Optimizing the Efficiency of Side-Channel Attacks with Advanced Stochastical Methods

Werner Schindler

Bundesamt für Sicherheit in der Informationstechnik (BSI), Germany

Bochum, September 21, 2004

Outline of the talk

- Introduction
- A brief excursion into statistical decision theory
- Montgomery's multiplication algorithm (stochastical model)
- Example 1: A timing attack on RSA without CRT
- Example 2: A timing attack on RSA with CRT
- Example 3: A combined timing and power attack
- Conclusions

Side-channel attacks: General Remarks

Examples: Timing attacks, power attacks, radiation attacks, acoustic attacks, combined attacks

Usually, the useful (key-dependent) information is covered by noise. An attacker clearly aims to exploit this information in an optimal way.

Vice versa, the understanding of the true risk potential of an attack enables the choice of appropriate and reliable countermeasures.

This focus of this talk lies on the methodology but not on new attacks.

Methodology: General Remarks

In side-channel attacks the secret key is guessed in small portions. This process can be interpreted as a sequence of ***statistical decision problems***.

Statistical decision theory

- quantifies the impact of different pieces of information on the optimal decision
- enables a formal search for an optimal decision strategy

Statistical decision theory (I)

The statistician (in our context: the attacker)

- observes $\omega \in \Omega$ (here: result of a measurement)
- interprets ω as a realization of (i.e., a value assumed by) a random variable X with unknown distribution p_θ with $\theta \in \Theta$ (parameter space)
- based on ω decides for a parameter $\theta^* \in \Theta$

Note: A false decision causes a “damage” (loss).

Statistical decision theory (II)

Formally, a statistical decision problem is given by a 5-tupel $(\Theta, \Omega, s, D, A)$

Θ : set of admissible hypotheses

Ω : observation space

$s : \Theta \times A \rightarrow [0, \infty]$, loss function; quantifies the „harm“ of a wrong decision (here: false guess of a key part)

D : set of all admissible decision strategies

A : set of all possible decisions (here: $A = \Theta$ are finite)

Examples

<u>Example I:</u>

The attacker guesses single RSA key bits on basis of N timing or power measurements. Then $\Theta = A = \{0,1\}$ and $\Omega = R^N$.

<u>Example II:</u>

The attacker attacks a 6-bit DES subkey that affects a single S-box in the first round. Here $\Theta = A = \{0,1\}^6$.

Expected loss of wrong decisions

Depending on the concrete scenario it may be easier to detect and correct particular types of guessing errors than others (see Example 3).

The loss function $s : \Theta \times A \rightarrow [0, \infty]$ quantifies the consequences of wrong guesses. Clearly, $s(\theta, \theta) = 0$

A decision strategy τ is a mapping $\tau: \Omega \rightarrow A$

Expected loss for t **if** $q \in Q$ **is the true parameter:**

risk function

$$r(\theta, \tau) = \int_{\Omega} s(\theta, \tau(\omega))\, p_{\theta}(d\omega)$$

What is optimal?

Goal: Choose a decision strategy that minimizes the expected loss.

Unfortunately, there do not exist decision strategies that are simultaneously optimal for all admissible hypotheses.

However, usually there is some knowledge about the distribution of the admissible hypotheses, quantified by the a priori distribution η on the parameter space Θ.

Examples

Example I continued (bitwise guessing of an RSA exponent):

Assume that k exponent bits remain to be guessed and that the attacker knows that r of them equal 1.
Then $\eta = (\eta(0),\eta(1)) = ((k-r)/k, r/k)$.

Example II continued (attacking a 6 bit-DES subkey):

Here $\eta(x) = 2^{-6}$ for all $x \in \{0,1\}^6$.

