A Combined Timing and Power Attack

Werner Schindler

Bundesamt für Sicherheit in der Informationstechnik (BSI)
Godesberger Allee 185-189, 53175 Bonn, Germany
Werner.Schindler@bsi.bund.de

Abstract. In [9]Walter and Thompson introduced a new side-channel attack on the secret exponents of modular exponentiations which uses techniques from timing attacks to exploit specific information gained by a power attack. Walter and Thompson assumed that the attacked device uses a particular table method combined with Montgomery's algorithm. In the present paper their attack is optimized and generalized. For 2-bit tables this leads to a reduction of the necessary sample size to 20 per cent. The original attack cannot be applied if 4-bit tables are used, a case of particular practical interest, whereas the optimized attack gets by with 500 measurements. The optimized version can straightforwardly be adapted to other table methods, other multiplication algorithms and inexact timings. Moreover, it is shown that the countermeasures proposed in [9] do not prevent the optimized attack if unsuitable parameters are chosen.

Keywords: Timing attack, power attack, Montgomery's algorithm.

1 Introduction

In the past half-decade side-channel attacks have attracted enourmous attention as they have turned out to constitute serious threats for cryptosystems, especially if the cryptographic operations run on smart cards. Various types of timing attacks were introduced and optimized ([2], [1], [7], [8]), and a large number of papers were devoted to power attacks ([4] etc.). However, these timing attacks do not work if appropriate blinding techniques are used and various countermeasures against power attacks have been proposed. In [9] a new type of side-channel attack was introduced which uses techniques typical for timing attacks to exploit specific timing information stemming from a power attack (or, which made no difference in this context, from a radiation attack (cf. [3], for example)). In particular, this combined attack may be successful even if the attacked device could resist both, a "pure" timing attack (because blinding techniques are used) and a "pure" power attack (because appropriate countermeasures are employed). We optimize and, moreover, generalize this combined attack considerably.

As in [9], we attack the secret exponents of modular exponentiation (e.g., secret RSA exponents). We also assume that b-bit-tables (cf. [5], Alg. 14.82) and Montgomery's algorithm are used to carry out the modular exponentiations. (A b-bit table stores the powers of the basis up to the exponent $2^b - 1$.) We assume

D. Naccache and P. Paillier (Eds.): PKC 2002, LNCS 2274, pp. 263–279, 2002.

that the analysis of the power consumption enables the attacker to determine the times needed by the particular Montgomery multiplications (e.g., due to small peaks at the beginning of the modular multiplications), i.e. whether an extra reduction is necessary. As in [9] the focus of this paper is not the power measurement part but the guessing of the secret exponent.

The attacker uses a sample of modular exponentiations to guess the type of the particular Montgomery multiplications (squaring or a multiplication with a particular table entry) which in turn yields the secret exponent. In [9] for the most favourable parameter values a sample size of "less than 1000" turned out to be sufficient to recover a secret 384-bit exponent if a 2-bit table was used. For these parameters our optimized attack gets by with 200 samples. The original attack cannot be applied if 4-bit tables are used which is of particular practical interest. In contrast, even for non-optimal parameters our attack gets by with 500 samples to recover a 512-bit exponent, or a 1024-bit RSA exponent if the Chinese Remainder Theorem (CRT) is used, resp. Our approach can directly be adapted to other table methods and other modular multiplication algorithms. Even the times needed for the particular modular multiplications need not be determined exactly. Finally, we show that the countermeasures proposed in [9] do not prevent our attack if unsuitable parameters are used.

The paper is organized as follows: In Sect. 2 we recall the definition and basic facts concerning Montgomery's algorithm and table methods, and general assumptions are formulated. In Sect. 3 we explain the central ideas of our attack. The optimal decision strategy is derived in Sect. 7. The preliminary steps are discussed in Sects. 4-6. Experimental results are presented and discussed in Sect. 8. In Sect. 9 we compare the optimized attack with the original one. Sect. 10 considers generalizations of our attack under weaker assumptions. Then we propose effective countermeasures and conclude with final remarks.

2 General Assumptions

We assume that the attacked cryptographic device uses a b-bit table to compute $y^d (\mathrm{mod}\, M)$. Modular multiplications are calculated with Montgomery's algorithm (cf. [6]). We begin with some basic definitions.

Definition 1. As usually, for an integer b the term $b(\mathrm{mod}\ M)$ denotes the smallest nonnegative integer which is congruent to b modulo M. Further, $Z_M := \{0, 1, \ldots, M-1\}$, and for $x \in \mathbb{R}$ the term $\lceil x \rceil$ denotes the smallest integer $\geq x$.

The most elementary variant of Montgomery's algorithm transfers the modular multiplications to a modulus $R > M$ with $gcd(R, M) = 1$. Usually, R is a power of 2 whose exponent perfectly fits to the device's hardware architecture (cf. Remark 1(i)). Further, $R^{-1} \in Z_M$ denotes the multiplicative inverse of R in Z_M, i.e. $RR^{-1} \equiv 1 \pmod{M}$. The integer $M^* \in Z_R$ satisfies the integer equation $RR^{-1} - MM^* = 1$. For input $a', b' \in Z_M$ Montgomery's multiplication algorithm returns $\mathrm{MM}(a', b') := a'b'R^{-1} (\mathrm{mod}\, M)$.

Montgomery's algorithm

```
z:=a'b'
r:=(z(mod R)M*) (mod R)
s:=(z+rM)/R
if s≥M then s:=s-M
return s
```

In particular, $\mathrm{MM}(aR(\mathrm{mod}\, M), bR(\mathrm{mod}\, M)) = abR(\mathrm{mod}\, M)$. The subtraction in line 4 is called *extra reduction*. For fixed values M and R the time for a Montgomery multiplication can only attain two different values, namely c_0 if no extra reduction has to be carried out and $c_0 + c_{\mathrm{ER}}$ else. Let $(d_{w-1}, \ldots, d_0)_2$ denote the binary representation of the secret exponent d where, as usually, d_{w-1} denotes its most significant bit. Therefrom we derive $\lceil w/b \rceil$ b-bit integers $D_{\lceil w/b \rceil - 1}, \ldots, D_0$ with $D_j := \sum_{k=0}^{b-1} d_{bj+k} 2^k$. Combining a b-bit table with Montgomery's algorithm gives Algorithm 1. First, the table entries $u_1, \ldots, u_{2^b-1}$ are computed.

Algorithm 1

```
u_1:= MM(y, R^2(mod M))      (= yR(mod M))
for j:=2 to 2^(b-1) do   u_j:=MM(u_{j-1},u_1)
temp:=u_{D_{⌈w/b⌉-1}}
for i:=⌈w/b⌉-2 downto 0 do {
  for j:=1 to b do   temp:=MM(temp,temp)
  if (D_i>0) temp:=MM(temp,u_{D_i})
return MM(temp,1)            (=y^d(mod M))
```

The attacker analyzes the power consumption needed for the modular exponentiations ${y_{(1)}}^d (\mathrm{mod}\, M), \ldots, {y_{(N)}}^d (\mathrm{mod}\, M)$ for a given sample $y_{(1)}, \ldots, y_{(N)}$. The target of the attack is the secret exponent d. Next, we formulate the general assumptions. Note that our attack does also work under considerably weaker assumptions (cf. Sects. 10, 11). In particular, assumption e) may completely be dropped.

General Assumptions

a) The attacked device uses Algorithm 1 to compute $y^d (\mathrm{mod}\, M)$.
b) By observing the power consumption the attacker is able to determine the times needed by the particular Montgomery multiplications within Algorithm 1, i.e. he can decide whether an extra reduction is carried out.
c) The attacker has no knowledge about the base $y_{(k)}$ nor about the table entries $u_{1(k)}, \ldots, u_{2^b-1(k)}$ $(1 \le k \le N)$.
d) Algorithm 1 uses the same secret exponent d for all modular exponentiations ${y_{(1)}}^d (\mathrm{mod}\, M), \ldots, {y_{(N)}}^d (\mathrm{mod}\, M)$
e) The attacker knows the ratios M/R and $(R^2 (\mathrm{mod}\, M))/M$.

