Introduction to Side Channel Analysis*

Dr. Torsten Schütze
Robert Bosch GmbH
Corporate Sector Research and Advance Engineering
Software (CR/AEA)

September 28, 2009

*Work on this topic was done while the author was at Siemens AG, CT IC 3. The material has been presented at previous Summer Schools of the Studienstiftung des Deutschen Volkes.

Outline: Introduction to Side Channel Analysis

Not for distribution!

1. **Introduction: What are Side Channel Attacks?**
2. **Simple Power Analysis (SPA)**
3. **Timing Analysis (TA)**
4. **Differential Power Analysis (DPA)**
5. **Template Attacks**
6. **Differential Fault Analysis (DFA)**
7. **Electromagnetic Analysis (SEMA, DEMA)**
8. **Micro-Architectural Analysis**
9. **Side Channel Attacks — Conclusions**

The Dangerous Bend Sign: to mark any part that is "more obscure than others" with a "special symbol . . . to warn about esoterica" [D. Knuth]

1. Introduction: What are Side Channel Attacks?

The name of the game

> In cryptography, a **side channel attack** is any attack based on information gained from the physical *implementation* of a cryptosystem, rather than the weaknesses in the mathematical algorithms (compare cryptanalysis).
>
> From Wikipedia, the free encyclopedia.

Why are they important?

- Most efficient threats against smart cards, embedded systems and cryptographic hardware
- Much more efficient than classical methods of cryptanalysis
- Resistance to attacks mandatory for security certification (cf. Digital Tachograph / ITSEC E3 high)

Example: Attacks against DES

	Brute Force Search	DPA Analysis
time:	22 h (1999!)	approx. 20 min
hardware:	DES Cracker ($250000), 100000 computers	oscilloscope, standard equipment
Triple DES:	not feasible	approx. 1 h

Theory vs. Practice — What are cryptographic side channels?

Theoretical Cryptography — Algorithm:

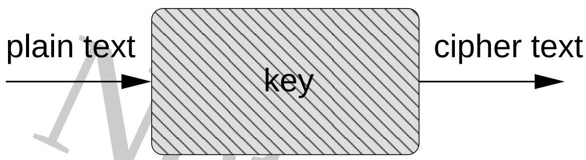
- Mathematical specification of algorithms (exact definition of input, output and attack potential)
- Attacker exploits theoretical weaknesses of algorithm

Practical Cryptography — Implementation:

- Even strong algorithms can be implemented the wrong way
- Attacker gathers additional information from physical *side channels*
  - run-time of operations
  - power consumption
  - electromagnetic radiation
  - behaviour under fault conditions

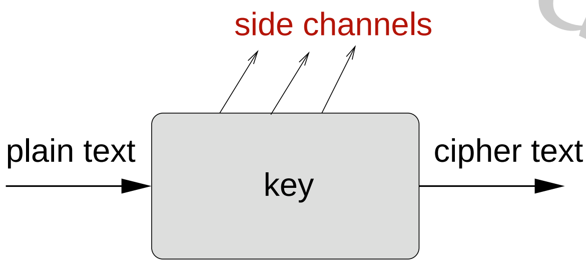
  and exploits them for attacking the implementation
- Meaningful modeling of side channels is difficult

Encryption (Black Box Model)



For state-of-the-art algorithms, brute force search is the most efficient attack method $\Longrightarrow$ Use large key space such that brute force is impractical!

Encryption (Side Channel Model)



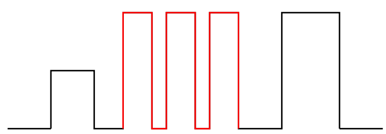
Using side channel information, one can attack interim results which depend only on a few key bits (sub key), i. e., brute force attack on sub keys becomes possible!

Example: Verification of a secret PIN

```
const char top_secret[] = "44250382";
printf("Enter␣PIN:␣");
scanf("%8s", input);
for (i = 0; i < strlen(top_secret); i++)
if (top_secret[i] != input[i]) {
  printf("Wrong␣PIN.\n");
  return FALSE;
}
printf("PIN␣correct.\n");
return TRUE;
```

Possible side channels:

- Run-time between input and response
- Power consumption profile of loop:



Theory: 8-digit PIN, 10 potential trials per digit $\Longrightarrow 10^8$ experiments for brute force search

Practice: unskilled implementation — *perform* 10 *experiments successively per digit*
$\Longrightarrow 10 + \cdots + 10 = 80$ potential experiments

Effort reduction from 100.000.000 to 80!

The most important side channels and their attacks

- **Run-time**
  - **Timing Analysis (TA):** statistical evaluation of run-time variations
  - **Cache Attacks:** information leakage through memory caches
- **Power consumption**
  - **Simple Power Analysis (SPA):** *direct* interpretation of power consumption during *one* cryptographic operation
  - **Differential Power Analysis (DPA):** statistical evaluation of power traces from *many* operations
  - **Template Attacks:** combine techniques from SPA and DPA. Training device (full control of attacker) for exact characterization, *direct key extraction* on target device
- **Electromagnetic fields**
  - **Electromagnetic Analysis (SEMA, DEMA):** evaluation of electromagnetic radiation during cryptographic operations
- **Behaviour under fault conditions**
  - **Differential Fault Analysis (DFA):** exploit faulty results

History of Side Channel Attacks

Cryptography in military/authorities

- Measurement of emissions and application of countermeasures standard practice for a long time
- Focus: correlation between *plain text* and *side channel information*
- Extent of expertise not clear, only a few documents freely available, see TEMPEST

Rediscovery of attack technique by P. Kocher et al.

- 1996 Timing Analysis, 1997 Differential Fault Analysis, 1997 Differential Power Analysis
- Basic idea: find correlation of *key material* with *side channel information*

Status of SCA

- Side channel attacks and countermeasures are active research area
- Many publications about attacks on implementations of common algorithms
- Resistance to side channel attacks is mandatory for security certification with mechanism strength “high”

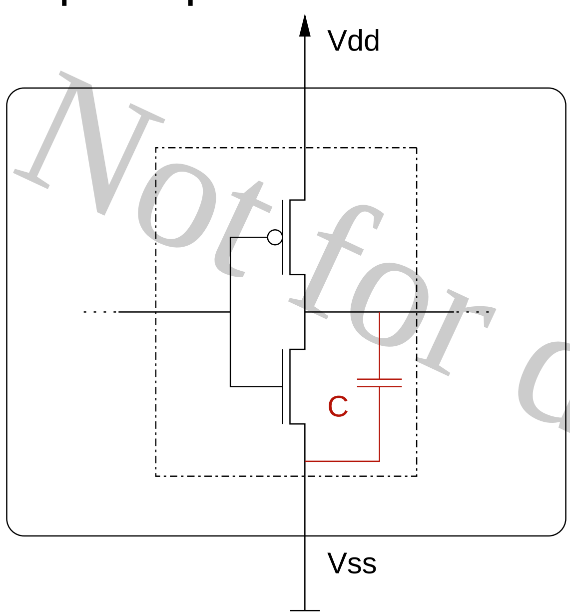
2. Simple Power Analysis (SPA)

Methodology, Assumptions and Goals:

- Collect *one power profile* and *directly extract/compute the secret key*
- Requires oscilloscope and standard equipment $\Rightarrow$ relatively easy to perform
- Normally, the attacker requires *detailed knowledge of implementation* to understand the power profile
- SPA gives information on:
  - data dependent switches, calls of subprograms
  - Hamming weights/Hamming distances of registers as well as data on address and data busses
- Exact power leakage model is architecture dependent!