Expected loss of wrong decisions

Expected loss for t **against the a priori distribution** η **:**

$$E_\eta(r(\theta, \tau)) = \sum_{q_i \in Q} \int_\Omega s(\theta_i, \tau(\omega))\, p_{\theta i}(d\omega)\ \ \eta(\theta_i)$$

Theorem 1: Let $p_{\theta i} = f_{\theta i} * \mu$ for all $\theta_i \in \Theta$, i.e. $p_{\theta i}$ has the μ-density $f_{\theta i}$. The decision strategy $\tau^*: \Omega \to A$, given by

$\tau^*(\omega) := a^*$ if

$$\sum_{q_i \in Q} s(\theta_i, a^*)\, \eta(\theta_i)\, f_{\theta i}(\omega) = \min_{a \in A} \left\{ \sum_{q_i \in Q} s(\theta_i, a)\, \eta(\theta_i)\, f_{\theta i}(\omega) \right\}$$

is optimal against the a priori distribution η.

The attacker

- computes $p_{\theta i} = f_{\theta i} * \mu$ for all $\theta_i \in \Theta$ (most difficult task)
- selects an appropriate loss function s(.,.)
- determines the a priori distribution η
- applies Theorem 1 to derive the optimal decision strategy

The distributions $p_{\theta i}$ have the most significant impact on the optimal decision strategy.

Montgomery‘s multiplication algorithm (I)

n (modulus)

$n < R$ Montgomery‘s constant

Input: $a,b \in Z_n$

Output: $MM(a,b;n) := abR^{-1} \pmod n$

In particular: $MM(aR,bR;n) := abR \pmod n$

Montgomery‘s multiplication algorithm (II)

ws word size (tailored to the given hardware)

$r = 2^{ws}$, $R = r^t$

$n' := -n^{-1} \pmod r$, $RR^{-1} \pmod n - nN^* = 1$ (in Z)

Input: $a = (a_{t-1}, ..., a_0)_r$, $b = (b_{t-1}, ..., b_0)_r < n$

Montgomery's multiplication algorithm (III)

$s := 0$

for i=0 to t-1 do {

$u_i := (s_0 + a_i b_0)\, n'$ (mod r)

$s := (s + a_i b + u_i n) / r$

}

if $(s \geq n)$ then $s := s - n$ ⟵ **EXTRA REDUCTION**

return s = MM(a,b;n) $(= abR^{-1} \pmod n)$

Montgomery's multiplication algorithm (IV)

$\text{Time}(MM(a,b;n)) \in \{c, c+c_{ER}\}$ (at least) for essentially all input values of the same size.

time for an extra reduction

For random input values the time needed for a Montgomery operation can be viewed as a two-valued random variable.

Montgomery's multiplication algorithm (V)

Regardless of the wordsize ws,

$$\frac{(MM(a,b;n))}{n} = \left(\frac{a}{n} \frac{b}{n} \frac{n}{R} + \frac{abN^* \pmod R}{R} \right) \pmod 1$$

The term within the brackets corresponds to the value of s before the ER step.

Distribution for random input ?

Montgomery's multiplication algorithm (VI)

Theorem 2: Let the r.v. B be equidistributed on Z_n. Then the intermediate result before the ER step is (in good approximation) distributed as

$$\frac{n}{R}\frac{a}{n}\,U + V \qquad \text{for MM(a,B;n)}$$

$$\frac{n}{R}\,U^2 + V \qquad \text{for MM(B,B;n)}$$

where U and V denote independent and equidistributed random variables on [0,1).

Mod. exponentiation algorithm with Montgomery

1) $\underline{y_1} := MM(y, R^2; n)$

2) modular exponentiation algorithm (replace the multiplications and squarings with the respective MM operations)

   a) table initialization (if necessary)

   b) exponentiation phase

3) return temp := MM(temp,1,n) $(= y^d \pmod n)$

Stochastical model (I)

Interpret the normalized intermediate values

$$\frac{temp_0}{n} = \frac{y_t}{n}, \frac{temp_1}{n}, \frac{temp_2}{n}, \ldots$$

within the exponentiation phase as realizations of [0,1)-valued random variables $S_0, S_1, \ldots$

Stochastical model (II)

$$S_i := \begin{cases} S_{i-1}\cdot(y_t/n)\cdot g + V_i \pmod 1 & \text{mult. with table entry } y_t \\ S_{i-1}^2\cdot g + V_i \pmod 1 & \text{squaring} \end{cases}$$

$g := n/R$; $V_1, V_2, \ldots$ random variables, independent and equidistributed on $[0,1)$

Consequence: The random variables $S_1, S_2, \ldots$ are independent and equidistributed on $[0,1)$.

