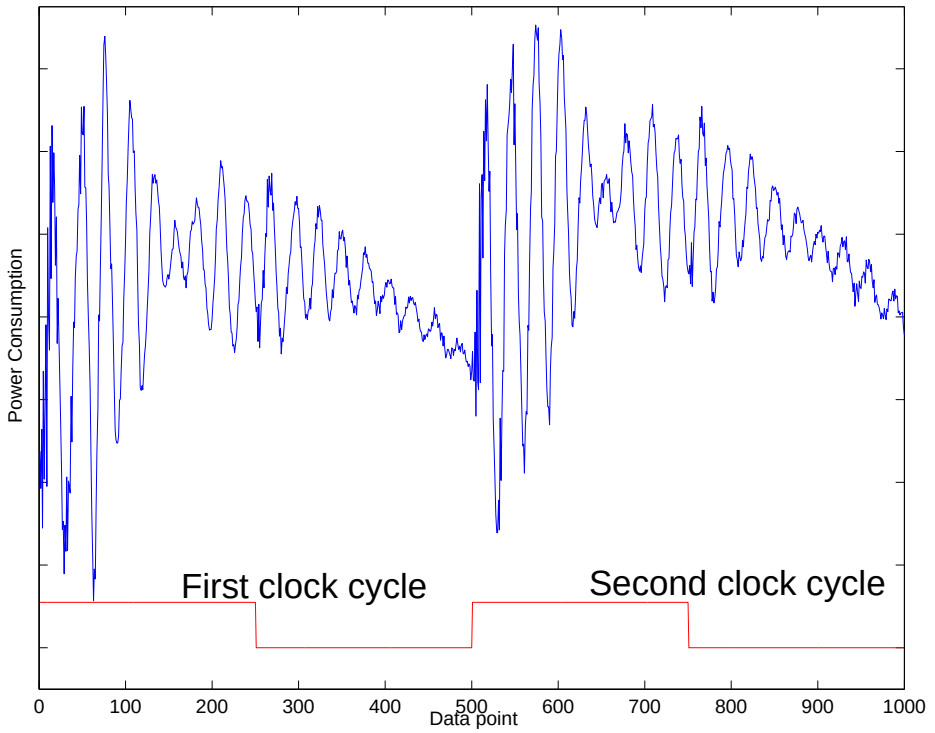


(b)

Correlation between M_1 and all the columns of M_2: (a) with 10 000 plaintexts (b) as a function of the number of measurements

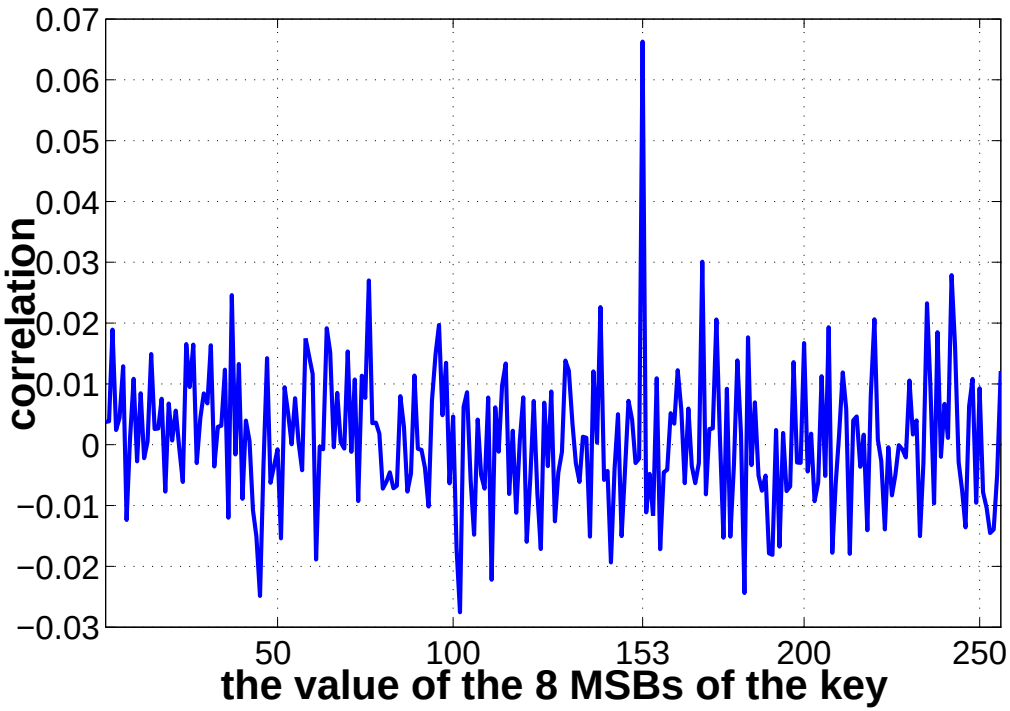
A DPA Using Measured Data (1/2)

1. Encrypted the same N plaintexts with the same key as used in the first step.
2. Measured the current consumption during the first two clock cycles.
3. Produced a $N \times 1000$ matrix, M_3.

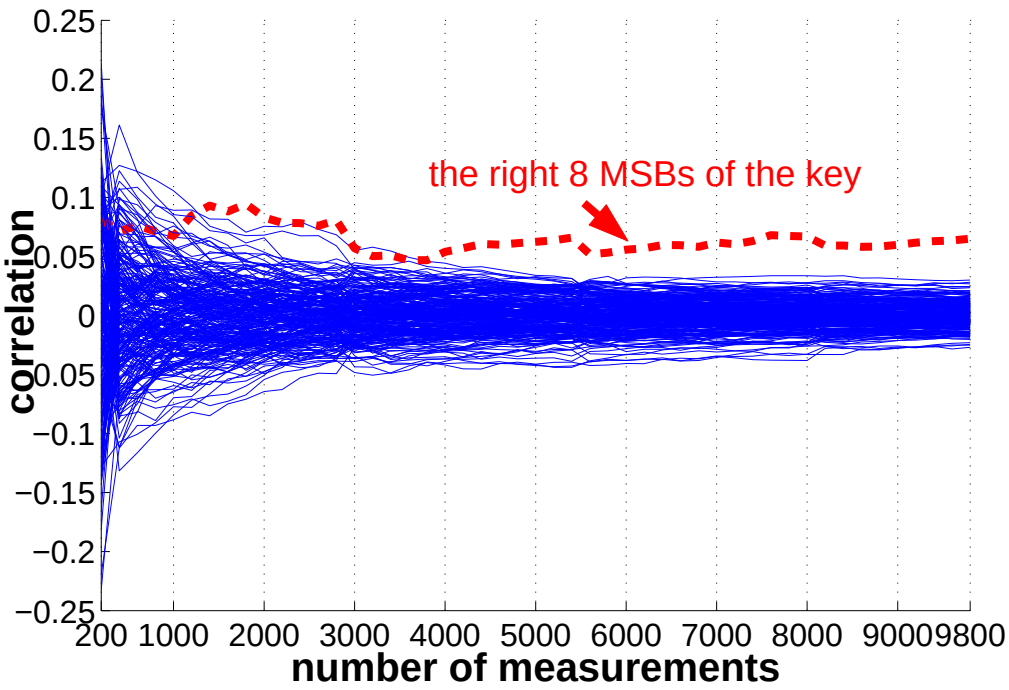


A DPA Using Measured Data (2/2)

1. Calculated the mean value of the measurement data in the second clock cycle: $M_4(i) = E(M_3(i, D+1:2D))$.
2. Correlation analysis: $c_i = C(M_4, M_2(1:N, i)) \quad i = 0, \ldots, 2^L - 1$.



(a)



(b)

Software Countermeasures

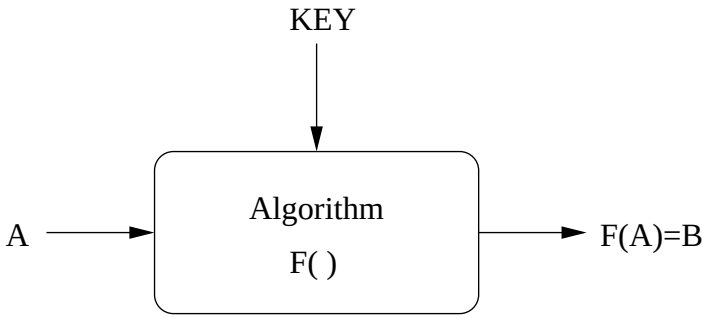
- Time randomization:
  - operations occur during random intervals
  - no-operations (NOPs)
  - dummy variables and instructions
  - data balancing (representation of the data is done in order to make the Hamming weight constant)
- Permuting the execution
- Masking techniques

Hardware Countermeasures

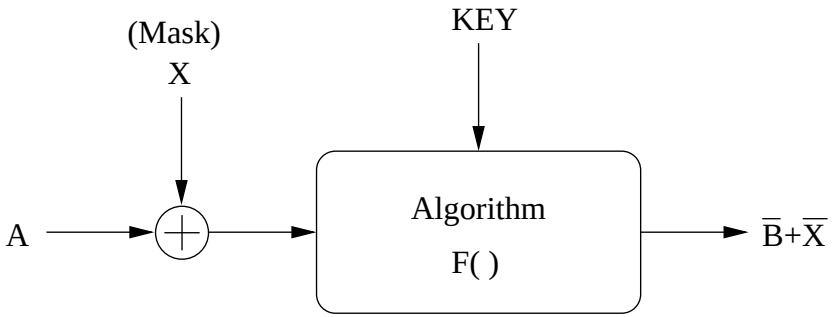
- Increasing the measurement noise
- power signal filtering
- novel circuit designs
  - detachable power supplies
  - Securing algorithm at the logic level
  - a family of masked gates which is theoretically secure in the presence of glitches
  - masked and dual-rail pre-charge logic style
  - Asynchronous circuits

Hardware Implementations of Data Masking on AES

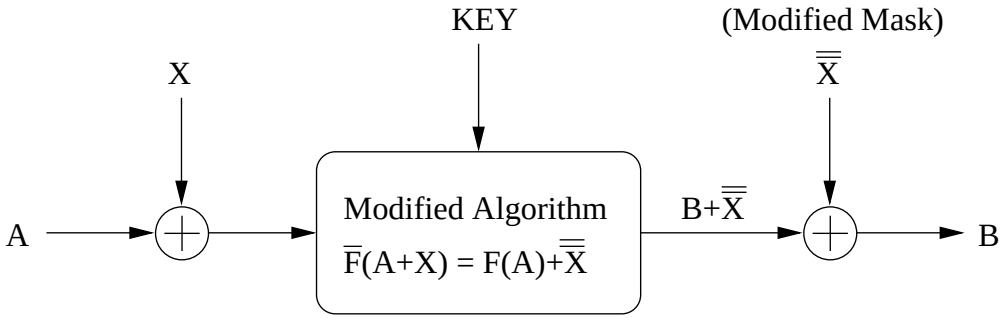
a.)