Power side channel is intrinsic to CMOS technology

Example: Simplest circuit = inverter



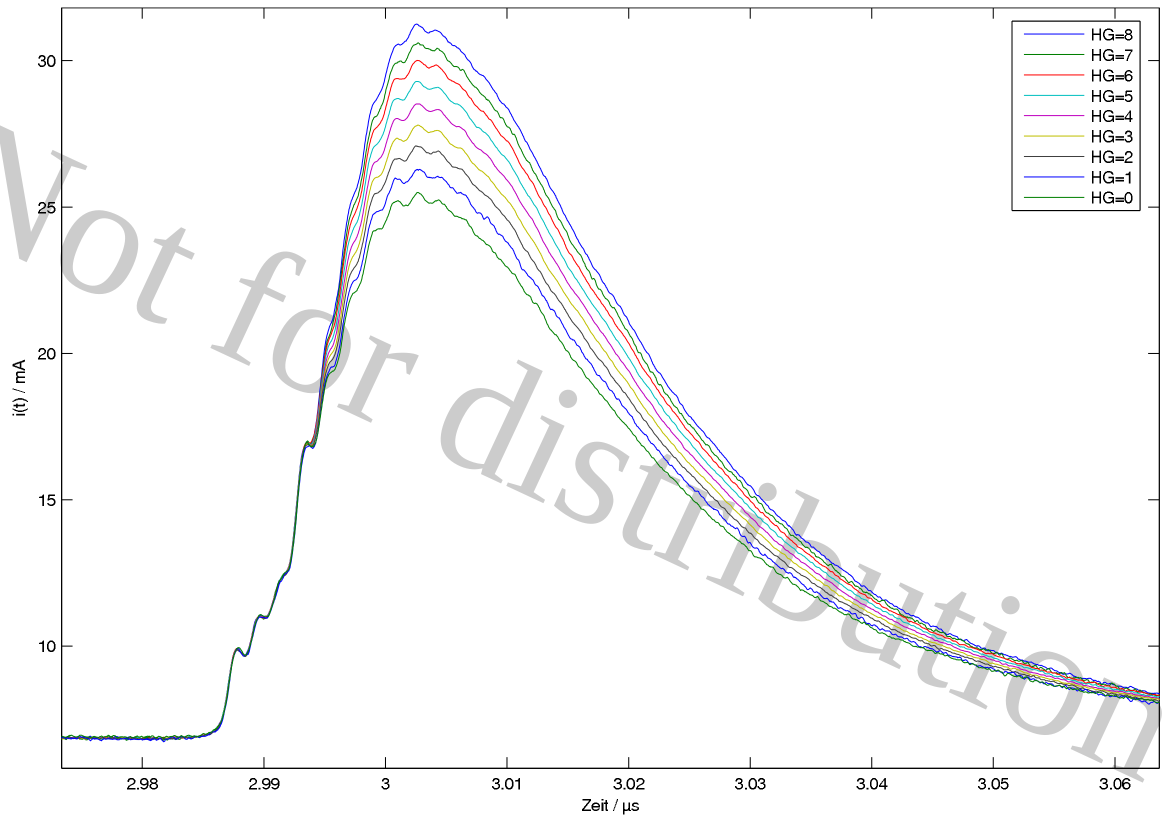
Parasitic capacity C (geometry of wires) results in charging current

Flow through a CMOS-Inverter

transition	current flow
$0 \to 0$	only leakage current
$0 \to 1$	leakage current, short circuit and charging current
$1 \to 0$	leakage current, short circuit current
$1 \to 1$	only leakage current

For different Hamming weights consider above circuit diagram in parallel
$\Longrightarrow$ charging current is data dependent!

Data dependency of power profiles from Hamming weight of data



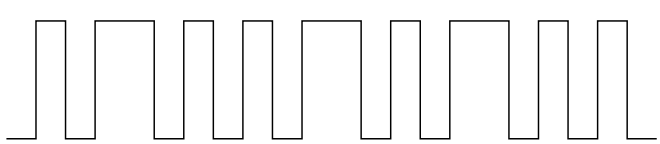
Example: SPA of a simple RSA implementation

RSA implementation with Square-and-Multiply Algorithm (left-to-right binary exponentiation):

$c = b^e \bmod m$, base b, secret exponent $e = (e_1, \ldots, e_k)$, modulus m

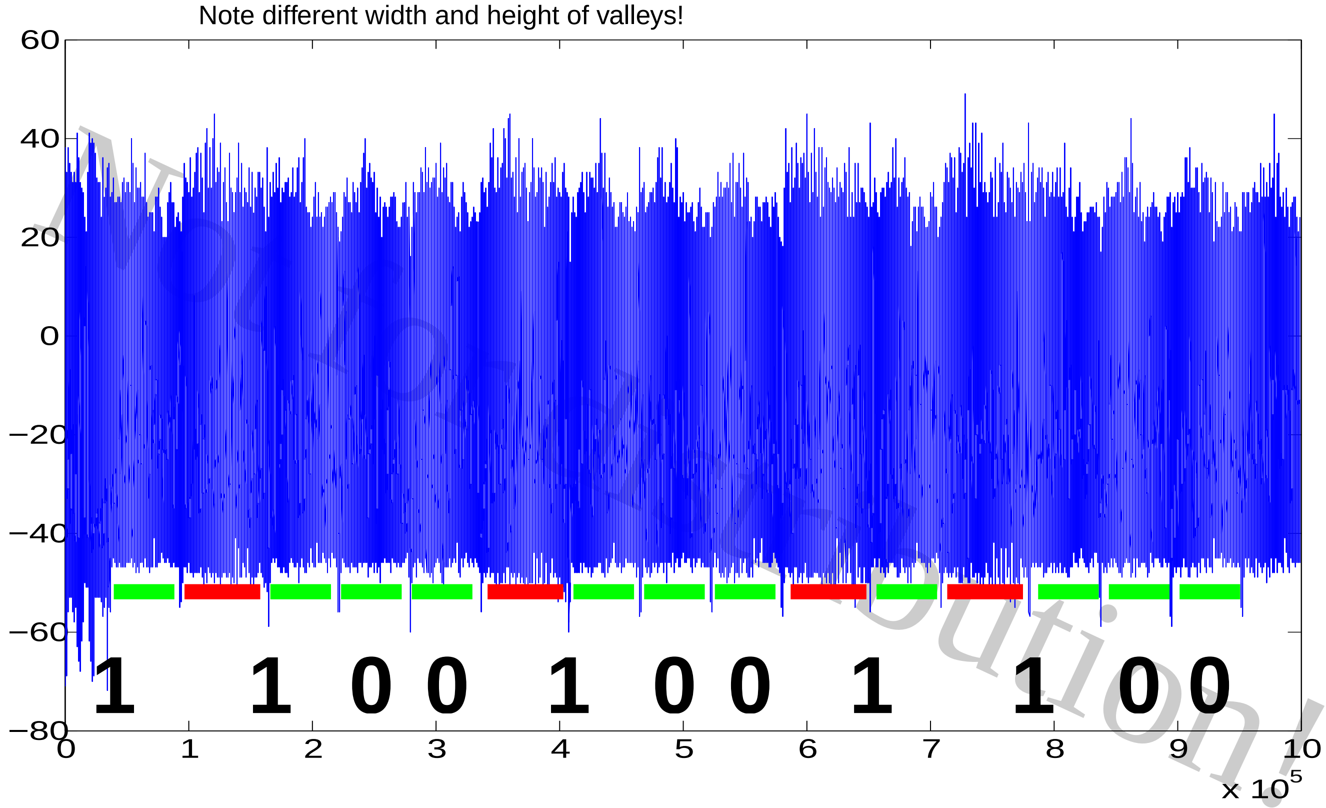
```
c = 1;
for (i = 1; i <= k; i++) {
  c = mod_square(c, m); /* c <- c*c mod m */
  if (e[i] == 1)
    c = mod_mult(c, b, m); /* c <- c*b mod m */
}
```

Schematic power profile for Square-and-Multiply:



Power profile of Square-and-Multiply Algorithm on ST10F168

Note different width and height of valleys!