Stochastical model (III)

Theorem 3: $\mathrm{Time}(MM(S_i,S_i;n)) = c+c_{ER}*W_i$ and

$\mathrm{Time}(MM(S_i,y_t;n)) = c+c_{ER}*W_i$ where

W_i is a $\{0,1\}$-valued random variable

$$W_i := \begin{cases} 1_{S_i<S_{i-1}}\,(y_t / n)\,g & \text{mult. with table entry } y_t \\ 1_{S_i<S_{i-1}}{}^2 g & \text{squaring} \end{cases}$$

Study the stochastic process $W_1, W_2, \ldots$

Properties of the stochastic process $W_1, W_2, \ldots$

$$E(W_i) := \begin{cases} \frac{1}{2} \frac{y_t}{n} \frac{n}{R} \\ \frac{1}{3} \frac{n}{R} \end{cases}$$

mult. with table entry y_t

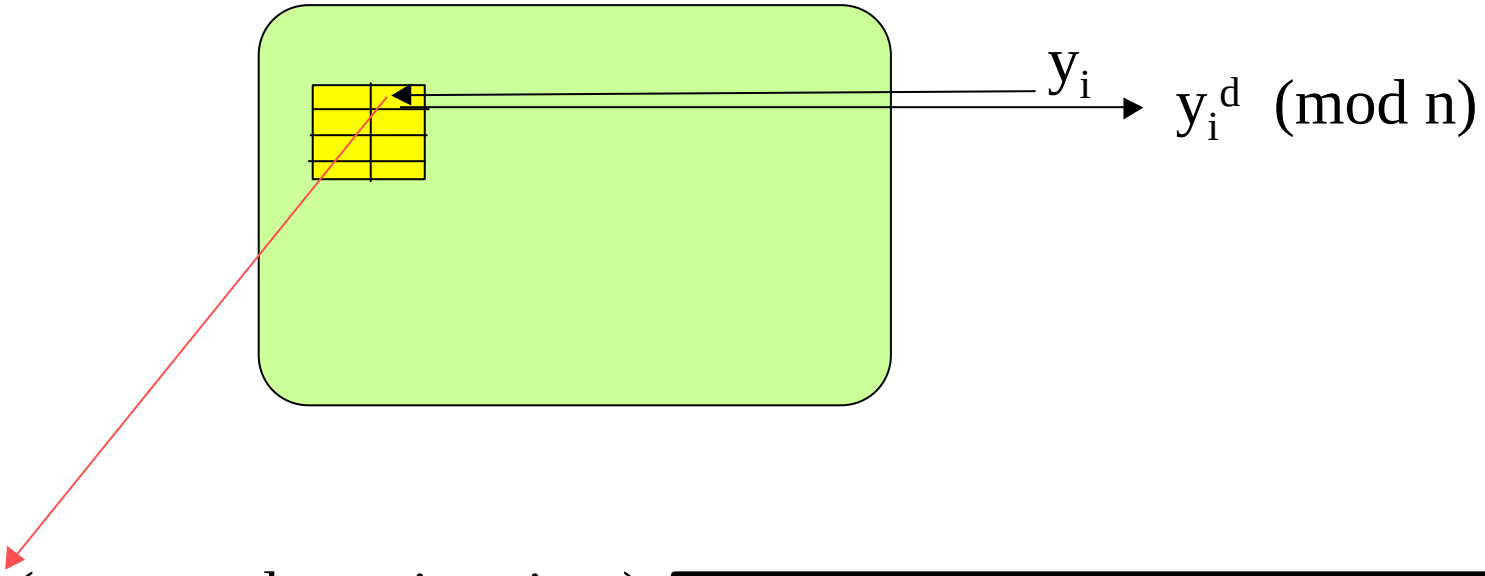
squaring

probability for an ER

W_i and W_{i+1} are negatively correlated

W_i and W_h are independent if $|i-h| > 1$

Timing attacks – basic idea



Timing attacks exploit time differences required for different input values.

