Research Article

Margaux Dugardin, Werner Schindler, and Sylvain Guilley*

Stochastic methods defeat regular RSA exponentiation algorithms with combined blinding methods

https://doi.org/10.1515/jmc-2020-0010
received March 1, 2020; accepted February 26, 2021

Abstract: Extra-reductions occurring in Montgomery multiplications disclose side-channel information which can be exploited even in stringent contexts. In this article, we derive stochastic attacks to defeat Rivest-Shamir-Adleman (RSA) with Montgomery ladder regular exponentiation coupled with base blinding. Namely, we leverage on precharacterized multivariate probability mass functions of extra-reductions between pairs of (multiplication, square) in one iteration of the RSA algorithm and that of the next one(s) to build a maximum likelihood distinguisher. The efficiency of our attack (in terms of required traces) is more than double compared to the state-of-the-art. In addition to this result, we also apply our method to the case of regular exponentiation, base blinding, and modulus blinding. Quite surprisingly, modulus blinding does not make our attack impossible, and so even for large sizes of the modulus randomizing element. At the cost of larger sample sizes our attacks tolerate noisy measurements. Fortunately, effective countermeasures exist.

Keywords: RSA, Montgomery multiplication, side-channel analysis, optimization, extra-reduction, regular exponentiation, maximum likelihood, base blinding, modulus blinding

MSC 2020: 94A60, 60G99

1 Introduction

It has been noted by Kocher [13] as early as 1996 that asymmetric cryptographic algorithms are prone to side-channel attacks. Countermeasures have been developed in a view to make these attacks either impossible or at least much harder to perform. There are several countermeasure principles. One first class consists in balancing the control-flow so that execution traces perfectly superimpose whatever the value of the secrets. A second important class of countermeasures consists in deceiving correlation attempts by attacker with side-channel traces. The strategy consists in randomizing algorithm inputs or internal parameters, so that the computation is carried out on unpredictable data. Obviously, the randomization is restricted, since it must be possible to unravel the injected randomness at the end of the computation.

In this article, we focus on the Rivest-Shamir-Adleman (RSA) cryptosystem, while it uses its secret exponent k. Despite the balancing and randomization countermeasures, attackers will desperately persist at recovering k. But in order to bypass protections, the attacker needs to resort to more evolved strategies.

*** Corresponding author: Sylvain Guilley,** Secure-IC S.A.S., Tour Montparnasse, 27th floor, 75015, Paris, France; LTCI, Telecom Paris, IMT, Institut Polytechnique de Paris, 75013, Paris, France; École Normale Supérieure, Département d'Informatique (ENS/DI), CNRS, PSL University, 75005, Paris, France, e-mail: sylvain.guilley@secure-ic.com, sylvain.guilley@telecom-paristech.fr
Margaux Dugardin: Secure-IC S.A.S., ZAC des Champs Blancs, 15, rue Claude Chappe Bât. B, 35510, Cesson-Sévigné, France; LTCI, Telecom Paris, IMT, Institut Polytechnique de Paris, 75013 Paris, France, e-mail: margaux.dugardin@telecom-paristech.fr
Werner Schindler: Bundesamt für Sicherheit in der Informationstechnik (BSI), Godesberger Allee 185-189, 53175, Bonn, Germany, e-mail: Werner.Schindler@bsi.bund.de

We make a difference between attacks which can be carried out in one *single* trace and those which require *multiple* traces (since there is not enough information in a single trace). An attack which succeeds with one single trace can overcome any algorithmic countermeasure:[1] basically, against randomizing countermeasures, it will recover the randomized version of some sensitive value, but this randomized value is still sufficient for the adversary to behave as if he knows the secret. As an example, in the case of exponent blinding, instead of computing $m^k \bmod N$ (where m is the base, k is the secret exponent, and N is the modulus), the side-channel protected RSA computes $m^{k+\varphi(N)} \bmod N$ (where φ is the Euler totient function). Those two quantities are equal, owing to the Fermat little theorem, hence, it does not matter if the attacker recovers $k' = k + \varphi(N)$ in lieu of k: in both cases he can forge valid signatures or decrypt messages correctly. Indeed, k' is equivalent to k for the purpose of signature generation or decryption. When attacks require some kind of averaging, then randomization countermeasures do work in concealing the secret, at least if the randomness is refreshed at each new computation. However, the balancing countermeasures do not deceive an attacker which averages traces, because the averaging of always the same execution allows for the attacker to increase the signal-to-noise ratio (SNR).

In practice, the attacks which succeed in a single trace are the more dangerous, and implementers defend their implementation in the first place. The so-called simple power analysis (SPA [14, §2]) introduced in 1999 allows us to read out the exponent in one trace. Therefore, the usual countermeasure consists in the implementation of a regular exponentiation algorithm. In RSA, the so-called "*regular algorithm*" is a method to compute the modular exponentiation using a key-independent sequence of *squaring* and *multiplication* operations. Examples of regular exponentiation algorithms are the Montgomery ladder (treated in this paper), the square and multiply always algorithm, or fixed window exponentiation with explicit multiplication also if the exponent bits in the current window are all equal to zero [15, Algorithm 14.82].

Thus, it is a protection against the simple trace analysis, where the attacker attempts to derive the exponent by observing one (or several identical) computation. The regular exponentiation countermeasure against SPA plugs the leak, but in the meantime takes care to properly align traces corresponding to various executions. This is at the advantage of the adversary, in that such unfortunate alignment opens the door to differential power analyses, as discussed in [14, §5], to template attacks [5], or to machine learning attacks [19]. Those attacks, provided they require to collect several traces from the same inputs (for averaging in order to increase the SNR), are combated by randomizing countermeasures. For instance, the input of the RSA (its base) can be randomized at the input, while being consistently derandomized at the output. Another option to randomize the intermediate computations is to randomize the modulus (so-called "*modular extension*"). This second option also allows us to perform a sanity check for the computation, which is incidentally a countermeasure against fault injection attacks [7]. We insist that all three countermeasures might well be stacked one on top of each other, so as to thwart simple power attacks, differential power attacks, and perturbation attacks, altogether. As an alternative to regular exponentiation algorithm, or even as a complement to it, the secret exponent can be protected by blinding, as explained earlier.

2 Previous work and our contributions

2.1 State-of-the-art

We analyze in this article possible remaining biases, namely, extra-reductions inherent to the modular multiplication algorithm.

1 For the sake of being accurate, let us precise that this assertion holds true for most scenarios, but might become wrong for some pathological counterexamples where the overall attack requires some additional work (e.g., some search) which, e.g., increases in the exponent length so that an attack becomes infeasible when the exponent becomes longer by exponent blinding. However, such countermeasures are not realistic from an industrial standpoint owing to the excessive overhead they incur, thus they can safely be ignored in our argumentation.

Given two integers a and b, the classical modular multiplication $a \times b \bmod p$ computes the multiplication $a \times b$ followed by the modular reduction by p. Montgomery Modular Multiplication (MMM) transforms a and b into special representations known as their Montgomery forms.

Definition 2.1. (Montgomery transformation [16]) For any modulus p, the Montgomery form of $a \in \mathbb{F}_p$ is $\phi(a) = a \times R \bmod p$ for some constant R greater than and co-prime with p.

In order to ease the computation, R is usually chosen as the smallest power of two greater than p, that is, $R = 2^{\lceil \log_2(p) \rceil}$. Using the Montgomery form of integers, modular multiplications used in modular exponentiation algorithms can be carried out using the MMM:

Definition 2.2. (MMM [16]) Let $\phi(a)$ and $\phi(b)$ be two elements of $\mathbb{F}_p$ in the Montgomery form. The MMM of $\phi(a)$ and $\phi(b)$ is $\phi(a) \times \phi(b) \times R^{-1} \bmod p$.

Proposition 2.3. (MMM correction [15, §14.36]) *The output of the MMM of $\phi(a)$ and $\phi(b)$ is $\phi(ab)$.*

Algorithm 1 shows that the MMM can be implemented in two steps:
(i) compute $D = \phi(a) \times \phi(b)$, then
(ii) reduce D using Montgomery reduction which returns $\phi(c)$.

In Algorithm 1, the pair (R^{-1}, v) is such that $RR^{-1} - vp = 1$.

Algorithm 1. Montgomery reduction (Algorithm 14.32 of [15])

input : $D = \phi(a) \times \phi(b)$
output : $\phi(c) = \phi(a) \times \phi(b) \times R^{-1} \bmod p$

1 $m \leftarrow (D \bmod R) \times v \bmod R$;
2 $U \leftarrow (D + m \times p)/R$; // Invariant: $0 \leq U < 2p$
3 **if** $U \geq p$ **then**
4 $\quad C \leftarrow U - p$; // Extra-reduction
5 $C \leftarrow U$;
6 **return** C;

Definition 2.4. (Extra-reduction) In Algorithm 1, when the intermediate value U is greater than p, a subtraction named extra-reduction occurs so as to have a result C of the Montgomery multiplication (MM) between 0 and $p - 1$. We set $X = 1$ in the presence of the extra-reduction, and $X = 0$ in its absence.

As we shall explain, this side channel is induced by the choice of moduli represented on a bitwidth, which is exactly divisible by the bitwidth of the computers, namely, this bitwidth is typically a power of two, such as 16, 32, or 64. This bias has given rise to the so-called extra-reduction analysis (ERA). An overview of known ERAs is provided in Table 1. Specifically, this table shows which countermeasure can be bypassed by which attack. The classification criteria in Table 1 are listed as follows:

- the implementation uses the Chinese Remainder Theorem (CRT), i.e., the moduli p and q are unknown to the attacker,
- the protection against differential power analysis named the base blinding,
- the protection against SPA protection named the regular exponentiation algorithm,
- the compensation of the extra-reduction by a fake operation, which is named constant time nonstraight line algorithm (N-SLA), i.e., constant operations have their fixed values identified by software.[2] In

2 For example, the reduction is always carried out with a value computed in Boolean logic (hence straight line) as either the modulus or the constant zero (case of OpenSSL), or a dummy operation of same duration as a reduction is executed if the reduction shall not be carried out (mbedTLS [6]). These two strategies are described in Appendix A, page 20, of [10]. However,

Table 1: Summary of capability of extra-reduction analyses published before December 2020

	With RSA-CRT	Basis blinding	Regular algorithm	Constant time N-SLA	Constant time SLA	Exponent blinding	Modular extension
ERA-1a	✗	✗	✗	✗	✗	✗	✗
[9,13,22,25]	No	No	No	No	No	No	No
ERA-1b	✓	✗	✓	✗	✗	✗	✗
[3,6,8,20]	Yes	No	Yes	No	No	No	No
ERA-2	✓	✗	✗	✗	✗	✓	✗
[23,24]	Yes	No	No	No	No	Yes	No
ERA-L1	✓	✓	✓	✗	✗	✗	✗
[1,2,21,26,30]	Yes/No	Yes	Yes	No	No	No	No
ERA-L2	✓	✓	✓	✓	✗	✗	✗
[10,11]	Yes/No	Yes	Yes	Yes	No	No	No
This work	✓	✓	✓	✓	✗	✗	✓
	Yes/No	Yes	Yes	Yes	No	No	Yes

principle (at least with a reasonable probability), these countermeasures might be detected and nullified by a suitable side-channel attack. In Table 1, we assume that such side-channel attacks exist,

- identical execution times are ensured by avoiding extra-reductions at all, which is named constant time straight line algorithm (SLA). Obviously, the attacks listed in Table 1 cannot work in this case, see also Section 5,
- the protection against differential power analysis named the exponent blinding, and
- the fault and differential protection named modular extension.

The algorithms from ERA-1a, ERA-1b, and ERA-2 are pure (global) timing attacks. Of course, by definition, pure timing attacks cannot overcome constant time implementations. While the pure timing attacks are very different for CRT implementations and for non-CRT implementations the local timing attacks from ERA-L1 and ERA-L2 work for the CRT and non-CRT implementations as well. More precisely, these local attacks are a little bit easier to perform on non-CRT implementations because the ratio p/R (and sometimes also the value $R^2 (\mathrm{mod}\, p)/p$) does not have to be estimated there. For these reasons, we did not distinguish between CRT and not CRT there. The pioneer papers [9,30] are significantly less efficient than their successors in the respective ERA (up to factor 50) and less general [30]. The difference between ERA-L1 and ERA-L2 is that with ERA-L2, the attacker is capable of probing the cache to distinguish between two different execution paths of otherwise identical duration and power leakage, whereas with ERA-L1, the attacker is restricted to observe the duration or the power leakage. Arguably, this difference resides more in the side-channel collection than in its analysis.

Remark. The terminology in Table 1 shall be considered with attention. Indeed, historically, ERA-1a, ERA-1b, and ERA-2 are pure timing attacks discovered in this order. Similarly, ERA-L1 and ERA-L2 are local timing attacks, discovered in this order. But some papers about ERA-1b were published after the papers from ERA-L1 and vice versa.

In [10,11], side-channel attacks on RSA, with CRT and without CRT, were investigated using leakage information of the presence or absence of the extra-reductions in MMM. The side-channel information was used to identify, which MMs require extra reductions. Two exponentiation algorithms were considered, namely, the always square and multiply exponentiation and the Montgomery ladder. The overall attacks split into many individual decisions whether $(k_i = k_{i-1})$ or $(k_i \neq k_{i-1})$, where k_i and k_{i-1} denote subsequent key bits. The

both countermeasures rely on a test, hence a branching in the control flow, which can be detected by a cache-timing analysis (see [2]) or by a power/electromagnetic side-channel analysis (empowered by a two-class clustering algorithm; see Figure 7 of [10]).

presented attacks were successful but for these decisions only two – one squaring and one multiplication – out of four Montgomery operations (squaring or multiplication) were exploited. However, the approach is too complex: the derivation of the probability mass function (PMF) of values for multiple operations becomes mathematically intractable when the number of operations analyzed jointly is strictly greater than two.

2.2 Novel contributions

For these reasons, in this article, we resort to another way to estimate the distribution of the extra-reduction which does not need the estimation of PMF values. We leverage on a previous work of Schindler [21]: this paper simplifies the characterization of the extra-reduction distribution using two elegant properties of MMM.

Using sophisticated stochastic methods, we solve the problem and improve the efficiency of [10,11], in the presence of regular exponent and base blinding.

Moreover, we extend the results to the case where the modulus is itself randomized. We show that ERA remains a powerful side-channel despite the stacking of three protections, namely, regular exponentiation and base and modulus blinding. We performed our experiments on 1024-bit RSA moduli as this allows a fair comparison of the attack efficiency with the experimental results in [10,11].

This manuscript contains joint research work from the years 2016–2018. We mention that parts of an intermediate version of this paper have found input in the PhD thesis of the lead author.

2.3 Outline

The rest of this paper is organized as follows. We start by giving our optimized attack in Section 3. Namely, we recapitulate in Section 3.1 the background to optimize the state-of-the-art when RSA uses a regular algorithm (we focus on the so-called Montgomery ladder) and base blinding. The core of our attack is presented in Section 3.2. Evaluation with both perfect and noisy measurements is conducted in Section 4, where we also consider the "modulus extension" as a third countermeasure on top of regular exponentiation and base blinding. Eventually, countermeasures are addressed in Section 5, and conclusions are derived in Section 6. Some formal computation results are given in Appendix A.