General countermeasures against SPA

- Decrease signal-to-noise ratio (random noise, dual rail logic, pre-charge logic)
- Hide implementation details
- Avoid key dependent branches, try to achieve uniform behaviour $\Longrightarrow$
  - redundancy: always-square-and-multiply
  - uniform algorithm: Montgomery Powering Ladder
  - defensive programming of branches
- Effective countermeasures (for a medium/high security level) are only possible with hardware and software together

3. Timing Analysis (TA)

- Uses key dependent run-time variations
- Requires *exact emulation of run-time!*
- Easier to implement than DPA, but less powerful

Example: Attack over Ethernet-LAN on OpenSSL = Remote Timing Attack

- D. Boneh/D. Brumley: *Remote Timing Attacks are Practical*, Usenix 2003.
- Based on W. Schindler: *A Timing Attack against RSA-CRT*, CHES 2000.
- Attack on RSA-CRT, Sliding Windows, Montgomery Multiplication, Karatsuba
- Requires for 1024 bit RSA: 1 mio decryptions, 2 h
- Attack can be drastically improved (personal communication with W. Schindler, papers in Statist. Decisions 2002 and 12th ACM Conf. on CCS 2005)

$\Longrightarrow$ Boneh/Brumley paper resulted in a number of Prio 1 Security Advisories e. g. for OpenSSL
$\Longrightarrow$ **Side Channel Resistance is important even for distant servers!!**

Math refresher: Background on stochastic/statistics

Let X be a random variable with cumulative distribution function F_X and probability density function f_X.

Sample: $(x_1, \ldots, x_n)$ realization of random variable $(X_1, \ldots, X_n)$

Standard assumption: random variables X_i, $i = 1, \ldots, n$ are i.i.d. (independent, identically distributed)

Expected value: $\mathrm{E}\,X := \int_{-\infty}^{\infty} x\, dF_X(x)$, continuous $\int_{-\infty}^{\infty} x f_X(x)\, dx$,
discrete $\sum_i x_i P(X = x_i)$

Variance: $\mathrm{Var}\,X := \mathrm{E}(X - \mathrm{E}\,X)^2 = \int_{-\infty}^{\infty} (x - \mathrm{E}\,X)^2\, dF_X(x)$
continuous $\int_{-\infty}^{\infty} (x - \mathrm{E}\,X)^2 f_X(x)\, dx$,
discrete $\sum_i (x_i - \mathrm{E}\,X)^2 P(X = x_i)$

expected value = measure of location, variance = measure of dispersion

Math refresher: Background on stochastic/statistics (2)

- Unbiased estimators for expected value and variance

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \qquad \text{mean}$$

$$\hat{\sigma}^2 = \frac{1}{n-1}\sum_{i=1}^{n} \left(X_i - \bar{X}\right)^2 \qquad \text{sample (empirical) variance}$$

- Normal distribution $N(\mu, \sigma^2)$
  probability density function:

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

$\mathrm{E}\,X = \mu$ expected value, $\mathrm{Var}\,X = \sigma^2$ variance

Example: Timing Analysis Square-and-Multiply (Kocher[96])

Assumption: bits $e_1, \ldots, e_{d-1}$ of exponent are already known, e_d will be attacked

1. Measure run-time T_i for N modular exponentiations with random (known) base b_i, $i = 1, \ldots, N$
$$T_i = T\left(b_i^e \bmod m\right)$$
2. For $e_d = 0$: Compute (via emulation) time $T_{i,d}^0$ until bit e_d is reached
   Compute variance $V_0 := \mathrm{Var}(T_i - T_{i,d}^0)$
3. For $e_d = 1$: Compute (via emulation) time $T_{i,d}^1$ until bit e_d is reached
   Compute variance $V_1 := \mathrm{Var}(T_i - T_{i,d}^1)$
4. Decision rule
$$e_d = \begin{cases} 0 & \text{if } V_0 < V_1, \\ 1 & \text{else.} \end{cases}$$

Why does the attack work?

- **Observation:** Wrong guess of bit e_d leads to increasing empirical variance!
- **Proof:** Elementary operations with variances, see next slide

Proof: Why does the Timing Attack work?

- Time to compute $b_i^e \bmod m$: $T_i = c_i + \sum_{j=1}^{k} t_{i,j}$
- $t_{i,j}$ — time for j-th iteration of exponentiation for base b_i, $\tilde{t}_{i,d}$ — emulated time for step d, c_i — measurement error, noise
- Can be computed via emulation:

$$T_{i,d} = \sum_{j=1}^{d-1} t_{i,j} + \tilde{t}_{i,d} = \sum_{j=1}^{d} t_{i,j} \quad \text{if bit } e_d \text{ is known}$$

- Assumption: run-times are i.i.d. random variables
- Distinction of cases

  Case 1: guess for e_d is correct

$$\begin{aligned}\mathrm{Var}(T_i - T_{i,d}) &= \mathrm{Var}\left(c_i + \sum_{j=1}^{k} t_{i,j} - \sum_{j=1}^{d} t_{i,j}\right) = \mathrm{Var}\left(c_i + \sum_{j=d+1}^{k} t_{i,j}\right)\\ &= \mathrm{Var}(c) + (k-d)\,\mathrm{Var}(t)\end{aligned}$$

Proof: Why does the Timing Attack work? (2)

- Case 2: guess for e_d is wrong

$$\begin{aligned}\mathrm{Var}(T_i - T_{i,d}) &= \mathrm{Var}\left(c_i + \sum_{j=1}^{k} t_{i,j} - (\tilde{t}_{i,d} + \sum_{j=1}^{d-1} t_{i,j})\right)\\ &= \mathrm{Var}\left(c_i - \tilde{t}_{i,d} + \sum_{j=d}^{k} t_{i,j}\right)\\ &= \mathrm{Var}(c) + (k-d)\,\mathrm{Var}(t) + 2\,\mathrm{Var}(t)\end{aligned}$$

Note: $\mathrm{Var}(t) + \mathrm{Var}(-\tilde{t}) = \mathrm{Var}(t) + (-1)^2\,\mathrm{Var}(\tilde{t}) = 2\,\mathrm{Var}(t)$

- **Observation:** Wrong guess of bit e_d leads to increasing empirical variance!

General Countermeasures against TA

- Algorithms with constant run-time
- Random variations of run-time/random wait states
- Masking of data (prohibits emulation of run-times)

  $\Longrightarrow$ “base point blinding” = Chaum’s blind signatures for $c = b^e \bmod m$:

  - Choose random number λ and compute $\mu = (\lambda^{-1})^e \bmod m$
  - Blinding of message $\hat{b} = \lambda b \bmod m$
  - Exponentiation $\hat{c} = \hat{b}^e \bmod m$
  - Unblinding $c = \mu \hat{c} \bmod m$
  - Efficient Updating of (λ, μ) necessary

  Chaum’s blind signatures have been developed for e-Money in 1982
- Masking of exponent and modulus, see DPA

4. Differential Power Analysis (DPA)

Methodology, Assumptions and Goals:

- *Several measurements of power consumption* while performing cryptographic operations and *statistical evaluation* of data dependent variations to extract secret keys
- Requires oscilloscope to acquire power profiles, PC with evaluation software, standard equipment
- *Does not require detailed knowledge of implementation*: It suffices, if certain values are computed at any time somehow or other
- DPA uses statistical methods to increase/amplify the weak signals of individual bits of sub keys in the power traces

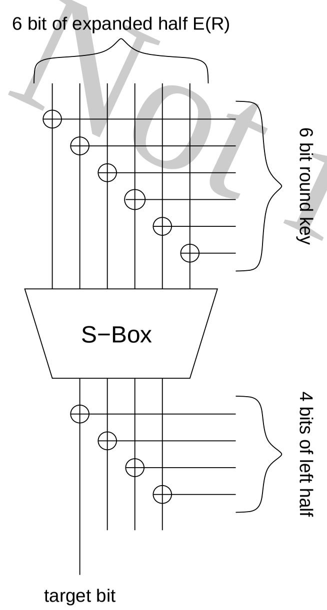
Basic Prerequisites for Differential Power Analysis

1. Physical access to processor/board to measure power consumption during cryptographic operations
2. Known plaintext or known ciphertext
3. **One intermediate result that occurs during the cryptographic operation is a function of the cipher-/plaintext and a small number of key bits.**
4. The power consumption of the device is data dependent.