Timing attacks: Kocher (1996), Quisquater et al. (1998), Schindler (2000, 2002), Schindler, Koeune, Quisquater (2001), Brumley, Boneh (2003)

Example 1:
A timing attack against RSA without CRT

Assumptions:

- The device computes $y^d \pmod n$ with the square & multiply exp. algorithm and Montgomery‘s algorithm
- d, n fixed; no blinding

s&m with Montgomery's algorithm

computes $y \longmapsto y^d \pmod n$

$d = (d_{w-1},\ldots,d_0)_2$

temp := y_1 = MM(y,R^2;n)

for i=w-2 down to 0 do {

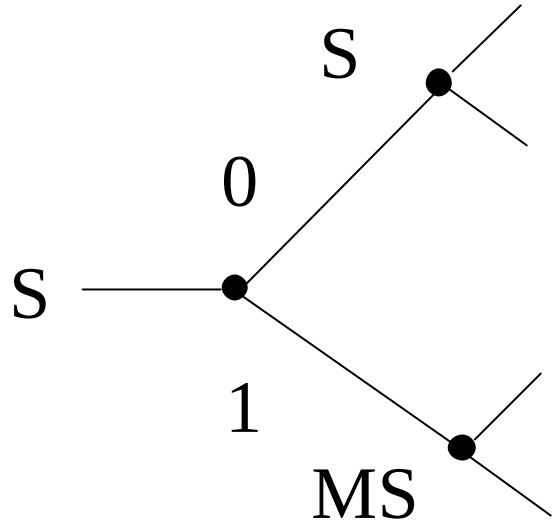
    temp := MM(temp,temp;n)

    if (d_i = 1) then temp := MM(temp,y_1;n)

    }

return MM(temp,1;n) (= $y^d \pmod n$)

$d_{w-1} = 1$ d_{w-2} d_k d_0



s&m with Montgomery's algorithm

computes $y \longmapsto y^d \pmod n$

$d = (d_{w-1},\ldots,d_0)_2$

temp := $\underline{y_1}$= MM(y,R^2;n)

for i=w-2 down to 0 do {

temp := MM(temp,temp;n)

