# Approximate Divisor Multiples – Factoring with Only a Third of the Secret CRT-Exponents

Alexander May[1] [*], Julian Nowakowski[1], and Santanu Sarkar[2] [**]

[1] Ruhr-University Bochum, Bochum, Germany
{alex.may,julian.nowakowski}@rub.de
[2] Indian Institute of Technology Madras, Chennai, India
sarkar.santanu.bir@gmail.com

**Abstract.** We address Partial Key Exposure attacks on CRT-RSA on secret exponents $d_p, d_q$ with small public exponent $e$. For constant $e$ it is known that the knowledge of half of the bits of one of $d_p, d_q$ suffices to factor the RSA modulus $N$ by Coppersmith's famous *factoring with a hint* result. We extend this setting to non-constant $e$. Somewhat surprisingly, our attack shows that RSA with $e$ of size $N^{\frac{1}{12}}$ is most vulnerable to Partial Key Exposure, since in this case only a third of the bits of both $d_p, d_q$ suffices to factor $N$ in polynomial time, knowing either most significant bits (MSB) or least significant bits (LSB).

Let $ed_p = 1 + k(p - 1)$ and $ed_q = 1 + \ell(q - 1)$. On the technical side, we find the factorization of $N$ in a novel two-step approach. In a first step we recover $k$ and $\ell$ in polynomial time, in the MSB case completely elementary and in the LSB case using Coppersmith's lattice-based method. We then obtain the prime factorization of $N$ by computing the root of a univariate polynomial modulo $kp$ for our known $k$. This can be seen as an extension of Howgrave-Graham's *approximate divisor* algorithm to the case of *approximate divisor multiples* for some known multiple $k$ of an unknown divisor $p$ of $N$. The point of *approximate divisor multiples* is that the unknown that is recoverable in polynomial time grows linearly with the size of the multiple $k$.

Our resulting Partial Key Exposure attack with known MSBs is completely rigorous, whereas in the LSB case we rely on a standard Coppersmith-type heuristic. We experimentally verify our heuristic, thereby showing that in practice we reach our asymptotic bounds already using small lattice dimensions. Thus, our attack is highly efficient.

**Keywords:** Coppersmith's method, CRT-RSA, Partial Key Exposure

## 1 Introduction

*RSA.* As opposed to other cryptosystems, RSA has the disadvantage that it suffers from *Partial Key Exposure* (PKE) attacks. Given only a constant fraction

[*] Funded by DFG under Germany's Excellence Strategy - EXC-2092 CASA - 390781972.

[**] Funded by a Humboldt Research Fellowship for experienced researchers, while visiting Ruhr-University Bochum.

of the secret key, in many settings RSA can be broken in polynomial time. Coppersmith's *factoring with a hint* [6] that factors RSA moduli $N = pq$ in polynomial time given only half of the bits of $p$ can be considered the pioneer work in the area, from which many other results where derived [5,2,11,13]. A direct application of *factoring with a hint* is the Boneh-Durfee-Frankel (BDF) attack [5], that factors $N$ given only a quarter of the least significant bits (LSB) of the RSA secret exponent, provided that $e$ is constant. For larger values of $e$ and known most significant bits (MSB) of $d$, the BDF attack requires the knowledge of a larger fraction of $d$.

In the case of full-size $e$, denoted $e \approx N$, and small $d$, it was shown by Ernst, May, Jochemsz, de Weger [9], Aono [1] and Takayasu and Kunihiro [20] that there exist Partial Key Exposure attacks that require no bits for $d \leq N^{0.284}$ respectively $d \leq N^{0.292}$, coinciding with the results of Boneh and Durfee [4], and work up to full-size $d$, coinciding with the result of Coron and May [15,8].

*CRT-RSA.* In practice, basically all RSA implementations use CRT secret exponents $d_p = d \bmod p-1, d_q = d \bmod q-1$ in combination with a small public exponent $e$. The first PKE for CRT-RSA was shown by Blömer and May [2] – also derived from *factoring with a hint* – that factors $N$ given half of either LSBs or MSBs of one of $d_p, d_q$, provided that $e$ is constant.

Sarkar and Maitra [18] and later Takayasu-Kunihiro [19] showed that there exist Partial Key Exposure attacks for all $e$ up to full-size, where naturally the larger $e$ the more LSBs/MSBs of $d_p, d_q$ one has to know, see Figure 1.

In the small CRT-exponent setting, May, Nowakowski and Sarkar [17] recently showed that there exist PKE for known LSB that require no knowledge of bits for $d_p, d_q \leq N^{0.122}$, coinciding with Takayasu, Lu, Peng [21], and work up to full-size CRT-exponents.