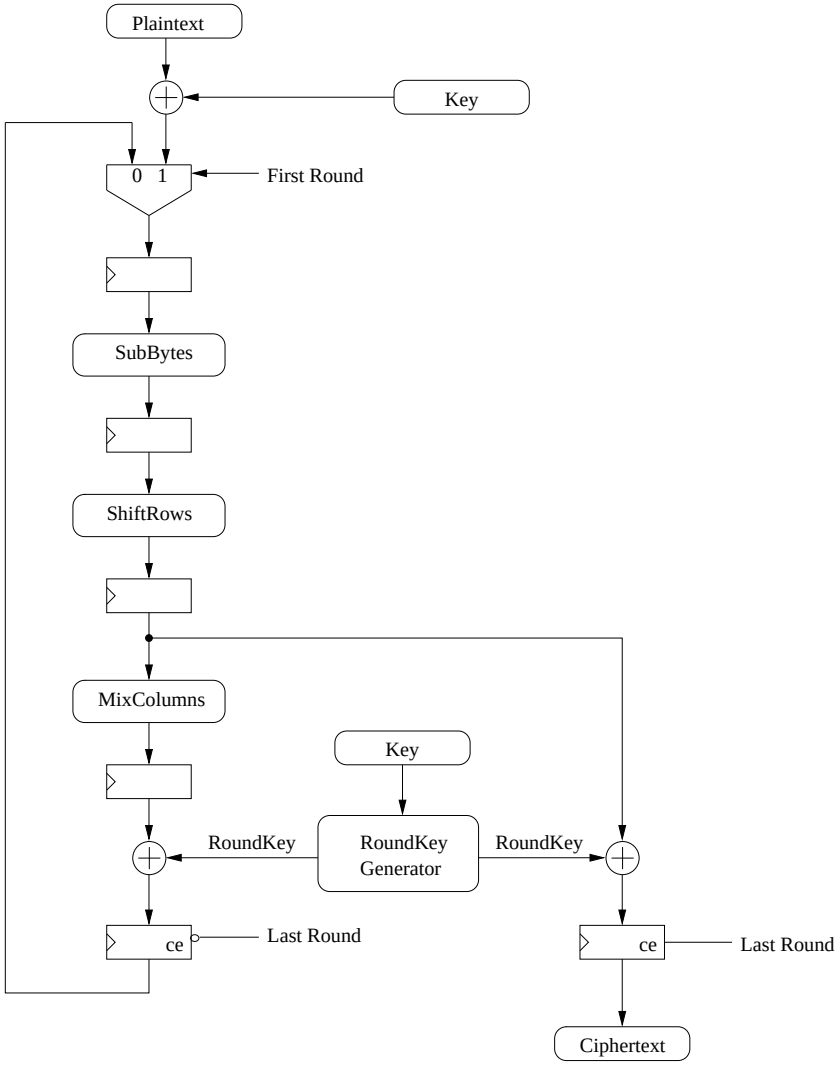
b.)



c.)



Implementation of AES Without Countermeasure



$SubBytes()$ Transformation

The S-Box includes two transformations:

1. multiplicative inversion over $GF\left(2^8\right)$
2. affine transformation

Table Method

Steps:

- output of the S-Box is calculated beforehand for all possible inputs
- these values are written in a ROM

Properties:

- fastest
- area of the ROM is the highest

Composite Field Method

- $GF\left(2^k\right) = GF\left(\left(2^n\right)^m\right)$
- $\{GF\left(2^n\right), Q(y)\}$ and $GF\left(\left(2^n\right)^m\right), P(x)\}$ forms a composite field
- $GF\left(2^n\right)$=subfield of $GF\left(2^k\right)$

Aim: to simplify the mathematical operations

1. a transformation to the subfield
2. mathematical operations
3. inverse transformation to the composite field

In AES
$\{GF\left(2^4\right), Q(y) = y^4 + y + 1\}$ and $\{GF\left(\left(2^4\right)^2\right), P(x) = x^2 + x + \lambda\}$

SBox with Composite Field Method

$E \in GF\left(2^8\right)$ and $E' \in GF\left(\left(2^4\right)^2\right)$

$E' = TE$

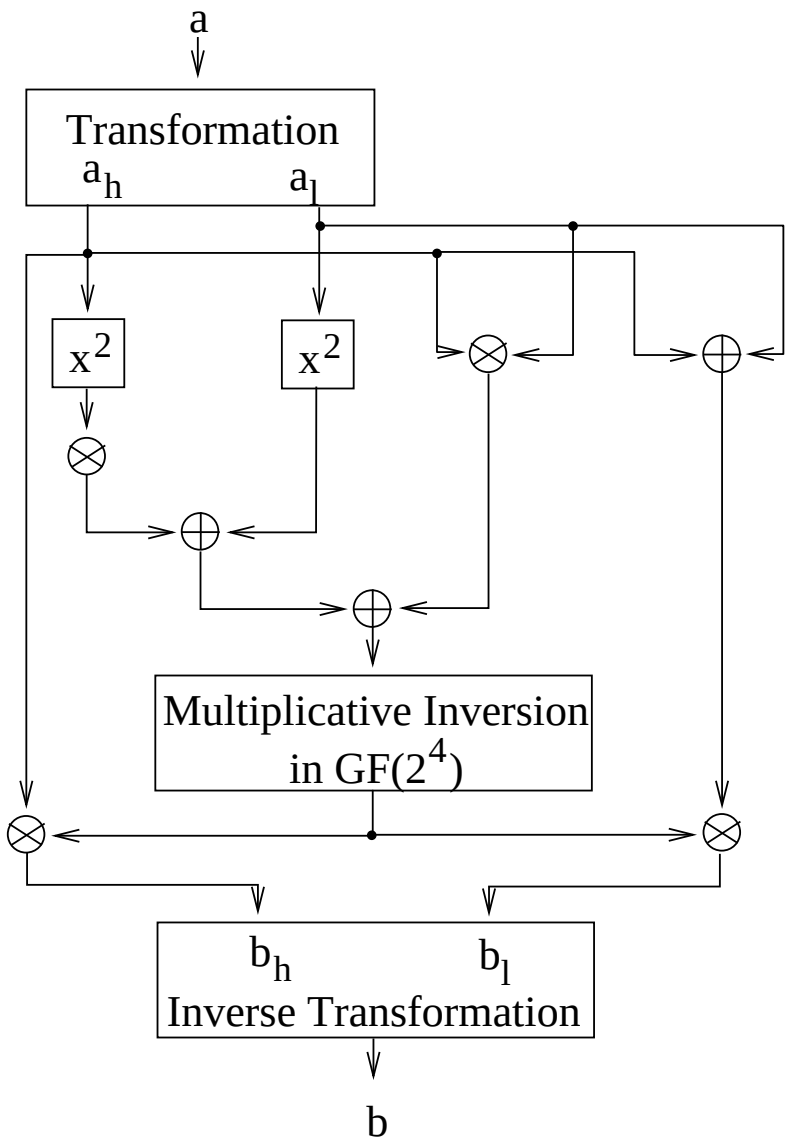
$$T = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad T^{-1} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$\forall a \in GF\left(\left(2^4\right)^2\right), a_h, a_l \in GF\left(2^4\right)$

$a = a_h x + a_l$

SBox with Composite Field Method

$a = a_h x + a_l$
$a^{-1} = b = b_h x + b_l$
$\delta = (a_h + a_l) \times a_l + \lambda \times a_h^2$
$b_h = a_h \times \delta^{-1}$
$b_l = (a_h + a_l) \times \delta^{-1}$



SBox with Composite Field Method

- The multiplicative inversion in $GF\left(2^4\right)$ is reduced to the multiplicative inversion in $GF\left(2^2\right)$.
- The multiplicative inversion in $GF\left(2^2\right)$ is linear according to $GF\left(2\right)$ and is equal to the square operation.
- Transform matrixes from $GF\left(2^4\right)$ to $GF\left(\left(2^2\right)^2\right)$ [**?**]:

$$T = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} T^{-1} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

SBox with Composite Field Method

The area and latency results of three different implementation of S-Box

	# of LUT	# of Slices	Minimum period (ns)
Table method	192	106	5.469
$GF\left(\left(2^4\right)^2\right)$	76	44	13.423
$GF\left(\left(\left(2^2\right)^2\right)^2\right)$	76	44	13.098

$ShiftRows()$ Transformation

- only changes the position of the bytes in the state
- A combinational circuit is not needed for the implementation of this transformation
- by wiring some outputs of one register to some inputs in different positions of another register

$MixColumn()$ Transformation

Let $\tilde{a} = \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix}$ and $\tilde{b} = \begin{bmatrix} b_3 \\ b_2 \\ b_1 \\ b_0 \end{bmatrix}$ be two columns of state.

$$\tilde{b} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \tilde{a}$$

- $a_i = a_{i,7}x^7 + \cdots + a_{i,0}$ and $b_i = b_{i,7}x^7 + \cdots + b_{i,0}$
- The irreducible polynomial in $GF\left(2^8\right)$ is $m(x) = x^8 + x^4 + x^3 + x + 1$.
- Costs 37 slices (64 LUTs) on Virtex-E 1000 FPGA.