S. Mangard, PhD thesis, 2004

Example: Standard DPA on DES Implementation

Attack first output bit of first S-box in first round of DES Encryption: $(R_0, L_0 \oplus P(S(E(R_0) \oplus K_0)))$



- R_0 and L_0 are known to the attacker (known plaintext attack), trial and error for round key K_0 (only $2^6 = 64$ values!)
- For all possible values of K_0: divide measured power traces into two sets, set C^0 ("target bit is 0") and C^1 ("target bit is 1")
- Compute difference D between mean value of power traces from C^0 and mean value of power traces from C^1
- Value of K_0 with highest peak of D is most probable round key for this S-box
- Repeat attack with other S-boxes

Described attack: difference of means, Kocher attack (1998)

Example: Standard DPA on DES Implementation — detailed —

1. Encrypt N random known plaintexts Y_i, $i = 1, \ldots, N$
   Collect discrete time-power-profile C_i

$$C_{ij} = \{ \text{ power consumption for plaintext } Y_i \text{ at time } t_j \}$$

   Amplification of signal building mean $\bar{C}$

$$\bar{C}_j = \frac{1}{N} \sum_{i=1}^{N} C_{ij}$$

2. Choose target bit, e. g., first output bit of first S-Box in first DES round.
   Note: value b of target bit depends only on 6 sub key bits (and plaintext)!
3. Statistical hypotheses: $H_0 : k_6 = k_6^0$ "guess sub key"
   Important: Under H_0 we can compute b for known plaintext Y_i

$$\text{Classification} \qquad C^0 := \{C_{ij} | b = 0\}, \quad C^1 := \{C_{ij} | b = 1\}$$

Example: Standard DPA on DES Implementation — detailed — (2)

4. Compute mean $\bar{C}^0 = \{\frac{1}{N_0} \sum_{i=1}^{N_0} C_{ij} | b = 0\}$, $N_0 = \#\{C_{i.} | b = 0\}$,
   mean $\bar{C}^1$ analogously

   Case 1: "H_0 is wrong" $\Longrightarrow$ $\bar{C}^0$ and $\bar{C}^1$ are statistically indistinguishable
   (computed value b is wrong in approx. 50%)
   Case 2: "H_0 is correct" $\Longrightarrow$ $\bar{C}^0$ and $\bar{C}^1$ are correlated
   $\Longrightarrow$ high peaks in difference of mean values $\bar{C}^0 - \bar{C}^1$ (bias profile, differential trace)

   Decision rule:
   If $\bar{C}^0$ and $\bar{C}^1$ are with small probability of error statistically indistinguishable $\Longrightarrow$ Reject H_0 and select different hypotheses H_0 (2^6 possible values)
   $\Longrightarrow$ Else "sub key bits k_6^0 found"
5. Repeat steps 1–4 for remaining 7 S-Boxes $\Longrightarrow$ 48 sub key bits found

Consolidation: Some statistics related to Kocher attack

Two sets of power traces = random variables $(X_1, \dots, X_{N_0})$ and $(Y_1, \dots, Y_{N_1})$; X_i and Y_i are independent and identically distributed with unknown cumulative distribution function $X_i \sim F_X$ and $Y_i \sim F_Y$, respectively.
Measured power profiles = realized samples of a random experiment with results x_i and y_i:

$$x_i \in \mathbb{R}^M, i = 1, \dots, N_0 \quad \text{and} \quad y_i \in \mathbb{R}^M, i = 1, \dots, N_1.$$

1. Nonparametric test = no distribution assumption on F_X and F_Y

$$H_0 : F_X = F_Y \quad \Longleftrightarrow \quad H_1 : \exists x : \quad F_X(x) \neq F_Y(x).$$

- If H_0 cannot be rejected, then the hypotheses (sub key guess) was not correct with overwhelming probability.
- If H_0 with probability of error α is rejected, then the measurements come from different distributions and the sub key is found.

Possible statistical tests: Kolmogorov-Smirnov-Test, Wilcoxon-Rank-Sum-Test

Consolidation: Some statistics related to Kocher attack (2)

2. Parametric test, e. g., normal distribution with equal unknown variances $\sigma^2 = \sigma_X^2 = \sigma_Y^2$ and different means μ_X, μ_Y, i.e., $X_i \sim \mathrm{N}(\mu_X, \sigma_X^2)$ and $Y_i \sim \mathrm{N}(\mu_Y, \sigma_Y^2)$.

$$H_0 : \mu_X = \mu_Y \quad \Longleftrightarrow \quad H_1 : \mu_X \neq \mu_Y$$

Check hypotheses with two-sample t-test, test statistics

$$T = \frac{\bar{X} - \bar{Y}}{S_P} \sqrt{\frac{N_0 N_1}{N_0 + N_1}}$$

with

$$S_P^2 = \frac{1}{N_0 + N_1 - 2}((N_0 - 1)S_x^2 + (N_1 - 1)S_y^2),$$

$$S_x^2 = \frac{1}{N_0 - 1} \sum_{i=1}^{N_0} (x_i - \bar{X})^2, \quad S_y^2 = \frac{1}{N_1 - 1} \sum_{i=1}^{N_1} (y_i - \bar{Y})^2.$$

Consolidation: Some statistics related to Kocher attack (3)

- Under H_0 we have $T \sim t_{N_0+N_1-2}$, i. e., rejection of H_0, if $|T| \geq t_{N_0+N_1-2,1-\frac{\alpha}{2}}$
- Kocher attack = neglect S_P, N_0 and N_1 in test statistics T (constant values under assumption of equal variances)
- $\Longrightarrow$ Difference of means
- Difference curve with highest peak comes with overwhelming probability from correct key hypotheses

Alternative method: Correlation Power Attack

- Measurement of N power traces for known plaintext Y_i: $C_{ij}, i = 1, \ldots, N, j = 1, \ldots, M$
- Compute target bit b for all N plaintexts and all 64 key hypotheses:

$$B_{ki}, k = 1, \ldots, 64, i = 1, \ldots, N$$

- Compute Pearson's correlation coefficient $\rho(X, Y) := \frac{\mathrm{Cov}(X,Y)}{\sqrt{\mathrm{Var}(X)\,\mathrm{Var}(Y)}}$:

$$\rho_{kj}(B_{ki}, C_{ij}) = \frac{\mathrm{E}(B_{ki} \cdot C_{ij}) - \mathrm{E}(B_{ki}) \cdot \mathrm{E}(C_{ij})}{\sqrt{\mathrm{Var}(B_{ki}) \cdot \mathrm{Var}(C_{ij})}}$$

- Every row of matrix ρ will be interpreted as correlation curve.
  Correlation curve with highest peak belongs to most probable key!
- Formalization: E. Brier, C. Clavier, F. Olivier: *Correlation Power Analysis with a Leakage Model*, CHES 2004.