Remark 1. (i) Many implementations use a more efficient multiprecision variant of Montgomery's algorithm than listed above (cf. [5], Algorithm 14.36). Whether an extra reduction is necessary, however, merely depends on the parameters a', b', M and R but not on the chosen variant of Montgomery's algorithm or any

hardware characteristics ([8], Remark 1). (The latter aspects, of course, influence the absolute values of the time constants c_0 and c_{ER}.)
(ii) Re: GA a): Our attack can easily be adapted to other table methods (cf. Sect. 10).
(iii) Re: GA c): This could be the consequence of a standard blinding technique which prevents "ordinary" timing attacks, namely the pre-multiplication of the base y with a register value v_a and a post-multiplication of $(yv_a)^d (\bmod M)$ with $v_b = v_a^{-d} (\bmod M)$. Before the next exponentiation both register values are updated via $v_a := v_a^2 (\bmod M)$ and $v_b := v_b^2 (\bmod M)$ (cf. [2], Sect. 10). Note that if the attacker knew the bases $y_{(1)}, y_{(2)}, \ldots$ a very small sample size would be sufficient to recover the secret key. In fact, the attacker could successively guess the b-bit blocks $D_{\lceil w/b \rceil - 1}, D_{\lceil w/b \rceil - 2}, \ldots$, compare the corresponding extra reduction / not extra reduction patterns with the observed ones and exclude wrong assumptions (cf. [9], Subsect. 3.3).
(iv) Re GA d): cf. Sect. 11.
(v) Re GA e): If the ratios M/R and $(R^2 (\bmod M))/M$ are unknown they can be guessed efficiently (cf. Sect. 10).

3 The Central Ideas of Our Attack

In this section we sketch the fundamental ideas of our attack. Technical details will be treated in the following sections.

After the table entries $u_1, \ldots, u_{2^b-1}$ have been computed ("initialization phase") the "computation phase" begins. The "type" $T(i)$ of the i^{th} Montgomery multiplication within the computation phase is determined by the secret exponent d. We distinguish between squarings (i.e., $T(i) = `S`$) and multiplications of the temp value with table entry u_j (i.e. $T(i) = `M_j`$). The attacker guesses the sequence $T(1), T(2), \ldots$. If all guessings are correct this in turn yields d besides its most significant block. The most significant block can be guessed in a similar manner or determined by exhaustive search.

The probability that a squaring requires an extra reduction equals $M/3R$ while it is linear in the ratio u_j/M if $T(i) = `M_j`$ (cf. Sect. 4). If $T(i) = `S`$ and if the sample size N is sufficiently large the ratio "# extra reductions in the i^{th} Montgomery multiplication (counted over the whole sample) / N" should approximately equal $M/3R$. If $T(i) = `M_j`$ the probability for an extra reduction depends on the particular base $y_{(k)}$, or more precisely, on the (unknown!) table entry $u_{j(k)}$. Our attack exploits the differences between these probabilities. The "source" of our attack is the initialization phase as the attacker knows the types of the $2^b - 1$ Montgomery multiplications. The observed extra reductions within the initialization phase are indicators for the magnitude of the ratios $u_{1(k)}/M, \ldots, u_{2^b-1(k)}/M$. To simplify further notation we introduce *er-values* $q_{j(k)}$ and set $\boldsymbol{q}'_{(k)} := (q'_{1(k)}, \ldots, q'_{2^b-1(k)})$ where $q'_{j(k)} := 1$ if the computation of the j^{th} table entry, $u_{j(k)}$, requires an extra reduction for base $y_{(k)}$ and $q'_{j(k)} := 0$ else. Conditional to the observed *er-vector* $\boldsymbol{q}'_{(k)}$ the attacker first computes the joint conditional probability density for the vector $(u_{1(k)}/M, \ldots, u_{2^b-1(k)}/M) \in$

$[0,1)^{2^b-1}$, denoted by $g(u_{1(k)}/M,\ldots,u_{2^b-1(k)}/M \mid \boldsymbol{q}'_{(k)})$. Different er-vectors $\boldsymbol{q}'_{(k)} \neq \boldsymbol{q}''_{(k)}$ yield different conditional densities $g(\cdot \mid \boldsymbol{q}'_{(k)}) \neq g(\cdot \mid \boldsymbol{q}''_{(k)})$.

To illustrate the essential ideas we consider the most elementary variant where the attacker estimates the types $T(1), T(2), \ldots$ separately and independent from the others. (Since the extra reductions of subsequent Montgomery multiplications are not independent it yet may be reasonable to estimate the types of f consecutive Montgomery multiplications simultaneously, cf. Sects. 4,7,8.) Based on the observed er-values the attacker decides for that alternative $\theta \in \Theta := \{`S`, `M_1`, \ldots, `M_{2^b-1}`\}$ which appears to be the most likely one (to be precised later). In a first step the attacker uses the conditional densities $g(\cdot \mid \boldsymbol{q}'_{(k)})$ from above to compute the conditional probabilities $p_{\theta'}(q_{i(k)} \mid \boldsymbol{q}'_{(k)})$ for all $k \leq N$ and each $\theta' \in \Theta$. (The index θ' means that the conditional probability holds under the hypothesis $T(i) = \theta'$.) Analogously as above, $q_{i(k)} := 1$ iff the i^{th} Montgomery multiplication in the computation phase requires an extra reduction for base $y_{(k)}$.

The straight-forward approach, of course, was to decide for $\theta \in \Theta$ if the product $\prod_{k \leq N} p_{\theta'}(q_{i(k)} \mid \boldsymbol{q}'_{(k)})$ (= joint conditional probability for $\left(q_{i(k)} \mid \boldsymbol{q}'_{(k)}\right)_{k \leq N}$ under the hypothesis $T(i) = \theta'$) is maximal for $\theta' = \theta$ (maximum likelihood estimator). Although already efficient the maximum likelihood decision strategy can still be improved by considering two further criteria. First, not all admissible hypotheses occur with the same probability. In fact, it is much more likely that a randomly chosen Montgomery multiplication within the computation phase is a squaring than a multiplication with any particular table entry. Loosely speaking, compared with $\prod_{k \leq N} p_{(`M_j`)}(q_{i(k)} \mid \boldsymbol{q}'_{(k)})$ the term $\prod_{k \leq N} p_{(`S`)}(q_{i(k)} \mid \boldsymbol{q}'_{(k)})$ gets some "bonus".

Unlike for the timing attacks in [2], [1], [7] and [8] a false estimator $\widetilde{\theta}$ does not imply that the subsequent estimators are worthless. Indeed, the correction of false estimators is possible after the types of all Montgomery multiplications have been guessed. In fact, squarings occur in subsequences whose lengths are multiples of b and each subsequence is followed by exactly one Montgomery multiplication with a table entry. An (isolated) erroneous estimator 'M_j' instead of 'S' (or vice versa) can thus easily be detected and localized ("local errors") whereas the detection, localization and correction of an erroneous estimator 'M_j' instead of 'M_t' requires much greater efforts ("global errors"). The optimal decision strategy takes the different kinds of possible estimation errors into account. Loosely speaking, it "prefers" local errors instead of global errors.

4 Conditional Probabilities

Recall that the attacker wants to estimate the types of the Montgomery multiplications in the computation phase on basis of the observed er-values $q'_{j(k)}$ and $q_{i(k)}$ within the initialization and computation phase, resp. In Sect. 4 we derive explicit formulas for the conditional probabilities mentioned in the previous section. We first introduce some further definitions.