$AddRoundKey()$ Transformation

- Bitwise xor of 128-bits of the round key and 128-bits of the state
- Implemented by using 128 two input XOR gates
- 74 slices (128 LUTs) are used for this operation

Implementation with Masking Countermeasure

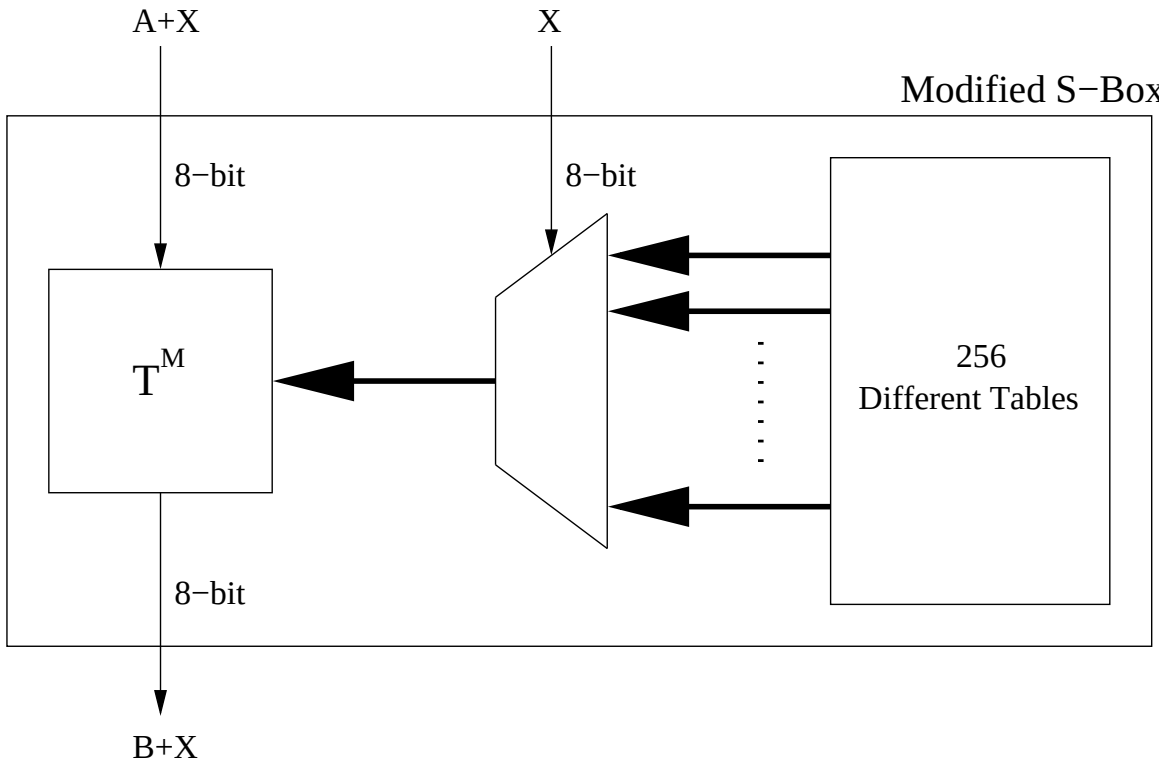
Modified Table:

$b = T[a]$

$T^M[a \oplus x] = T[a] \oplus \bar{x}$

T_i^M have to be calculated for all possible, $2^8 = 256$, values of the mask.

1 table=106 Slices and 256 tables $= 256 \times 106 = 26404$ Slices.



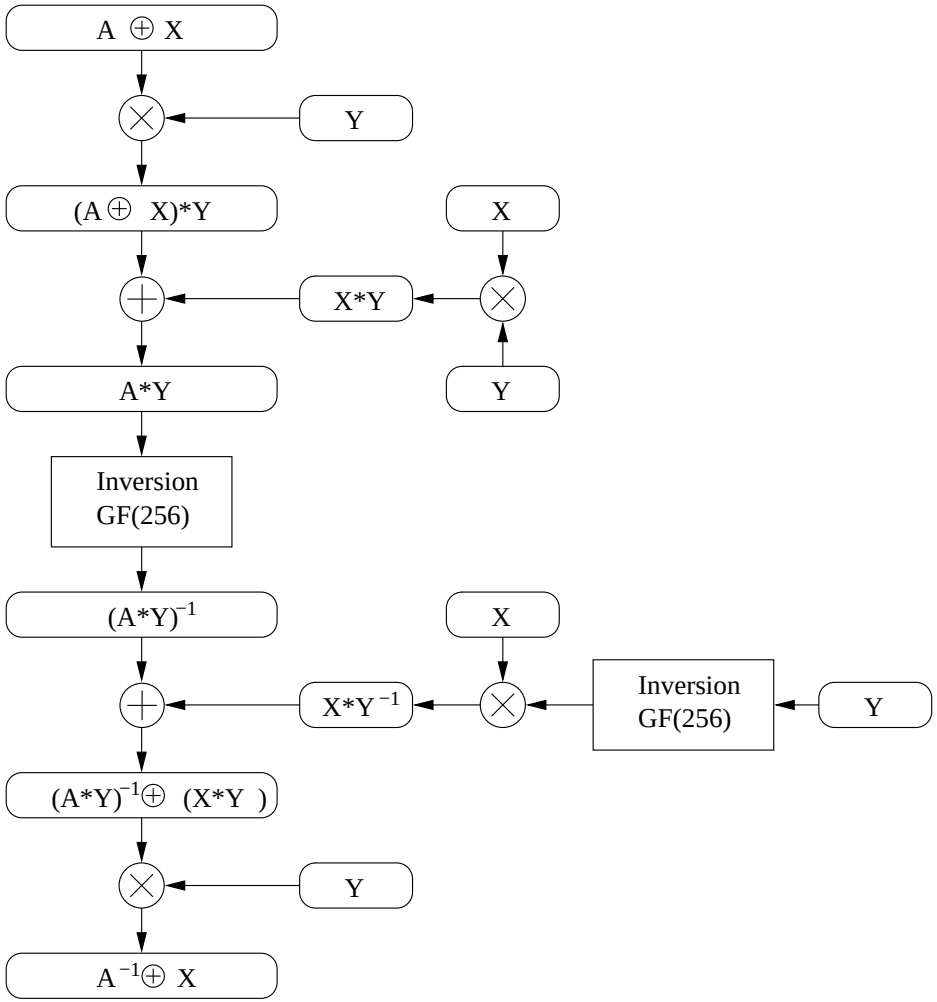
Additive and Multiplicative Masking

Akkar and Giraud propose to modify the S-Boxes in order to make them act as linear to the additive mask in [2].

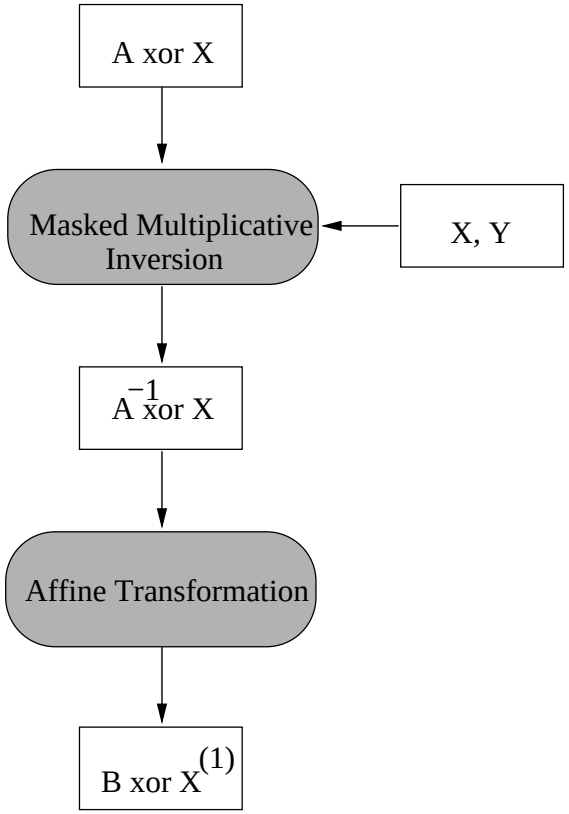
$(a \times x)^{-1} = a^{-1} \times x^{-1}$

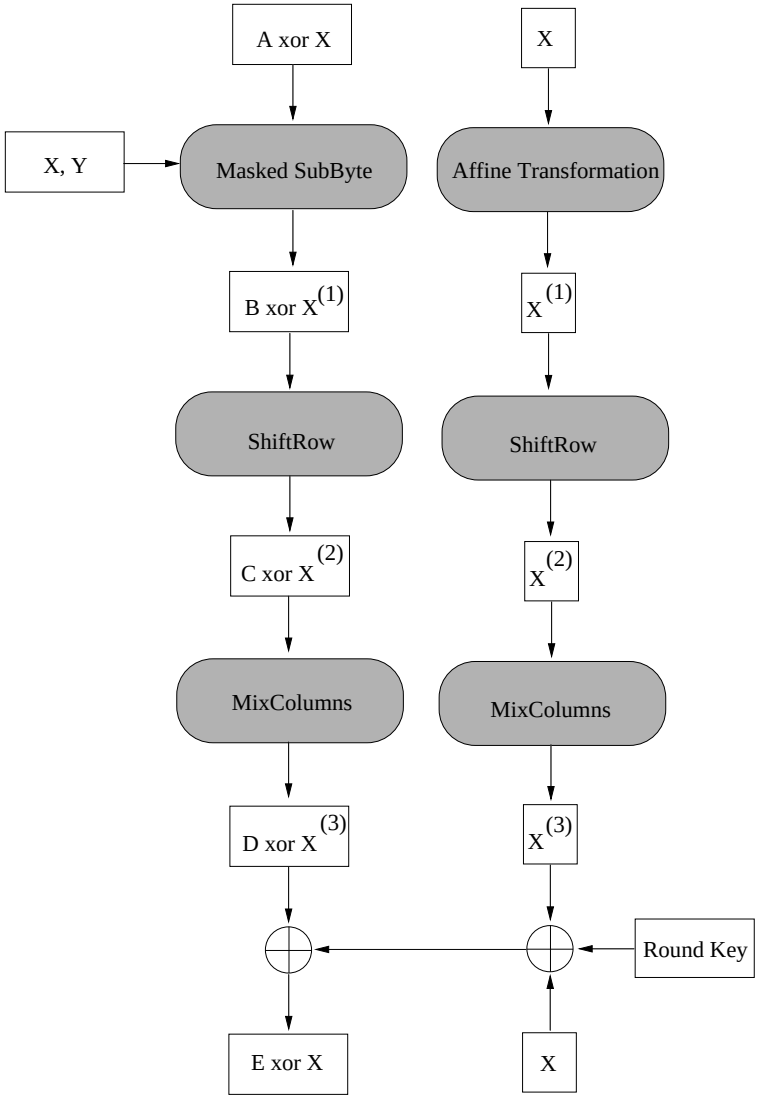
- additive mask throughout the algorithm
- transform from the additive to the multiplicative mask before the multiplicative inversion
- transform back to the additive mask after completing the multiplicative inversion

Additive and Multiplicative Masking



Additive and Multiplicative Masking





Additive and Multiplicative Masking

- The original multiplicative inversion is kept unchanged.
- 4175 Slices (7628 LUTs) and 1291 flip-flops were used. Minimum clock period was 22.836 ns. Throughput is 140,13 Mb/s.

Additive Masking

Oswald *et. al* propose a masking method which is not weak against zero value attacks and has lower complexity than Akkar and Giraud method in [53].