Modifications of Kocher Attack

Multiple Bit DPA

- Consider 4 target bits $b = (b_1, \ldots, b_4)$
- Build sets C^i depending on the Hamming weight of b:

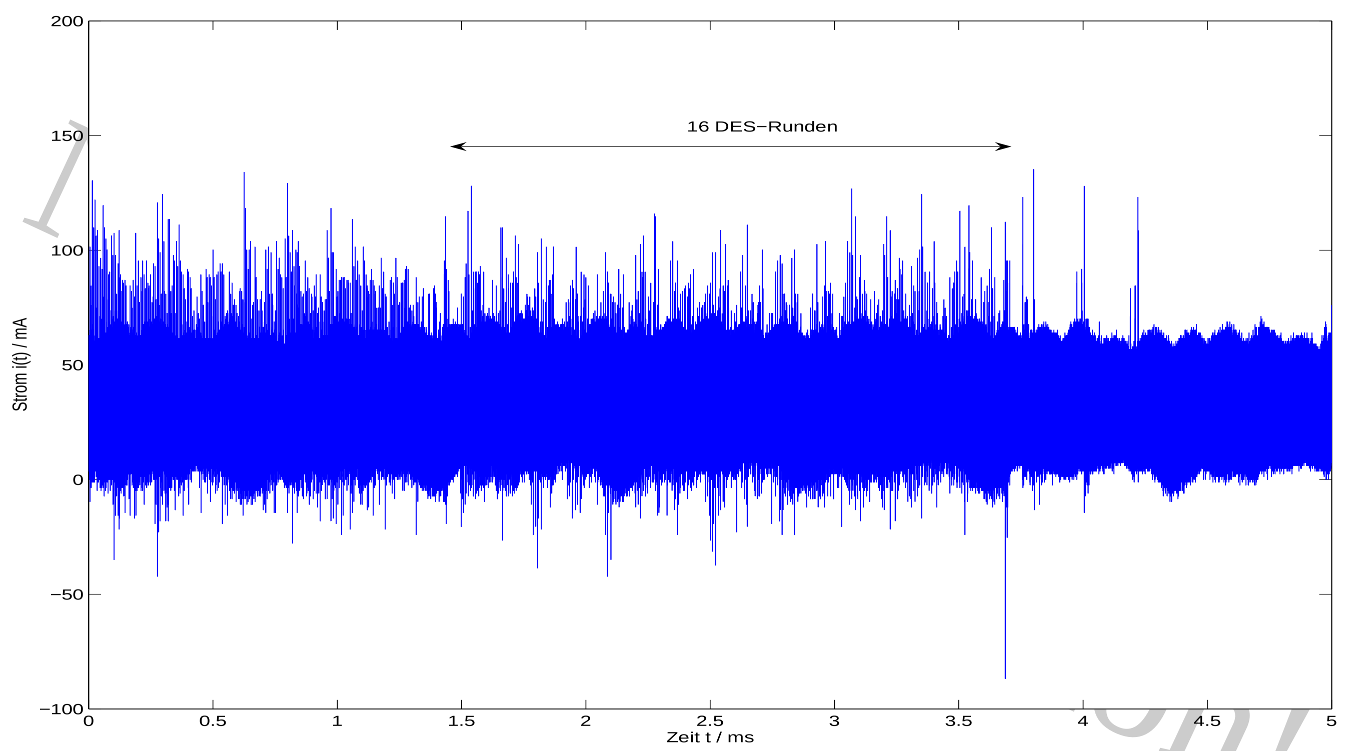
$$C^0 := \{C_{ij} | \operatorname{hw}(b) \leq l\}, C^1 := \{C_{ij} | \operatorname{hw}(b) \geq u\}, C^2 := \{C_{ij} | l < \operatorname{hw}(b) < u\}$$

- C^2 will be discarded: Multiple Bit DPA needs more samples $\Longrightarrow$ but better for low signal-to-noise ratio

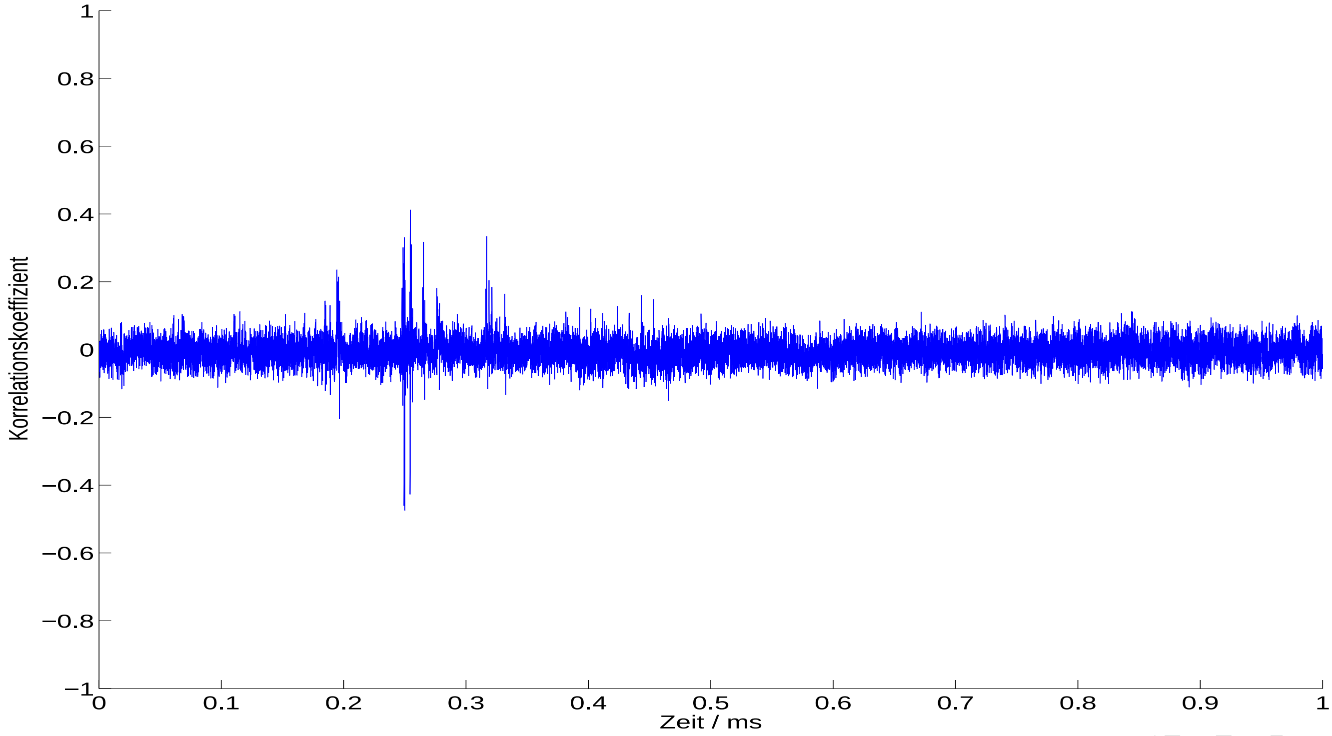
Higher Order DPA

- All attacks presented so far are first order attacks, i. e., one evaluation point / target bits is attacked, no detailed knowledge about implementation is necessary.
- Improvement: Second and Higher Order DPA
  - Attack, e.g., target bits of different rounds together
  - Requires detailed knowledge of implementation
  - Can overcome masking countermeasures, see later
  - Example: Thomas Messerges, *Using Second-Order Power Analysis to Attack DPA Resistant Software*, CHES 2000.