Definition 2. *A realization of a random variable X is a value assumed by X. Further, $\mathcal{T} := \{`S`, `M_1`, \ldots, `M_{2^b-1}`\}$. Analogously to $\boldsymbol{q}'_{(k)}$ the term $\boldsymbol{q}_{i,\ldots,i+f-1(k)}$ abbreviates $(q_{i(k)}, \ldots, q_{i+f-1(k)})$. For $A \subseteq B$ the indicator function $1_A \colon B \to \mathbb{R}$ is defined by $1_A(x) := 1$ if $x \in A$ and $:= 0$ else.*

Lemma 1. (i) $\frac{\mathrm{MM}(a'b')}{M} = \left(\frac{a'}{M} \frac{b'}{M} \frac{M}{R} + \frac{a'b'M^* \pmod R}{R} \right) \pmod 1$.
(ii) In particular, an extra reduction is necessary iff $\frac{\mathrm{MM}(a'b')}{M} < \frac{a'}{M} \frac{b'}{M} \frac{M}{R}$.
(iii) Let the random variables $V'_1, \ldots, V'_{2^b-1}$ and $V_1, V_2, \ldots$ be independent and equidistributed on $[0,1)$ while $S'_0, \ldots, S'_{2^b-1}$ and $S_0, S_1, \ldots$ are defined recursively. In particular, S'_0 is $[0,1)$-valued and

$$S'_i := \begin{cases} \left(S'_0(R^2 \pmod M)/M)(M/R) + V'_1\right) \pmod 1 & \text{for } i = 1 \\ \left(S'_{i-1} S'_1 M/R + V'_i\right) \pmod 1 & \text{for } 2 \le i \le 2^b - 1 \end{cases} \tag{1}$$

Similarly, $S_0 := S'_r$ where r temporarily stands for the most significant b-bit block of d, i.e. $r = D_{\lceil w/b \rceil - 1}$. Further, for $i \ge 1$

$$S_i := \begin{cases} \left(S^2_{i-1} M/R + V_i\right) \pmod 1 & \text{if } T(i) = `S` \\ \left(S_{i-1} S'_j M/R + V_i\right) \pmod 1 & \text{if } T(i) = `M_j` \end{cases} \tag{2}$$

Analogously, let the $\{0,1\}$-valued random variables $W'_1, \ldots, W'_{2^b-1}$ and $W_1, W_2, \ldots$ be defined by

$$W'_i := \begin{cases} 1_{S'_1 < S'_0(R^2 \,(\bmod\, M)/M)(M/R)} & \text{for } i = 1 \\ 1_{S'_i < S'_{i-1} S'_1 M/R} & \text{for } 2 \le i \le 2^b - 1 \quad \textit{and} \end{cases} \tag{3}$$

$$W_i := \begin{cases} 1_{S_i < S^2_{i-1} M/R} & \text{if } T(i) = `S` \\ 1_{S_i < S_{i-1} S'_j M/R} & \text{if } T(i) = `M_j` \end{cases} \tag{4}$$

Then the random variables $S'_0, S'_1, \ldots, S'_{2^b-1}$ as well as $S_0, S_1, \ldots$ are independent. The random variables $S'_1, \ldots, S'_{2^b-1}, S_1, S_2 \ldots$ are independent and equidistributed on $[0,1)$. Further, W_i and W_h are independent if $|i - h| > 1$.

Proof. Assertion (i) follows immediately from the definition of Montgomery's algorithm whereas (ii) is a consequence from (i) as the second summand of its right-hand side lies in $[0,1)$. The assertions in (iii) concerning the random variables S'_i and S_i follow from the fact that the random variables $V'_1, V'_2, \ldots$ and $V_1, V_2, \ldots$ are independent and equidistributed on $[0,1)$. The final assertion in (iii) follows from the definition of the random variables W_i.

Clearly, whether $\mathrm{MM}(a', b')$ requires an extra reduction depends deterministically on a' and b'. On the other hand, even small deviations in a' or b' usually cause "vast" deviations in the second summand of the right-hand side in Lemma 1(i). Recall that we neither know the base y nor the factors of any Montgomery multiplication. Assume for the moment that $a' := R^2 (\mathrm{mod}\, M)$ and $y/M \in I_j$ (cf. the first line in Alg. 1) where $I_j := [j2^{-v}, (j+1)2^{-v})$ denotes a small interval (e.g. $v = 16$) and further, that the random variable B is

equidistributed on the set $(\mathrm{Z_M/M}) \cap \mathrm{I_j}$. For realistic modulus size M the random variable $C := (a'BM^*) (\mathrm{mod}\, R)/R$ should fulfil $\mathrm{Prob}(C \in I_i) \approx 2^{-v}$ for all $i \leq 2^v - 1$ while the sum $a'B/R + C$ should similarly be distributed as if both summands were independent (cf. the proof of Lemma A.3(iii) in [7]). In particular, the remainder of the sum $(\mathrm{mod}\, 1)$ then is "almost" equidistributed on $[0, 1)$. An extra reduction is necessary iff $(a'B/R + C) (\mathrm{mod}\, 1) < a'B/R$. However, this is a formal analogon to the definitions of S'_1 and W'_1. We can continue this analogy and derive the following mathematical model.

Mathematical Model. We interpret the components of the er-vector $\boldsymbol{q}'_{(k)} = (q'_{1(k)}, \ldots, q'_{2^b-1(k)})$ as realizations of the random variables $W'_1, \ldots, W'_{2^b-1}$ with $S'_0 = y_{(k)}/M$. Similarly, we interpret $q_{1(k)}, q_{2(k)}, \ldots$ as realizations of $W_1, W_2, \ldots$.

Consequently, we have to study the stochastic processes $W'_1, \ldots, W'_{2^b-1}$ and $W_1, W_2, \ldots$. As a first result, the probability for an extra reduction in the i^{th} Montgomery multiplication equals

$$\mathrm{Prob}(W_i = 1) = \begin{cases} \frac{1}{3}\frac{M}{R} & \text{if } T(i) = \text{`}S\text{`} \\ \frac{u_j}{2M}\frac{M}{R} & \text{if } T(i) = \text{`}M_j\text{`}. \end{cases} \tag{5}$$

Remark 2. The random variables $W_1, W_2, \ldots$ are not independent but W_i and W_{i+1} are negative correlated. We point out that the random variables $W_1, W_2, \ldots$ are similarly defined as in [8] (cf. Theorem 2) or [7] (cf. Lemma 6.3). However, the requirements on the mathematical model are considerably higher than in [7] or [8] as we there were primarily interested in the variance of $W_1 + W_2 + \cdots$ to apply a version of the central limit theorem for dependent random variables which holds under relatively weak conditions. However, the experimental results (cf. Sect. 8) confirm the suitability of our mathematical model retrospectively. Lemma 2(ii) provides concrete formulas for the conditional probability densities $g(\cdot \mid \boldsymbol{q}'_{(k)})$ mentioned in the previous section.