- all the intermediate values are masked with additive mask
- the multiplicative inversion is modified in a way that it produces the right output and preserve the mask value
- The multiplicative inversion is performed by $GF\left(\left(\left(2^2\right)^2\right)^2\right)$

$(a+m) = (a_h + m_h)\, x + (a_l + m_l)$

Additive Masking

$$
\begin{array}{lcl}
a^{-1} & = & b = b_h x + b_l \\
b_h & = & a_h \times \bar{d} \\
b_l & = & (a_h + a_l) \times \bar{d} \\
d & = & (a_h + a_l) \times a_l + \lambda a_h^2 \\
\bar{d} & = & d^{-1}
\end{array}
$$

$$
\begin{array}{lcl}
(a+m)^{-1} & = & (b+m) = (b_h+m_h)\, x + (b_l+m_l) \\
(b_h+m_h) & = & \left(a_h \times \bar{d}\right) + m_h = f_{b_h}\left((a_h+m_h), (a_l+m_l), (d+m_d), m_h, m_l, m_d\right) \\
(b_l+m_l) & = & (a_h+a_l) \times \bar{d} + m_l = f_{b_l}\left((a_h+m_h), (a_l+m_l), (d+m_d), m_h, m_l, m_d\right) \\
(d+m_d) & = & (a_h+a_l) \times a_l + \lambda \times a_h^2 + m_d \\
 & = & f_d\left((a_h+m_h), (a_l+m_l), (d+m_d), m_h, m_l, m_d\right) \\
\left(\bar{d}+\bar{m}_d\right) & = & d^{-1} + m_d = f_{\bar{d}}\left((a_h+m_h), (a_l+m_l), (d+m_d), m_h, m_l, m_d\right)
\end{array}
$$

Only the masked values $((a_h + m_h), (a_l + m_l))$ and the masks can be input.

Implementation Results

	# of Slices	# of LUTs	# of FF	Min Period (ns)
Akkar [2]	4175	7628	1291	22.836
IAIK [53]	3580	6722	1292	20.769

Electromagnetic Attacks

The sudden current pulse that occurs during the transition of the output of a CMOS gate, causes a sudden variation of the electromagnetic field surrounding the chip.

The electromotive force across the sensor (Lentz' law) relates to the variation of magnetic flux as follows [57]:

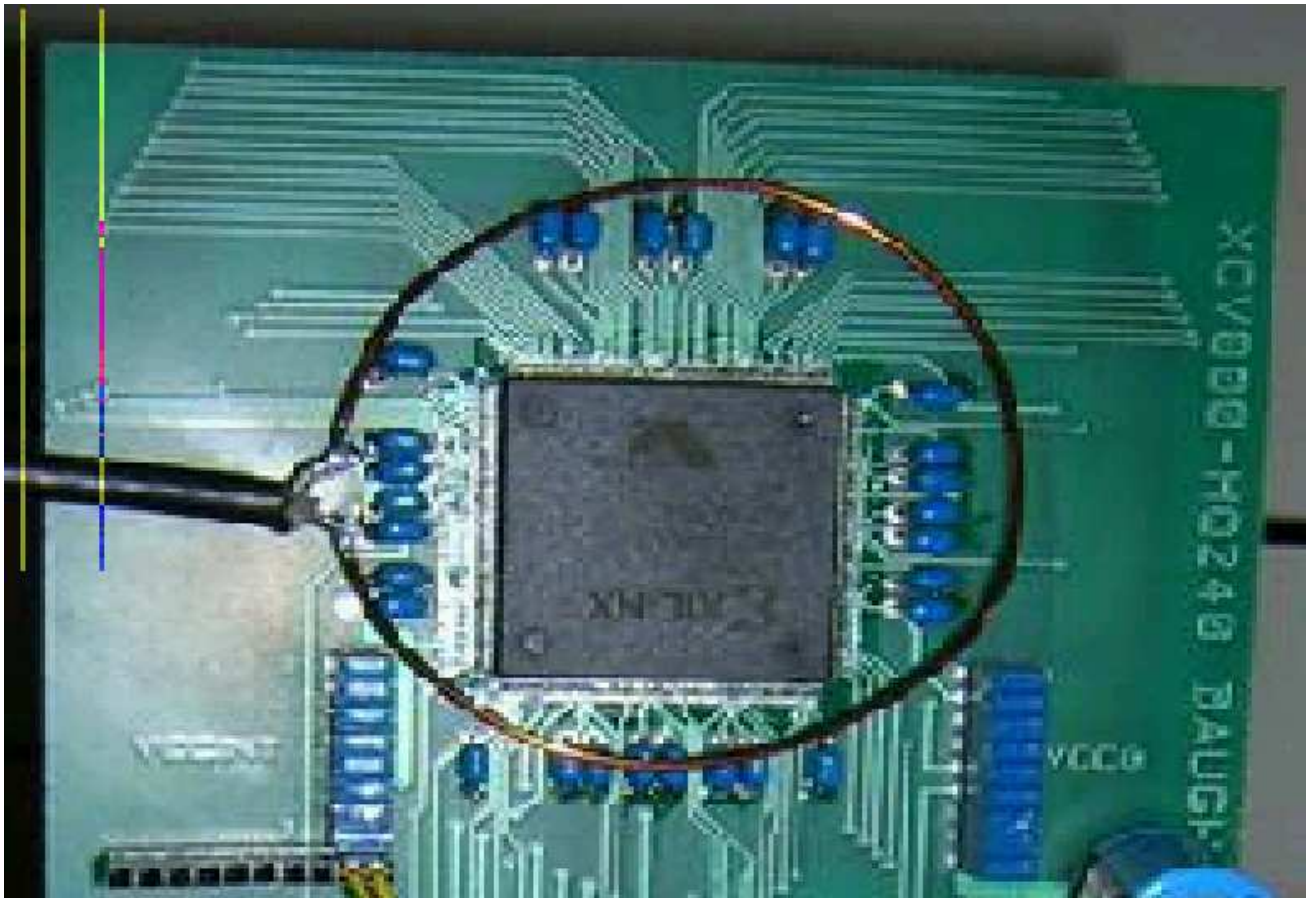
$$V = -\frac{\mathrm{d}\phi}{\mathrm{d}t} \quad \text{and} \quad \phi = \iint \vec{B} \cdot d\vec{A}\,,$$

The Biot-Savart Law relates magnetic fields to the currents which are their sources.

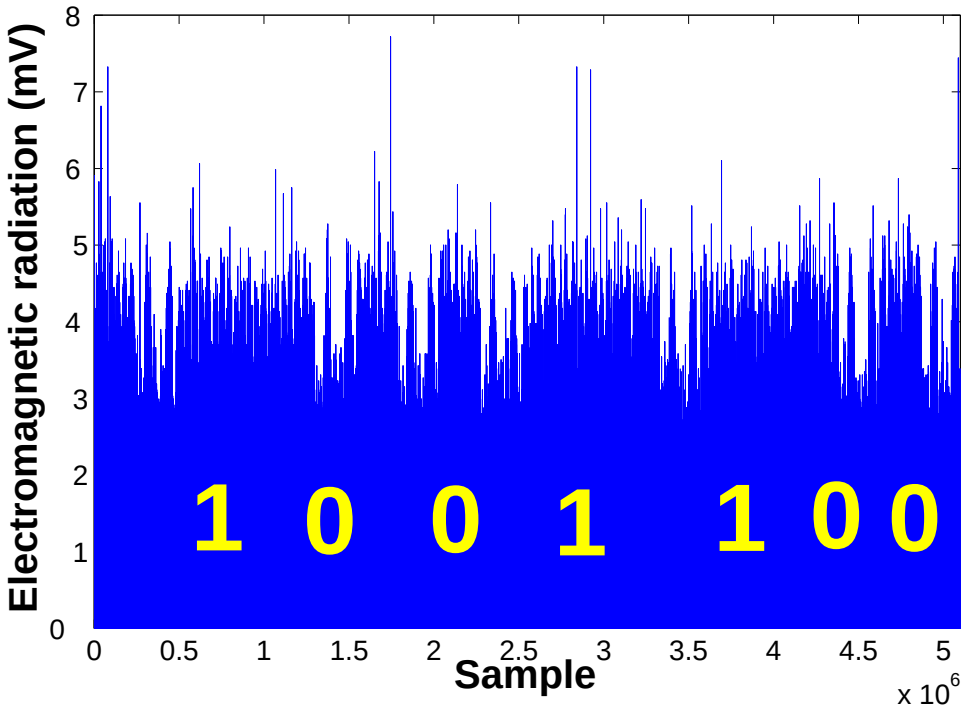
$$\mathrm{d}\vec{B} = \frac{\mu_0 I \mathrm{d}\vec{L} \times \vec{\hat{r}}}{4\pi r^2}\,,$$

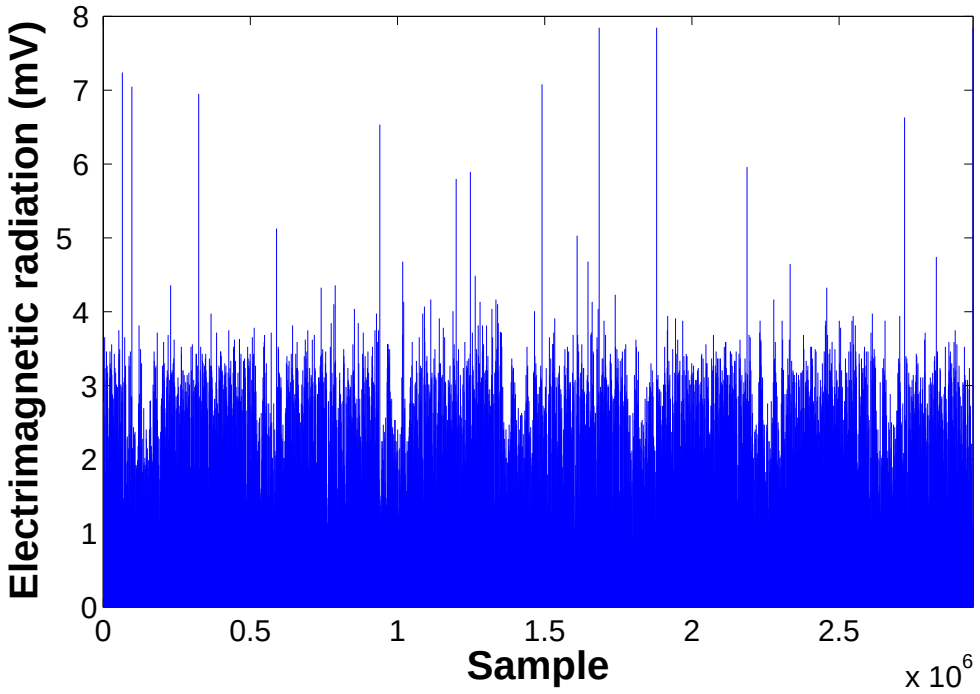
where $\mathrm{d}\vec{L}$ is length of conductor carrying electric current I and $\vec{\hat{r}}$ is unit vector to specify the direction of the vector distance r from the current to the field point.