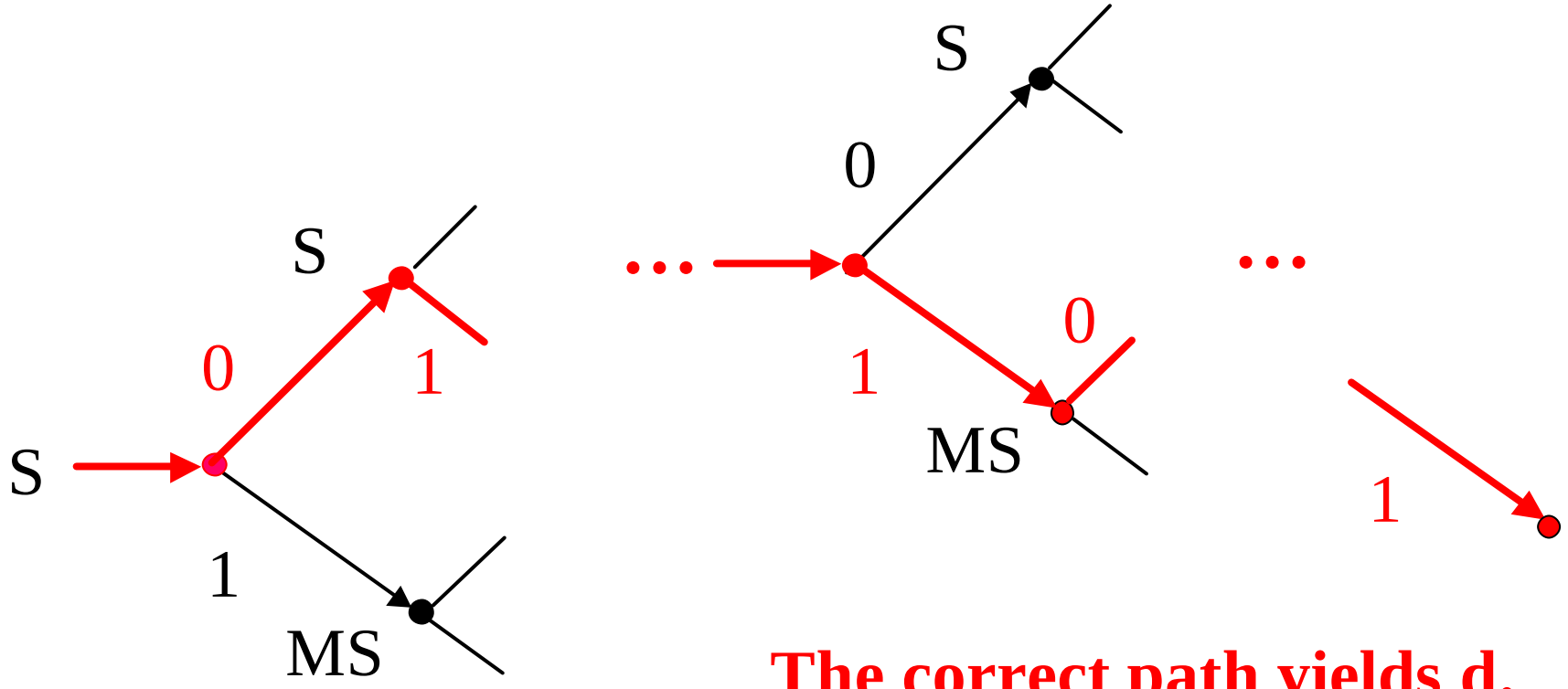
---

if (d_i = 1) then temp := MM(temp,$\underline{y_1}$;n)

}

return MM(temp,1;n) (= y^d (mod n))

$d_{w-1} = 1$ d_{w-2} d_k d_0



The correct path yields d.

Guessing the key bit d_k (I)

- $\Theta = A = \{0,1\}$
- loss function: $s(0,1) = s(1,0) = 1$
- a priori distribution
  $\eta(1) = \text{hamming weight } (d_k, \ldots, d_1) / k$
  (since $d_0 = 0$)

Guessing the key bit d_k (II)

- observation: $(t_i, u_i, v_i, q_i)_{\,i \le N}$

  t_i : remaining running time for the bits k, ..., 0

  u_i : time for the next mult with y_1 if d_k=1

  v_i : time for the next squaring if d_k=1

  q_i : time for the next squaring if d_k=0

The computation of the probability densities $f_0((t_i, u_i, v_i, q_i)_{\,i \le N})$ and $f_1((t_i, u_i, v_i, q_i)_{\,i \le N})$ requires the understanding of the stochastical process $W_1, W_2, \ldots$

Error detection and correction

All decisions that follow a wrong decision are meaningless.

Efficient error detection and correction strategies are desirable. This also requires stochastical methods.

Efficiency of the optimized attack

Attacked Device: Cascade Chip (ARM7M Risc processor, s&q, Montgomery's algorithm, **unprotected**)

512 bit key:

Quisquater et al. (Cardis 1998):

200.000 – 300.000 time measurements

Schindler, Koeune, Quisquater (2001):

5000 time measurements

Success rate: 74 %

Simulation: success rate 85 %

Possible countermeasures

- constant running times for all modular multiplications
- blinding techniques (→ Kocher)
- use the Chinese Remainder Theorem (???)

Applying the CRT without further countermeasures prevents this type of timing attack, but allows another, even more efficient timing attack!

Example 2:
A timing attack against RSA with CRT

Timing Attacks on RSA with CRT and Montgomery's algorithm: Schindler (2000), Brumley, Boneh (2003; remote attacks on OpenSSL implementations)

- The attack from Example 1 cannot be transferred.
- Key observation: Prob(ER in MM(S_i,$\underline{y_1}$;p)) is linear in ($\underline{y_1}$/p)
- Basic idea: The attacker is able to decide whether $[t_1,t_2]$ contains a multiple of p or q.

„Factoring by timing differences" (chosen-input attack)

Efficiency and Countermeasures

1024 bit RSA (s&q): about 300 time measurements

(under optimal conditions)

This attack and, vice versa, the implied security threat had not been detected without the understanding of the stochastic process $W_1, W_2, \ldots$

Countermeasures: see Example 1

Example 3:
A combined timing and power attack

The attacked device (smart card)

- computes $y^d \pmod n$
- uses Montgomery‘s algorithm
- uses a b-bit - table
  (table entry j: $M_j := (y^j R) \pmod n$ for $j=1,...,2^b-1$)

Modular exponentiation with tables

Example: b = 4

Secret exponent d = 1001001100001011...

d = 1001 0011 0000 1011 ...