Lemma 2. *(i) For* $1 \leq i \leq 2^b - 1$ *and* $w \in \{0, 1\}$ *let* $\mathcal{C}'(i; w) := \{(s'_0, \ldots, s'_{2^b-1}) \in [0, 1)^{2^b} \mid w'_i = w\}$. *Then* $\mathcal{C}'(i; 0) = [0, 1)^{2^b} \setminus \mathcal{C}'(i; 1)$. *In particular,* $\mathcal{C}'(1; 1) := \{(s'_0, \ldots, s'_{2^b-1}) \in [0, 1)^{2^b} \mid s'_1 < s'_0(R^2 (\mathrm{mod}\, M))/R\}$ *and further* $\mathcal{C}'(i; 1) := \{(s'_0, \ldots, s'_{2^b-1}) \in [0, 1)^{2^b} \mid s'_i < s'_{i-1} s'_1 M/R\}$ *for* $i > 1$.
(ii) Let the random variable S'_0 *be equidistributed on* $[0, 1)$. *The distribution of the random vector* $(S'_1, \ldots, S'_{2^b-1})$ *conditional to* $(W'_1 = w'_1, \ldots, W'_{2^b-1} = w'_{2^b-1})$ *has the joint conditional probability density* $g(s'_1, \ldots, s'_{2^b-1} \mid w'_1, \ldots, w'_{2^b-1}) :=$

$$\frac{\int_0^1 1_{\bigcap_{i=1}^{2^b-1} \mathcal{C}'(i;w'_i)}(s'_0, s'_1, \ldots, s'_{2^b-1})\, ds'_0}{\int_{[0,1)^{2^b}} 1_{\bigcap_{i=1}^{2^b-1} \mathcal{C}'(i;w'_i)}(s'_0, s'_1, \ldots, s'_{2^b-1})\, ds'_0 ds'_1 \cdots ds'_{2^b-1}} \tag{6}$$

on $[0, 1)^{2^b-1}$.
(iii) For $w \in \{0, 1\}$ *and* $i \leq m \leq i + f - 1$ *let* $\mathcal{C}_f(i, m; w, t) := \{(s_{i-1}, \ldots, s_{i+f-1})$

$\in [0,1)^{f+1} \mid w_m = w, T(m) = t\}$. *Then* $\mathcal{C}_f(i,m;0,t) = [0,1)^{f+1} \setminus \mathcal{C}_f(i,m;1,t)$. *In particular,* $\mathcal{C}_f(i,m;1,\text{`}S\text{`}) = \{(s_{i-1},\ldots,s_{i+f-1}) \in [0,1)^{f+1} \mid s_m < s_{m-1}^2 M/R\}$ *and* $\mathcal{C}_f(i,m;1,\text{`}M_j\text{`}) = \{(s_{i-1},\ldots,s_{i+f-1}) \in [0,1)^{f+1} \mid s_m < s_{m-1}s'_j M/R\}$.
(iv)

$$\text{Prob}(W_i = w_i,\ldots,W_{i+f-1} = w_{i+f-1} \mid W'_1 = w'_1,\ldots,W'_{2^b-1} = w'_{2^b-1}) = \tag{7}$$

$$\int\limits_{[0,1)^{2^b+f}} g(s'_1,\ldots,s'_{2^b-1} \mid w'_1,\ldots,w'_{2^b-1}) \cdot 1_{\bigcap_{m=i}^{i+f-1} \mathcal{C}_f(i,m;w_i,T(m))}(s_{i-1},\ldots,s_{i+f-1}) \;\times$$

$$\times \qquad\qquad ds'_1 \cdots ds'_{2^b-1} ds_{i-1} \cdots ds_{i+f-1}.$$

(v) If $S_0 = S'_r$ *then*

$$\text{Prob}(W_1 = 1 \mid W'_1 = w'_1,\ldots,W'_{2^b-1} = w'_{2^b-1}) = \tag{8}$$

$$\int\limits_{[0,1)^{2^b-1}} g(s'_1,\ldots,s'_{2^b-1} \mid w'_1,\ldots,w'_{2^b-1}) \cdot {s'_r}^2 \frac{M}{R}\, ds'_1 \cdots ds'_{2^b-1}.$$

Proof. Assertions (i) and (iii) follow immediately from the definition of the random variables W'_i and W_i. Clearly, $\{(s'_0,\ldots,s'_{2^b-1}) \in [0,1)^{2^b} \mid W'_1 = w'_1,\ldots, W'_{2^b-1} = w'_{2^b-1}\} = \bigcap_{i=1}^{2^b-1} \mathcal{C}'(i,w'_i)$. Equation (6) follows from the fact that the random variables $S'_0,\ldots,S'_{2^b-1}$ are independent and equidistributed on $[0,1)$ and the definitions of conditional distributions and a marginal densities. As the random variables $S_{i-1},\ldots,S_{i+f-1}$ are independent and equidistributed on $[0,1)$ assertion (iv) follows from the identity $\{(s_{i-1},\ldots,s_{i+f-1}) \in [0,1)^{f+1} \mid W_i = w_i,\ldots,W_{i+f-1} = w'_{i+f-1}\} = \bigcap_{m=i}^{i+f-1} \mathcal{C}_f(i,m;w_m,T(m))$. The first Montgomery multiplication in the computation phase is a squaring. Hence $\text{Prob}(W_1 = 1) = \text{Prob}({S'_r}^2 M/R + V_1 \geq 1) = {S'_r}^2 M/R$ which proves (v).

Example 1. Let $b = 2$, $f = 1$, $W'_1 = W'_3 = 1$ and $W'_2 = 0$. Then the denominator of $g(s_1,s_2,s_3 \mid 1,0,1)$ equals

$$\int_0^1 \int_0^{s'_0 \frac{R^2 (\bmod M)}{M} \frac{M}{R}} \int_{{s'_1}^2 \frac{M}{R}}^1 \int_0^{s'_1 s'_2 \frac{M}{R}} 1\, ds'_3 ds'_2 ds'_1 ds'_0 \;. \tag{9}$$

The nominator of $g(s_1,s_2,s_3 \mid 1,0,1)$ is a weighted indicator function which, however, need not be evaluated explicitely. If $T(i) = \text{`}M_2\text{`}$ inserting this nominator in (7) gives

$$\text{Prob}(W_i = 1 \mid W'_1 = 1, W'_2 = 0, W'_3 = 1) = \tag{10}$$

$$\frac{\int_0^1 \int_0^{s'_0 \frac{R^2 (\bmod M)}{M} \frac{M}{R}} \int_{{s'_1}^2 \frac{M}{R}}^1 \int_0^{s'_1 s'_2 \frac{M}{R}} \int_0^1 \int_0^{s_{i-1} s'_2 \frac{M}{R}} 1\, ds_i ds_{i-1} ds'_3 ds'_2 ds'_1 ds'_0}{\int_0^1 \int_0^{s'_0 \frac{R^2 (\bmod M)}{M} \frac{M}{R}} \int_{{s'_1}^2 \frac{M}{R}}^1 \int_0^{s'_1 s'_2 \frac{M}{R}} 1\, ds'_3 ds'_2 ds'_1 ds'_0}$$

The calculation of (10), or more generally, of (7) is elementary as it requires no more than the evaluation of 1-dimensional integrals of polynomials. If $T(i) = \text{`}S\text{`}$ then $\text{Prob}(W_i = 1 \mid W'_1 = w'_1, W'_2 = w'_2, W'_3 = w'_3) = M/3R$ for all w'_1, w'_2, w'_3.