*Discussion of Takayasu-Kunihiro (TK) [19].* From Figure 1, one observes that the TK attack converges for small $e$ to only a known $\frac{1}{3}$-fraction of the bits of $d_p, d_q$. We strongly question this result of the TK attack in the small $e$ regime. First, we show that knowledge of a known $\frac{1}{3}$-fraction of $p$ (LSBs/MSBs) implies knowledge of a $\frac{1}{3}$-fraction of $d_p, d_q$ (for constant $e$), which in turn by the TK-result would imply polynomial time factoring (see Theorem 4). Thus, if TK works in the small $e$ setting, then this result immediately implies that we can factor with only $\frac{1}{3}$ of the bits of $p$, a major improvement over Coppersmith's famous *factoring with hint.* Second, we give also strong experimental evidence that TK fails to recover the factorization in the small $e$ regime.

Notice that we do not question the TK analysis in general. The point is that the TK attack uses the standard Coppersmith-type heuristic (see Assumption 1) for extracting roots of multivariate polynomials. Takayasu and Kunihiro did not provide experimental verification of this heuristic in [19]. Our experimental results in Section 5 show that the heuristic systematically fails when less than half of the bits of $d_p, d_q$ are known. We conjecture that the TK attack works for $e \leq N^{\frac{1}{8}}$ (asymptotically, for sufficiently large lattice dimension), if at least half of the bits of $d_p, d_q$ are known, see also Figure 1. We also believe that the

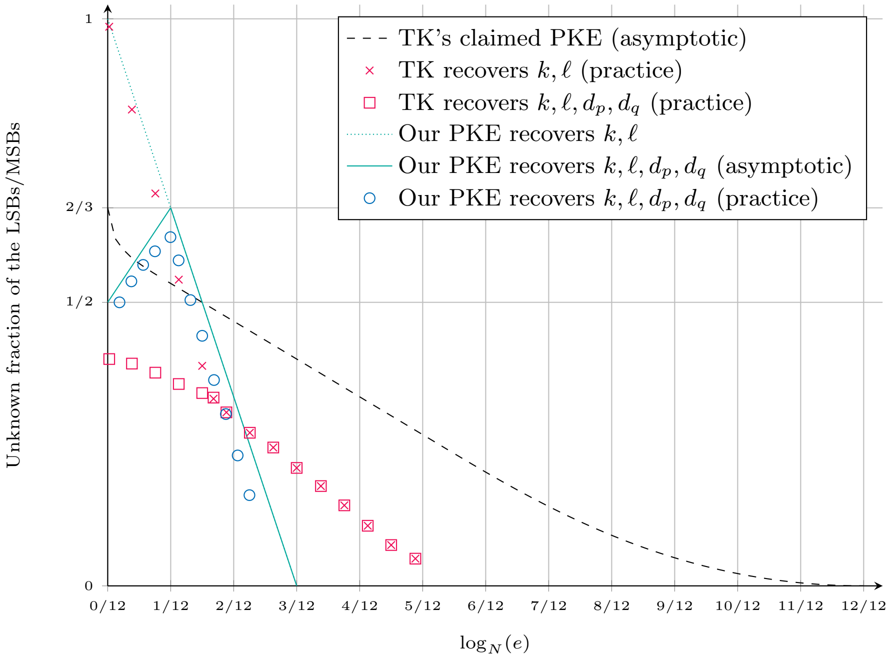

**Fig. 1.** Comparison between the Takayasu-Kunihiro attack and our attack.

asymptotic TK attack area is correct for $e \geq N^{\frac{1}{8}}$, thereby coinciding with our new attack at the point $e = N^{\frac{1}{8}}$.

*Our contributions.* We take as starting point the two CRT-exponent equations

$$\begin{aligned} ed_p &= 1 + k(p-1), \\ ed_q &= 1 + \ell(q-1). \end{aligned} \tag{1}$$

We then introduce a novel two-step procedure to factor $N$. In a first step, for $e \leq N^{\frac{1}{4}}$ we efficiently recover $k, \ell$ given sufficiently many LSBs/MSBs of $d_p, d_q$. For constant-size $e$ we do not need any bits of $d_p, d_q$, whereas for $e = N^{\frac{1}{4}}$ we need all bits. We would like to stress that as opposed to the results of Blmer, May [2] our algorithm really requires the knowledge of LSBs/MSBs of *both* $d_p, d_q$. It is open how to recover $k$ efficiently given only bits of $d_p$.

Our algorithm for recovering $k$ and $\ell$ in the case of MSBs of $d_p, d_q$ is completely elementary and rigorous, whereas we recover $k, \ell$ in the LSB case using Coppersmith's method under the usual heuristic for extracting roots for multivariate modular polynomial equations. We verify our heuristic experimentally in Section 5.