M_9 $SSSSM_3$ SSSS $SSSSM_{11}$ ...

9-th table entry

S = squaring

M_j = multiplication with the j-th table entry

Further assumptions

- Base blinding prevents pure timing attacks
- Hardware countermeasures prevent pure power attacks.
- But: The power consumption reveals the end of the particular Montgomery operations.

Solution: Combine the timing and the power information.

Phases of the attack

The attacker

- observes modular exponentiations (n,d fixed)
- guesses independently the types T(1),T(2),... $\in$ $\{S, M_1, M_2, ..., M_{2^b-1}\}$ of the Montgomery operations within the exponentiation phase
- corrects wrong guesses
- T(1),T(2),... $\Rightarrow$ d

Example: (1 = extra reduction / 0 = no extra eduction)

sample	table initialization 1 2 .... 2^b-1	exponentiation phase 1 2 ..... i .....
1	1 0 1	0 0 .... 1 .....
2	0 1..... 1	0 0 .... 1 .....
⋮	⋮ ⋮ ⋮	⋮ ⋮ ⋮
N-1	0 0 0	1 0 .... 0 .....
N	0 0 1	0 1 .... 1 .....

„source“of the attack

guess T(i)

A priori distribution

A squaring is $b2^b$ times as probable as, e.g., a multiplication with the table entry M_6.

Different types of guessing errors

Example: b = 4

correct sequence:

... S, M_3, S, S, S, S , M_{12}, S, S, S, S, M_1, S ...

possible guesses:

a): ... S, M_3, S, S, S,M_{11}, M_{12}, S, S, S, S, M_1, S ...

b): ... S, M_3, S, S, S, S, S, S, S, S, S, M_1, S ...

c): ... S, M_3, S, S, S, S, M_{14}, S, S, S, S, M_1, S ...

a): localization + correction is easy.

b): localization is easy, correction is not obvious.

c): localization is difficult, correction is not obvious.

Optimal decision strategy

The optimal decision strategy „prefers“ errors of type a) and b).

The original attack (Walter & Thompson, 2001) considers only the case b=2.

For **b = 2** the optimized attack (Schindler 2002) reduces the sample size by factor 5.

The optimized attack works for each table size. It can also be applied against CRT implementations.

Efficiency

b=4: 512-bit exponent, sample size: 550, n/R ≈ 0.7

Optimized attack	maximum likelihood estimator (neglects different types of errors & a priori distribution)
success rate: 95 %	success rate: 76 %
av. # type c)-errors: < 0.3 (per trial)	av. # type c)-errors: 0.8 (per trial)

Efficiency

b=4: 512-bit exponent, sample size: 450, n/R ≈ 0.7

Optimized attack	maximum likelihood estimator (neglects different types of errors & a priori distribution)
success rate: 67 %	success rate: 13 %
av. # type c)-errors: 0.8 (per trial)	av. # type c)-errors: 2.4 (per trial)

Countermeasures

Exponent blinding prevents this attack.

Base blinding techniques, which prevent pure timing attacks, are yet not effective.

Power attacks

How can the presented concept fruitfully be transferred to to power attacks?

Presently, Kerstin Lemke (University of Bochum) and I work at a joint project on this topic.

Concluding remarks

To rate the true risk potential of a side-channel attack and to give effective and reliable countermeasures, the source of the attack has to be understood and the attack should be optimized.

Advanced stochastical methods (stochastic processes, statistical decision theory) have turned out to be useful tools to achieve this goal.

Contact

Bundesamt für Sicherheit in der
Informationstechnik (BSI)

Werner Schindler
Godesberger Allee 185-189
53175 Bonn

Tel: +49 (0)1888-9582-652
Fax: +49 (0)1888-9582-90652

Werner.Schindler@bsi.bund.de
www.bsi.bund.de
www.bsi-fuer-buerger.de