Theorem 1. *Let* $\theta = (\omega_i, \ldots, \omega_{i+f-1}) \in \mathcal{T}^f$. *If* $T(i) = \omega_i, \ldots, T(i+f-1) = \omega_{i+f-1}$ *then* $p_\theta\left(\boldsymbol{q}_{i,\ldots,i+f-1(k)1\le k\le N} \mid \boldsymbol{q}'_{(k)1\le k\le N}\right)$ *denotes the conditional probability for the er-vector* $\boldsymbol{q}_{i,\ldots,i+f-1(k)1\le k\le N}$ *if* $\boldsymbol{q}'_{(k)1\le k\le N}$ *was observed in the initialization phase. In particular,*

$$p_\theta\left(\boldsymbol{q}_{i,\ldots,i+f-1(k)1\le k\le N} \mid \boldsymbol{q}'_{(k)1\le k\le N}\right) \approx \prod_{k=1}^{N} \int_{[0,1)^{2^b+f}} g(s'_1, \ldots, s'_{2^b-1} \mid q'_{1(k)}, \ldots, q'_{2^b-1(k)}) \;\times$$

$$\times \quad 1_{\bigcap_{m=i}^{i+f-1} \mathcal{C}_{f(i,m;w_i,\omega_m)}}(s_{i-1}, \ldots, s_{i+f-1})\, ds'_1 \cdots ds'_{2^b-1} ds_{i-1} \cdots ds_{i+f-1}. \quad (11)$$

If $D_{\lceil w/b\rceil - 1} = r$ *then*

$$\text{Prob}\left((q_{1(k)})_{1\le k\le N} \mid \boldsymbol{q}'_{(k)1\le k\le N}\right) \approx \quad (12)$$

$$\prod_{k=1}^{N} \int_{[0,1)^{2^b-1}} g(s'_1, \ldots, s'_{2^b-1} \mid q'_{1(k)}, \ldots, q'_{2^b-1(k)}) {s'_r}^2 \frac{M}{R}\, ds'_1 \cdots ds'_{2^b-1}.$$

Proof. According to our mathematical model we interpret the observed er-vectors $\boldsymbol{q}'_{(k)}$ and $\boldsymbol{q}_{i,\ldots,i+f-1(k)}$ as realizations of random variables $W'_{1(k)}, \ldots, W'_{2^b-1(k)}$ and $W_{i(k)}, \ldots, W_{i+f-1(k)}$, resp., which correspond $T(i) = \omega_i, \ldots, T(i+f-1) = \omega_{i+f-1}$ (cf. Lemma 1(iii)). Theorem 1 is an immediate consequence of Lemma 2.

Remark 3. (i) In the proof of Theorem 1 we tacitly assumed that the values $y_{(1)}/M, \ldots, y_{(N)}/M \in [0,1)$ behave like realizations of independent and equidistributed random variables on $[0,1)$. This assumption surely is justified if the blinding technique described in Remark 1(iii) is applied or for RSA encryptions with (pseudo-)random padding (with or without blinding, resp.), for example.
(ii) Consider RSA-based signatures with fixed padding (i.e., integrity and information bytes ‖ fixed padding bytes ‖ hash value) for which no blinding is applied. Then the ratios $y_{(1)}/M, \ldots, y_{(N)}/M$ are almost constant ($\approx c'$). Consequently, the conditional probability $g(\cdot \mid \cdot)$ from Lemma 2(ii) does not fit to the changed situation. To obtain the needed analogon (which has to be inserted in (11) and (12)) the integrals $\int_0^1 \ldots ds'_0$ in (6) have to be replaced by integrals with respect to the Dirac measure with total mass on $c' \in [0,1)$. Equivalently, we may completely drop the integration with respect to s'_0, set $s'_0 := c'$ in $\mathcal{C}'(i;w)$ and project $\mathcal{C}'(i;w)$ onto the components $s'_1, \ldots, s'_{2^b-1}$. If the CRT is used $M = p_i$ for a particular prime factor of the RSA modulus. The (unknown) ratio M/R can be guessed as described in Sect. 10 and $c'(R^2 (\mathrm{mod}\, M)/R)(M/R) \approx (q'_{1(1)} + \cdots q'_{1(N)})/N$. If the attacked device does not use the CRT the attack is trivial anyway as no blinding is applied (cf. Remark 1(iii)).
(iii) If the blinding technique from Remark 1(iii) is applied or if (pseudo-)random padding is used (cf. (i)) the attacker has to derive the needed timing information

from the power traces of single exponentiations (SPA). In the context of (ii) also DPA may be applied.

5 A Priori Distribution

Assume that the attacker wants to estimate $T(i), \ldots, T(i+f-1)$ simultaneously ($f \geq 1$). Based on the observed er-vectors he decides for that alternative $\theta \in \Theta \subseteq \mathcal{T}^f$ which appears to be the most likely one. For a randomly chosen position i, however, not all $\theta \in \Theta$ occur with equal probability. Therefore, we derive a probability distribution η on Θ which at least approximates the exact probabilities. For simplicity, we assume $f \leq b+1$.

Recall that squarings occur in subsequences whose lengths are multiples of b and are interrupted by single multiplications with table values. Hence the set of all admissible hypotheses Θ equals

$$\Theta = \theta_0 \cup \{\theta_{m,j} \mid 1 \leq m \leq f; 1 \leq j \leq 2^b - 1\} \qquad \text{for } f \leq b+1 \tag{13}$$

where $\theta_0 := (\text{`}S\text{`}, \ldots, \text{`}S\text{`})$ means that $T(i) = \text{`}S\text{`}, \ldots T(i+f-1) = \text{`}S\text{`}$. Analogously, $\theta_{m,j} := (\text{`}S\text{`}, \ldots, \text{`}S\text{`}, \text{`}M_j\text{`}, \text{`}S\text{`}, \ldots, \text{`}S\text{`})$ means $T(i+m-1) = \text{`}M_j\text{`}$ but $T(v) = \text{`}S\text{`}$ for $v \neq i+m-1$).

First, let us derive an approximator $\eta_{m,j}$ of the (exact) probability that $(T(i), \ldots, T(i+f-1)) = \theta_{m,j}$ for randomly chosen i. Without any knowledge about d it is reasonable to assume $\eta_{1,1} = \cdots = \eta_{f,2^b-1}$. In average, $2^{-b}(\lceil w/b \rceil - 1)$ many multiplications with the table entry u_j are carried out within the computation phase and thus (almost) as many blocks of f consecutive Montgomery multiplications of type $\theta_{m,j}$ exist. Further, in the computation phase about $(b + (2^b-1)2^{-b})(\lceil w/b \rceil - 1)$ Montgomery multiplications are carried out in average. (The exact number depends on the concrete value of d.) Altogether, we hence set

$$\eta_{1,1} := \cdots = \eta_{f,2^b-1} := \frac{2^{-b}\,(\lceil w/b \rceil - 1)}{(b + (2^b-1)2^{-b})\,(\lceil w/b \rceil - 1)} = \frac{1}{b2^b + (2^b-1)} \qquad \text{and}$$

$$\eta_0 := 1 - \frac{(2^b-1)\,f}{b2^b + (2^b-1)} = \frac{b2^b - (f-1)(2^b-1)}{b2^b + (2^b-1)}. \tag{14}$$

6 Error Detection and Correction

After the attacker has guessed the types $T(1), T(2), \ldots$ of all Montgomery multiplications he therefrom determines an estimator $\widetilde{d}$ for d. Then he computes $y^{\widetilde{d}} \pmod M$ for a reference base y to check whether $\widetilde{d} = d$. Either $y^d \pmod M$ itself is known or the exponentiation of $y^{\widetilde{d}} \pmod M$ with the public exponent e (RSA) gives y if $\widetilde{d}$ is correct. (For RSA implementations using the CRT (cf. Sect. 10) M equals a particular prime factor p_i of the modulus $n = p_1 p_2$, and the attack yields an estimator $\widetilde{d}$ for $d \pmod{(p_i - 1)}$. If $\widetilde{d}$ is correct, $\gcd(y^d \pmod n) -$

$y^{\widetilde{d}} (\mathrm{mod}\, n), p_1 p_2) = p_i$ and $\gcd(y - y^{\widetilde{de}} (\mathrm{mod}\, n), p_1 p_2) = p_i$.) If $\widetilde{d}$ turns out to be wrong the attacker has to correct false estimators.

It seems to be inconsequent to consider error detection and error correction strategies before the decision strategy itself has been derived. However, it will turn out to be useful to classify the estimation errors first.