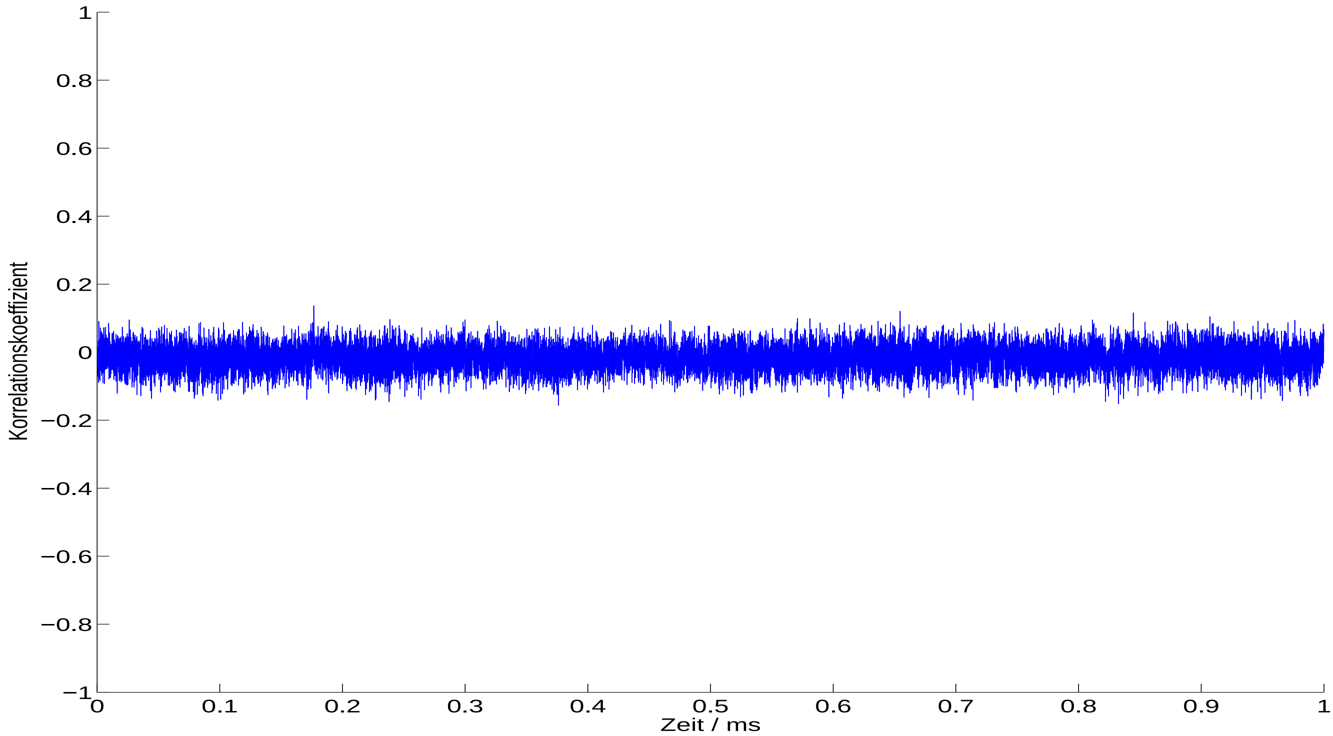
Example: Power trace of a DES computation on ST10F168



Correlation curve, 1000 measurements (insecure implementation)



Correlation curve, 1000 measurements (secured implementation)



Example: Masking of DES due to Akkar/Giraud

Unsecured DES:

$$L_{i+1} = R_i \quad \text{and} \quad R_{i+1} = L_i \oplus P\left(S\left(E\left(R_i\right) \oplus K_i\right)\right)$$

Secured DES due to Akkar/Giraud [2001]:

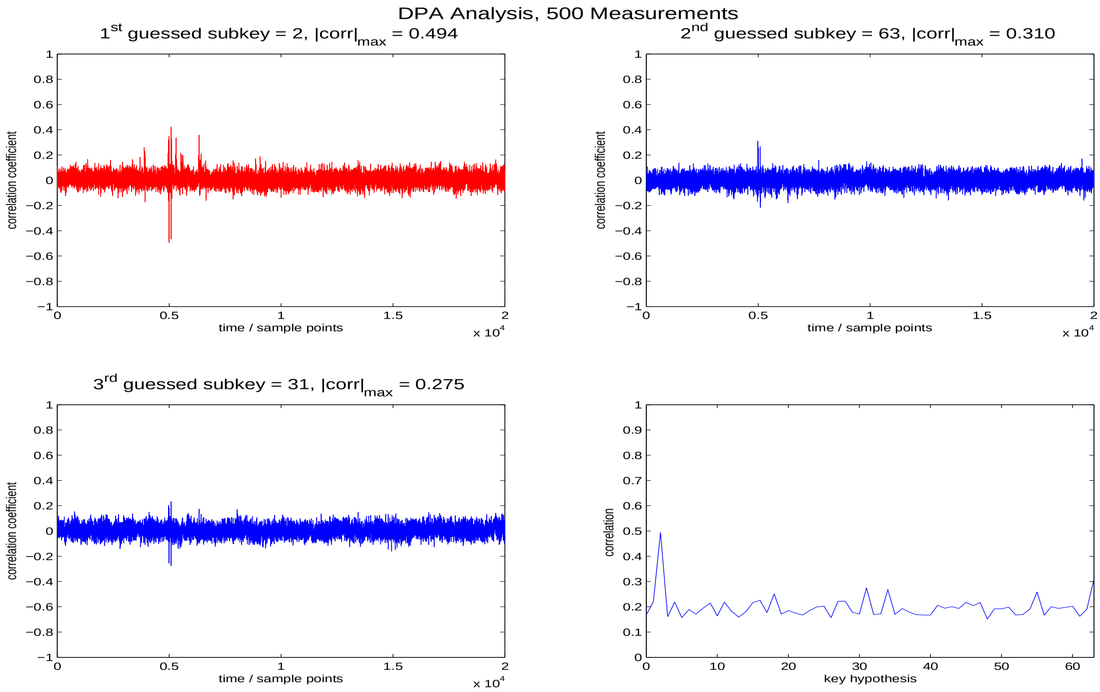
- Generate two random 32-Bit masks X_L and X_R; $L'_1 = L_1 \oplus X_L$, $R'_1 = R_1 \oplus X_R$
- Modified round function
  $L'_{i+1} = R'_i \oplus (X_L \oplus X_R) \quad \text{and} \quad R'_{i+1} = L'_i \oplus P\left(SM\left(E\left(R'_i\right) \oplus K_i\right)\right)$
- Masked S-Box:
  $$SM(A) = S(A \oplus E(X_R)) \oplus P^{-1}(X_L \oplus X_R)$$
- Every round is masked by the same value, i. e.,
  $L'_i = L_i \oplus X_L$, $R'_i = R_i \oplus X_R$
- After the last round the mask has to be removed, i. e.,
  $L_{16} = L'_{16} \oplus X_L$, $R_{16} = R'_{16} \oplus X_R$
- Main problem: modified S-Box is variable, i. e., RAM instead of SP-Box in ROM

General countermeasures against DPA

- Decrease signal-to-noise ratio $\Longrightarrow$ more measurements required
- Random Wait States $\Longrightarrow$ make alignment of power traces on time scale more difficult $\Longrightarrow$ signal processing to overcome
- Randomization = De-correlate processed data from known inputs/outputs
  - Randomization of Algorithms (random addition/subtraction chains, etc.) $\Longrightarrow$ not very powerful
  - Randomization of data/Masking
    - Compute $b^{e+\text{rand}\,\varphi(m)} \bmod m$ instead of $b^e \bmod m$ for RSA
    - Use projective coordinates for elliptic curves
    - mask input for DES and AES
- In general: hardware and software countermeasures together required