3 The optimized attack: the stochastic background

In this section, we optimize the attack from [10,11]. We begin with definitions and we formulate the target of our attack in Section 3.1. In Section 3.2, we analyze the stochastic properties of the MM, and in Lemma 3.4 we develop a formula for the joint probability of several extra-reductions. The following subsections treat the estimation of two parameters, which are usually unknown, and the maximum likelihood estimator is derived.

3.1 Definitions and target of the attack

In this paper, we only consider the Montgomery ladder (left-to-right), which is described in Algorithm 2. Unlike [10,11] we do not consider the square and always multiply algorithm (cf. Algorithm 1.1 in [11]). It is obvious how the applied mathematical methods can be transferred to the square and always multiply exponentiation algorithm.

We assume that the message m has been blinded (message blinding, a.k.a. base blinding). The attack applies to both RSA with CRT and RSA without CRT. We further assume that the arithmetic operations apply the Montgomery's multiplication algorithm [17]. As in [10,11] we assume that a side-channel attack yields (possibly noisy) information about whether or not MMs need extra-reductions. The applied mathematical techniques are similar to that in [1,2,21], where attacks on different variants of fixed window exponentiation algorithms [2,21] and the sliding window exponentiation algorithm [1] were analyzed thoroughly.

To avoid clumsy formulations we always target RSA with CRT in the following, where p denotes one prime factor of the RSA modulus n. We note that the attack on RSA without CRT works identically and is even simpler since there is no need to estimate the ratio n/R (which is the ratio of two public parameters).

Definition 3.1 describes the notations, necessary to understand this paper.

Definition 3.1. For $i = l-1, l-2, \ldots, 0$, and $j = 0, 1$, the term $r_{i,j}$ denotes the value of register R_j after the key bit k_i has been processed. Furthermore, $s_{i,j} := r_{i,j}/p \in [0, 1)$ stands for the normalized register values. For $i = l-2, \ldots, 0$, we set $w_{i(M)} = 1$ if the first Montgomery operation for key bit k_i ("**m**ultiplication") needs an extra-reduction (ER) and $w_{i(M)} = 0$ otherwise. Analogously, $w_{i(Q)} = 1$ if the second Montgomery operation for key bit k_i ("s**q**uaring," or "**Q**uadrierung" in German – we apply "Q" in place of "S" to prevent confusion with the stochastic process $S_{i;j}$ defined below) needs an ER and $w_{i(Q)} = 0$ otherwise. We recall that in the context of random variables the abbreviation "iid" stands for "independent and identically distributed." The indicator function $1_A(x)$ assumes the value 1 if $x \in A$ and 0 else. For $b \in \mathbb{Z}$, the term $b(\bmod p)$ denotes the unique element in $\mathbb{Z}_p = \{0, 1, \ldots, p-1\}$, which is congruent to b modulo p. The letter R denotes the Montgomery constant $R = 2^x$ for some integer $x \geq \lceil \log_2 p \rceil$. (Usually, $x = \lceil \log_2 p \rceil$.) When b is a real number, the term $b(\bmod p)$ denotes the real number $b - \lfloor b/p \rfloor$. Finally, for $a, b \in \mathbb{Z}_p$ we define $\mathrm{MM}(a, b; p) := abR^{-1} (\bmod p)$ (MM, as per Definition 2.2).

Algorithm 2. Left-to-right Montgomery ladder with MM algorithm

Input: m, $k = (k_{l-1}k_{l-2} \ldots k_0)_2$, p $\quad (k_{l-1} = 1 \text{ and } k_0 = 1)$
Output: $m^k \bmod p$

1 $R_0 \leftarrow \mathrm{MM}(m, R^2; p)$
2 $R_1 \leftarrow \mathrm{MM}(R_0, R_0; p)$ // First Square
3 **for** $i = l-2$ **down to** 0 **do**
4 $\quad R_{\neg k_i} \leftarrow \mathrm{MM}(R_0, R_1; p)$ // $i(M)$
5 $\quad R_{k_i} \leftarrow \mathrm{MM}(R_{k_i}, R_{k_i}; p)$ // $i(Q)$
6 **return** $\mathrm{MM}(R_0, 1; p)$

We note that $\mathrm{MM}(m, R^2; p) \equiv mR (\bmod p)$ and $\mathrm{MM}(R_0, 1; p) \equiv R_0 R^{-1} (\bmod p)$ (cf. lines 1 and 6 of Algorithm 2). Besides, the key k is chosen of full length (hence $k_{l-1} = 1$) and must be coprime with $p-1$, which is even (as p is a prime number); therefore, k is odd (hence $k_0 = 1$). This gives for free two bits of information to an attacker. The index l may be determined by an SPA. Moreover, it suffices to recover the exponent k for the exponentiation modulo p: if d denotes the secret RSA key and if $y = x^d (\bmod n)$, then $\gcd(x^k (\bmod n), y) = p$, which factorizes the modulus n (see, e.g., [21], Section 6).

3.2 The core of our attack

We interpret the $s_{i,j}$ as realizations of random variables $S_{i,j}$, i.e., values taken on by $S_{i,j}$, which assume values in $[0, 1)$. Analogously, we view $w_{i(M)}$ and $w_{i(Q)}$ as realizations of $\{0, 1\}$-valued random variables $W_{i(M)}$ and $W_{i(Q)}$.

Lemmas 3.2(i) and (ii) collect known stochastic properties of Montgomery's multiplication algorithm, while Assertions (iii) and (iv) follow the strategy that has proven successful for fixed-window exponentiation in [2,21].

Lemma 3.2. (MM)

(*i*) MM(a, b;p) *requires an extra-reduction iff*

$$\left(\frac{a}{p}\frac{b}{p}\frac{p}{R} + \frac{abp \pmod R}{R} \geq 1\right) \quad \textit{iff} \quad \left(\frac{\mathrm{MM}(a,b,p)}{p} < \frac{a}{p}\frac{b}{p}\frac{p}{R}\right). \tag{3.1}$$

(*ii*) *Assume that* $a \in \mathbb{Z}_p$ *and that the random variable* B *is uniformly distributed on* $\mathbb{Z}_p$. *Furthermore*, U *and* V *denote independent random variables, which are uniformly distributed on* $[0, 1)$. *Then approximately*

$$\mathbb{P}\left(\frac{a}{p}\frac{B}{p}\frac{p}{R} + \frac{aBp \pmod R}{R} \geq 1\right) = \mathbb{P}\left(\frac{a}{p}\frac{p}{R}U + V \geq 1\right) = \frac{p}{2R}\frac{a}{p}, \tag{3.2}$$

$$\mathbb{P}\left(\frac{B}{p}\frac{B}{p}\frac{p}{R} + \frac{B^2p \pmod R}{R} \geq 1\right) = \mathbb{P}\left(\frac{p}{R}U^2 + V \geq 1\right) = \frac{p}{3R}. \tag{3.3}$$

(*iii*) *The random variables* $S_{l,0}, S_{l,1}, S_{l-1,0}, \ldots, S_{0,0}, S_{0,1}$ *may be viewed as iid uniformly distributed on* $[0, 1)$.

(*iv*) *For* $i = l - 1, \ldots, 0$, *we have*

$$W_{i(M)} = \begin{cases} 1_{\left\{S_{i,1} < S_{i+1,0}S_{i+1,1}\frac{p}{R}\right\}} & \textit{if } k_i = 0 \\ 1_{\left\{S_{i,0} < S_{i+1,0}S_{i+1,1}\frac{p}{R}\right\}} & \textit{if } k_i = 1, \end{cases} \tag{3.4}$$

$$W_{i(Q)} = \begin{cases} 1_{\left\{S_{i,0} < S_{i+1,0}^2\frac{p}{R}\right\}} & \textit{if } k_i = 0 \\ 1_{\left\{S_{i,1} < S_{i+1,1}^2\frac{p}{R}\right\}} & \textit{if } k_i = 1. \end{cases} \tag{3.5}$$

(*v*) *For the indicator functions, we obtain*

$$1_{\{W_{i(M)}=1\}} = W_{i(M)}, \quad 1_{\{W_{i(M)}=0\}} = 1 - W_{i(M)}, \tag{3.6}$$

$$1_{\{W_{i(Q)}=1\}} = W_{i(Q)}, \quad 1_{\{W_{i(Q)}=0\}} = 1 - W_{i(Q)}. \tag{3.7}$$

Proof. Assertions (i) and (ii) are shown in [22] (see Lemma A.3 and its proof at page 209). The core idea of the approximate representations (3.2) and (3.3) is that a small deviation of the random variable B (resp. of B/p) causes only a small deviation of the first summand but implies an "uncontrolled large" deviation of the second summand over the unit interval. We note that if U and V are independent, then U and $(a/R)U + V \pmod 1$ are independent, too. Since the base m (Algorithm 2) has been base-blinded, we may assume that $s_{l,0} = r_{l,0}/p = m/p$ is a realization of a random variable $S_{l,0}$, which is uniformly distributed on the unit interval $[0, 1)$. Following (3.3) we further assume that $S_{1,0}$ is also uniformly distributed on $[0, 1)$ and that $S_{l,0}$ and $S_{l,1}$ are independent (see also Remark 3.3). Now let us assume that the random variables $S_{l,0}, S_{l,1}, S_{l-1,0}, \ldots, S_{t+1,1}$ are iid uniformly distributed on $[0, 1)$. If $(k_i, k_{i-1}) = (0, 0)$ we may replace (a/p), U (approximation of B/p), and V in (3.2) by $S_{i+1,0}$, $S_{i+1,1}$, and $V_{i,0}$, and analogously U and V in (3.3) by $S_{i+1,0}$ and $V_{i,1}$, where $V_{i,0}$ and $V_{i,1}$ are uniformly distributed on $[0, 1)$ and independent of $S_{l,0}, \ldots, S_{i+1,1}$. Furthermore, the assumption that $V_{i,0}$ and $V_{i,1}$ are independent seems to be reasonable since $S_{i+1,1}$ and $S_{i,1}$ are independent. This assumption finally implies that the random variables $S_{l,0}, \ldots, S_{i,1}$ are independent. Formula (3.4) follows from (3.1) if we replace the terms (a/p) and (B/p) by $S_{i+1,0}$ and $S_{i+1,1}$ (cf. (3.2)), and further MM(a, b;p)/p by $S_{i,1}$. Analogously, to verify (3.5) one replaces in (3.1) the terms (B/p) and MM(a, b;p)/p by $S_{i+1,0}$ and $S_{i,1}$, respectively. The cases $(k_i, k_{i-1}) \in \{(1, 0), (0, 1), (1, 1)\}$ are similar. Assertion (v) follows immediately from the definition of indicator functions. This completes the proof of Lemma 3.2. □

Remark 3.3. (The independence assumption) A central assertion of Lemma 3.2, which is used in Lemma 3.4, is that random variables $S_{i,j}$ may be viewed iid uniformly distributed on $[0, 1)$. This property has been deduced from the (approximate) stochastic representations (3.2) and (3.3). In a strict sense, this claim is certainly not correct, e.g., because the normalized register values $r_{i,j}/p$ only assume values in the finite set $\mathbb{Z}_p/p \subseteq [0, 1)$, and to mention just one missing number theoretical property, the $r_{i,j}$ cannot assume non-quadratic residua in $\mathbb{Z}_p$. However, this is not relevant for our purposes since we are only interested in the (joint) probabilities of extra reductions. These events can be characterized by "metric" conditions in $\mathbb{R}$ (cf. (3.1), (3.2), (3.3)). It should be noted that the iid assumption on the normalized intermediate random variables of the exponentiation algorithm (here: the $S_{i,j}$) has been proven successful, e.g., in [2, 3,20–22], and it will turn out to be successful in the following, too.

The overall attack consists of many independent decisions (which nevertheless influence each other). Each of these attack steps (decisions) considers all MM simultaneously, which are carried out when u consecutive key bits $(k_i, \ldots, k_{i-u+1})$ are processed. Lemma 3.4 is the core of our attack. It provides the probabilities, which are needed later in Lemma 4.6 (maximum likelihood decision strategy).

Lemma 3.4. *Let $u \geq 2$ and $\overrightarrow{\theta} = (\theta_1, \ldots, \theta_u) \in \{0, 1\}^u$.*

(*i*) *The term* (3.8) *quantifies the probability that the extra-reduction vector $(w_{i(M)}, w_{i(Q)}, \ldots, w_{i-u+1(M)}, w_{i-u+1(Q)})$ occurs if $(k_i, \ldots, k_{i-u+1}) = (\theta_1, \ldots, \theta_u)$. The probabilities are expressed by integrals over $[0, 1)^{2u+2}$. The index $\overrightarrow{\theta}$ shows the dependency on θ.*

$$\begin{aligned}&\mathbb{P}_{\overrightarrow{\theta}}(W_{i(M)} = w_{i(M)}, W_{i(Q)} = w_{i(Q)}, \ldots, W_{i-u+1(M)} = w_{i-u+1(M)},\\ &W_{i-u+1(Q)} = w_{i-u+1(Q)}) = \int_0^1 \int_0^1 \int_{a_1}^{b_1} \int_{a_2}^{b_2} \cdots \int_{a_{2u-1}}^{b_{2u-1}} \int_{a_{2u}}^{b_{2u}} 1 \, \mathrm{d}s_{i-u+1,1} \mathrm{d}s_{i-u+1,0} \ldots \mathrm{d}s_{i,1} \mathrm{d}s_{i,0} \mathrm{d}s_{i+1,1} \mathrm{d}s_{i+1,0}.\end{aligned} \tag{3.8}$$

Note: When the key bit k_j (for $j \in \{i, i-1, \ldots, i-u+1\}$) is processed the register value R_v ($v \in \{0, 1\}$) corresponds to the integration variable $s_{j,v}$. The integration boundaries (a_{2j}, b_{2j}) and (a_{2j-1}, b_{2j-1}) correspond to the integration with regard to the variables $s_{i-j+1,1}$ and $s_{i-j+1,0}$, respectively ($j = 1, \ldots, u$). The integration boundaries depend on the hypothesis $\overrightarrow{\theta} = (\theta_1, \ldots, \theta_u)$ and the observed extra-reduction vector $(w_{i(M)}, w_{i(Q)}, \ldots, w_{i-u+1(M)}, w_{i-u+1(Q)})$. More precisely, for $j \in \{1, \ldots, u\}$ we have

If $\theta_j = 0$, then

$$(a_{2j}, b_{2j}) = \begin{cases} \left(0, s_{i-j+2,0} s_{i-j+2,1} \dfrac{p}{R}\right) & \text{if } w_{i-j+1(M)} = 1 \\ \left(s_{i-j+2,0} s_{i-j+2,1} \dfrac{p}{R}, 1\right) & \text{if } w_{i-j+1(M)} = 0 \end{cases} \tag{3.9}$$

$$(a_{2j-1}, b_{2j-1}) = \begin{cases} \left(0, s_{i-j+2,0}^2 \dfrac{p}{R}\right) & \text{if } w_{i-j+1(Q)} = 1 \\ \left(s_{i-j+2,0}^2 \dfrac{p}{R}, 1\right) & \text{if } w_{i-j+1(Q)} = 0. \end{cases} \tag{3.10}$$