Example 2. Let $b = 4$ and let the correct type sequence be given by
. $\ldots, `S', `M_3', `S', `S', `S', `S', \quad `M_{12}', `S', `S', `S', `S', `M_1', `S', \ldots$ whereas a), b) and c) are possible estimation sequences
a) $\ldots, `S', `M_3', `S', `S', `S', `M_{11}', `M_{12}', `S', `S', `S', `S', `M_1', `S', \ldots$
b) $\ldots, `S', `M_3', `S', `S', `S', `S', \quad `S', \quad `S', `S', `S', `S', `M_1', `S', \ldots$
c) $\ldots, `S', `M_3', `S', `S', `S', `S', \quad `M_{14}', `S', `S', `S', `S', `M_1', `S', \ldots$

The subsequences a), b) and c) contain exactly one false estimator. The error in a) ($`M_{11}$') can easily be located and corrected as the number of squarings between two multiplications with a table entry must be a multiple of $b = 4$. Similarly, as in b) nine squarings occur between $`M_3$' and $`M_1$' the fifth Montgomery multiplication cannot be a squaring. Its correction, however, is not as obvious as for a). For $f = 1$, i.e. if the attacker has guessed the types of all Montgomery multiplications separately, he first tries that alternative which appeared to be most likely one after $`S$'. (For $f > 1$ the situation is similar.) For sequence c) it is even not obvious that an error ($`M_{14}$' instead of $`M_{12}$') had occurred. It has to be searched exhaustively over all positions with type estimator $\neq `S$' (cf. Sect. 8). Of course, it is reasonable to start at those positions where the respective decisions have been "close".

Suggestively, we denote errors as in sequences a) and b) as *local errors*, or more precisely, as *local-a errors* and *local-b errors*, resp. Errors as in sequence c) are called *global errors*. The detection and localization of the local errors may be interpreted as a decoding problem. Therefore, we derive a $0-1$ sequence from the estimated type values by replacing $`S$' by 0 and $`M_j$' by 1. The code words are the $0-1$ sequences of the same length with isolated ones and subsequences of zeroes whose lengths are multiples of b. We decide for that code word with minimal Hamming distance. If the local errors occur "isolated" (as in Example 2) they can be localized separately, although their positions occasionally are not obvious. If the attacker has guessed $3b+1$ consecutive squarings, for example, then either the $(b+1)^{th}$ or the $(2b+1)^{th}$ estimator is wrong. Usually, especially for $b > 2$, also "neighboured" local errors can successfully be localized. Note, however, that this need not always be the case. For example, let $b = 2$ and assume that $\ldots, `S', `S', `S', `S', \ldots$ has been estimated instead of the correct subsequence $\ldots, `M_3', `S', `S', `M_2', \ldots$. Then both local errors will not be detected. Especially for $b = 2$ the sample size N hence should be chosen sufficiently large (if possible!) that neighboured local errors are unlikely. The correction of local-b errors and the localization and correction of global errors has to be done simultaneously over all local-b error positions and all positions with estimator $\neq `S$', resp. The optimal decision strategy which will be derived in the next section considers the

different error types. Roughly speaking, it clearly tries to avoid estimation errors but "favours" local errors instead of global errors.

7 The Optimal Decision Strategy

In the previous sections we have done the necessary preliminary work. Now we are going to put the pieces together to derive the optimal decision strategy to guess the types of f consecutive Montgomery multiplications simultaneously ($1 \le f \le b+1$). Therefore, we interpret the estimation of $T(i), \ldots, T(i+f-1)$ as a statistical decision problem.

Roughly speaking, in a statistical decision problem the statistician observes a sample $\omega \in \Omega$ which he interprets as a realization of a random variable X with unknown distribution p_θ. On basis of this observation he tries to estimate the parameter $\theta \in \Theta$ where Θ denotes the parameter space, i.e. the set of all admissible hypotheses (= possible parameters). Formally, a statistical decision problem is described by a 5-tupel $(\Theta, \Omega, s, \Delta, A)$ where A denotes the set of all possible alternatives the statistician can decide for, and Ω denotes the observation space. In our case the observations are the er-vectors $(\boldsymbol{q}'_{(k)}, \boldsymbol{q}_{i\ldots,i+f-1(k)})_{1\le k\le N}$ and thus $\Omega = (\{0,1\}^{2^b-1+f})^N$. The parameter space Θ was defined in Sect. 5, and further $A := \Theta$. Applying a deterministic decision strategy $\tau\colon \Omega \to A$ means that the statistician decides for $\tau(\omega)$ upon observation ω. The loss function $s\colon \Theta \times A \to [0,\infty)$, quantifies the "damage" of a wrong decision, i.e. if the statistician decides for $\theta' \in A$ although $\theta \in \Theta$ was the correct parameter. For our attack potential errors must be "punished" by the loss function with regard to the effort which is necessary for their detection, localization and correction (cf. Sect. 8). Clearly, $s(\theta,\theta) = 0$ for all $\theta \in \Theta$ (correct decisions).

Optimal Decision Strategy. Let the a priori distribution η be defined as in Sect. 5. Let $\tau_{opt}((\boldsymbol{q}'_{(k)}, \boldsymbol{q}_{i,\ldots,i+f-1(k)})_{1\le k\le N}) := \theta^*$ if the sum

$$\sum_{\theta\in\Theta} s(\theta,\theta') p_\theta \left((\boldsymbol{q}_{i,\ldots,i+f-1(k)} \mid \boldsymbol{q}'_{(k)})_{1\le k\le N} \right) \eta(\theta) \tag{15}$$

is minimal for $\theta' = \theta^*$. Then τ_{opt} causes the minimal expected loss of all decision strategies. That is, τ_{opt} is optimal among all decision strategies which estimate $T(i), \ldots, T(i+f-1)$ simultaneously.

Proof. After reordering the sums the expected loss for a deterministic decision strategy τ equals $\sum_{\omega\in\{0,1\}^{2^b-1+f}} \sum_{\theta\in\Theta} s(\theta,\tau(\omega)) p_\theta(\omega)\eta(\theta)$. The optimal decision strategy minimizes the inner sum for each $\omega := (\boldsymbol{q}_{i,\ldots,i+f-1(k)}, \boldsymbol{q}'_{(k)})$. Further, $p_\theta\left((\boldsymbol{q}_{i,\ldots,i+f-1(k)}, \boldsymbol{q}'_{(k)})_{1\le k\le N}\right) = p_\theta\left((\boldsymbol{q}_{i,\ldots,i+f-1(k)} \mid \boldsymbol{q}'_{(k)})_{1\le k\le N}\right) \cdot \mathrm{Prob}\left((\boldsymbol{q}'_{(k)})_{1\le k\le n}\right)$. The last term is independent of θ which proves the assertion.

Remark 4. (i) The optimal decision strategy may not be unique. As Θ is finite it suffices to consider deterministic decision strategies.
(ii) The differentiation of the error types, i.e. using different $s(\cdot,\cdot)$-values for local-a, local-b and global errors, does not reduce the total number of errors but reduces the number of global errors at the expense of the local ones which are easier to detect, to localize and to correct.
(iii) The types $T(1), T(2), \ldots$ determine d besides its most significant b-bit block. The latter can be estimated using (12) (maximum likelihood estimator!).