Upon recovering $k, \ell$, we use in a second step only $k$ and Equation (1) to construct an approximation of $kp$ (either LSBs or MSBs). We then show that such an *approximate divisor multiple* for some *known* multiple $k$ allows reconstruction of $kp$ with a smaller portion of known bits than in the standard *approximate divisor* case with $k = 1$ aka *factoring with a hint* [6,11]. We would like to stress that the *approximate divisor multiple* setting was already addressed in [3, Theorem 13]. However, our new result (Theorem 3) improves over [3, Theorem 13], and to the best of our knowledge we are the first to give an application of this setting.

Let us illustrate our new result for *approximate divisor multiples* with a small numerical example. Assume that $N = pq$ with $p, q$ of equal bit-size, i.e. $p$ is roughly of size $N^{\frac{1}{2}}$. If we know an approximation of $p$, then *factoring with a hint* tells us that we can recover the unknown remaining part of $p$, as long as it is bounded by $N^{\frac{1}{4}}$. This implies that we have to know half of the bits of $p$, and the remaining half can be efficiently recovered.

Now in the *approximate divisor multiple* setting assume that we know $k \approx N^{\frac{1}{4}}$ and in addition an approximation of $kp \approx N^{\frac{3}{4}}$. Our result shows that the amount of required known bits is then still only $N^{\frac{1}{4}}$, whereas we can efficiently recover the unknown remaining part of $kp$ of size $N^{\frac{1}{2}}$. In other words, in the *approximate divisor multiple* setting we only need a *third* of $kp$ as opposed to a *half* of $p$ in *factoring with a hint*. However, the total amount $N^{\frac{1}{4}}$ of known bits is identical in both settings.

This effect helps us to improve Partial Key Exposure Attacks in the public exponent regime $e \leq N^{\frac{1}{12}}$, and explains the *bump shape* in Figure 1. Since $k$ grows with larger $e$, we obtain approximations of $kp$ that allow for more efficient factorizations. This leads to the – maybe somewhat couterintertuitive – result that for $e \approx N^{\frac{1}{12}}$ we need the least amount of $d_p, d_q$ to efficiently factor. Namely, in this case only a third of the bits of $d_p, d_q$ is sufficient to factor $N$. For $e > N^{\frac{1}{12}}$ our two-step approach requires again more bits, since the reconstruction of $k, \ell$ in the first step requires more than a third of the bits of $d_p, d_q$, see Figure 1. Eventually, based on our conjecture we expect that for $e \geq N^{\frac{1}{8}}$ our approach is superseded by the (heuristic) Takayasu-Kunihiro attack.

Notice that our MSB attack is rigorous in both steps, leading to a fully provable factorization algorithm. In the LSB case we require the usual Coppersmith-type heuristic only for the first step that computes $k, \ell$. We verify the validity of this heuristic for our LSB case, and for TK in the case $e \geq N^{\frac{1}{8}}$ experimentally in Section 5.

*Discussion of Our Result.* Since our result suggests that RSA with $e \approx N^{\frac{1}{12}}$ is most vulnerable to Partial Key Exposure attacks, this strange behaviour might ask for further discussion. In our opinion, constant-size $e$ is still weakest in the PKE setting. In fact, whereas the Blmer-May result requires only half of either $d_p$ *or* $d_q$, we require at least a third of $d_p$ *and* $d_q$. As a (rough) numerical example for $n$-bit $d_p$, $d_q$ one requires in Blmer-May $\frac{1}{2}n$ bits to factor $N$, whereas we require $\frac{2}{3}n$ bits. Nevertheless, there might be side-channel scenarios that allow for easier recovery of a third of both $d_p, d_q$ than for a half of a single CRT-exponent. In such a case, our results indicate that $e \approx N^{\frac{1}{12}}$ is indeed weakest.

## 2 Coppersmith's Method

We briefly recall Coppersmith's lattice-based method for computing small modular roots of polynomials [7]. For a more thorough introduction we refer to [16].

Suppose we are given a $k$-variate polynomial $f(x_1, \ldots, x_k)$, which has a small root $r = (r_1, \ldots, r_k)$ modulo some integer $M$, where for $i = 1, \ldots, k$ and known bounds $X_i$ we have $|r_i| \leq X_i$. Our goal is to compute $r$ in polynomial time.

We fix an $m \in \mathbb{N}$ and define a collection of so-called *shift-polynomials*

$$g_{[i_0,\ldots,i_k]}(x_1, \ldots, x_k) := f^{i_0}(x_1, \ldots, x_k) \cdot x_1^{i_1} \cdot \ldots \cdot x_k^{i_k} \cdot M^{m-i_0}$$

with indices $i_0, \ldots, i_k \in \mathbb{N}$. By construction the shift-polynomials have the root $r$ modulo $M^m$.