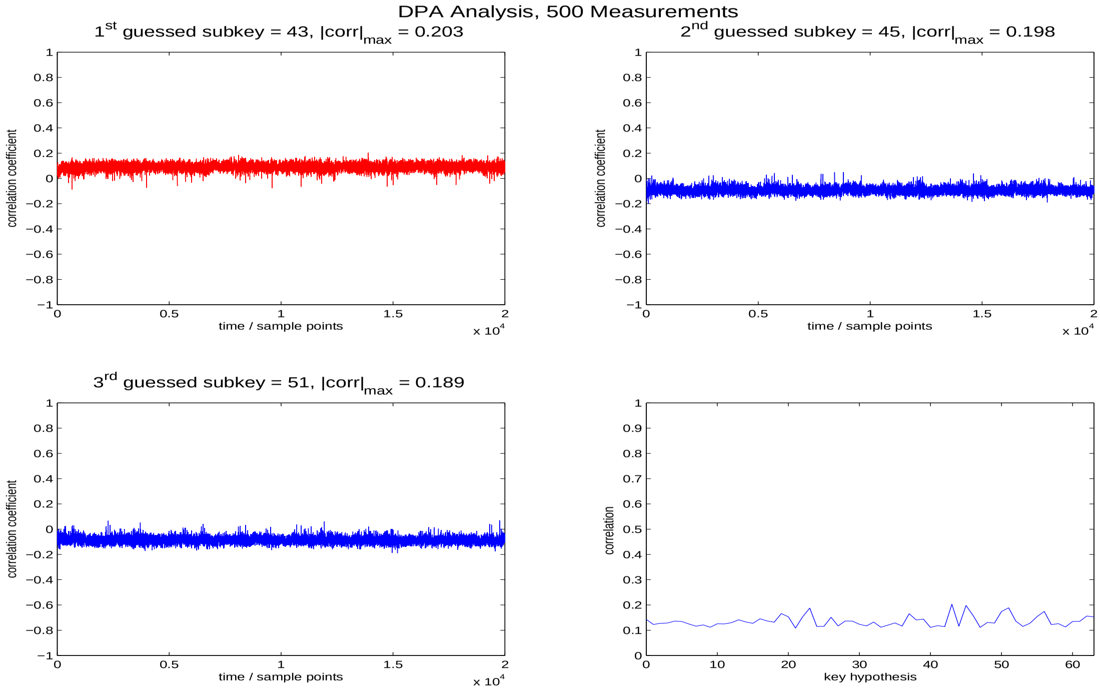
It's demo time!

Demo: Correlation Power Attack on DES implementation



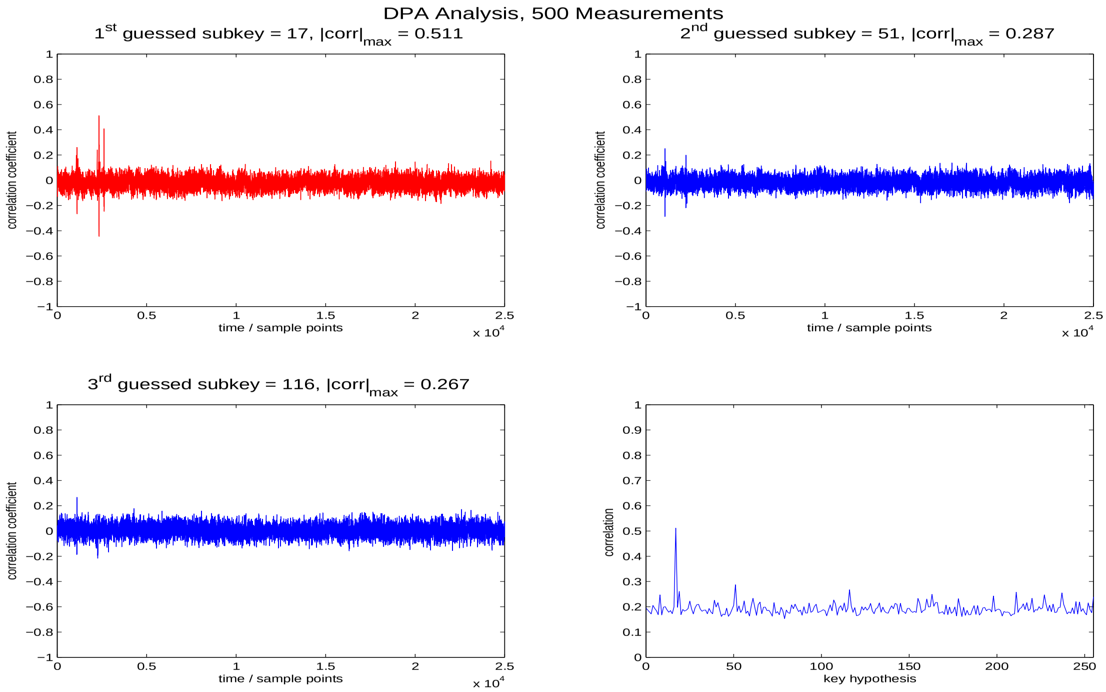
Insecure DES implementation on ST10F168 processor, 500 measurements, Source: ©Siemens CT

Demo: Correlation Power Attack on secured DES implementation



Secured DES implementation on ST10F168 processor, 500 measurements, Source: ⓒSiemens CT

Demo: Correlation Power Attack on AES implementation



Insecure AES implementation on ST10F168 processor, 500 measurements, Source: ⓒSiemens CT

5. Template Attacks

Methodology, Assumptions and Goals:

- Template Attacks combine techniques from SPA and DPA, they build a noise model and use this model during the attack
  $\Longrightarrow$ optimal attack in an information theoretical sense
- Prerequisite: Access to Training Device
- **Training Device — Profiling Step**
  - Identical to target device except the secret cryptographic key used
  - Under "full control" of attacker, i. e., key is known or can be loaded
  - Training Device is used to characterize the power leakage model with statistical methods (estimation/learning methods)
- **Target Device — Key Extraction Step**
  - Contains unknown secret key
  - Direct interpretation of *one measurement* to extract key utilizing the model from first step
    $\Longrightarrow$ can be used even for ephemeral keys or masked keys

Template Attacks (2)

Stochastic Model:

Target of evaluation: block cipher without masking

$x \in \{0,1\}^p$ known plaintext or ciphertext

$k \in \{0,1\}^s$ sub key, t time

$$\underbrace{I_t(x,k)}_{\text{random variable}} = \underbrace{h_t(x,k)}_{\text{deterministic part}} + \underbrace{R_t}_{\text{random variable, } E(R_t)=0}$$

Statistical standard notation: X, I, R, K random variables; x, i, r, k realizations/samples of random variable

Profiling Step: Use known x and k to characterize $h_t(x,k)$ and distribution of R_t for $t = t_1, t_2, \ldots, t_m$

Naïve Approach: Estimate $E(I_t(x,k))$ independently for all $(x,k) \in \{0,1\}^p \times \{0,1\}^s$

Key Extraction: Use characterization of $h_t(.,.)$ and noise distribution to find unknown key k

Template Attacks (3)

Some Well-known Template Attacks

- Chari/Rao/Rohatgi [2003]: estimate $\mathrm{E}(I_t(x,k))$ and covariance matrix of noise $\mathrm{Cov}(R_t)$
- Schindler [2005]: linear stochastic model for $h_t(x,k)$; estimator in low-dimensional (optimal) sub-spaces, details see CHES 2005 paper, Attention, statistics!
- CHES 2006: Lemke-Rust et al. *Templates vs. Stochastic Methods*, Standaert et al. *Template Attacks in Principal Subspaces*

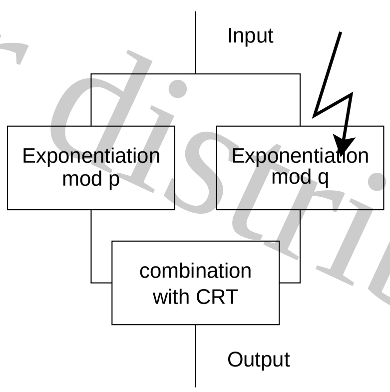
Template Attacks — Conclusions

- More efficient than SPA and DPA, statistical countermeasures against DPA can be overcome $\Longrightarrow$ give attacker no full control even on training device, e.g., no control over masking
- Today besides higher order DPA and micro-architectural side channel attacks most active research area in SCA community.

6. Differential Fault Analysis (DFA)

- Exploit faulty results which have been (actively) induced by errors
- Sources of errors: x-rays and ion beams, UV- and laser light, voltage glitches, temperature, electromagnetic induction, etc.