If $\theta_j = 1$, then

$$(a_{2j}, b_{2j}) = \begin{cases} \left(0, s_{i-j+2,1}^2 \dfrac{p}{R}\right) & \text{if } w_{i-j+1(Q)} = 1 \\ \left(s_{i-j+2,1}^2 \dfrac{p}{R}, 1\right) & \text{if } w_{i-j+1(Q)} = 0 \end{cases} \tag{3.11}$$

and

$$(a_{2j-1}, b_{2j-1}) = \begin{cases} \left(0, s_{i-j+2,0} s_{i-j+2,1} \dfrac{p}{R}\right) & \text{if } w_{i-j+1(M)} = 1 \\ \left(s_{i-j+2,0} s_{i-j+2,1} \dfrac{p}{R}, 1\right) & \text{if } w_{i-j+1(M)} = 0. \end{cases} \tag{3.12}$$

(*ii*) *Let* $\overrightarrow{1} := (1, \dots, 1)$ (*with* u *components*). *For each hypothesis* $\overrightarrow{\theta} \in \{0,1\}^u$ *and each extra-reduction vector* $(w_{i(M)}, w_{i(Q)}, \dots, w_{i-t+1(M)}, w_{i-u+1(Q)})$, *we have*

$$\mathbb{P}_{\overrightarrow{\theta}}(W_{i(M)} = w_{i(M)}, \dots, W_{i-u+1(Q)} = w_{i-u+1(Q)}) \tag{3.13}$$

$$\mathbb{P}_{\overrightarrow{1}-\overrightarrow{\theta}}(W_{i(M)} = w_{i(M)}, \dots, W_{i-u+1(Q)} = w_{i-u+1(Q)}). \tag{3.14}$$

Proof. By Lemma 3.2(iv), the random variables $W_{i(M)}, W_{i(Q)}, \dots, W_{i-u+1(M)}, W_{i-u+1(Q)}$ can be expressed by indicator functions, which depend on the random variables $S_{i+1,1}, S_{i+1,0}, \dots, S_{i-u+1,1}, S_{i-u+1,0}$. This allows us to express the probability (3.8) by an integral over $[0,1)^{2u+2}$ of a product of indicator functions. Furthermore, for $j < u$ the indicator functions $1_{\{W_{i-j+1(M)}=w_{i-j+1(M)}\}}$ and $1_{\{W_{i-j+1(Q)}=w_{i-j+1(Q)}\}}$ actually only depend on $s_{i+1,1}$, $s_{i+1,0}, \dots, s_{i-u+2,1}, s_{i-u+2,0}$, while $1_{\{W_{i-u+1(M)}=w_{i-u+1(M)}\}}$ and $1_{\{W_{i-u+1(Q)}=w_{i-u+1(Q)}\}}$ merely depend on $s_{i-u+2,1}$, $s_{i-u+2,0}$, $s_{i-u+1,1}$, $s_{i-u+1,0}$. This allows us to express (3.8) in the form

$$\int_{[0,1)^{2u}} \prod_{j=1}^{u-1} (1_{\{W_{i-j+1(M)}=w_{i-j+1(M)}\}} \cdot 1_{\{W_{i-j+1(Q)}=w_{i-j+1(Q)}\}}) \cdot \left(\int_{a_{2u-1}}^{b_{2u-1}} \int_{a_{2u}}^{b_{2u}} 1 \, \mathrm{d}s_{i-u+1,1} \mathrm{d}s_{i-u+1,0} \right) \mathrm{d}s_{i-u+2,1} \dots \mathrm{d}s_{i+2,0} \tag{3.15}$$

with suitable integration boundaries a_{2u-1}, b_{2u-1}, a_{2u}, b_{2u}. These integration boundaries follow immediately from Lemma 3.2(iv) and (ii) with $i - u + 1$ in place of i. This verifies the formula (3.9) to (3.12) for $j = u$. The integral over $[0,1)^{2u}$ can be transformed in the same way into a sequence of one-dimensional integrals. Since the integration boundaries $a_1, b_1, \dots, a_{2u-2}, b_{2u-2}$ depend only on the left-hand indicator functions, i.e., on the observations $w_{i(M)}, w_{i(Q)}, \dots, w_{i-u+2(M)}, w_{i-u+2(Q)}$ Lemma 3.4(i) can be verified by induction on u.

We first note that

$$\begin{aligned} &\phi : [0,1)^{2u+2} \to [0,1)^{2u+2}, \\ &\phi(s_{i+1,1}, s_{i+1,0}, \dots, s_{i-u+1,1}, s_{i-u+1,0}) := (s_{i+1,0}, s_{i+1,1}, \dots, s_{i-u+1,1}, s_{i-u+1,0}) \end{aligned}$$

(swapping the right-hand indices from 0 to 1 and vice versa) defines a volume-preserving diffeomorphism on $[0,1)^{2u+2}$. As already pointed out above the probabilities (3.13) and (3.14) can be expressed by integrals over $[0,1)^{2u+2}$ of indicator functions

$$\prod_{j=1}^{u} 1_{[\overrightarrow{\theta}]\{W_{i-j+1(M)}=w_{i-j+1(M)}\}} \cdot 1_{[\overrightarrow{\theta}]\{W_{i-j+1(Q)}=w_{i-j+1(Q)}\}}$$

and

$$\prod_{j=1}^{u} 1_{[\overrightarrow{1}-\overrightarrow{\theta}]\{W_{i-j+1(M)}=w_{i-j+1(M)}\}} \cdot 1_{[\overrightarrow{1}-\overrightarrow{\theta}]\{W_{i-j+1(Q)}=w_{i-j+1(Q)}\}}$$

respectively. The terms $\overrightarrow{\theta}$ and $\overrightarrow{1} - \overrightarrow{\theta}$ indicate the hypotheses. From Lemma 3.2(iv), we conclude that $1_{[\overrightarrow{1}-\overrightarrow{\theta}]\{W_{i-j+1(M)}=w_{i-j+1(M)}\}} = 1_{[\overrightarrow{\theta}]\{W_{i-j+1(M)}=w_{i-j+1(M)}\}} \circ \phi$ and $1_{[\overrightarrow{1}-\overrightarrow{\theta}]\{W_{i-j+1(Q)}=w_{i-j+1(Q)}\}} = 1_{[\overrightarrow{\theta}]\{W_{i-j+1(Q)}=w_{i-j+1(Q)}\}} \circ \phi$ for all $j \le u$, which completes the proof of Assertion (ii). □

Lemma 3.4(ii) says that the information contained in the extra-reduction vectors $(w_{i(M)}, \dots, w_{i-u+1(Q)})$ does not allow us to distinguish between the hypotheses $\overrightarrow{\theta}$ and $\overrightarrow{1} - \overrightarrow{\theta}$. This means that we can only determine the set $\{\overrightarrow{\theta}, \overrightarrow{1} - \overrightarrow{\theta}\}$, as depicted in Figure 1.

In particular, it would be pointless to consider the case $u = 1$. For $u = 2$ one can distinguish between the cases $(k_i, k_{i-1}) \in \{(0,0), (1,1)\}$ and $(k_i, k_{i-1}) \in \{(0,1), (1,0)\}$, or equivalently, between $k_i = k_{i-1}$ and $k_i \neq k_{i-1}$. For $u \ge 2$, the parameter $\overrightarrow{\theta} \in \{(\theta_1, \dots, \theta_u), (1-\theta_1, \dots, 1-\theta_u)\}$ corresponds to

$$(k_i \oplus k_{i-1} = \theta_1 \oplus \theta_2, \dots, k_{i-u+2} \oplus k_{i-u+1} = \theta_{u-1} \oplus \theta_u), \tag{3.16}$$

where "$\oplus$" denotes the addition modulo 2. For the sake of clarity, we precise that the components of vector $\overrightarrow{1} - \overrightarrow{\theta}$ can also be written as $(1 \oplus \theta_i) = \neg\theta_i$ for $1 \le i \le u$.

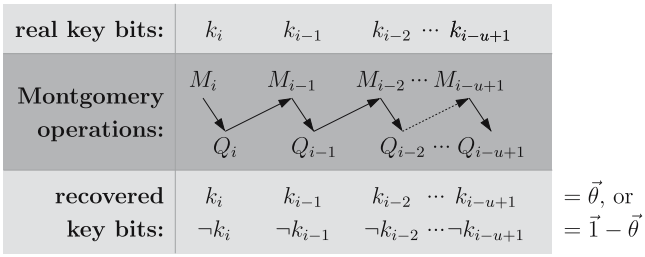

Figure 1: Information collected during the presented attack on u pairs of extra-reductions.

Remark 3.5.

(i) Lemma 3.4 can be applied to all u-tuples $(k_i, \dots, k_{i+u-1})$ for $i = l-1, \dots, u-1$. Combining the information from all u-tuples only provides the vector $(k_{l-1} \oplus k_{l-2}, \dots, k_1 \oplus k_0)$. This information determines the whole key $k = (k_l = 1, k_{l-1}, \dots, k_0)$ since k is odd due to $\gcd(k, \varphi(p)) = 1$ (where we recall that φ is Euler totient function).

(ii) The probabilities in Lemma 3.4 do not depend on the index i. By Lemma 3.4(ii), it suffices to compute at most 2^{3u-1} probabilities of type (3.8). (Note that 2^{2u} different extra-reduction vectors exist and one has to distinguish between 2^{u-1} hypotheses.) Example 3.6 illustrates the calculation of one particular probability, and the appendix contains two tables with all probabilities for $u = 2$.

(iii) For $u = 2$, our attack aims at pairs of consecutive key bits (k_i, k_{i-1}). This is like the original attack in [10,11], but the original attack only exploits the extra reductions $(w_{Q(i)}, w_{M(i-1)})$ while our attack considers $(w_{M(i)}, w_{Q(i)}, w_{M(i-1)}, w_{Q(i-1)})$. The probabilities, which are applied in the original attack, are the marginal probabilities of the probability (3.8) with regard to $(w_{M(i)}, w_{Q(i-1)})$. Obviously, the original attack exploits less information than the new attack for $u = 2$, and experiments confirm that for $u = 2$ our new attack reduces by a factor greater than 2 the number of queries (cf. Figure 3).

Example 3.6. Let $(\theta_1, \theta_2) = (0, 1)$ and $(w_{i(M)}, w_{i(Q)}, w_{i-1(M)}, w_{i-1(Q)}) = (1, 1, 0, 1)$. By Lemma 3.4(i),

$$\begin{aligned}
&\mathbb{P}_\theta(W_{i(M)} = 1, W_{i(Q)} = 1, W_{i-1(M)} = 0, W_{i-1(Q)} = 1) \\
&= \int_0^1 \int_0^1 \int_0^{s_{i+1,0}s_{i+1,1}\frac{p}{R}} \int_0^{s_{i+1,1}^2\frac{p}{R}} \int_0^{s_{i,0}^2\frac{p}{R}} \int_{s_{i,0}s_{i,1}\frac{p}{R}}^1 1 \, \mathrm{d}s_{i-1,1}\mathrm{d}s_{i-1,0}\mathrm{d}s_{i,1}\mathrm{d}s_{i,0}\mathrm{d}s_{i+1,1}\mathrm{d}s_{i+1,0} \\
&= \cdots = \int_0^1 \int_0^1 \int_0^{s_{i+1,0}s_{i+1,1}\frac{p}{R}} \int_0^{s_{i+1,1}^2\frac{p}{R}} \left(s_{i,0}^2\frac{p}{R} - s_{i,0}^3 s_{i,1}\left(\frac{p}{R}\right)^2\right)\mathrm{d}s_{i,1}\mathrm{d}s_{i,0}\mathrm{d}s_{i+1,1}\mathrm{d}s_{i+1,0} \qquad (3.17) \\
&= \cdots = \int_0^1 \int_0^1 \left(\frac{1}{3}s_{i+1,0}^3 s_{i+1,1}^5\left(\frac{p}{R}\right)^5 - \frac{1}{8}s_{i+1,0}^4 s_{i+1,1}^8\left(\frac{p}{R}\right)^8\right)\mathrm{d}s_{i+1,1}\mathrm{d}s_{i+1,0} \\
&= \cdots = \frac{1}{3\cdot 4\cdot 6}\left(\frac{p}{R}\right)^5 - \frac{1}{8\cdot 5\cdot 9}\left(\frac{p}{R}\right)^8 = \frac{1}{72}\left(\frac{p}{R}\right)^5 - \frac{1}{360}\left(\frac{p}{R}\right)^8.
\end{aligned}$$

Corollary 3.7. *For $u = 2$ by applying the law of total probability on $\mathbb{P}_\theta$ in* (3.8)*, the joint probability for maximum likelihood described in* [10, 11, Theorem 2] *can be recovered.*

Remark 3.8. The two approaches in previous work [10, 11] and this work are independent and both allow us to derive a maximum likelihood key distinguisher. Here, we are not interested in the values manipulated by the multiplication and square operations, but only with the necessary and sufficient conditions for the existence of extra-reductions, allowing an analysis of larger dimensions.

4 Perfect and noisy measurements

The attacker gets access to side-channel information about each bit k_i ($l-2 \geq i > 0$) of the exponent k through the noised distribution of the pair of extra-reductions $(W_{i(M)}, W_{i(Q)})$. The noise consists in two binary random variables $(N_{i(M)}, N_{i(Q)})$. Additionally, the random variables $N_{i(M)}$ and $N_{i(Q)}$ are assumed independent and identically distributed (iid), as is usually the case of measurement noise of different operations in a side-channel trace. Namely, we denote by p_{noise} the probability

$$p_{\text{noise}} = \mathbb{P}(N_{i(M)} = 1) = \mathbb{P}(N_{i(Q)} = 1) \quad \text{for all } i.$$

Thus, the attacker garners an iid sequence $(\underline{y_{i(M)}}, \underline{y_{i(Q)}}) = (y_{i(M);n}, y_{i(Q);n})_{n=1,\dots,\mathcal{N}}$, where for each query n and exponent index $i \in \{l-1, \dots, 0\}$, $y_{i(M);n} = x_{i(M);n} \oplus n_{i(M);n}$ and $y_{i(Q);n} = x_{i(Q);n} \oplus n_{i(Q);n}$. This means that $W_{i(M)}$ and $Y_{i(M)}$ are, respectively, the input and the output of a binary symmetric channel (BSC) of parameter p_{noise}. Similarly, $W_{i(Q)}$ and $Y_{i(Q)}$ are also input and output of an independent identical BSC parallel to the first one.

In practical cases, detecting an extra-reduction using only one acquisition can lead to errors. Let us model the attack setup, taking into account that the detection of presence/absence of extra-reductions is a random variable, due to some noise. The random variables Markov chain for index i is given as follows:

Secret	$\longrightarrow$	Bias	$\longrightarrow$	Observable
$K = k_i$	$\longrightarrow$	$(W_{i(M)}, W_{i(Q)})$	$\longrightarrow$	$(Y_{i(M)}, Y_{i(Q)}) = (W_{i(M)} \oplus N_{i(M)}, W_{i(Q)} \oplus N_{i(Q)})$.

The probabilities (3.8) depend on the unknown ratio p/R. The crucial observation is that the attacker knows the position of all squarings and all multiplications. Lemma 4.2 provides concrete formula, which allows us to estimate p/R. Of course, this estimation step is only necessary for RSA with CRT but not for RSA without CRT. We begin with a lemma, which will be needed.