The measurement setup

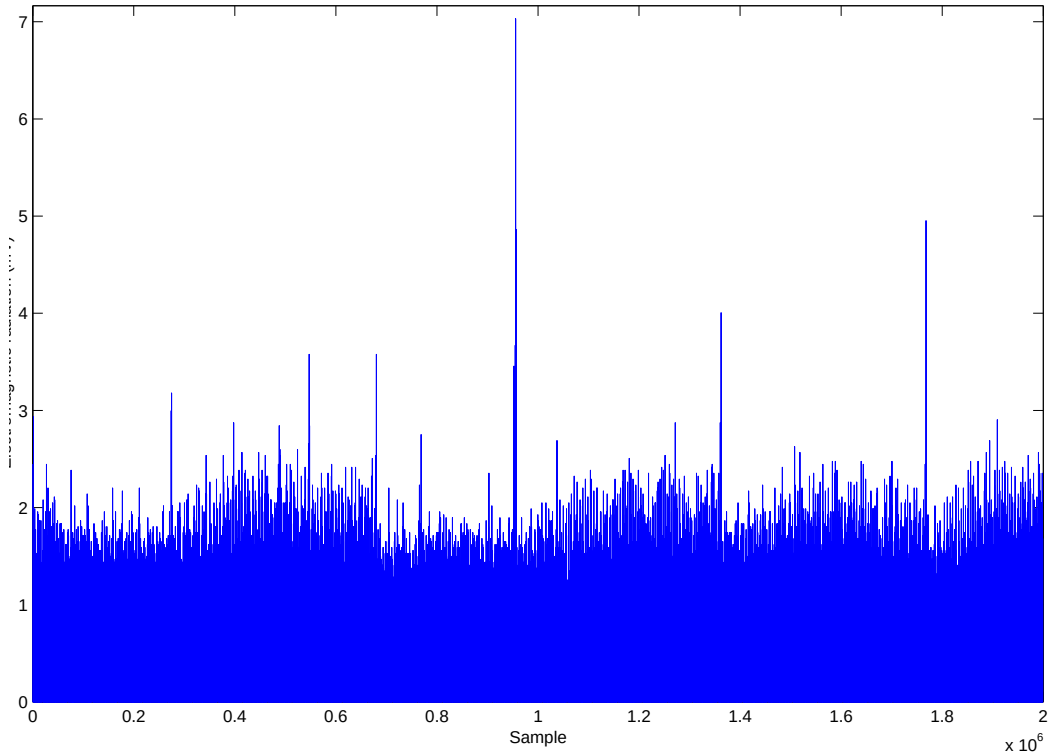


SEMA on the FPGA Implementation of ECC

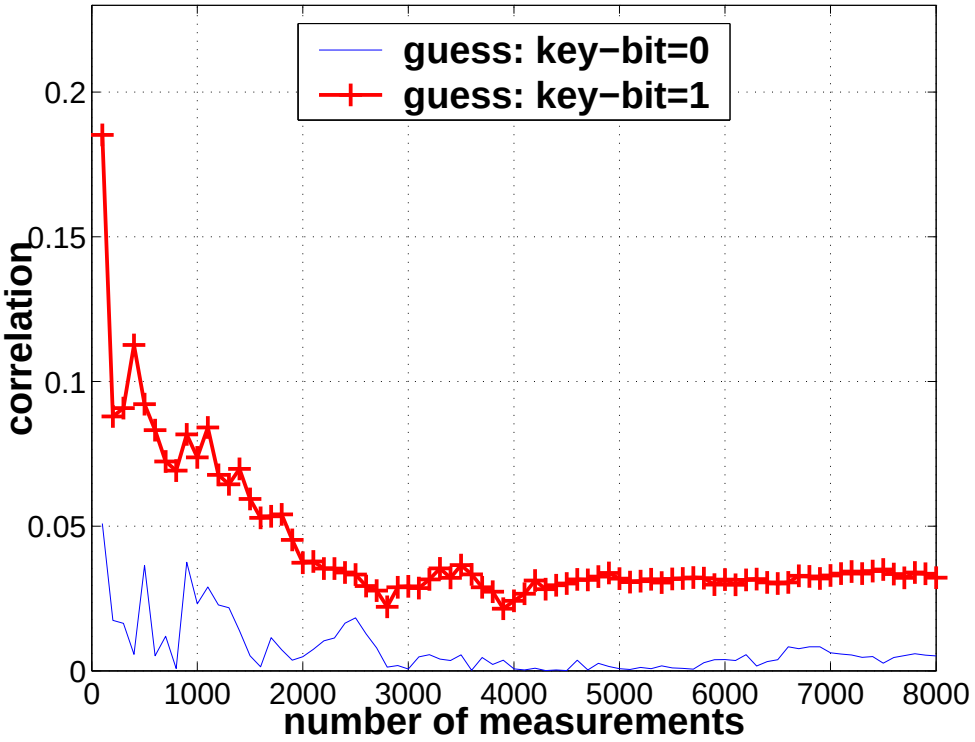


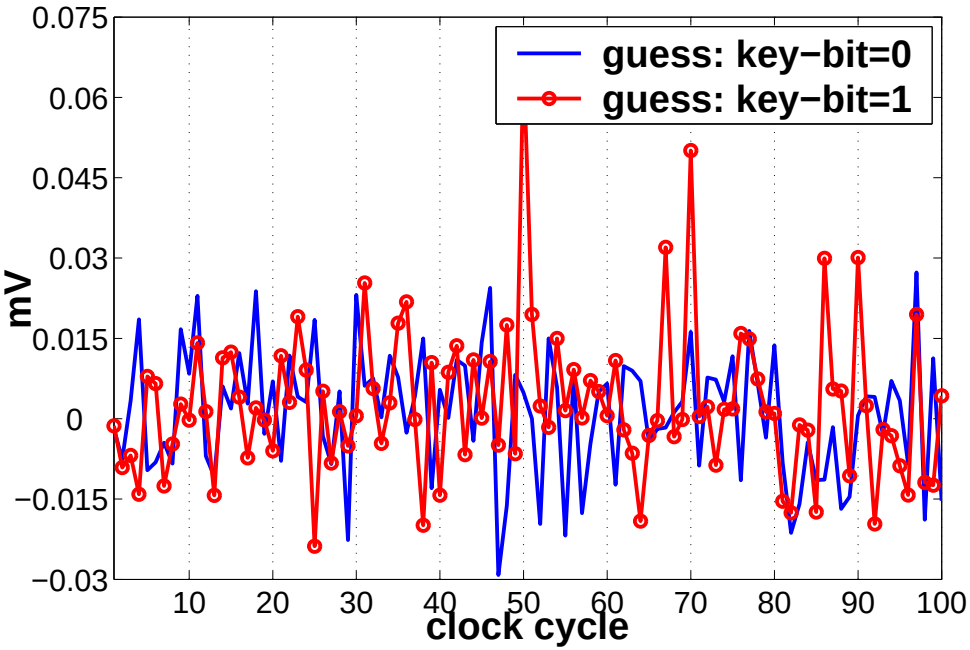


DEMA on the FPGA Implementation of ECC



DEMA on an FPGA Implementation of ECC





Countermeasures

Very few articles describe countermeasures against an EMA analysis. A complete shielding of Smart Card controllers, known from devices used in electronic data processing, is possible, but an attacker could simply remove the shield prior to analysis, making this countermeasure worthless [27].

With these presumptions in mind, EMA countermeasures have to reach much further than the commonly known PA defense systems, due to the fact that EMA attacks may provide information about small chip areas, whereas the PA measurement only yields data concerning the supply current of the complete chip.

The EM SideChannel(s)

EM emanations arise as a consequence of current flows within the control, I/O, data processing or other parts of a device [**?**]. These flows and resulting emanations may be intentional or unintentional. Each current carrying component of the device not only produces its own emanations based on its physical and electrical characteristics but also affects the emanations from other components due to coupling and circuit geometry.

Types of EM Emanations

1. Direct Emanations:

   These result from intentional current flows. Many of these consist of short bursts of current with sharp rising edges resulting in emanations observable over a wide frequency band. Often, components at higher frequencies are more useful to the attacker due to noise and interference prevalent in the lower bands. In complex circuits, isolating direct emanations may require use of tiny field probes positioned very close to the signal source and/or special filters to minimize interference: getting good results may require decapsulating the chip packaging.

2. Unintentional Emanations:

   Increased miniaturization and complexity of modern CMOS devices results in electrical and electromagnetic coupling between components in close proximity. Small couplings, typically ignored by circuit designers, provide a rich source of compromising emanations. These emanations manifest themselves as modulations of carrier signals generated, present

or “introduced” within the device. One strong source of carrier signals is the ubiquitous harmonicrich “squarewave” clock signal. Other sources include communication related signals. Ways in which modulation occurs include:

(a) Amplitude Modulation:

Nonlinear coupling between a carrier signal and a data signal results in the generation and emanation of an Amplitude Modulated (AM) signal. The data signal can be extracted via AM demodulation using a receiver tuned to the carrier frequency.

(b) Angle Modulation:

Coupling of circuits also results in Angle Modulated Signals (FM or Phase modulation). For instance, while signal generation circuits should ideally be decoupled from data processing circuits, this is rarely achieved in practice. For example, if these circuits draw upon a limited energy source the generated signal will often be angle modulated by the data signal. The data signal is recoverable by angle demodulation of the generated signal.

Acoustic Attacks

Recently, Shamir and Tromer present their results using the sound of a central processing unit (CPU) as a side-channel information in [59]. The oldest eavesdropping channel, namely acoustic emanations, has received little attention. Shamir and Tromer's preliminary analysis of acoustic emanations from personal computers shows them to be a surprisingly rich source of information on CPU activity.

Several desktop and laptop computers have been tested and in all cases it was possible to distinguish an idle CPU from a busy CPU. For some computers, it was also possible to distinguish various patterns of CPU operations and memory access. This can be observed for artificial cases (e.g., loops of various CPU instructions), and also for real-life cases (e.g., RSA decryption).

A low-frequency (KHz) acoustic source can yield information on a much faster (GHz) CPU in two ways. First, when the CPU is carrying out a long operation, it may create a characteristic acoustic spectral signature. Second, temporal information about the length of each operation is learnt and this can be used to mount TA, especially when the attacker can affect the input to the operation.

Countermeasures

One obvious countermeasure is to use sound dampening equipment, such as "sound-proof" boxes, that is designed to sufficiently attenuate all relevant frequencies. Conversely, a sufficiently strong wide-band noise source can mask the informative signals, though ergonomic concerns may render this unattractive. Careful circuit design and high-quality electronic components can probably reduce the emanations. Alternatively, one can employ known algorithmic techniques to reduce the usefulness of the emanations to attacker. These techniques ensure the rough-scale behavior of the algorithm is independent of the inputs it receives; they usually carry some performance penalty, but are often already used to thwart other side-channel attacks.