Famous Example: Bellcore Attack on RSA in CRT-Mode



Details on Bellcore Attack

Garner's Algorithm on RSA in CRT-Mode

$N = pq, S = m^d \bmod N, ed = 1 \bmod (p-1)(q-1)$

Precomputation: $d_p = d \bmod p-1, d_q = d \bmod q-1, P_q = p^{-1} \bmod q$

Exponentiation: $S_p = m^{d_p} \bmod p, S_q = m^{d_q} \bmod q$

Combination: $S = S_p + [((S_p - S_q) \times P_q) \bmod p] \times q$

Attack: faulty medium result $\tilde{S}_q \Longrightarrow$ faulty end result $\tilde{S}$

It holds: $S - \tilde{S} \neq 0$, but $S - \tilde{S} = 0 \bmod q$

$\Longrightarrow$ Factorization

$$\gcd\left(\left(m - \tilde{S}^e\right) \bmod N, N\right) = q$$

= Lenstra Attack [1997]

Differential Fault Analysis — Conclusions

Reconstruction of private signature key possible through (transient fault model):

- Boneh/DeMillo/Lipton [1997]: one wrong signature, one correct signature
- Lenstra [1997]: one wrong signature, original message

Biham/Shamir [1997]: Extension to symmetric methods, persistent fault model

General Countermeasures against DFA

- Introduce redundancies/invariants
- Comparison of results:
  - after encryption try decryption
  - after signature generation try signature verification
  - compute back-wards some/all rounds
- Special methods for RSA-CRT
- Many proposals for ECC countermeasures
- Note! Simple repetition of calculation is often not sufficient!

7. Electromagnetic Analysis (SEMA, DEMA)

Methodology, Assumptions and Goals:

- Measure electromagnetic field during cryptographic operations
- Analysis in simplest case as in SPA and DPA

  Differences:
  - No galvanic connection necessary, only contact-less measurement in near and far field
  - Measurement can be performed at any place in the field (interesting regions on chips as busses, crypto coprocessors, etc.)
  - Lots of data, but, in general, bad signal-to-noise ratio
  - Additional dimension per measurement: TA: scalar; SPA/DPA: time dependent vector; SEMA/DEMA: time dependent three-dimensional information
- Efficient usage of this information difficult, but large potential (signal processing know-how)

8. Micro-Architectural Analysis

- Newly evolving area of side-channel cryptanalysis
- Studies the effects of common processor components and their functionalities on the security of software cryptosystems
- Especially dangerous for virtualization technologies as Intel's LaGrande Technology (LT) and Vanderpool Technology (VT), AMD's Pacifica, ARM's Trustzone, and software based virtualization mechanisms like VMWare
- Two important classes of MA attacks, both utilize data dependent run-time

  1. cache analysis
  2. branch prediction analysis

Micro-Architectural Analysis — Cache Analysis

- Access times to different types of processor memory differ by several order of magnitudes
  $\Longrightarrow$ cache architectures, e. g., L1 code & data cache, L2 cache, Translation Lookaside Buffer
  $\Longrightarrow$ data depending timing differences due to cache access
- Fast AES implementations, for example, Barreto's method with four 1024-byte tables T_i, use table lookups, i. e., variable-index array lookups like $T_0[k[0] \oplus p[0]]$
- Main misconception: "Table lookup: not vulnerable to timing attacks" (NIST, 2001)
- But: above instruction leaks information about key $\Longrightarrow$ unprotected implementations can be broken in several milliseconds

Short history of cache attacks

- D. Page, *Theoretical Use of Cache Memory as a Cryptanalytic Side-Channel*, 2002.
- Y. Tsunoo et al., *Cryptanalysis of DES implemented on computers with cache*, CHES 2003.
- D.J. Bernstein, *Cache-timing attacks on AES*, 2004–2005.
- D.A. Osvik, A. Shamir, E. Tromer, *Cache Attacks and Countermeasures: The Case of AES*, CT-RSA 2006.
- M. Neve, *Cache-based Vulnerabilities and SPAM Analysis*, PhD Thesis, UCL, July 2006.

Micro-Architectural Analysis — Branch Prediction Analysis

- Modern pipelined processors use branch predictors, i. e., guess whether a conditional branch will be taken or not
- Timing information from branch prediction can be used in a classical timing analysis
  $\Longrightarrow$ Branch Prediction Analysis (many measurements)
- Spy process running simultaneously with crypto algorithm
  $\Longrightarrow$ analyze CPU's Branch Predictor states
  $\Longrightarrow$ *single crypto operation* reveals most key bits, e.g., 500 from 512 RSA key bits
  $\Longrightarrow$ Simple Branch Prediction Analysis

Short history of branch prediction analysis

- O. Aciiçmez, J.P. Seifert, C.K. Koç, *Predicting secret keys via branch prediction*, CT-RSA 2007.
- O. Aciiçmez, C.K. Koç, J.P. Seifert, *On the Power of Simple Branch Prediction Analysis*, 2006.
- O. Aciiçmez, W. Schindler, *A Major Vulnerability in RSA Implementations due to MicroArchitectural Analysis Threat*, 2007.

Micro-Architectural Analysis — Countermeasures and Conclusion

- Use logical operations instead of table lookups: very slow
- Preload tables: not always possible; masking and hiding: possible
- Normalize cache: very CPU specific
- Use vendor specific secured AES commands:
  Intel introduces new AES instructions for next generations processors, [Rev. 2.0, April 2009]
  - four instructions (`AESENC`, `AESENCLAST`, `AESDEC`, and `AESDELAST`) for encryption and decryption; two instructions (`AESIMC` and `AESKEYGENASSIST`) for key expansion
  - run in data independent time and do not use table lookups $\Longrightarrow$ secure against MAA
  - substantial performance speedup, especially in parallelizable modes of operation
- Micro-Architectural Analysis = essentially timing attack variant, but with much smaller effects caused by Micro-Architecture
- Difficult to protect without detailled processor knowledge
- Breaks many assumptions on compartments etc. in virtualized systems

9. Side Channel Attacks — Conclusions

Side Channel Attacks are todays most efficient threat against smart cards, embedded systems, and cryptographic hardware!

- Side Channel Analysis can be performed with standard equipment
- Without special countermeasures most implementations of cryptographic methods are vulnerable
- Efficient countermeasures are possible, but only as interaction between hardware and software
- Countermeasures cost performance and memory
- Many attacks and countermeasures are not published
- Difficult patent situation
- Secure implementation of cryptographic algorithms is experts work

Side Channel Analysis — Further Information

- E. Hess; N. Janssen; B. Meyer; T. Schütze:
  Information Leakage Attacks Against Smart Card Implementations of Cryptographic Algorithms and Countermeasures – A Survey. Proceedings of EUROSMART Security Conference 2000,
  `http://www.torsten-schuetze.de/reports/leakage.pdf`
- M. Aigner; E. Oswald: *Power Analysis Tutorial*, Technical report, IAIK Graz, 2000,
  `http://www.iaik.tu-graz.ac.at/aboutus/people/oswald/papers/dpa_tutorial.pdf`
- J. J. Quisquater; F. Koeune:
  State-of-the-art regarding side channel attacks, Technical report, Oct 2002,
  `http://www.ipa.go.jp/security/enc/CRYPTREC/fy15/doc/1047_Side_Channel_report.pdf`
- YongBin Zhou; DengGuo Feng:
  Side-Channel Attacks: Ten Years After Its Publication and the Impacts on Cryptographic Module Security Testing. NIST Physical Security Testing Workshop, Hawaii, Sep 2005.
  `http://csrc.nist.gov/cryptval/physec/physecdoc.html`
- Proceedings of CHES Conferences (Cryptographic Hardware and Embedded Systems)
- S. Mangard; E. Oswald; T. Popp:
  Power Analysis Attacks: Revealing the Secrets of Smart Cards. Springer, April 2007,
  `http://www.dpabook.org`