Lemma 4.1. *It is*

$$E(W_{i(M)}) = \mathbb{P}(W_{i(M)} = 1) = \frac{1}{4} \cdot \frac{p}{R}, \tag{4.1}$$

$$E(W_{i(Q)}) = \mathbb{P}(W_{i(Q)} = 1) = \frac{1}{3} \cdot \frac{p}{R}, \tag{4.2}$$

$$\frac{p}{R} = 3E(W_{i(Q)}) = 2E(W_{i(M)}) + 1.5E(W_{i(Q)}). \tag{4.3}$$

Proof. Since $W_{i(M)}$ and $W_{i(Q)}$ assume values in $\{0, 1\}$ the left-hand side equations in (4.1) and (4.2) are obvious, while the right-hand side equation follow immediately from (3.4) and (3.5), respectively. For $k_i = 0$, for instance,

$$\mathbb{P}(W_{i(M)} = 1) = \int_0^1 \int_0^1 \int_0^{s_{i+1,0} s_{i+1,1} p/R} 1 \, \mathrm{d}s_{i+1,1} \mathrm{d}s_{i+1,0} \mathrm{d}s_{i,1} = \frac{1}{4} \cdot \frac{p}{R}.$$

We note that the probability (4.2) was already verified in [20][3] and, for instance, in [11], respectively, the latter by other mathematical methods. Formula (4.3) follows directly from (4.1) and (4.2). □

The ER-values $w_{i(M)}$ and $w_{i(Q)}$ are determined (or more precisely: guessed) on the basis of single-trace template attacks. In particular, their guesses $\widetilde{w}_{i(M)}$ and $\widetilde{w}_{i(Q)}$ might be incorrect with some probability. We

3 Actually, (4.2) was proven in [20] but not (4.1). In [20], the square and multiply algorithm was considered where multiplications with a fixed value (MM transformed basis mod p) are carried out. Formula (4.1) considers the case of two random factors.

denote the corresponding random variables (referring to the guessed ER values) by $\widetilde{W}_{i(M)}$ and $\widetilde{W}_{i(Q)}$. In the following, we assume that

$$\begin{aligned}&\mathbb{P}(\widetilde{W}_{i(M)} = v | W_{i(M)} = 1 - v) = \mathbb{P}(\widetilde{W}_{i(Q)} = v | W_{i(Q)} = 1 - v) = p_{\text{noise}}\\ &\quad \text{for } i \in \{0, \dots, l-1\} \text{ and } v \in \{0, 1\},\end{aligned} \tag{4.4}$$

and similarly for the initialization of the registers R_0 and R_1 in Algorithm 2. In other words, the probability of guessing an ER value incorrectly is $p_{\text{noise}} \geq 0$, independently of the true value. Of course, $p_{\text{noise}} = 0$ characterizes a perfect side-channel measurement. Lemma 4.2(iii) is the generalization of (4.3) for noisy measurements. As noted in Lemma 4.4, this allows the estimation of p/R and p_{noise}.

Lemma 4.2.

$$\frac{p}{R} = \frac{12E(\widetilde{W}_{i(Q)}) - 12E(\widetilde{W}_{i(M)})}{1 + 6E(\widetilde{W}_{i(Q)}) - 8E(\widetilde{W}_{i(M)})}, \tag{4.5}$$

$$p_{\text{noise}} = 4E(\widetilde{W}_{i(M)}) - 3E(\widetilde{W}_{i(Q)}). \tag{4.6}$$

Proof. Since $\widetilde{W}_{i(Q)}$ is $\{0, 1\}$-valued, we obtain

$$\begin{aligned}E(\widetilde{W}_{i(Q)}) &= \mathbb{P}(\widetilde{W}_{i(Q)} = 1)\\ &= \mathbb{P}(\widetilde{W}_{i(Q)} = 1 | W_{i(Q)} = 1)\mathbb{P}(W_{i(Q)} = 1) + \mathbb{P}(\widetilde{W}_{i(Q)} = 1 | W_{i(Q)} = 0)\mathbb{P}(W_{i(Q)} = 0)\\ &= (1 - p_{\text{noise}})\frac{p}{3R} + p_{\text{noise}}\left(1 - \frac{p}{3R}\right),\end{aligned}$$

and similarly

$$\begin{aligned}E(\widetilde{W}_{i(M)}) &= \mathbb{P}(\widetilde{W}_{i(M)} = 1)\\ &= \mathbb{P}(\widetilde{W}_{i(M)} = 1 | W_{i(M)} = 1)\mathbb{P}(W_{i(M)} = 1) + \mathbb{P}(\widetilde{W}_{i(M)} = 1 | W_{i(M)} = 0)\mathbb{P}(W_{i(M)} = 0)\\ &= (1 - p_{\text{noise}})\frac{p}{4R} + p_{\text{noise}}\left(1 - \frac{p}{4R}\right).\end{aligned}$$

Solving these equations for (p/R) and p_{noise} yields (4.5) and (4.6). □

In Lemma 4.3, $(e_{1(M)}, e_{1(Q)}, \dots, e_{u(M)}, e_{u(Q)}) \in \{0, 1\}^{2u}$ represents the "error vector" and ham corresponds to the Hamming weight of a value. The nonzero entries give the positions at which the guessed extra-reduction vector $(\widetilde{w}_{i(M)}, \widetilde{w}_{i(Q)}, \dots, \widetilde{w}_{i-u+1(M)}, \widetilde{w}_{i-u+1(Q)})$ are incorrect.

Lemma 4.3.

(*i*)

$$\begin{aligned}&\mathbb{P}_{\overrightarrow{\theta}}(\widetilde{W}_{i-j+1(M)} = \widetilde{w}_{i-j+1(M)}, \widetilde{W}_{i-j+1(Q)} = \widetilde{w}_{i-j+1(Q)} | j = 1, \dots, u)\\ &= \sum_{\substack{0 \leq e_{j(M)}, e_{j(Q)} \leq 1\\ 1 \leq j \leq u}} \mathbb{P}_{\overrightarrow{\theta}}(W_{i-j+1(M)} = \widetilde{w}_{i-j+1(M)} \oplus e_{j(M)}, W_{i-j+1(Q)} = \widetilde{w}_{i-j+1(Q)} \oplus e_{j-i+1(Q)} | j = 1, \dots, u)\\ &\quad \times p_{\text{noise}}^{\text{ham}(e_{1(M)}, \dots, e_{u(Q)})}(1 - p_{\text{noise}})^{2u - \text{ham}(e_{1(M)}, \dots, e_{u(Q)})}.\end{aligned} \tag{4.7}$$

(*ii*) *For each hypothesis* $\overrightarrow{\theta} \in \{0, 1\}^u$ *and each* (*guessed*) *extra-reduction vector* $(\widetilde{w}_{i(M)}, \widetilde{w}_{i(Q)}, \dots, \widetilde{w}_{i-u+1(M)}, \widetilde{w}_{i-u+1(Q)})$, *we have*

$$\mathbb{P}_{\overrightarrow{\theta}}(\widetilde{W}_{i(M)} = \widetilde{w}_{i(M)}, \dots, \widetilde{W}_{i-u+1(Q)} = \widetilde{w}_{i-u+1(Q)}) = \mathbb{P}_{\overrightarrow{1} - \overrightarrow{\theta}}(\widetilde{W}_{i(M)} = \widetilde{w}_{i(M)}, \dots, \widetilde{W}_{i-u+1(Q)} = \widetilde{w}_{i-u+1(Q)}). \tag{4.8}$$

Proof. The term $p_{\text{noise}}^{\text{ham}(e_{1(M)}, \dots, e_{u(Q)})}(1 - p_{\text{noise}})^{2u - \text{ham}(e_{1(M)}, \dots, e_{u(Q)})}$ quantifies the probability for the error vector $(e_{1(M)}, e_{1(Q)}, \dots, e_{u(M)}, e_{u(Q)})$. This fact and the definition of the conditional probability imply (4.7). Assertion (ii) follows immediately from (i) and Lemma 3.4(ii), applied to the particular right-hand probabilities in (4.7). □

The last lemma of this section explains how to estimate the ratio p/R and the probability p_{noise}.

Lemma 4.4. *Assume that the attacker has observed $\mathcal{N}$ side-channel traces. Then*

$$\widetilde{\mu}_M := \frac{1}{\mathcal{N}l}\sum_{n=1}^{\mathcal{N}}\sum_{i=0}^{l-1}\widetilde{w}_{i(M);n} \tag{4.9}$$

provides an estimator for $E(\widetilde{W}_{i(M);n})$ and analogously

$$\widetilde{\mu}_Q := \frac{1}{\mathcal{N}l}\sum_{n=1}^{\mathcal{N}}\sum_{i=0}^{l-1}\widetilde{w}_{i(Q);n} \tag{4.10}$$

for $E(\widetilde{W}_{i(Q);n})$. The index n refers to the numbering of the side-channel traces.

(ii) Substituting $\widetilde{\mu}_M$ and $\widetilde{\mu}_Q$ for $E(\widetilde{W}_{i(M);n})$ and $E(\widetilde{W}_{i(Q);n})$ into (4.5) *and* (4.6) *yields estimates $\widetilde{p/R}$ and $\widetilde{p}_{\text{noise}}$.*

(iii) For perfect measurements alternatively (4.3) *might be used to estimate p/R. Compared to the mid-term the right-hand term considers twice as many MM and thus should provide a more precise estimate.*

Proof. Straightforward. □

Example 4.5. (Estimation of p/R and p_{noise}) For different exponents of 512-bit length, we estimate $\widetilde{p/R}$ and $\widetilde{p_{\text{noise}}}$ for two moduli (`RSA-1024-p` and `RSA-1024-q` defined in [11, Section 2.2]) and different values of p_{noise} depending on the number of side-channel traces $\mathcal{N}$. For each value of $\mathcal{N}$ between 0 and 500, we compute $\widetilde{p/R}$ using (4.5) and $\widetilde{p_{\text{noise}}}$ using (4.6) for the different exponents and the found values are represented using a box plot (deciles/quartile/median values) in Figure 2.

4.1 The optimal decision strategy

Lemma 4.6 provides the optimal decision strategy for the individual decisions, i.e., for guessing the parameter set $\{\vec{\theta}, 1-\vec{\theta}\}$ for the particular u-tuples $(k_i,\dots,k_{i-u+1})$. The decision strategy exploits the information from the observed (guessed) ER-vectors from $\mathcal{N}$ side-channel traces. For $p_{\text{noise}} = 0$, Lemma 4.6 describes the situation in the case of perfect measurements.

Lemma 4.6. (Maximum likelihood estimator) *Assume that the key k has been selected randomly and that the attacker has no information on the subkey $(k_i, \dots, k_{i-u+1})$. Let*

$$\hat{\vec{\theta}} := \underset{\vec{\theta}\in\{0,1\}^u}{\operatorname{argmax}} \prod_{n=1}^{\mathcal{N}} \mathbb{P}_{\vec{\theta}}(\widetilde{W}_{i-j+1(M);n} = \widetilde{w}_{i-j+1(M);n},\ \widetilde{W}_{i-j+1(Q);n} = \widetilde{w}_{i-j+1(Q);n} | j = 1,\dots,u). \tag{4.11}$$

(i) $\hat{\vec{\theta}}$ maximizes the right-hand side of (4.11) *iff $\vec{1} - \hat{\vec{\theta}}$ maximizes the right-hand side of* (4.11)*. It thus suffices to compute the right-hand term of* (4.11) *for all $\vec{\theta} \in \{\{0,1\}^u | \theta_u = 0\}$, or, without loss of generality, by fixing one arbitrary bit within $\{0,1\}^u$.*

(ii) The attacker decides for

$$(k_i \oplus k_{i-1} = \hat{\theta}_1 \oplus \hat{\theta}_2, \dots, k_{i-u+2} \oplus k_{i-u+1} = \hat{\theta}_{u-1} \oplus \hat{\theta}_u). \tag{4.12}$$

This is the optimal decision strategy.

Proof. The first assertion of (i) follows from Lemma 3.4(ii), and the second is an immediate consequence of the first. With regard to the assumptions on k and on the subkey $(k_i,\dots,k_{i-u+1})$ we interpret the unknown subkey $(k_i,\dots,k_{i-u+1})$ as a realization of random variable, which is uniformly distributed on $\{0,1\}^u$. Then $(k_i \oplus k_{i-1}, \dots, k_{i-u+2} \oplus k_{i-u+1})$ may be viewed as a realization of a random variable, which is uniformly

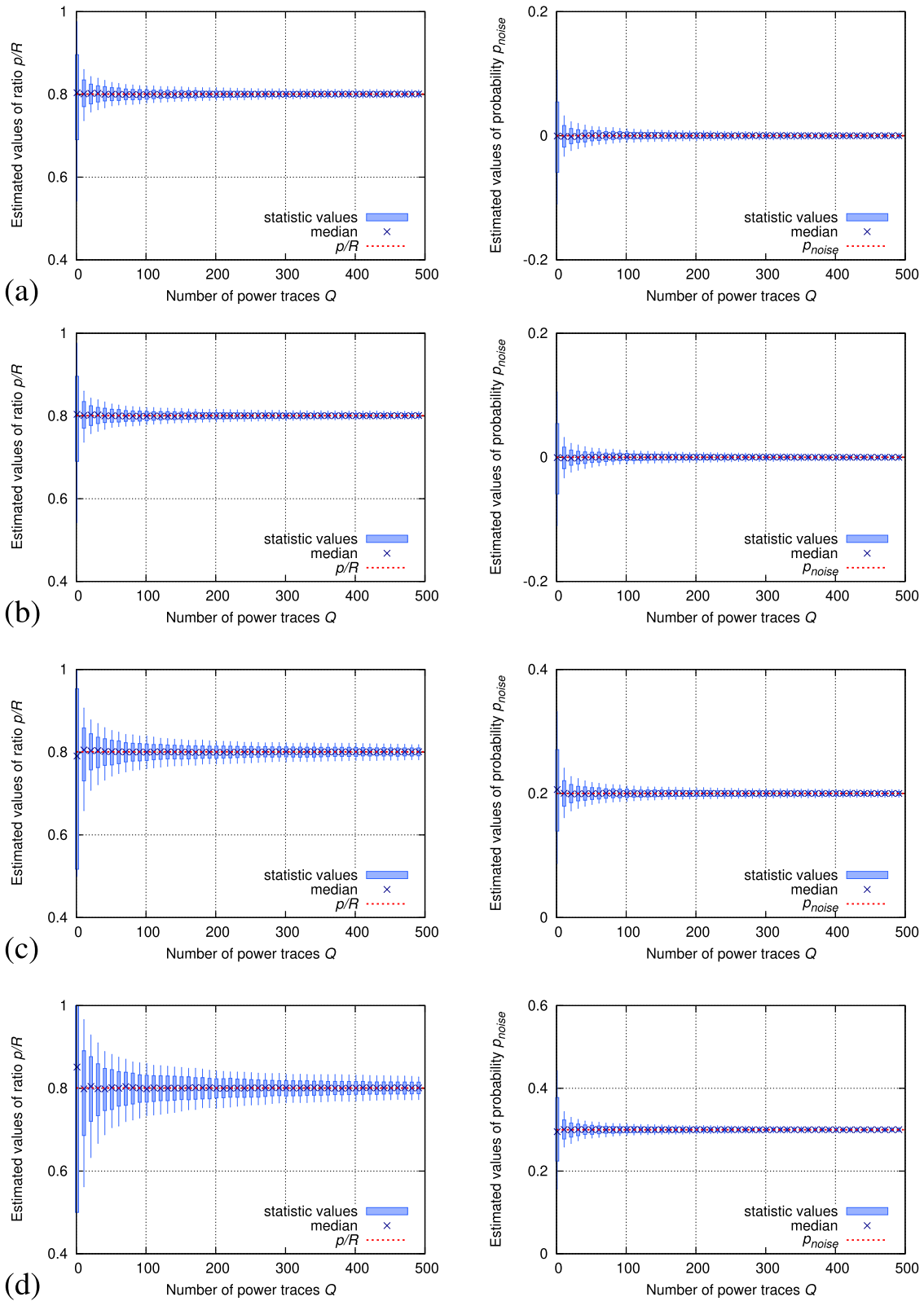

Figure 2: Statistic box plot to estimate the ratio p/R and the probability p_{noise} in function of side-channel traces $\mathcal{N}$ using 1.000 randomly selected exponent values. (a) $p/R \simeq 0.800907$ and $p_{\text{noise}} \simeq 0.00$, (b) $p/R \simeq 0.789290$ and $p_{\text{noise}} \simeq 0.00$, (c) $p/R \simeq 0.800907$ and $p_{\text{noise}} \simeq 0.20$, (d) $p/R \simeq 0.789290$ and $p_{\text{noise}} \simeq 0.30$.