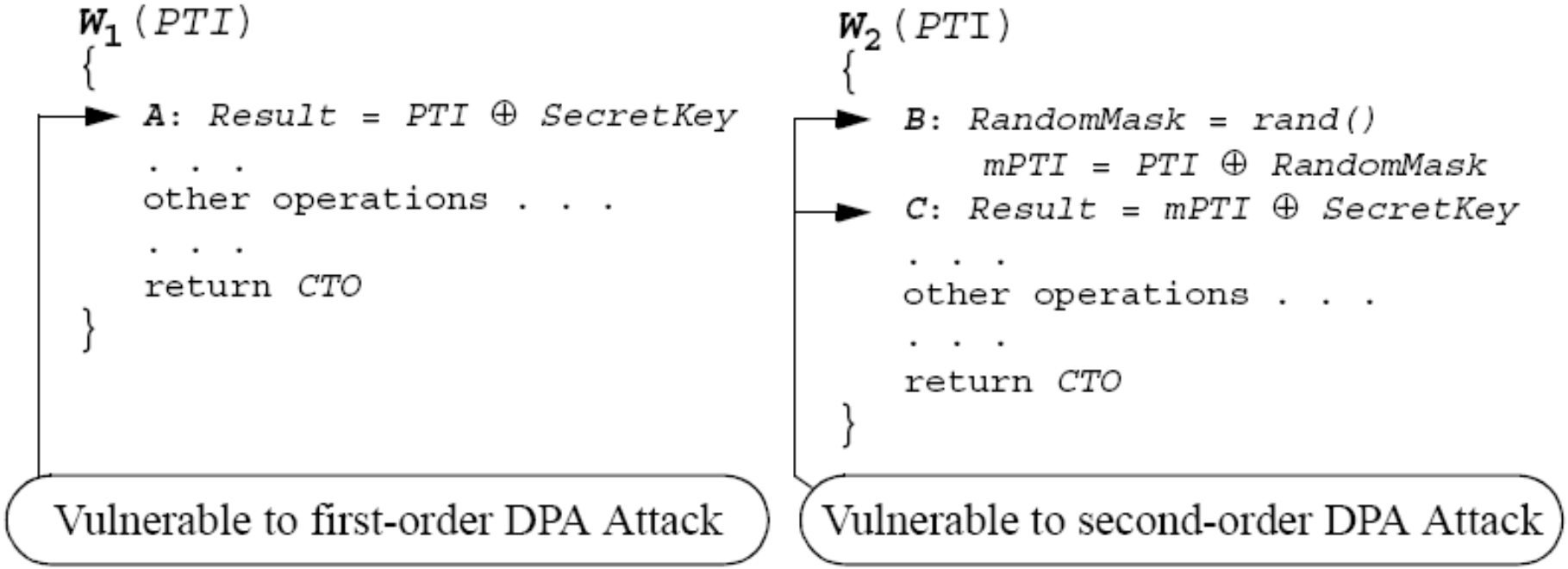
High Order Attacks

The attacker calculates joint statistical properties of the power consumption at multiple sample times within the power signals.

Definition 1. An nth-order DPA attack makes use of n different samples in the power consumption signal that correspond to n different intermediate values calculated during the execution of an algorithm [43].

- $P[j]$: the power consumption at a particular time j
- $P[j] = \varepsilon.d[j] + L + n$
- $d[j]$: Hamming weight of the intermediate data result at time j
- ε: the incremental amount of power for each extra '1' in the Hamming weight
- L: the additive constant portion of the total power
- n: the noise

Example Data-Whitening Routines



When considered jointly, the operations at lines B and C are vulnerable to a second-order DPA attack.

First-Order DPA Attack

Proposition 1. When the W_1 algorithm is implemented in an N-bit processor, where there is a linear relationship between the instantaneous power consumption and the Hamming weight of the data being processed, the following DPA attack is sound:

1. *Repeat for i equal to* 0 *through* $N-1$ {
2. *Repeat for* $b = 0$ *to* 1 {
3. *Calculate the average power signal* $A_b[j]$ *by repeating the following:*{
4. *Set the ith bit of the PTI input to b.*
5. *Set the remaining PTI bits to random values.*
6. *Collect the algorithm's power signal.* } }
7. *Create the DPA bias signal* $T[j] = A_0[j] - A_1[j]$.
8. $T[j]$ *will have a positive bias spike when the ith secret key bit is a one, and will have a negative DPA bias spike when* ith *secret key bit is a zero.* }

Second-Order DPA attack

Proposition 2. When the W_2 algorithm is implemented in an N-bit processor, where there is a linear relationship between the instantaneous power consumption and the Hamming weight of the data being processed, the following second-order DPA attack is sound:

1. *Repeat for i equal to* 0 *through* $N-1$ {
2. *Repeat for* $b = 0$ *to* 1 {
3. *Calculate average statistic* $\bar{S}_b = \left| P_B - P_C \right|$ *by repeating the following:*{
4. *Set the ith bit of the PTI input to b.*
5. *Set the remaining PTI bits to random values.*
6. *Collect the algorithm's instantaneous power consumption as lines B and C. Call these values* P_B *and* P_C, *respectively.* } }
7. *Calculate the DPA bias statistic* $T = \bar{S}_0 - \bar{S}_1$.
8. *If* $T > 0$ *then the ith key bit is a one, otherwise it is a zero.*

Proof of the Second-Order DPA attack

$P_B = d_B\varepsilon_B + L_B$ and $P_C = d_C\varepsilon_C + L_C$

- d_B: the Hamming weight of the data RandomMask at line B
- d_C: the Hamming weight of the data Result at line C

to simplify the proof, we initially assume that

- $L_B = L_C$ and $\varepsilon_B = \varepsilon_C$

The experimental results confirmed that above assumptions are true for the implementation considered. In the general case these equalities may not hold.

Proof of the Second-Order DPA attack

$|P_B - P_C| = \varepsilon |d_B - d_C|$

- k_i: ith bit of the variable SecretKey
- r_i: ith bit of the random variable RandomMask
- p_i: ith bit of PTI

$E[d_B | r_i = 1] = E[d_C | r_i \oplus k_i \oplus p_i = 1] = (N+1)/2$
$E[d_B | r_i = 0] = E[d_C | r_i \oplus k_i \oplus p_i = 0] = (N-1)/2$
If $p_i = 0$
$\overline{S_0} = \frac{1}{2}E[\varepsilon | d_B - d_C || r_i = 1, k_i = 0] + \frac{1}{2}E[\varepsilon | d_B - d_C || r_i = 1, k_i = 1] + \frac{1}{2}E[\varepsilon | d_B - d_C || r_i = 0, k_i = 1] + \frac{1}{2}E[\varepsilon | d_B - d_C || r_i = 0, k_i = 0]$
if $k_i = 0$
$\overline{S_0} = \frac{1}{2}E[\varepsilon | d_B - d_C || r_i = 1, k_i = 0] + \frac{1}{2}E[\varepsilon | d_B - d_C || r_i = 0, k_i = 0] = 0$
If $p_i = 1$
$\overline{S_1} = \frac{1}{2}E[\varepsilon | d_B - d_C || r_i = 1, k_i = 1] + \frac{1}{2}E[\varepsilon | d_B - d_C || r_i = 1, k_i = 0] + \frac{1}{2}E[\varepsilon | d_B - d_C || r_i = 0, k_i = 0] + \frac{1}{2}E[\varepsilon | d_B - d_C || r_i = 0, k_i = 1]$

if $k_i = 0$

$\overline{S_1} = \frac{1}{2}E[\varepsilon|d_B - d_C||r_i = 1, k_i = 0] + \frac{1}{2}E[\varepsilon|d_B - d_C||r_i = 0, k_i = 0] = \frac{1}{2}\varepsilon|(N+1)/2 - (N-1)/2| + \frac{1}{2}\varepsilon|(N-1)/2 - (N+1)/2| = \varepsilon$

$T = \overline{S_0} - \overline{S_1} = -\varepsilon$

In the case where $k_i = 1$, $\overline{S_0} = \varepsilon$, $\overline{S_1} = 0$.

When $T < 0$ $k_i = 0$ and when $T > 0$ $k_i = 1$

Hence, the sign of T indicates the value of k_i.

When the equality assumption of εs is not true, the situation can be handled through a process of normalization.

Instead of calculating $\overline{S_0}$ and $\overline{S_1}$ by directly using P_B and P_C, normalized versions of P_B and P_C can be used.

By using normalized values, the equality assumption is effectively forced to be true.