8 Experimental Results

The er-values $q'_{1(k)}, \ldots, q'_{2^b-1(k)}$ and $q_{1(k)}, q_{2(k)}, \ldots$ depend on $y_{(k)}, d, M, R$ and b but not on implementation details (cf. Remark 1(i)). As GA b) further assumes that the attacker can determine the er-values exactly (cf. Sect. 10) Alg. 1 was emulated on a computer with pseudorandom moduli and pseudorandom bases $y_{(1)}, \ldots, y_{(N)}$. Program output was $q'_{1(k)}, \ldots, q'_{2^b-1(k)}$ and $q_{1(k)}, q_{2(k)}, \ldots$. In the first phase of the attack type estimators $\widetilde{T}(1), \widetilde{T}(2), \ldots$ for $T(1), T(2), \ldots$ were derived using the optimal decision strategy from Sect. 7. For $(b = 2, f = 1)$ we used the loss function values $s(\text{`}S\text{`}, \text{`}M_j\text{`}) = 1$ (local-a error), $s(\text{`}S\text{`}, \text{`}M_j\text{`}) = 1.5$ (local-b error) and $s(\text{`}M_j\text{`}, \text{`}M_t\text{`}) = 4.0$ for $j \neq t$ (global error). For $(b = 4, f = 1)$ we defined $s(\text{`}S\text{`}, \text{`}M_j\text{`}) := 1$, $s(\text{`}S\text{`}, \text{`}M_j\text{`}) := 1.5$ and $s(\text{`}M_j\text{`}, \text{`}M_t\text{`}) := 8.0$. For $f > 1$ we used the loss function $s_f((\omega_1, \ldots, \omega_f), (\omega'_1, \ldots, \omega'_f)) := \sum_{j=1}^{f} s(\omega_j, \omega'_j)$.

Table 1. Average number of errors per 100 type estimators

	local-a errors	local-b errors	global errors
($b = 2$, $M/R \approx 0.99$, f=1, $N = 250$)	0.8	0.6	0.02
($b = 2$, $M/R \approx 0.99$, f=3, $N = 200$)	0.4	0.3	0.05
($b = 2$, $M/R \approx 0.7$, f=1, $N = 375$)	0.6	0.6	0.05
($b = 2$, $M/R \approx 0.7$, f=3, $N = 300$)	0.4	0.4	0.06
($b = 4$, $M/R \approx 0.7$, f=1, $N = 500$)	0.3	0.7	0.07
($b = 4$, $M/R \approx 0.7$, f=1, $N = 550$)	0.2	0.5	0.05

In the second phase the local errors were localized which in particular enabled the immediate correction of the local-a errors. As described in Sect. 6 we interpreted the localization of the local errors as a decoding problem where we searched for the code word(s) with minimal Hamming distance. Clearly, the type estimation and the error correction are easiest for $b = 2$ as there exist the fewest alternatives. In contrast, the detection and localization of local errors is most difficult for $b = 2$ as the subsequences of consecutive squarings are shorter than for $b > 2$. For $b = 4$ and $w = 512$, for example, (e.g., 1024-bit RSA using the CRT; cf. Sect. 10) about 632 Montgomery multiplications are carried out in

the computation phase. For ($b = 4$, $M/R \approx 0.7$, $f = 1$, $N = 500$) the attacker makes about 1.9 local-a, 4.4 local-b and 0.4 global errors in average. From the attacker's point of view the case $M/R \approx 0.99$ is the most favourable one. If $M/R \approx 0.5$ twice as many samples are needed to ensure similar success rates. For fixed parameters b, M/R, f and N the probabilities for wrong estimators do not depend on $w = \lceil log_2(d) \rceil$. The total number of errors (especially the global ones) hence increases linear in w. In particular, the necessary minimal sample size increases as $\log_2(M)$ increases.

Finally, the local-b errors and possible global errors have to be corrected. Therefore, the estimators which have been identified as local-b errors are replaced simultaneously by alternatives which had been ranked on positions 2 to 4 (behind the false estimator) in the type estimation phase, the most probable combinations of candidates first. If no global errors have been made this approach leads to a quick success. In fact, simulations using the parameters from Table 1 showed that for ($f = 1$, $b = 2$) and ($f = 1$, $b = 4$) the correct estimators usually are on rank 2, resp. nearly always on rank 2 or 3. In particular, for $b = 2$ the local-b errors sometimes can be corrected in the first attempt if no global errors have been made. If this procedure does not yield the searched exponent d the attacker presumably has made a global error. He then just repeats the steps from above, additionally changing one 'M_j'-estimator (= candidate for a global error), beginning at the position with the "closest" decision in the estimation phase. Provided that the local errors have been localized correctly this strategy can correct one global error. (The correction of more than one global error clearly is also possible but costly.)

Using the parameters from Table 1 for $w = 384$ (as in [9]) and ($b = 2$, $f = 1$) the attack was successful in about 90 per cent of the trials for both, $M/R \approx 0.99$ and $M/R \approx 0.7$. (To be precise, we did not actually carry out the final phase of the attack. To save computation time we resigned on checking the particular combinations of candidates by exponentiating a reference base y (cf. the first paragraph of Sect. 6). In fact, an attack was viewed as successful if our program could localize all local errors correctly and if at most one global error had been made (cf. the previous paragraph).) For ($b = 2$, $f = 3$) the success rate was 92 per cent. For ($b = 4$, $f = 1$) (with $N = 500$, resp. $N = 550$) and $w = 512$ about 93 per cent of the trials, resp. more than 95 per cent of the trials were successful. First simulations confirm that estimating $f > 1$ types simultaneously will also reduce the necessary sample size for $b = 4$.

9 A Brief Comparison with the Original Attack

In [9] Walter and Thompson treat exclusively the case $b = 2$. Based on $\sum_{k=1}^{N} q_{i(k)}$ they first decide whether $T(i) =$ 'S' or $T(i) \neq$ 'S'. To distinguish between 'M_1', 'M_2' and 'M_3' they use the subset $A := \{y_{(k)} \mid q'_{2(k)} = 1\} \subset \{y_{(1)}, \ldots, y_{(N)}\}$ and, additionally, a subset $A_0 \subset A$. The number of extra reductions (counted over A or A_0, resp.) needed for the i^{th} Montgomery multiplication is plotted for all i for which $T(i) \neq$ 'S' is assumed. In the most favourable case, i.e. for

$M/R \approx 0.99$, for sample size $N = 1000$ these numbers fall into three more or less separated subsets corresponding to 'M_1', 'M_2' and 'M_3', resp. (cf. [9], Figs. 2 and 3). (Implicitely, Walter and Thompson exploit that the random variables W'_1, W'_2 and W'_3 are negative correlated.) Walter and Thompson assume that about 500 time measurements should also be sufficient (cf. [9], Sect. 4). Using the techniques from [9], however, this prognosis seems to be rather optimistic. First of all, there are no clear-cut decision rules but decisions are made by eye. Even for $N = 1000$ some global errors occur (cf. [9], Sect. 4) and for $N = 500$ the situation will be considerably less comfortable. Finally, as the random variables W'_i and W'_j are independent if $|i-j| > 1$ the original attack cannot be transferred to $b > 2$.

10 Weakening the Assumptions and Generalizations

The general assumptions GA a)-e) are fulfilled, for example, if an RSA implementation uses the standard blinding technique against timing attacks described in Remark 1(iii) but not the CRT. If the CRT is used, the ratios M/R and $(R^2(\mathrm{mod}\, M))/M$ are yet unknown. (Then M equals a particular prime factor p_i. As already pointed out in Sect. 6 it is sufficient to determine $d(\mathrm{mod}\ (p_1-1))$ or $d(\mathrm{mod}\ (p_2-1))$.) Let H temporarily denote the total number of Montgomery multiplications in the computation phase. As the least significant block D_0 of d is non-zero, $T(1) = \cdots = T(b) = T(H-b) = \cdots = T(H-1) =$ 'S', and further, the second Montgomery multiplication in the initialization phase is a squaring, too. As the probability for an extra reduction in a squaring equals $M/3R$ the attacker uses these $(2b+1)N$ squarings to estimate the ratio M/R. Similarly, from the first Montgomery multiplication in the initialization phase an estimator for $(R^2(\mathrm{mod}\, M))/M$ can be derived. Compared with the scenario from the preceding sections, i.e. that M/R and $(R^2(\mathrm{mod}\, M))/M$ are known, the additional estimation steps cause a lower success rate for equal sample size N. For example, using the parameter values from Table 1 the success rate for ($b = 2$, $M/R \approx 0.7$, $f = 1$, $N = 375$), resp. for ($b = 4$, $M/R \approx 0.7$, $f = 1$, $N = 500$), reduces from 90 to 83 per cent, resp. from 93 to 87 per cent.