distributed on $\{0, 1\}^{u-1}$. Furthermore, $(k_i, \ldots, k_{i-u+1}) \in \{\overrightarrow{\theta}, \overrightarrow{1} - \overrightarrow{\theta}\}$ iff $(k_i \oplus k_{i-1} = \theta_i \oplus \theta_{i-1}, \ldots, k_{i-u+2} \oplus k_{i-u+1} = \theta_{i-u+2} \oplus \theta_{i-u+1})$. Hence, (4.11) yields the maximum likelihood estimator for the transformed subkey $(k_i \oplus k_{i-1}, \ldots, k_{i-u+2} \oplus k_{i-u+1})$. If we assume that each false decision is equally bad the optimal decision

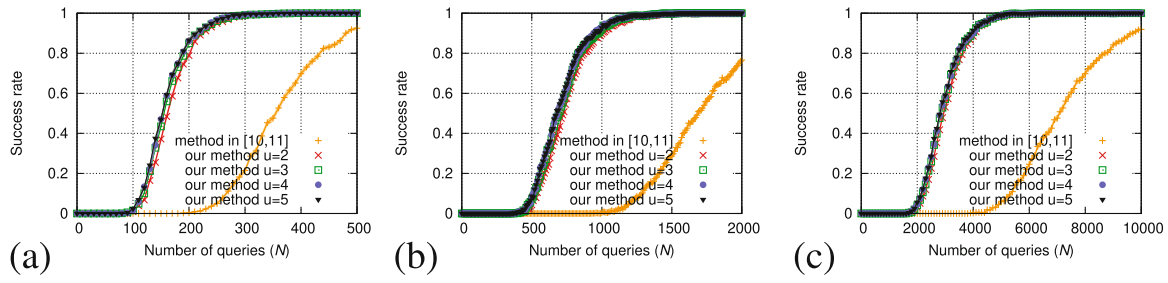

Figure 3: Success rate for an entire exponent using 1.000 randomly selected exponent values depending on the number of side-channel traces $\mathcal{N}$ with different noise probabilities p_{noise}: (a) $p_{\text{noise}} = 0.00$, (b) $p_{\text{noise}} = 0.10$, (c) $p_{\text{noise}} = 0.20$.

strategy (Bayes strategy against the uniform distribution on $\{0,1\}^{u-1}$, identical loss for all types of errors) is given by the maximum likelihood estimator, which completes the proof of Lemma 4.6. □

Remark 4.7.

(i) Lemma 4.6 assumes that p/R and p_{noise} are known. Substituting $\widetilde{p/R}$ and $\widetilde{p}_{\text{noise}}$ into (4.11) yields estimates for the probabilities

$$\mathbb{P}_{\vec{\theta}}(\widetilde{W}_{i(M)} = \widetilde{w}_{i(M)}, \ldots, \widetilde{W}_{i-u+1(Q)} = \widetilde{w}_{i-u+1(Q)}).$$

(ii) In the proof of Lemma 4.6, we assume that $(k_i, \ldots, k_{i-u+1})$ is a realization of a uniformly distributed random variable on $\{0,1\}^u$. This assumption may not be justified for $i = l-2$ and in particular not for $i = u-1$ since $k_0 = 1$. Since we are only interested in the distribution on $\{0,1\}^{u-1}$ (cf. equation (4.12)), this relaxes the uniformity condition.

(iii) In the proof of Lemma 4.6, we assume that each false decision is equally bad. This assumption is reasonable if all transformed subkeys

$$(k_i \oplus k_{i-1}, \ldots, k_{i-u+2} \oplus k_{i-u+1})$$

are treated independently.

4.2 Attack summary and success rate

The decision strategy in Lemma 4.6 is based on the observed extra-reductions for each multiply and square operation for $\mathcal{N}$ calls of the cryptographic operation with a static key k of l-bit length ($k_{l-1} = 1$ and $k_0 = 1$, as described in Algorithm 2). For each $2u$-tuple of (noisily) observed extra reductions

$$(\tilde{y}_{i(M)}, \tilde{y}_{i(Q)}, \ldots, \tilde{y}_{i-u+1(M)}, \tilde{y}_{i-u+1(Q)}) \in \{0,1\}^{2u},$$

the attacker estimates the $\vec{\theta}_i$ value using the maximum likelihood estimator like described in Lemma 4.6 using only the probabilities $\mathbb{P}_\theta$ (for $u = 2$ and $p_{\text{noise}} = 0$ the probabilities are given as polynomials in the ratio p/R in the informative Appendix A). Algorithm 3 permits us to retrieve the key bit values. It is a windowed algorithm, which recovers an estimation $\hat{k}$ of the secret key k by tuples of u bits. In Algorithm 3, i takes values $(u-1)$, $2(u-1)$, $3(u-1)$, etc. The first u-bit window considers the $2u$ Montgomery operations, which depend on the key bits $k_{u-1}, \ldots, k_0$. Due to Lemma 4.3, subsequent windows overlap in one bit position. Note that at lines 4 and 16 of Algorithm 3, the final value of i must be $(l-2)$, which might not be a multiple of $(u-1)$ depending on the values of l and u. Thence, the final value of i is adjusted to be equal to $(l-2)$. In this case, the last window consists in bits of indices $\{l-2, \ldots, l-u-1\}$, which overlaps the last but one window in more than one bit position. Alternatively, the final maximum likelihood can be computed for a smaller window (of length $< u$). Our first proposal saves the computation of additional probabilities (step 3 of Algorithm 3), hence it is adopted in Algorithm 3, and put in force at lines 7 and 17.

The last steps of Algorithm 3 consist in putting together pieces of $(u-1)$ bits of the key guess. Simple error correction can be applied at this stage, to fix easily one or two errors while rebuilding the full l bits of the secret exponent. For each trial only the loop from line 16 in Algorithm 3 has to be executed (with

modified guesses $\widehat{\vec{\theta}_i}$ for an index $i = i_1$ or for two indices $i \in \{i_1, i_2\}$) and the Euclidean algorithm, which is not costly. We point out that in Definition 4.8 we do not allow any false decision for the particular u-bit windows for the sake of a fair comparison with the attacks in [10,11]. If we did so this would increase our success rate to some extent (and those in [10,11] as well).

Algorithm 3. Optimal extra-reduction attack using maximum likelihood estimator

Input: $(\underline{\widetilde{w}_{l-2(M)}}, \underline{\tilde{w}_{l-2(Q)}}, \ldots, \underline{\tilde{w}_{0(M)}}, \underline{\tilde{w}_{0(Q)}})$, a set of $\mathcal{N}(l-1)$ pairs of noisy bits (extra-reductions)

Output: A guessed key value $\hat{k} \in \{0,1\}^l$

Attack phase

1 Estimate the ratio p/R and the probability p_{noise} (by their estimated values $\widetilde{p/R}$ and $\widetilde{p_{\text{noise}}}$ using Lemma 4.4)

2 **for each** $\vec{\theta} = (\theta_1, \ldots, \theta_u) \in \{0,1\}^u$ **with** $\theta_u = 0$ **do**

3 Compute the probability law $\mathbb{P}_{\vec{\theta}}(\widetilde{W}_{i(M)}, \widetilde{W}_{i(Q)}, \ldots, \widetilde{W}_{i-u+1(M)}, \widetilde{W}_{i-u+1(Q)})$ using the ratio $\widetilde{p/R}$ and the value $\widetilde{p_{\text{noise}}}$ by (4.7)

4 **for** $i = u-1$ **up to** $\left\lceil \frac{l-1}{u-1} \right\rceil (u-1)$ **by step** $(u-1)$ **do**

5 **if** $i > l-2$ **then**

6 $i \leftarrow l-2$ // The last window is left–justified on $[l-2, \ldots, l-2-(u-1)] = [l-2, \ldots, l-u-1]$

7 **for** $(v_1, \ldots, v_{2u}) \in \{0,1\}^{2u}$ **do** $\mathsf{Accum}(v_1, \ldots, v_{2u}) \leftarrow 0$

8 **for** $n = 1$ **to** $\mathcal{N}$ **do**

9 $\mathsf{Accum}(\tilde{w}_{i(M);n}, \tilde{w}_{i(Q);n}, \ldots, \tilde{w}_{i-u+1(M);n}, \tilde{w}_{i-u+1(Q);n})$++ // incrementing observed entry of Accum by 1

10 **for each** $\vec{\theta} \in \{0,1\}^u$ **with** $\theta_u = 0$ **do**

11 $T_{\vec{\theta}} \leftarrow 0$ // (Non–normalized) log-likelihood

12 **for each tuple** $(v_1, v_2, \ldots, v_{2u-1}, v_{2u}) \in \{0,1\}^{2u}$ **do**

13 $T_{\vec{\theta}} \leftarrow T_{\vec{\theta}} + \mathsf{Accum}(v_1, \ldots, v_{2u}) \times \ln(\mathbb{P}_{\vec{\theta}}(v_1, v_2, \ldots, v_{2u-1}, v_{2u}))$ // See Lemma 4.6

14 $\widehat{\vec{\theta}_i} = (\hat{\theta}_{i,1}, \ldots, \hat{\theta}_{i,u}) \leftarrow \operatorname{argmax}_{\vec{\theta}}(T_{\vec{\theta}})$ // see Lemma 4.6

Computation of the estimated key value

15 $\hat{k}_0 \leftarrow 1, \hat{k}_{l-1} \leftarrow 1$ // by definition of the key (see Alg. 2)

16 **for** $i = u-1$ **up to** $\left\lceil \frac{l-1}{u-1} \right\rceil (u-1)$ **by step** $(u-1)$ **do**

17 **if** $i > l-2$ **then** $i \leftarrow l-2$ // See line 6 of this Alg. 3

18 **for** $j = 1$ **to** u **do** $\hat{k}_{i-j+1} \leftarrow \hat{\theta}_{i,j} \oplus \hat{k}_{i-u+1}$

19 **return** $\hat{k} = (\hat{k}_{l-1}\hat{k}_{l-2} \ldots \hat{k}_0)_2$

In order to compare the previous work and this optimized method, we compute the success rate of those attacks. In this article, we define the success rate of a whole exponent value.

Definition 4.8. (Success rate of an attack) The success rate of an attack is the number of succeeded attacks over the number of experiments. The attack succeeds when all the key bits of entire exponent are found. As a corollary, if only one bit is badly guessed, then the attack is considered to have failed.

For different exponents of 512-bit length, we estimate the success rate of the attack for the modulo (`RSA-1024-q` defined in [11, Section 2.2]), for different probabilities p_{noise} and different values of u depending on the number of side-channel traces $\mathcal{N}$. Figure 3 shows a comparison between the attack described in [10,11] and our method for u between 2 and 5. Here one can observe that our method for different u values increases significantly the success rate compared to the state-of-the-art method described in [10,11]. The number of side-channel traces needed to succeed the attack is divided by a factor greater than 2. More precisely, our new method recovers the key with probability 80% using only ≈40% of the traces needed in [10,11]. This advantage does not depend on the size of the modulus p.

The gain obtained by the increasing u values is not significant.

4.3 The attack in the presence of several blinding techniques

We already know that base blinding (a.k.a. message blinding) does not prevent our attack. The reason is that our attack neither requires the knowledge of any register values R_0 and R_1 nor it needs chosen input values. In this section, we analyze the situation when in addition to base blinding either modulus blinding or exponent blinding is applied.

4.3.1 The combination of basis blinding with modulus blinding

In the first step, an odd modulus blinding factor $r \in \mathbb{Z}_{R_F;\text{odd}} \coloneqq \{z \in \mathbb{Z}_{R_F} | z \text{ is odd}\}$ is selected randomly, where $R_F = 2^F$ for a suitable exponent F, e.g., for $F = 64$. The modular exponentiation is calculated modulo (pr) (instead of modulo p), and the new Montgomery constant is the product $R^* \coloneqq RR_F$ in place of R. The input value (base) y is reduced modulo (pr), yielding y_{pr}, and then the product $m \equiv y_{pr} r_B \pmod{pr}$ is computed for some random value $r_B \in \mathbb{Z}_p$ (base blinding). The result of the modular exponentiation, $m^k \pmod{(pr)}$, is reduced modulo p, which yields $m^k \pmod{p}$. Finally, the effect of the base blinding is annihilated by the multiplication with $r_B^{-k} \pmod{p}$, providing the desired output $y^k \pmod{p} = (y \pmod{p})^k \pmod{p}$.

Remark 4.9.

(i) The modulus blinding factor r needs to be odd because Montgomery's multiplication algorithm requires that the modulus is coprime to R^*.

(ii) Of course, the annihilating term $r_B^{-k} \pmod{p}$ is not computed straightforward. First of all, this would be extremely inefficient, and further, k is a sensitive variable. Hence, it is better not to touch it more than necessary in computations. Hence, we recommended already to resort to a similar albeit less harmful strategy (cf. [13, §10]). If e denotes the public RSA exponent, then $ek \equiv 1 \bmod \varphi(p)$, and thus for $r_B \coloneqq (r'_B)^{-e} \pmod{p}$ (with randomly selected r'_B) we have $r_B^{-k} \equiv (r'_B)^{((-e)(-k))} \equiv r'_B \pmod{p}$. Such blinding, applied to Montgomery ladder regular exponentiation using MM (i.e., Algorithm 2), is illustrated in Algorithm 4. (The affectation "$\leftarrow_R$" stands for uniformly random assignment.) Moreover, once a pair (r_B, r_B^{-k}) has been found it can easily be updated by squaring both components modulo p [13, §10].

(iii) In this paper, we consider the case "first modulus blinding then base blinding." This countermeasure is represented in Algorithm 5. We point out that reversed order, "first base blinding then modulus blinding," can be attacked in the same way.