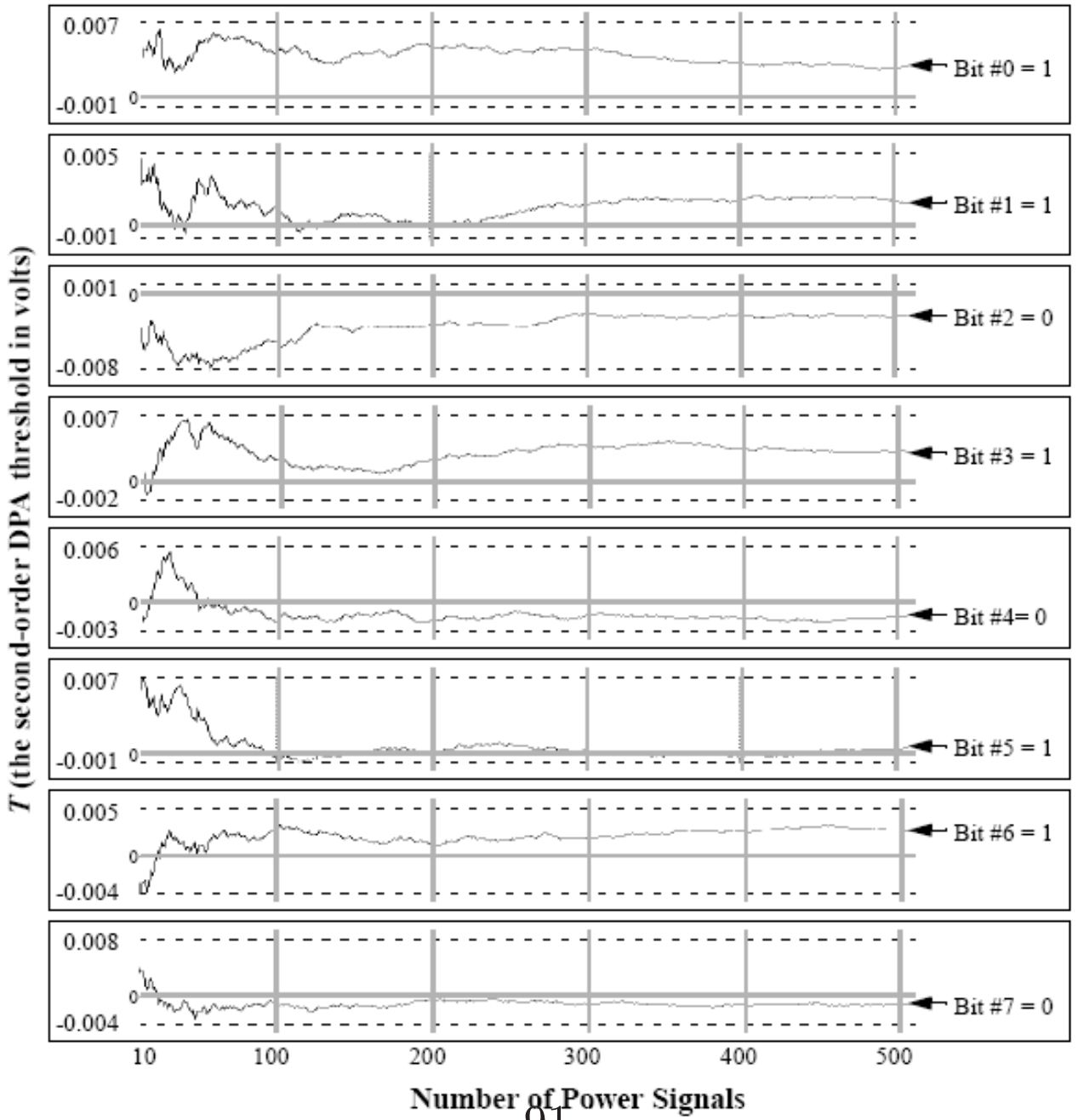
normalized$P_B = \frac{P_B - E[P_B]}{var[P_B]}$

Experimental Results

In a first-order DPA attack, knowledge of design information is not required. In a second-order DPA attack, however, knowledge of the algorithm code and the processor operation is much more important. Without such knowledge, attackers will not know which points in the power consumption signal are important.

In this example, the byte being attacked is equal to 0x6B

An interesting observation is that T converges at different rates for different bits in a byte. For some bits, T converged quickly; fewer than 50 power signals were needed. However, for other bits, T converged more slowly. For example, bit #5 requires about 2,500 power signals before T stabilizes to the correct sign. In general, the convergence of T in the second-order attack is slower and more erratic than in the first-order attack. Surprisingly, however, for some bits, T converges nearly as fast for both attacks.


0.007
-0.001
0
Bit #0 = 1
0.005
-0.001
0
Bit #1 = 1
0.001
0
-0.008
Bit #2 = 0
0.007
0
-0.002
Bit #3 = 1
0.006
0
-0.003
Bit #4= 0
0.007
-0.001
0
Bit #5 = 1
0.005
0
-0.004
Bit #6 = 1
0.008
0
-0.004
Bit #7 = 0
T (the second-order DPA threshold in volts)
10
100
200
300
400
500
Number of Power Signals

References

[1] M.-L. Akkar, R. Bevan, P. Dischamp, and D. Moyart. Power analysis, what is now possible... In Tatsuaki Okamoto, editor, *Proceedings of the 6th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology - ASIACRYPT*, volume 1976 of *Lecture Notes in Computer Science*, pages 489–502, Kyoto, Japan, December 3-7 2000. Springer-Verlag.

[2] M.-L. Akkar and C. Giraud. An implementation of DES and AES, secure against some attacks. In Ç. K. Koç, D. Naccache, and C. Paar, editors, *Proceedings of 3rd International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 2162 of *Lecture Notes in Computer Science*, pages 309–318, Paris, France, May 13-16 2001. Springer-Verlag.

[3] L. Batina. Power attacks on cryptographic algorithms. Mtd thesis, TU Eindhoven, April 2001.

[4] L. Batina and C. Jansen. Secret exponent information leakage for timing analyses. In B. Macq and J.-J. Quisquater, editors, *Proceedings of the 23rd Symposium on Information Theory in the Benelux*, pages 225–232, Louvain-la-Neuve, Belgium, May 29-31 2002. Werkgemeeschap voor Informatie-en-Communicatietheorie, Enschede, The Netherlands.

[5] E. Biham and A. Shamir. Power analysis of the key scheduling of the AES candidates. In *Proceedings of the Second Advanced Encryption Standard (AES) Candidate Conference*, Rome, Italy, 1999.

[6] D. Boneh, R. A. DeMillo, and R. J. Lipton. On the importance of checking cryptographic protocols for faults (extended abstract). In W. Fumy, editor, *Advances in Cryptology: Proceedings of EUROCRYPT'97*, volume 1233 of *Lecture Notes in Computer Science*, pages 37–51, Konstanz, Germany, May 11-15 1997. Springer-Verlag.

[7] J. Borst, B. Preneel, and J. Vandewalle. Power analysis: Methods and countermeasures. In J. Biemond, editor, *21st Symposium on Information Theory in the Benelux*, Wassenaar, The Netherlands, May 25-26 2000. Springer-Verlag.

[8] D. Brumley and D. Boneh. Remote timing attacks are practical. In *Proceedings of the 12th Usenix Security Symposium*, San Antonio, Texas, USA, June 9-14 2003.

[9] V. Carlier, H. Chabanne, E. Dottax, and H. Pelletier. Electromagnetic side channels of an FPGA implementation of AES. Cryptology ePrint Archive-2004/145, 2004. http://eprint.iacr.org/.

[10] J. Cathalo, F. Koeune, and J.-J. Quisquater. A new type of timing attack: Application to GPS. In C. Walter, Ç. K. Koç, and C. Paar, editors, *Proceedings of the 5th International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 2779 of *Lecture Notes in Computer Science*, pages 291–303, Cologne, Germany, September 7-10 2003. Springer-Verlag.

[11] G. M. Clarke and D. Cooke. *A basic course in statistics.* Arnold London, 4th edition, 1998.

[12] C. Clavier, J.-S. Coron, and N. Dabbous. Differential power analysis in the presence of hardware countermeasures. In Ç. K. Koç and C. Paar, editors, *Proceedings of the 2nd International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 1965 of *Lecture Notes in Computer Science*, pages 252–263, Worcester, Massachusetts, USA, August 17-18 2000. Springer-Verlag.

[13] J.-S. Coron. Resistance against differential power analysis for elliptic curve cryptosystems. In Ç. K. Koç and C. Paar, editors, *Proceedings of the 1st International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 1717 of *Lecture Notes in Computer Science*, pages 292–302, Worcester, Massachusetts, USA, August 12-13 1999. Springer-Verlag.

[14] J. Daemen, M. Peeters, and Gilles Van Assche. Bitslice ciphers and power analysis attacks. In B. Schneier, editor, *Proceedings of the 7th International Workshop on Fast Software Encryption (FSE)*, volume 1978 of *Lecture Notes in Computer Science*, pages 134–149, New York, NY, USA, April 10-12 2000. Springer-Verlag.

[15] E. De Mulder, P. Buysschaert, S. B. Ors, P. Delmotte, B. Preneel, G. Vandenbosch, and I. Verbauwhede. Electromagnetic analysis attack on a fpga implementation of an elliptic curve cryptosystem. In *Proceedings of the International Conference on "Computer as a tool (EUROCON)*, Sava Center, Belgrade, Serbia & Montenegro, November 21-24 2005. IEEE.

[16] E. De Mulder, S. B. Ors, B. Preneel, and I. Verbauwhede. Differential electromagnetic attack on an FPGA implementation of elliptic curve cryptosystems. In *Proceedings of the World Automation Congress (WAC) 2006, the 5th International Forum on Multimedia and Image Processing (IFMIP)*, page in print, Budapest, Hungary, July 24-27 2006.

[17] J. F. Dhem. *Design of an efficient public-key cryptographic library for RISC-based smart cards*. PhD thesis, Universite Catholiqué de Louvain, UCL Crypto Group, Laboratoire de microelectronique (DICE), May 1998.

[18] J. F. Dhem, F. Koeune, P.A. Leroux, P. Mestre, J.-J. Quisquater, and J. L. Willems. A practical implementation of the timing attack. Technical Report CG-1998/1, UCL Crypto Group, Universite Catholique de Louvain, Belgium, 1998.

[19] E.Brier, C.Clavier, and F.Olivier. Optimal statistical power analysis. IACR e-print archive 2003/152, 2003.

[20] K. Gandolfi, C. Mourtel, and F. Olivier. Electromagnetic analysis: Concrete results. In Ç. K. Koç, D. Naccache, and C. Paar, editors, *Proceedings of 3rd International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 2162 of *Lecture Notes in Computer Science*, pages 255–265, Paris, France, May 13-16 2001. Springer-Verlag.

[21] L. Goubin and J. Patari. DES and differential power analysis the "duplication" method. In Ç. K. Koç and C. Paar, editors, *Proceedings of the 1st International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 1717 of *Lecture Notes in Computer Science*, pages 158–172, Worcester, Massachusetts, USA, August 12-13 1999. Springer-Verlag.

[22] G. Hachez, F. Koeune, and J.-J. Quisquater. Timing attack: what can be achieved by a powerful adversary? In A. Barbé, E. C. van der Meulen, and P. Vanroose, editors, *Proceedings of the 20th symposium on Information Theory in the Benelux*, pages 63–70, May 1999.

[23] G. Hachez and J.-J. Quisquater. Montgomery exponentiation with no final subtractions: Improved results. In Ç. K. Koç and C. Paar, editors, *Proceedings of 2nd International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 1965 of *Lecture Notes in Computer Science*, pages 293–301, Worcester, Massachusetts, USA, August 17-18 2000.

[24] H. Handschuh and H. M. Heys. A timing attack on RC5. In S. E. Tavares and H. Meijer, editors, *Proceedings of Selected Areas in Cryptography (SAC)*, volume 1556 of *Lecture Notes in Computer Science*, pages 306–318, Kingston, Ontario, Canada, August 17-18 1998. Springer-Verlag.

[25] M. A. Hasan. Power analysis attacks and algorithmic approaches to their countermeasures for Koblitz curve cryptosystems. In Ç. K. Koç and C. Paar, editors, *Proceedings of the 2nd International Workshop on Cryptograpic Hardware and Embedded Systems (CHES)*, volume 1965 of *Lecture Notes in Computer Science*, pages 93–108, Worcester, Massachusetts, USA, August 17-18 2000. Springer-Verlag.