Ga a) assumes that the attacked device uses a particular table method. However, our method can be transferred to other table methods (cf. [5], Sect. 14.83 and 14.85) in an obvious manner. Our attack can also be applied if another modular multiplication algorithm is used than Montgomery's (e.g. a simple shift-and-add algorithm) provided that the (random) time needed for a modular multiplication of a fixed factor with a random cofactor depends significantly on the fixed factor. Similarly as in Sect. 4 the attacker interprets the times needed by the multiplications in the initialization and the computation phase as realizations of suitably defined random variables $T'_1, \ldots, T'_{2^b-1}$, resp., $T_1, T_2, \ldots$. If these random variables are continuously distributed (e.g. normally distributed, cf. [2]) the conditional probabilities $\mathrm{Prob}(W_i = w_i, \ldots, W_{i+f-1} = w_{i+f-1} \mid W'_1 = w'_1, \ldots, W'_{2^b-1} = w'_{2^b-1})$ from Sect. 4 correspond to conditional probability densities $f(T_i = t_i, \ldots, T_{i+f-1} = t_{i+f-1} \mid T'_1 = t'_1, \ldots, T'_{2^b-1} = t'_{2^b-1})$ where

$t'_1, \cdots, t'_{2^b-1}$ and $t_1, t_2, \ldots$ denote the times needed for the particular modular multiplications. (Recall that for Montgomery's algorithm $q_i \mapsto c + q_i\, c_{\mathrm{ER}}$ defines a bijection between the er-values and the running times.)

Moreover, the attacker may only be able to derive inexact values $\widetilde{t}'_j := t_j + t'_{\mathrm{E}rr;j}$ and $\widetilde{t}_i := t_i + t_{\mathrm{E}rr;i}$ from the power trace instead of t_j or t_i, resp., possibly a consequence of countermeasures against power attacks. The attacker then has to study the random variables $T'_1 + T'_{\mathrm{E}rr;1}, \ldots, T'_{2^b-1} + T'_{\mathrm{E}rr;2^b-1}$ and $T_1 + T_{\mathrm{E}rr;1}, T_2 + T_{\mathrm{E}rr;2}, \ldots$ instead of $T'_1, \ldots, T'_{2^b-1}$ and $T_1, T_2, \ldots$. The general approach yet remains unchanged.

11 Countermeasures

A standard blinding technique which prevents pure timing attacks is described in Remark 1(iii). However, it neither prevents the optimized nor the original attack from [9]. Walter and Thompson hence propose to apply another blinding technique which was also discussed in [2]. Namely, the base y shall not be exponentiated with the secret exponent d itself but with $d' := d + r\phi(M)$ where r is a non-negative pseudorandom integer which is renewed after each exponentiation and $\phi(\cdot)$ denotes the well-known Euler function. In fact, this violates assumption GA d) but does not necessarily prevent our attack.

Assume, for example, that for efficiency reasons $0 \le r < 2^4$. If d' has w' binary digits $(\lceil w'/b \rceil - 1)b$ many squarings are carried out within the computation phase. The number $\#(d')$ of multiplications with table entries and hence the total number of Montgomery multiplications, however, depend on d' rather than on w'. If two exponentiations require a different number of Montgomery multiplications the used exponents must be different. (Recall that the attacker knows the total number of Montgomery multiplications within the computation phase.) Consequently, the attacker divides the sample $y_{(1)}, \ldots, y_{(N)}$ into subsamples with respect to the number of Montgomery multiplications. Then he attacks each subsample separately as described in the previous sections. If there is a subsample which belongs to a unique exponent d' this attack will be successful.

The number $\#(d')$ may be viewed as a realization of a normally distributed random variable X with mean $(\lceil w'/b \rceil - 1)(1 - 2^{-b})$ and variance $(\lceil w'/b \rceil - 1)(1 - 2^{-b})2^{-b}$. If $513 \le w' < 516$ (e.g. for 1024-bit RSA using the CRT and $d' > 2^{512}$) and $b = 4$, for example, we have $\mathrm{Prob}(X = 120) \approx 0.145$, $\mathrm{Prob}(X = 119) = \mathrm{Prob}(X = 121) \approx 0.135$, $\mathrm{Prob}(X = 118) = \mathrm{Prob}(X = 122) \approx 0.114$, $\mathrm{Prob}(X = 117) = \mathrm{Prob}(X = 123) \approx 0.079$, $\mathrm{Prob}(X = 116) = \mathrm{Prob}(X = 124) \approx 0.051$ etc. Hence it is likely that at least one subsample belongs to a unique exponent d'. (In particular, note that for $d' = d$ not 512 but only 508 squarings are necessary.) Although the efficiency of our attack is reduced by factor 16 it is still practically feasible. To prevent the attack the range of the pseudorandom numbers should be chosen sufficiently large, e.g. $r \le 2^{16}$. Additionally, a lower bound for r may be chosen such that the number of squarings in the computation phase is constant for all admissible r's.

12 Final Remarks

Using suitable stochastical methods the efficieny of the original attack introduced by Walter and Thompson ([9]) was improved by factor 5 for 2-bit tables. Unlike the original attack the optimized attack works and is also practically feasible for $b > 2$. It can be adapted to other table methods, other modular multiplication algorithms and inexact timings in a straightforward manner. Finally, for unsuitable parameters the countermeasures proposed in [9] were shown to be insufficient.

References

1. J.-F. Dhem, F. Koeune, P.-A. Leroux, P.-A. Mestré, J.-J. Quisquater, J.-L. Willems: A Practical Implementation of the Timing Attack. In: J.-J. Quisquater and B. Schneier (eds.): Smart Card – Research and Applications. Lecture Notes in Computer Science **1820**, Berlin, Springer (2000), 175–191.
2. P. Kocher: Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS and Other Systems. In: N. Koblitz (ed.): Advances in Cryptology – Crypto '96, Lecture Notes in Computer Science **1109**. Springer, Heidelberg (1996), 104–113.
3. K. Gandolfi, C. Mourtel, F. Olivier: Electromagnetic Analysis: Concrete Results. In: Ç.K. Koç, D. Naccache, C. Paar (eds.): Cryptographic Hardware and Embedded Systems — CHES 2001, Springer, Lecture Notes in Computer Science **2162**, Berlin (2001), 251–261.
4. P. Kocher, J. Jaffe, B. Jub: Differential Power Analysis. In: M. Wiener (ed.): Advances in Cryptology – Crypto '99. Lecture Notes in Computer Science **1666**, Berlin, Springer (1999), 388–397.
5. A.J. Menezes, P.C. van Oorschot, S.C. Vanstone: Handbook of Applied Cryptography, Boca Raton, CRC Press (1997).
6. P.L. Montgomery: Modular Multiplication without Trial Division, Math. Comp. **44**, no. 170, 519–521 (April 1985).
7. W. Schindler: Optimized Timing Attacks against Public Key Cryptosystems. To appear in Statistics & Decisions.
8. W. Schindler: A Timing Attack against RSA with the Chinese Remainder Theorem. In: Ç.K. Koç, C. Paar (eds.): Cryptographic Hardware and Embedded Systems — CHES 2000, Springer, Lecture Notes in Computer Science **1965**, Berlin (2000), 110–125.
9. C.D. Walter, S. Thompson: Distinguishing Exponent Digits by Observing Modular Subtractions. In: D. Naccache (ed.): Topics in Cryptology – CT-RSA 2001, Springer, Lecture Notes in Computer Science **2020**, Berlin (2000), 192–207.