Algorithm 4. Left-to-right Montgomery ladder exponentiation built on top of MM algorithm, with base blinding (attacked in this paper, in Section 4.1)

Input: m, $k = (k_{l-1}k_{l-2}\dots k_0)_2$, p ($k_{l-1} = 1$ and $k_0 = 1$)
Output: $m^k \bmod p$
1 $r'_B \leftarrow_{\mathcal{R}} \{1, 2, \dots, p-1\}$ // Nonzero base blinding factor
2 $R_0 \leftarrow \mathrm{MM}(m \cdot r'_B, R^2; p)$ // Basis blinding
3 $R_1 \leftarrow \mathrm{MM}(R_0, R_0; p)$ // First Square
4 **for** $i = l-2$ **down to** 0 **do**
5 $\quad R_{\neg k_i} \leftarrow \mathrm{MM}(R_0, R_1; p)$ // $i(M)$
6 $\quad R_{k_i} \leftarrow \mathrm{MM}(R_{k_i}, R_{k_i}; p)$ // $i(Q)$
7 **return** $\mathrm{MM}(R_0, r_B'^{-e} \bmod p; p)$ // where $e = k^{-1} \bmod \varphi(p)$

Algorithm 5. Left-to-right Montgomery ladder exponentiation built on top of MM algorithm, with base and modulus blinding (attacked in this paper, in Section 4.3). (Throughout this algorithm, the MM algorithm uses $R^* := RR_F$ as the Montgomery constant.)

Input: m, $k = (k_{l-1}k_{l-2}\dots k_0)_2$, p ($k_{l-1} = 1$ and $k_0 = 1$)
Output: $m^k \bmod p$
1 $r \leftarrow_{\mathcal{R}} \{1, 3 \dots, R_F - 1\}$ // Odd modulus blinding factor
2 $r'_B \leftarrow_{\mathcal{R}} \{1, 2, \dots, p-1\}$ // Nonzero base blinding factor
3 $R_0 \leftarrow \mathrm{MM}(m \cdot r'_B, (RR_F)^2; p \cdot r)$ // Basis & modulus blinding
4 $R_1 \leftarrow \mathrm{MM}(R_0, R_0; p \cdot r)$ // First Square
5 **for** $i = l-2$ **down to** 0 **do**
6 $\quad R_{\neg k_i} \leftarrow \mathrm{MM}(R_0, R_1; p \cdot r)$ // $i(M)$
7 $\quad R_{k_i} \leftarrow \mathrm{MM}(R_{k_i}, R_{k_i}; p \cdot r)$ // $i(Q)$
8 **return** $\mathrm{MM}(R_0, r_B'^{-e} \bmod p; p \cdot r) \bmod p$ // where $e = k^{-1} \bmod \varphi(p)$

For the case that only base blinding (or even no blinding technique at all) is applied, Lemma 3.4 provides concrete formulae that the extra-reduction vector $(w_{i(M)}, w_{i(Q)}, \dots, w_{i-u+1(M)}, w_{i-u+1(Q)})$ occurs if the relevant part of the secret exponent, $(k_i, \dots, k_{i-u+1})$, equals $\overrightarrow{\theta}$. These probabilities are polynomials in the ratio $\beta := p/R$. So far, the parameter β remained constant during the attack so that there was no need to mention it explicitly.

In this subsection, the ratio between the modulus and the Montgomery constant is no longer constant but depends on the selected modulus blinding value r. Hence, we extend the notation and write $\mathbb{P}^{\rightarrow}_{\overrightarrow{\theta};\beta^*}(W_{i(M)} = w_{i(M)}, \dots, W_{i-u+1(Q)} = w_{i-u+1(Q)})$ in place of $\mathbb{P}^{\rightarrow}_{\overrightarrow{\theta}}(W_{i(M)} = w_{i(M)}, \dots, W_{i-u+1(Q)} = w_{i-u+1(Q)})$ if $\beta^* := pr/R^*$. For given modulus blinding factor r one has $\beta^* = \alpha\beta$ with $\alpha := r/R_F$ and $\beta := p/R$.

However, the applied modulus blinding factor r is unknown. Relevant to our formulae is the normalized modulus blinding factor $\alpha := r/R_F$. We interpret α as a realization of a random variable A, which assumes values in the finite set $M_A := Z_{R_F;\mathrm{odd}}/R_F \subseteq [0, 1)$. Then

$$\mathbb{P}^{\rightarrow}_{\overrightarrow{\theta}}(W_{i(M)} = w_{i(M)}, \dots, W_{i-u+1(Q)} = w_{i-u+1(Q)}) = \sum_{\alpha \in M_A} \mathbb{P}(A = \alpha)\mathbb{P}^{\rightarrow}_{\overrightarrow{\theta};\alpha\beta}(W_{i(M)} = w_{i(M)}, \dots, W_{i-u+1(Q)} = w_{i-u+1(Q)}) \tag{4.13}$$

quantifies the probability for the extra-reduction vector $(w_{i(M)}, \dots, w_{i-u+1(Q)})$ under $\overrightarrow{\theta}$ with a randomly selected (normalized) modulus blinding factor α (selected according to the distribution of the random variable A). The probabilities $\mathbb{P}^{\rightarrow}_{\overrightarrow{\theta};\alpha\beta}(\cdot)$ are given by Lemma 3.4(i), and Assertion (ii) of Lemma 3.4 remains valid, too.

Usually the normalized blinding factors should be uniformly distributed on M_A, i.e., each value in M_A should occur with probability $1/M_A = 2/R_F$. For typical parameters F (e.g., for $F = 64$), the right-hand side of (4.13) can be replaced by

$$\mathbb{P}_{\vec{\theta}}(W_{i(M)} = w_{i(M)}, \ldots, W_{i-u+1(Q)} = w_{i-u+1(Q)}) = \int_0^1 \mathbb{P}_{\vec{\theta};\alpha\beta}(W_{i(M)} = w_{i(M)}, \ldots, W_{i-u+1(Q)} = w_{i-u+1(Q)})\,\mathrm{d}\alpha. \quad (4.14)$$

For reasonable parameter F, the deviation of the right-hand term from the exact probability (4.13) is negligible, which should justify the "=" sign. The evaluation of the integral is fairly easy since the integrand is a polynomial in $(\alpha\beta)$. In fact, for the integrand $\sum_j \gamma_j(\alpha\beta)^j$ the integral equals $\sum_j \gamma_j\beta^j/(j+1)$. Another protection strategy would be to select modulus blinding factors uniformly in $Z_{R_F;\mathrm{odd}} \cap [2^{F-1}, 2^F)$ so that all blinding factors have identical (maximal) length. In this case, A assumes each value in $M_A \cap [0.5, 1)$ with probability $4/R_F$, and (4.13) can be expressed by

$$\mathbb{P}_{\vec{\theta}}(W_{i(M)} = w_{i(M)}, \ldots, W_{i-u+1(Q)} = w_{i-u+1(Q)}) = 2\int_{0.5}^1 \mathbb{P}_{\vec{\theta};\alpha\beta}(W_{i(M)} = w_{i(M)}, \ldots, W_{i-u+1(Q)} = w_{i-u+1(Q)})\,\mathrm{d}\alpha. \quad (4.15)$$

In analogy to Section 4, the next step is to estimate β and p_{noise}. The equivalents to (4.1) and (4.2) are

$$E(W_{i(M)}) = \mathbb{P}(W_{i(M)} = 1) = \sum_{\alpha\in M_A} \mathbb{P}(A = \alpha)\frac{1}{4}\alpha\beta = E(A)\frac{\beta}{4}, \quad (4.16)$$

$$E(W_{i(Q)}) = \mathbb{P}(W_{i(Q)} = 1) = \sum_{\alpha\in M_A} \mathbb{P}(A = \alpha)\frac{1}{3}\alpha\beta = E(A)\frac{\beta}{3}. \quad (4.17)$$

Substituting $\mathbb{P}(W_{i(M)} = 1)$ and $\mathbb{P}(W_{i(Q)} = 1)$ in the proof of Lemma 4.2 by the right-hand terms of (4.16) and of (4.17) (in place of (4.1) and (4.2)) yields equivalents to the formulae (4.5) and (4.6) for the modulus blinding case. Note that the conditional probabilities $\mathbb{P}(\widetilde{W}_{i(Q)} = i|W_{i(Q)} = j)$ and $\mathbb{P}(\widetilde{W}_{i(M)} = i|W_{i(M)} = j)$ depend only on p_{noise} but not on α or β.

More precisely, a careful computation yields

$$\frac{p}{R} = \frac{1}{E(A)} \cdot \frac{12E(\widetilde{W}_{i(Q)}) - 12E(\widetilde{W}_{i(M)})}{1 + 6E(\widetilde{W}_{i(Q)}) - 8E(\widetilde{W}_{i(M)})}, \quad (4.18)$$

$$p_{\mathrm{noise}} = 4E(\widetilde{W}_{i(M)}) - 3E(\widetilde{W}_{i(Q)}). \quad (4.19)$$

The right-hand side of (4.18) differs from (4.5) by the factor $1/E(A)$, while (4.19) coincides with (4.6)

Above we have identified two strategies for the selection of modulus blinding factors, which are of particular interest. If A is uniformly distributed on M_A, then $E(A) = \int_0^1 \alpha\,\mathrm{d}\alpha = 0.5$. Similarly, if A is uniformly distributed on $M_A \cap [0.5, 1)$, then $E(A) = 2\int_{0.5}^1 \alpha\,\mathrm{d}\alpha = 0.75$.

Substituting (4.13) (resp., (4.14) or (4.15)) into Lemma 4.3(i) yields analogous assertions for the modulus blinding case. The estimation of $\widetilde{\mu}_M$ and $\widetilde{\mu}_S$ is done as in Lemma 4.4. For different power traces, the blinding factors are selected independently according to the same distribution so that the normalized blinding factors $\alpha_1, \alpha_2, \ldots$ for the power traces $1, 2, \ldots$ may be interpreted as realizations of iid random variables $A_1, A_2, \ldots$, where A_j is distributed as A. With the aforementioned considerations and Lemma 4.6 also applies to the modulus blinding scenario when $\mathbb{P}_{\vec{\theta}}(\widetilde{W}_{i(M)} = \widetilde{w}_{i(M)}, \ldots, \widetilde{W}_{i-u+1(Q)} = \widetilde{w}_{i-u+1(Q)})$ is calculated as in (4.7), combined with (4.13). Usually, the latter should coincide with (4.14) or (4.15).

Altogether, modulus blinding does not prevent our attack. For power trace j it yet reduces its efficiency since $(pr_j)/(RR_F) = \alpha_j\beta < \beta$, which lowers the probability for extra-reductions. Moreover, the applied blinding factor r_j is unknown, which results in averaged probabilities (4.13). Both can be compensated by increasing the sample size.

Remark 4.10. Alternatively to the attack just analyzed one might estimate the product $\alpha_j\beta$ separately for all power traces with formula (4.5), whereas (as above) p_{noise} is estimated only once at the beginning of the attack on the basis of all $\mathcal{N}$ power traces. The intention is to reduce the loss of efficiency caused by the use of averaged probabilities (4.13). Lemma 4.6 then could be applied as in Sections 3.2–4.1 with individual parameters $\alpha_j\beta$ for each power trace. On the negative side, the estimates of the products $\alpha_j\beta$ are less precise than the estimate of β in the scenario without modulus blinding since $\widetilde{\mu}_M$ and $\widetilde{\mu}_S$ depend only on the MM of single power traces, which undermines the intention of this attack variant.

4.3.2 Experimental results with modulus blinding

Figure 4 compares the success rate evolution of our attack, using (4.14), for the same three noise levels as in Figure 3, for $F \in \{8, 16, 32, 64\}$ with modulus randomization uniformly distributed in interval $[0, 2^F)$.

It can be seen that the value of F does not really impact on the success rate of the attack, which is in line with (4.14) and (4.15). It is corroborated by the fact that the attack success rate in the case of a modulus randomization factor uniformly distributed in $[2^{F-1}, 2^F)$ does not change significantly, by adapting the attack with (4.15). These success rates are shown in Figure 5. Note that the ratio p/R is the same in the results from Figures 4 and 5, because the modulus p (on 512 bits) is the same and the Montgomery constant is also the same, namely, $R^* = RR_F = 2^{512+F}$.

The success rate for some modulus randomization factors $F < 8$ could be derived from the exact formula (4.13). However, one shall take care that such small blinding factors should be of no practical relevance. For instance,

- when $F = 2$, there exists only two eligible random numbers, namely 1 and 3;
- when $F = 3$, the only four eligible random numbers are $\{1, 3, 5, 7\}$;
- when $F = 4$, the only eight eligible random numbers are $\{1, 3, 5, 7, 9, 11, 13, 15\}$.

If furthermore we demand that the blinding factors have full bit length (which corresponds to (4.15)) the situation is even worse. The sets then reduce to $\{3\}$, $\{5, 7\}$, and $\{9, 11, 13, 15\}$, respectively. However, such little sets of admissible modular blinding factors might allow other, even stronger attacks. Interestingly, the attacks work about with the same success rate as the original attacks [10,11] before our improvement in the absence of modulus blinding.

4.3.3 The combination of basis blinding with exponent blinding

Assume that base blinding is combined with (additive) exponent blinding, which means that the exponent k is replaced by $k + r_E\varphi(p)$ for some randomly selected exponent blinding factor $r_E \in \mathbb{Z}_{2^E}$. Our attack cannot be transferred to this situation since (4.11) assumes that $\overrightarrow{\theta}$ is the same for all $\mathcal{N}$ power traces.

It should be noted, however, that if (e.g.) single-trace template attacks provide significant advantage over blind guessing of the exponent bits k_i a successful attack may be possible anyway; see [27,28], for example, for details. The techniques developed in [27] obviously apply to the Montgomery ladder as well. The knowledge of the extra-reductions alone does not yet give sufficient advantage over blind guessing for single power traces. Sufficient advantage might be achieved by exploiting further features of the power traces but this is not within the scope of this paper.