[26] A. Hevia and M. A. Kiwi. Strength of two data encryption standard implementations under timing attacks. In C. L. Lucchesi and A. V. Moura, editors, *Proceedings of the 3rd Latin American Symposium on Theoretical Informatics*, volume 1380 of *Lecture Notes in Computer Science*, pages 192–205, Campinas, Brazil, April 20-24 1998. Springer-Verlag.

[27] P. Hofreiter and P. Laackmann. Electromagnetic espionage from smart cards attacks and countermeasures. Infineon Technologies AG, Technology Update, Smart Cards.

[28] K. Itoh, T. Izu, and M. Takenaka. Address-bit differential power analysis of cryptographic schemes OK-ECDH and OK-ECDSA. In B. S. Kaliski Jr., Ç. K. Koç, and C. Paar, editors, *Proceedings of 4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 2523 of *Lecture Notes in Computer Science*, pages 129–143, Redwood Shores, CA, USA, August 13-15 2002. Springer-Verlag.

[29] T. Izu and T. Takagi. A fast parallel elliptic curve multiplication resistant against side channel attacks. Technical Report CORR 2002-03, the Centre for Applied Cryptographic Research (CACR), University of Waterloo, 2002.

[30] T. Izu and T. Takagi. A fast parallel elliptic curve multiplication resistant against side channel attacks. In D. Naccache and P. Paillier, editors, *Proceedings of the 5th International Workshop on Practice and Theory in Public Key Cryptosystems (PKC)*, volume 2274 of *Lecture Notes in Computer Science*, pages 280–296, Paris, France, February 12-14 2002. Springer-Verlag.

[31] M. Janke and P. Laackmann. Power and timing analysis attacks against security controllers. Infineon Technologies AG, Technology Update, Smart Cards.

[32] M. Joye, A. K. Lenstra, and J.-J. Quisquater. Chinese remaindering based cryptosystem in the presence of faults. *Journal of Cryptology*, 4(12):241–245, 1999.

[33] S.-M. Kang and Y. Leblebici. *CMOS Digital Integrated Circuits: Analysis and Design.* McGraw Hill, 2002.

[34] P. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS and other systems. In N. Koblitz, editor, *Advances in Cryptology: Proceedings of CRYPTO'96*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113, Santa Barbara, CA, USA, August 18-22 1996. Springer-Verlag.

[35] P. Kocher, J. Jaffe, and B. Jun. Introduction to differential power analysis and related attacks. `http://www.cryptography.com/dpa/technical`, 1998.

[36] P. Kocher, J. Jaffe, and B. Jun. Differential power analysis. In M. Wiener, editor, *Advances in Cryptology: Proceedings of CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397, Santa Barbara, CA, USA, August 15-19 1999. Springer-Verlag.

[37] F. Koeune and J.-J. Quisquater. A timing attack against Rijndael. Technical Report CG-1999/1, UCL Crypto Group, Louvain-la-Neuve, 1999.

[38] O. Kommerling and M. G. Kuhn. Design principles for tamper resistant smartcard processors. In *Proceedings of the USENIX Workshop on Smartcard Technology*, Chicago, Illinois, USA, May 10-11 1999.

[39] S. Mangard. A simple power-analysis attack (SPA) attack on implementations of the AES key expansion. In P. J. Lee and C. H. Lim, editors, *Proceedings of 5th International Conference on Information Security and Crytography (ICISC)*, volume 2587 of *Lecture Notes in Computer Science*, pages 343–358, Seoul, Korea, November 2002. Springer-Verlag.

[40] L. T. Mc Daniel. An investigation of differential power analysis attacks on FPGA-based encryption systems. Master's thesis, Virginia Polytechnic Insitute, May 29 2003.

[41] B. Megarajan. Combinational power analysis on smart cards. Technical report, Department of Electrical & Computer Engineering, Oregon State University, Corvallis, Oregon, 2002.

[42] N. Mentens, P. Rommens, and M. Verhelst. Timing and power analysis attacks on the hardware implementation of elliptic curve cryptosystems over $GF(p)$ and $GF\ 2^m$. Master's thesis, Katholieke Universiteit Leuven, Departement Elektrotechniek - ESAT, Kasteelpark Arenberg 10, B 3001 Heverlee, Belgium, May 2003.

[43] T. S. Messerges. Using second-order power analysis to attack DPA resistant software. In Ç. K. Koç and C. Paar, editors, *Proceedings of the 2nd International Workshop on Cryptograpic Hardware and Embedded Systems (CHES)*, volume 1965 of *Lecture Notes in Computer Science*, pages 238–251, Worcester, Massachusetts, USA, August 17-18 2000. Springer-Verlag.

[44] T. S. Messerges. *Power Analysis Attacks and Countermeasures on Cryptographic Algorithms*. PhD thesis, University of Illinois, 2002.

[45] T. S. Messerges, E. A. Dabbish, and R. H. Sloan. Investigations of power analysis attacks on smartcards. In *Proceedings of the USENIX Workshop on Smartcard Technology*, Chicago, Illinois, USA, May 10-11 1999.

[46] T. S. Messerges, E. A. Dabbish, and R. H. Sloan. Power analysis attacks of modular exponentiation in smartcards. In Ç. K. Koç and C. Paar, editors, *Proceedings of the 1st International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 1717 of *Lecture Notes in Computer Science*, pages 144–157, Worcester, MA, USA, August 12-13 1999. Springer-Verlag.

[47] T. S. Messerges, E. A. Dabbish, and R. H. Sloan. Examining smart-card security under the threat of power analysis attacks. *IEEE Transactions on Computers*, 51(5):541–552, May 2002.

[48] K. Okeya and K. Sakurai. Power analysis breaks elliptic curve cryptosystems even secure against the timing attack. In B. Roy and E. Okamoto, editors, *Proceedings of the 1st International Conference in Cryptology in India (INDOCRYPT)*, volume 1977 of *Lecture Notes in Computer Science*, pages 178–190, Calcutta, India, December 10-13 2000. Springer-Verlag.

[49] K. Okeya and K. Sakurai. A multiple power analysis breaks the advanced version of the randomized addition-subtraction chains countermeasure against side channel attacks. In *Proceedings of the IEEE Information Theory Workshop (ITW)175-178*, pages 175–178, 2003.

[50] S. B. Ors, F. K. Gürkaynak, E. Oswald, and B. Preneel. Power-analysis attack on an ASIC AES implementation. In *Proceedings of the International Conference on Information Technology (ITCC)*, pages 546–552, Las Vegas, NV, USA, April 5-7 2004.

[51] S. B. Ors, E. Oswald, and B. Preneel. Power-analysis attacks on an FPGA – first experimental results. In C. Walter, Ç. K. Koç, and C. Paar, editors, *Proceedings of the 5th International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 2779 of *Lecture Notes in Computer Science*, pages 35–50, Cologne, Germany, September 7-10 2003. Springer-Verlag.

[52] E. Oswald. Enhancing simple power-analysis attacks on elliptic curve cryptosystems. In B. S. Kaliski Jr., Ç. K. Koç, and C. Paar, editors, *Proceedings of the 4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 2523 of *Lecture Notes in Computer Science*, pages 82–97, Redwood Shores, CA, USA, August 13-15 2002. Springer-Verlag.

[53] E. Oswald, S. Mangard, N. Pramstaller, and V. Rijmen. A side-channel analysis resistant description of the AES s-box. In H. Gilbert and H. Handschuh, editors, *Proceedings of 12th International Workshop on Fast Software Encryption (FSE)*, volume 3557 of *Lecture Notes in Computer Science*, pages 413–423, Paris, France, February 21-23 2005. Springer-Verlag.

[54] J.-J. Quisquater and D. Samyde. Electromagnetic analysis (EMA): Measures and counter-measures for smard cards. In I. Attali and T. Jensen, editors, *Proceedings of the International Conference on Research in Smart Cards: Smart Card Programming and Security (E-smart)*, volume 2140 of *Lecture Notes in Computer Science*, pages 200–210, Cannes, France, September 19-21 2001. Springer-Verlag.

[55] W. Schindler. A timing attack against RSA with the chinese remainder theorem. In C. Paar and Çetin Koç, editors, *Proceedings of the 2nd International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 1965 of *Lecture Notes in Computer Science*, pages 109–124, Worcester, Massachusetts, USA, Aug 17-18 2000. Springer-Verlag.

[56] W. Schindler, F. Koeune, and J.-J. Quisquater. Unleashing the full power of timing attack. Technical Report CG-2001/3, UCL Crypto Group, 2001.

[57] R. A. Serway. *Physics for scientists and engineers.* Saunders Golden sunburst series. Saunders college publishing, 1996.

[58] A. Shamir. Method and apparatus for protecting public key schemes from timing and fault attacks. US patent number 5,991,415, November 1999.

[59] A. Shamir and E. Tromer. Acoustic cryptanalysis. Preliminary proof-of-concept presentation, 2004. http://www.wisdom.weizmann.ac.il/~tromer/acoustic/.

[60] D. X. Song, D. Wagner, and X. Tian. Timing analysis of keystrokes and timing attacks on SSH. In *Proceedings of the tenth USENIX Security Symposium*, Washington, D.C., USA, August 13-17 2001.

[61] F.-X. Standaert, S. B. Örs, and B. Preneel. Power analysis attack on an FPGA implementation of AES. In *Proceedings of the 6th International Workshop on Cryptograpic Hardware and Embedded Systems (CHES)*, Lecture Notes in Computer Science, pages 30–44, Cambridge (Boston), USA, August 11-13 2004. Springer-Verlag.

[62] F.-X. Standaert, S. B. Örs, B. Preneel, and J.-J. Quisquater. Power analysis attacks against FPGA implementations of DES. In *Proceedings of International Conference on Field-Programmable Logic and its Applications (FPL)*, Lecture Notes in Computer Science, Antwerp, Belgium, August 30-September 01 2004. Springer-Verlag.

[63] F.-X. Standaert, L. van Oldeneel, D.Samyde, and J.-J. Quisquater. Power analysis of fpgas, how practical is the attack? In *Proceedings of International Conference on Field-Programmable Logic and its Applications (FPL)*, volume 2278 of *Lecture Notes in Computer Science*, pages 701–711, Lisbon, Portugal, September 1-3 2003. Springer-Verlag.

[64] C. D. Walter. Montgomery exponentiation needs no final subtraction. *Electronic letters*, 35(21):1831–1832, October 1999.

[65] C. D. Walter. MIST: An efficient, randomized exponentiation algorithm for resisting power analysis. In B. Preneel, editor, *Proceedings of RSA 2002 Cryptographers' Track*, volume 2271 of *Lecture Notes in Computer Science*, pages 53–66, San Jose, USA, February 18-2 2002. Springer Verlag.