5 Countermeasures

In Table 1 and in Section 4.3, several countermeasures were addressed and analyzed. In particular, even the combination of base blinding and exponent blinding does not prevent our attack. An option is to add

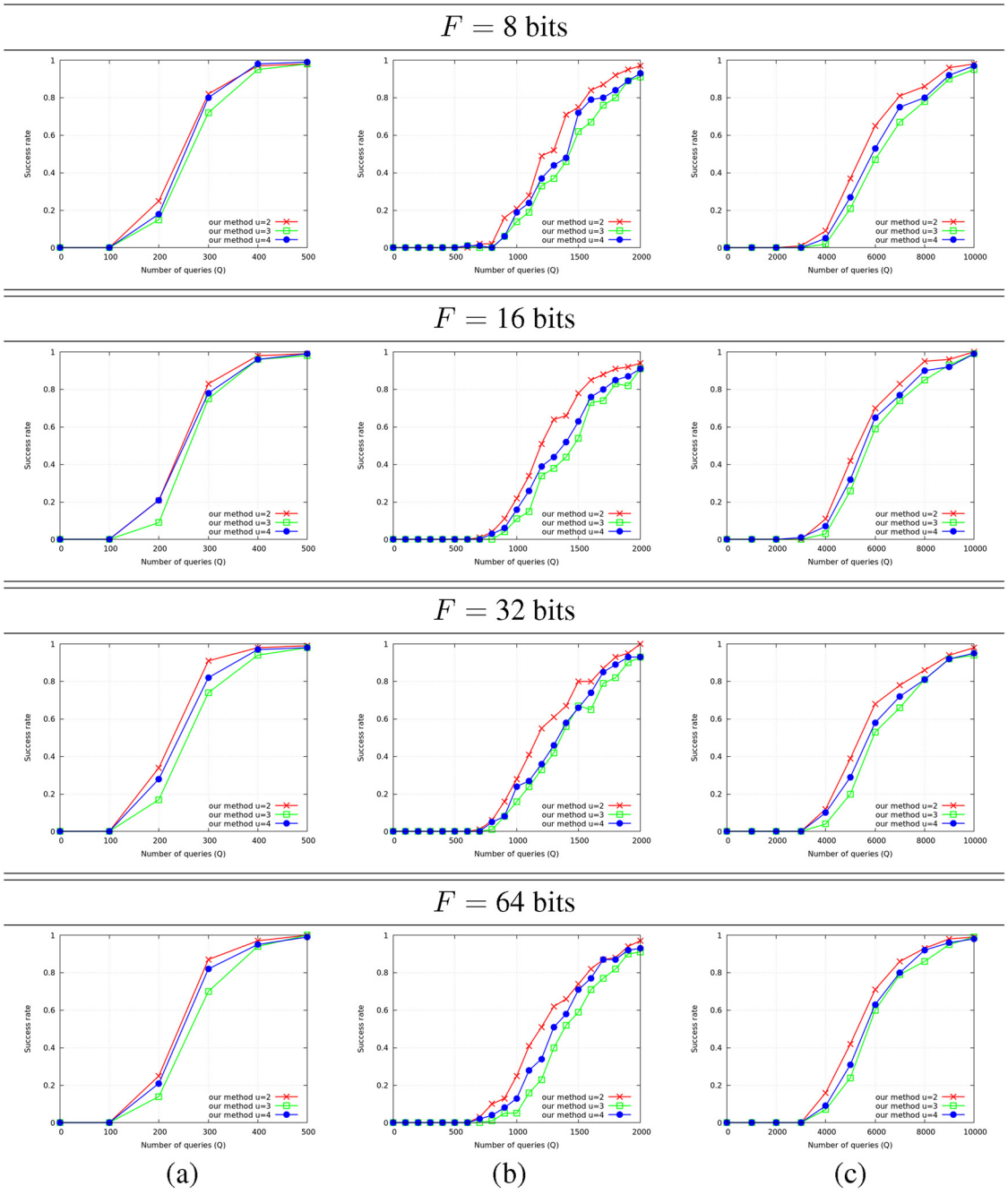

Figure 4: Success rate for an entire exponent depending on the number of side-channel trace $\mathcal{N}$ for different values of probability p_{noise} and for modulus randomization on F bits, for $F \in \{8, 16, 32, 64\}$ and modulus randomization uniform in $[0, 2^F)$. (a) $p_{\text{noise}} = 0.00$, (b) $p_{\text{noise}} = 0.10$, (c) $p_{\text{noise}} = 0.20$.

exponent blinding, resulting in the combination (base blinding and exponent blinding) or in (base blinding, modulus blinding, and exponent blinding). In the absence of additional leakage, to our best knowledge no attack is known (Section 4.3).

The most solid solution, of course, is to avoid extra-reductions at all. Following an idea of C. Walter one can completely resign on extra-reductions if the Montgomery constant R is not only larger than p but if $R > 4p$ [29], Theorems 3 and 6. In this case, the intermediate values of the Montgomery operations within the exponentiation algorithm are always between $[0, 2p)$ but they do not "explode." Currently, OpenSSL library uses another strategy. Indeed, most security standards prescribe that p be chosen with a size which is a multiple of the machine word size (typically 1024, 2048, 3072, and 4096 bits, which are all multiple of 32 and even 64 bits). Therefore, the abovementioned strategy of C. Walter requires that an extra limb (machine

word encoding on radix in the representation of a big number) shall be allocated for each intermediate variable, which is considered too high an overhead. For this reason, OpenSSL disguises the extra-reduction in a constant time SLA, a technique mentioned already in Section 2.1. Namely, a mask m of size l bits ($l = \lceil \log_2 p \rceil$ is the size of the modulus) is computed to be equal to $(1\ldots 1)_2$ (i.e., `0xFF...FF` in hexadecimal) when an extra-reduction is required or to $(0\ldots 0)_2$ (i.e., `0x00...00` in hexadecimal) when no extra-reduction is needed. Subsequently, the quantity $m \wedge p$ (word obtained by bitwise logical AND of bits from m and p) is subtracted from the result of the MM. This quantity is either 0 or p, depending on whether an extra-reduction is needed or not. This strategy implements an SLA. Such coding style is, as of today, believed secure against cache-timing attacks, because the control flow is data independent. However, the authors

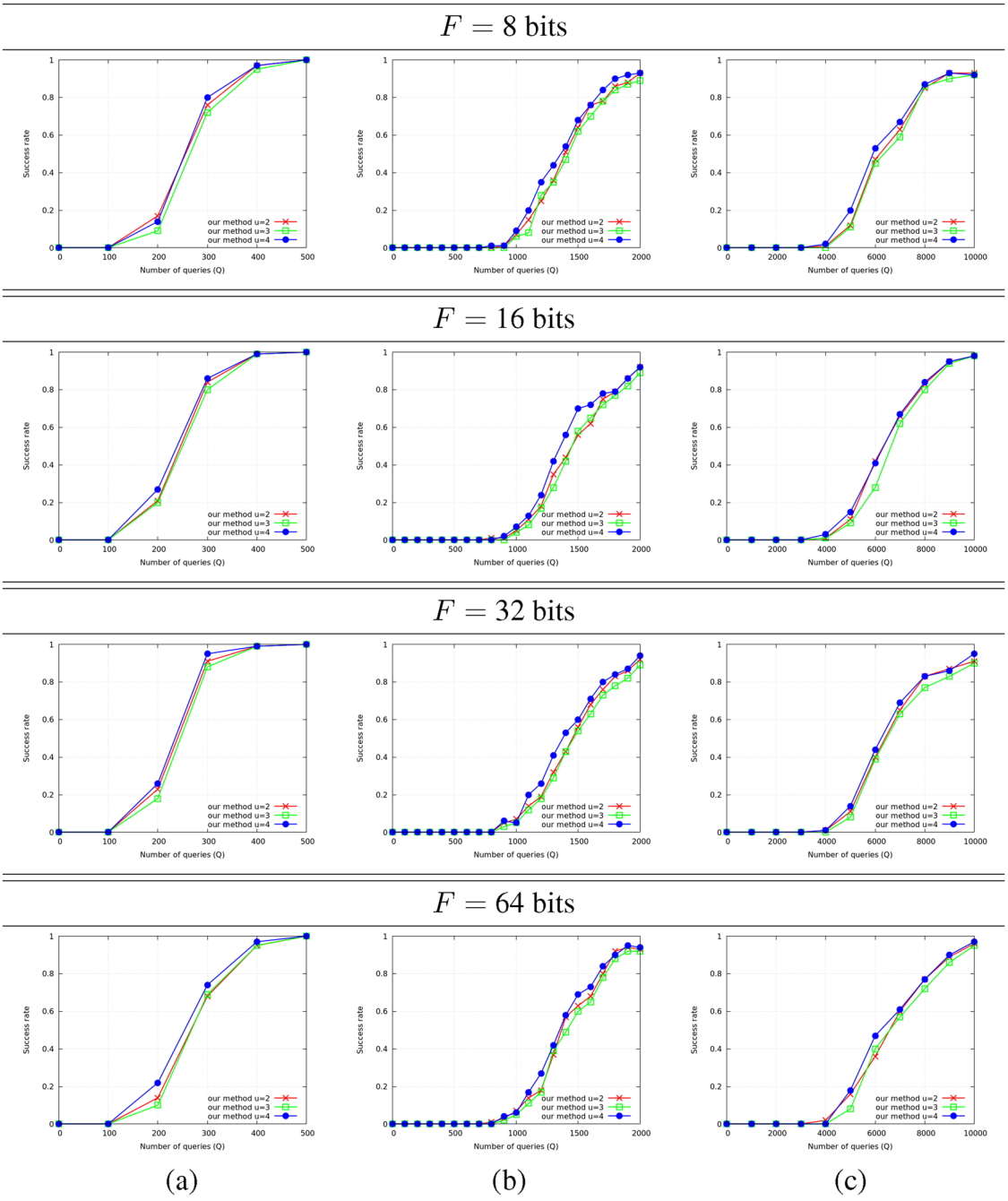

Figure 5: Success rate for an entire exponent depending of the number of side-channel trace $\mathcal{N}$ for different values of probability p_{noise} and for modulus randomization on F bits, for $F \in \{8, 16, 32, 64\}$ and modulus randomization uniform in $[2^{F-1}, 2^F)$: (a) $p_{\text{noise}} = 0.00$, (b) $p_{\text{noise}} = 0.10$, (c) $p_{\text{noise}} = 0.20$.

warn that the strategy of OpenSSL might not hide perfectly the extra-reduction if the attacker is able to partition power or electromagnetic side-channel traces based on the value of $m \wedge p$, since the absence of extra-reduction involves a remarkable subtraction with a big number equal to zero. Such bias has already been exploited in the past by attacks such as the Refined Power Analysis [12] or the Zero Power Analysis [4]. Note that OpenSSL is nowadays used in embedded systems (microcontrollers, internet of things devices, smartphones [5,18], etc.), which are indeed attackable with power and electromagnetic side-channel analyses.

6 Conclusion

In [10,11], ERA exploiting the dependency of two consecutive MMs was applied to attack RSA implementations, which use the Montgomery ladder or the always square and multiply exponentiation algorithm. Basis blinding does not prevent this attack. Although both attacks were successful they did not exploit all the available information. In this paper, we followed the strategy in [1,2,21], formulated, and analyzed a stochastic process, which was tailored to the stochastic behavior of the extra-reductions in Montgomery ladder. This sophisticated strategy allowed us to exploit all the given information in an optimal way. Practical experiments underlined that the new method reduces the sample size by a factor greater than 2 (to ≈40% of the original sample size). Our new attack can directly be transferred to the always square and multiply algorithm. Moreover, we presented a generalization of our attack, which cannot even be prevented by combination of base blinding with modulus blinding. This generalization of our attack is efficient, too.

Funding information: This work has benefited from a partial funding via TeamPlay (https://teamplay-h2020.eu/), a project from European Union's Horizon 2020 research and innovation program, under grant agreement no. 779882. The analysis methods have been integrated into Secure-IC Laboryzr tools https://www.secure-ic.com/solutions/laboryzr/.

Conflict of interest: Authors state no conflict of interest.

References

[1] Acıiçmez O, Schindler W. A major vulnerability in RSA implementations due to microarchitectural analysis threat. IACR Cryptology ePrint Archive 2007;2007:336.

[2] Acıiçmez O, Schindler W. A vulnerability in RSA implementations due to instruction cache analysis and its demonstration on OpenSSL. In: Malkin T, editor. Topics in Cryptology – CT-RSA 2008, Proceedings of the Cryptographers' Track at the RSA Conference 2008, San Francisco, CA, USA, April 8–11, 2008. Lecture Notes in Computer Science. vol. 4964, Berlin, Heidelberg: Springer; 1997. p. 256–73.

[3] Acıiçmez O, Schindler W, Koç ÇK. Improving Brumley and Boneh timing attack on unprotected SSL implementations. In: Atluri V, Meadows C, Juels A, editors. Proceedings of the 12th ACM Conference on Computer and Communications Security, CCS 2005, Alexandria, VA, USA, November 7–11, 2005. New York, NY, United States: Association for Computing Machinery; 2005. p. 139–46.

[4] Akishita T, Takagi T. Zero-value point attacks on elliptic curve cryptosystem. In: Boyd C, Mao W, editors. ISC. Lecture Notes in Computer Science. vol. 2851. Berlin, Heidelberg: Springer; 2003. p. 218–33.

[5] Alam M, Khan HA, Dey M, Sinha N, Callan RL, Zajic AG, et al. One&Done: A single-decryption EM-based attack on OpenSSL's constant-time blinded RSA. In: Enck W, Porter Felt A, editors. 27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, August 15–17, 2018. USENIX Association; 2018. p. 585–602.

[6] Arnaud C, Fouque PA. Timing attack against protected RSA-CRT implementation used in polarssl. In: Dawson E, editor. Topics in Cryptology - CT-RSA 2013. Proceedings of the Cryptographers' Track at the RSA Conference 2013, San Francisco, CA, USA, February 25-March 1, 2013. Lecture Notes in Computer Science. vol. 7779. Berlin, Heidelberg: Springer; 2013. p. 18–33.

[7] Baek YJ, Vasyltsov I. How to prevent DPA and fault attack in a unified way for ECC scalar multiplication - ring extension method. In: Dawson Ed, Wong SW, editors. Information Security Practice and Experience, Lecture Notes in Computer Science. vol. 4464. Berlin Heidelberg: Springer; 2007. p. 225–37.

[8] Brumley D, Boneh D. Remote timing attacks are practical. In: Proceedings of the 12th USENIX Security Symposium, Washington, D.C., USA, USENIX Association; August 4–8, 2003.
[9] Dhem J-F, Koeune F, Leroux P-A, Mestré P, Quisquater J, Willems J-L. A practical implementation of the timing attack. In: Quisquater J-J, Schneier B, editors. Smart Card Research and Applications, Proceedings of the International Conference, CARDIS '98, Louvain-la-Neuve, Belgium, September 14–16, 1998. Lecture Notes in Computer Science. vol. 1820. Berlin, Heidelberg: Springer; 1998, p. 167–82.
[10] Dugardin M, Guilley S, Danger J-L, Najm Z, Rioul O. Correlated extra-reductions defeat blinded regular exponentiation. In: Gierlichs B, Poschmann AY, editors. Cryptographic Hardware and Embedded Systems - Proceedings of the CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17–19. 2016, Lecture Notes in Computer Science. vol. 9813. Berlin, Heidelberg: Springer; 2016. p. 3–22.
[11] Dugardin M, Guilley S, Danger J-L, Najm Z, Rioul O. Correlated extra-reductions defeat blinded regular exponentiation - extended version. Cryptology ePrint Archive. Report 2016/597. June 6 2016. http://eprint.iacr.org/2016/597
[12] Goubin L. A refined power-analysis attack on elliptic curve cryptosystems. In: Proceedings of the 6th International Workshop on Theory and Practice in Public Key Cryptography: Public Key Cryptography, PKC '03. London, UK: Springer-Verlag; 2003. p. 199–210.
[13] Kocher PC. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: Koblitz N, editor. Advances in Cryptology - CRYPTO '96, Proceedings of the 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18–22, 1996. Lecture Notes in Computer Science. vol. 1109. Berlin, Heidelberg: Springer; 1996. p. 104–13.
[14] Kocher PC, Jaffe J, Jun B. Differential power analysis. In: Wiener MJ, editor. CRYPTO, Lecture Notes in Computer Science. vol. 1666. Berlin, Heidelberg: Springer; 1999. p. 388–97.
[15] Menezes AJ, van Oorschot PC, Vanstone SA. Handbook of Applied Cryptography. Boca Raton, USA: CRC Press; October 1996. http://www.cacr.math.uwaterloo.ca/hac/.
[16] Montgomery PL. Modular multiplication without trial division. Math Comput. April 1985;44(170):519–21.
[17] Montgomery PL. Modular multiplication without trial division. Math Comput. 1985;44(170): 519–21.
[18] Nakano Y, Souissi Y, Nguyen R, Sauvage L, Danger J-L, Guilley S, Kiyomoto S, et al. A pre-processing composition for secret key recovery on android smartphone. In: Naccache D, Sauveron D, editors. Information Security Theory and Practice. Proceedings of the Securing the Internet of Things - 8th IFIP WG 11.2 International Workshop, WISTP 2014, Heraklion, Crete, Greece, June 30–July 2, 2014. Lecture Notes in Computer Science. vol. 8501. Berlin, Heidelberg: Springer; 2014. p. 76–91.
[19] Perin G, Imbert L, Torres L, Maurine P. Attacking randomized exponentiations using unsupervised learning. In: Prouff E, editor. Constructive Side-Channel Analysis and Secure Design - 5th International Workshop, COSADE 2014, Paris, France, April 13–15, 2014. Revised Selected Papers. Lecture Notes in Computer Science. vol. 8622. Cham: Springer; 2014. p. 144–60.
[20] Schindler W. A timing attack against RSA with the Chinese remainder theorem. In: Koç ÇK, Paar C, editors. Cryptographic Hardware and Embedded Systems - Proceedings of the CHES 2000, Second International Workshop, Worcester, MA, USA, August 17–18, 2000. Lecture Notes in Computer Science. vol. 1965. Berlin, Heidelberg: Springer; 2000. p. 109–24.
[21] Schindler W. A combined timing and power attack. In: Naccache D, Paillier P, editors. Public key cryptography, Proceedings of the 5th International Workshop on Practice and Theory in Public Key Cryptosystems, PKC 2002, Paris, France, February 12–14, 2002. Lecture Notes in Computer Science. vol. 2274. Berlin, Heidelberg: Springer; 2002. p. 263–79.
[22] Schindler W. Optimized timing attacks against public key cryptosystems. Statistics and Decisions 2002;20(2):191–210.
[23] Schindler W. Exclusive exponent blinding may not suffice to prevent timing attacks on RSA. In: Güneysu T, Handschuh H, editors. Cryptographic Hardware and Embedded Systems - CHES 2015 - Proceedings of the 17th International Workshop, Saint-Malo, France, September 13–16, 2015. Lecture Notes in Computer Science. vol. 9293. Berlin, Heidelberg: Springer; 2015. p. 229–47.
[24] Schindler W. Exclusive exponent blinding is not enough to prevent any timing attack on RSA. J Cryptographic Eng. 2016;6(2):101–19.
[25] Schindler W, Koeune F, Quisquater J-J. Improving divide and conquer attacks against cryptosystems by better error detection/correction strategies. In: Honary B, editor. Cryptography and Coding, Proceedings of the 8th IMA International Conference, Cirencester, UK, December 17–19, 2001, Lecture Notes in Computer Science, vol. 2260. Berlin, Heidelberg: Springer; 2001. p. 245–67.
[26] Schindler W, Walter CD. More detail for a combined timing and power attack against implementations of RSA. In: Paterson KG, editor. Cryptography and Coding, Proceedings of the 9th IMA International Conference, Cirencester, UK, December 16–18, 2003. Lecture Notes in Computer Science. vol. 2898. Berlin, Heidelberg: Springer; 2003. p. 245–63.
[27] Schindler W, Wiemers A. Power attacks in the presence of exponent blinding. J. Cryptographic Eng. 2014;4(4):213–36.
[28] Schindler W, Wiemers A. Generic power attacks on RSA with CRT and exponent blinding: new results. J Cryptographic Eng. 2017;7(4):255–72.
[29] Walter CD. Precise bounds for montgomery modular multiplication and some potentially insecure RSA moduli. In: Preneel B, editor. Topics in Cryptology - CT-RSA 2002, Proceedings of the Cryptographer's Track at the RSA Conference, 2002, San Jose, CA, USA, February 18–22, 2002. Lecture Notes in Computer Science. vol. 2271. Berlin, Heidelberg: Springer; 2002. p. 30–9.
[30] Walter CD, Thompson S. Distinguishing exponent digits by observing modular subtractions. In: Naccache D, editor. Topics in Cryptology - CT-RSA 2001, Proceedings of the Cryptographer's Track at RSA Conference 2001, San Francisco, CA, USA, April 8–12, 2001. Lecture Notes in Computer Science. vol. 2020. Berlin, Heidelberg: Springer; 2001. p. 192–207.

Appendix A

Tables A.1 and A.2 contain all probabilities for $u = 2$ (with or without base blinding/no modulus blinding); cf. Lemma 3.4(i). These values have been computed with SageMath (http://www.sagemath.org/) computer algebra system, using formal computations and simplifications.

Table A.1: $\mathbb{P}_\theta(W_{i(M)} = w_{i(M)}, W_{i(Q)} = w_{i(Q)}, W_{i-1(M)} = w_{i-1(M)}, W_{i-1(Q)} = w_{i-1(Q)})$ for $\overrightarrow{\theta} \in \{(0,0),(1,1)\}$ (corresponds to the case $k_i = k_{i-1}$) (with or without base blinding/no modulus blinding)

$(w_{i(M)}, w_{i(Q)})$	$(w_{i-1(M)}, w_{i-1(Q)})$			
	(0,0)	**(0,1)**	**(1,0)**	**(1,1)**
(0,0)	$1 - \frac{7}{6}\frac{p}{R} + \frac{1}{3}\left(\frac{p}{R}\right)^2 + \frac{7}{90}\left(\frac{p}{R}\right)^3 + \frac{17}{504}\left(\frac{p}{R}\right)^4 - \frac{11}{336}\left(\frac{p}{R}\right)^5 - \frac{1}{72}\left(\frac{p}{R}\right)^6 + \frac{1}{264}\left(\frac{p}{R}\right)^8$	$\frac{1}{3}\frac{p}{R} - \frac{5}{24}\left(\frac{p}{R}\right)^2 - \frac{17}{504}\left(\frac{p}{R}\right)^4 + \frac{1}{48}\left(\frac{p}{R}\right)^5 + \frac{1}{72}\left(\frac{p}{R}\right)^6 - \frac{1}{264}\left(\frac{p}{R}\right)^8$	$\frac{1}{4}\frac{p}{R} - \frac{1}{8}\left(\frac{p}{R}\right)^2 - \frac{7}{90}\left(\frac{p}{R}\right)^3 + \frac{1}{72}\left(\frac{p}{R}\right)^4 + \frac{1}{84}\left(\frac{p}{R}\right)^5 + \frac{1}{72}\left(\frac{p}{R}\right)^6 - \frac{1}{264}\left(\frac{p}{R}\right)^8$	$\frac{1}{8}\left(\frac{p}{R}\right)^2 - \frac{1}{72}\left(\frac{p}{R}\right)^4 - \frac{1}{72}\left(\frac{p}{R}\right)^6 + \frac{1}{264}\left(\frac{p}{R}\right)^8$
(0,1)	$\frac{1}{3}\frac{p}{R} - \frac{1}{8}\left(\frac{p}{R}\right)^2 - \frac{1}{20}\left(\frac{p}{R}\right)^3 - \frac{1}{21}\left(\frac{p}{R}\right)^4 + \frac{11}{336}\left(\frac{p}{R}\right)^5 + \frac{1}{72}\left(\frac{p}{R}\right)^6 - \frac{1}{264}\left(\frac{p}{R}\right)^8$	$\frac{1}{21}\left(\frac{p}{R}\right)^4 - \frac{1}{48}\left(\frac{p}{R}\right)^5 - \frac{1}{72}\left(\frac{p}{R}\right)^6 + \frac{1}{264}\left(\frac{p}{R}\right)^8$	$\frac{1}{20}\left(\frac{p}{R}\right)^3 - \frac{1}{84}\left(\frac{p}{R}\right)^5 - \frac{1}{72}\left(\frac{p}{R}\right)^6 + \frac{1}{264}\left(\frac{p}{R}\right)^8$	$\frac{1}{72}\left(\frac{p}{R}\right)^6 - \frac{1}{264}\left(\frac{p}{R}\right)^8$
(1,0)	$\frac{1}{4}\frac{p}{R} - \frac{5}{24}\left(\frac{p}{R}\right)^2 - \frac{1}{36}\left(\frac{p}{R}\right)^3 + \frac{1}{72}\left(\frac{p}{R}\right)^4 + \frac{11}{336}\left(\frac{p}{R}\right)^5 - \frac{1}{264}\left(\frac{p}{R}\right)^8$	$\frac{1}{12}\left(\frac{p}{R}\right)^2 - \frac{1}{72}\left(\frac{p}{R}\right)^4 - \frac{1}{48}\left(\frac{p}{R}\right)^5 + \frac{1}{264}\left(\frac{p}{R}\right)^8$	$\frac{1}{36}\left(\frac{p}{R}\right)^3 - \frac{1}{72}\left(\frac{p}{R}\right)^4 - \frac{1}{84}\left(\frac{p}{R}\right)^5 + \frac{1}{264}\left(\frac{p}{R}\right)^8$	$\frac{1}{72}\left(\frac{p}{R}\right)^4 - \frac{1}{264}\left(\frac{p}{R}\right)^8$
(1,1)	$\frac{1}{8}\left(\frac{p}{R}\right)^2 - \frac{11}{336}\left(\frac{p}{R}\right)^5 + \frac{1}{264}\left(\frac{p}{R}\right)^8$	$\frac{1}{48}\left(\frac{p}{R}\right)^5 - \frac{1}{264}\left(\frac{p}{R}\right)^8$	$\frac{1}{84}\left(\frac{p}{R}\right)^5 - \frac{1}{264}\left(\frac{p}{R}\right)^8$	$\frac{1}{264}\left(\frac{p}{R}\right)^8$

Table A.2: $\mathbb{P}_\theta(W_{i(M)} = w_{i(M)}, W_{i(Q)} = w_{i(Q)}, W_{i-1(M)} = w_{i-1(M)}, W_{i-1(Q)} = w_{i-1(Q)})$ for $\overrightarrow{\theta} \in \{(0, 1), (1, 0)\}$ (corresponds to the case $k_i \neq k_{i-1}$) (with or without base blinding/no modulus blinding)

$(w_{i(M)}, w_{i(Q)})$	$(w_{i-1(M)}, w_{i-1(Q)})$			
	(0,0)	**(0,1)**	**(1,0)**	**(1,1)**
(0,0)	$1 - \frac{7}{6}\frac{p}{R} + \frac{13}{36}\left(\frac{p}{R}\right)^2 + \frac{7}{90}\left(\frac{p}{R}\right)^3 - \frac{1}{240}\left(\frac{p}{R}\right)^4 - \frac{13}{504}\left(\frac{p}{R}\right)^5 - \frac{1}{200}\left(\frac{p}{R}\right)^6 + \frac{1}{360}\left(\frac{p}{R}\right)^8$	$\frac{1}{3}\frac{p}{R} - \frac{17}{72}\left(\frac{p}{R}\right)^2 + \frac{1}{240}\left(\frac{p}{R}\right)^4 + \frac{1}{72}\left(\frac{p}{R}\right)^5 + \frac{1}{200}\left(\frac{p}{R}\right)^6 - \frac{1}{360}\left(\frac{p}{R}\right)^8$	$\frac{1}{4}\frac{p}{R} - \frac{1}{8}\left(\frac{p}{R}\right)^2 - \frac{7}{90}\left(\frac{p}{R}\right)^3 + \frac{1}{40}\left(\frac{p}{R}\right)^4 + \frac{1}{84}\left(\frac{p}{R}\right)^5 + \frac{1}{200}\left(\frac{p}{R}\right)^6 - \frac{1}{360}\left(\frac{p}{R}\right)^8$	$\frac{1}{8}\left(\frac{p}{R}\right)^2 - \frac{1}{40}\left(\frac{p}{R}\right)^4 - \frac{1}{200}\left(\frac{p}{R}\right)^6 + \frac{1}{360}\left(\frac{p}{R}\right)^8$
(0,1)	$\frac{1}{3}\frac{p}{R} - \frac{17}{72}\left(\frac{p}{R}\right)^2 - \frac{1}{20}\left(\frac{p}{R}\right)^3 + \frac{1}{40}\left(\frac{p}{R}\right)^4 + \frac{13}{504}\left(\frac{p}{R}\right)^5 - \frac{1}{360}\left(\frac{p}{R}\right)^8$	$\frac{1}{9}\left(\frac{p}{R}\right)^2 - \frac{1}{40}\left(\frac{p}{R}\right)^4 - \frac{1}{72}\left(\frac{p}{R}\right)^5 + \frac{1}{360}\left(\frac{p}{R}\right)^8$	$\frac{1}{20}\left(\frac{p}{R}\right)^3 - \frac{1}{40}\left(\frac{p}{R}\right)^4 - \frac{1}{84}\left(\frac{p}{R}\right)^5 + \frac{1}{360}\left(\frac{p}{R}\right)^8$	$\frac{1}{40}\left(\frac{p}{R}\right)^4 - \frac{1}{360}\left(\frac{p}{R}\right)^8$
(1,0)	$\frac{1}{4}\frac{p}{R} - \frac{1}{8}\left(\frac{p}{R}\right)^2 - \frac{1}{36}\left(\frac{p}{R}\right)^3 - \frac{1}{48}\left(\frac{p}{R}\right)^4 + \frac{13}{504}\left(\frac{p}{R}\right)^5 + \frac{1}{200}\left(\frac{p}{R}\right)^6 - \frac{1}{360}\left(\frac{p}{R}\right)^8$	$\frac{1}{48}\left(\frac{p}{R}\right)^4 - \frac{1}{72}\left(\frac{p}{R}\right)^5 - \frac{1}{200}\left(\frac{p}{R}\right)^6 - \frac{1}{360}\left(\frac{p}{R}\right)^8$	$\frac{1}{36}\left(\frac{p}{R}\right)^3 - \frac{1}{84}\left(\frac{p}{R}\right)^5 - \frac{1}{200}\left(\frac{p}{R}\right)^6 - \frac{1}{360}\left(\frac{p}{R}\right)^8$	$\frac{1}{200}\left(\frac{p}{R}\right)^6 - \frac{1}{360}\left(\frac{p}{R}\right)^8$
(1,1)	$\frac{1}{8}\left(\frac{p}{R}\right)^2 - \frac{13}{504}\left(\frac{p}{R}\right)^5 + \frac{1}{360}\left(\frac{p}{R}\right)^8$	$\frac{1}{72}\left(\frac{p}{R}\right)^5 - \frac{1}{360}\left(\frac{p}{R}\right)^8$	$\frac{1}{84}\left(\frac{p}{R}\right)^5 - \frac{1}{360}\left(\frac{p}{R}\right)^8$	$\frac{1}{360}\left(\frac{p}{R}\right)^8$