Next we select a subset of polynomials $g_{[i_0,\ldots,i_k]}(X_1x_1, \ldots, X_kx_k)$, whose coefficient vectors generate an $n$-dimensional lattice with triangular basis matrix $\mathbf{B}$. If $m$ is chosen sufficiently large and the so-called *enabling condition*

$$|\det \mathbf{B}| \leq M^{mn} \tag{2}$$

is satisfied, then we can compute $k$ polynomials $h_1(x_1, \ldots, x_k), \ldots, h_k(x_1, \ldots, x_k)$ in polynomial time, which have the root $r$ not only modulo $M^m$, but also over the integers.

If our polynomial $f$ is univariate, i.e. $k = 1$, then we can easily obtain $r$ from $h_1$, using standard techniques such as Newton's method. In the case of a multivariate polynomial $f$, i.e. $k > 1$, the polynomials $h_1, \ldots, h_k$ however do not necessarily reveal the root $r$. Nevertheless, in practice the polynomials $h_1, \ldots, h_k$ usually generate an ideal $\mathfrak{a} = \langle h_1, \ldots, h_k \rangle$ of zero-dimensional variety – in which case we can still obtain $r$ in polynomial time by computing the Grbner basis of $\mathfrak{a}$. There is, however, no provable guarantee that the variety of $\mathfrak{a}$ is zero-dimensional. Therefore, many Coppersmith-type results rely on the following heuristic assumption.

**Assumption 1** *Coppersmith's method for the multivariate setting yields polynomials, which generate an ideal of zero-dimensional variety.*

As Assumption 1 might fail in some instances (e.g. in the small $e$ regime of the TK attack), it is crucial to verify its validity experimentally.

## 3 Our Two-Step Partial Key Exposure Attack

Let $(N, e)$ be an RSA public key, where $N = pq$ and $e = N^\alpha$. As usual in practice, we assume that $p$ and $q$ have the same bit-size, which lets us bound $p, q = \Theta(N^{1/2})$. Let $d_p$ and $d_q$ be the corresponding CRT-exponents. We assume that $d_p$ and $d_q$ are full-size, i.e., $d_p, d_q = \Theta(N^{1/2})$. We write

$$d_p = d_p^{(M)} 2^i + d_p^{(L)},$$
$$d_q = d_q^{(M)} 2^i + d_q^{(L)},$$

for MSBs $d_p^{(M)}, d_q^{(M)}$ and LSBs $d_p^{(L)}, d_q^{(L)}$. We call knowledge of the bits of $d_p^{(M)}, d_q^{(M)}$ the *MSB case*, where knowledge of $d_p^{(L)}, d_q^{(L)}$ is called the *LSB case.*

Let us first state our main result that already explains the *bump shape* of our attack region in Figure 1.

**Theorem 1.** *Let $N$ be sufficiently large. Suppose we are given the public key $(N, e)$ with $e = N^\alpha$ and additionally the MSBs $d_p^{(M)}, d_q^{(M)}$ or the LSBs $d_p^{(M)}, d_q^{(M)}$ of the CRT-exponents $d_p, d_q$. If the unknown parts of $d_p$ and $d_q$ are upper bounded by $N^\delta$, where*

$$\delta < \min\left\{\frac{1}{4} + \alpha, \frac{1}{2} - 2\alpha\right\},$$

*then we can factor $N$ in polynomial time (under Assumption 1 for the LSB case).*

*Proof.* The CRT-exponents $d_p, d_q$ fulfill the RSA key equations

$$ed_p = k(p-1) + 1, \tag{3}$$

$$ed_q = \ell(q-1) + 1. \tag{4}$$

We follow a two-step strategy for factoring $N$.

*Step 1.* We show that we can compute the two parameters $k$ and $\ell$ both in the MSB case (Section 3.1) and the LSB case (Section 3.2) if the unknown parts of the CRT-exponents are upper bounded by $N^{\frac{1}{2}-2\alpha}$. The MSB algorithm is completely elementary, whereas the LSB algorithm is based on Coppersmith's (heuristic) method for multivariate polynomials, and therefore requires Assumption 1.

*Step 2.* Given $k$ and MSBs/LBSs of $d_p$ with unknown part bounded by $N^{\frac{1}{4}+\alpha}$, we then provide in Section 3.3 a completely rigorous factoring algorithm based on a novel result on *approximate divisor multiples* (Theorem 3) that we prove using Coppersmith's method for univariate modular polynomials. □

### 3.1 Step 1: Computing $(k, \ell)$, Given MSBs

We first show how to compute $(k, \ell)$, given the MSBs of the CRT-exponents. For this scenario, our algorithm is particularly simple and efficient, as it uses only elementary arithmetic.

**Lemma 1** **($(k, \ell)$ from MSB).** *Let $N$ be sufficiently large. Suppose we are given the public key $(N, e)$ with $e = N^\alpha$ and the MSBs $d_p^{(M)}, d_q^{(M)}$. If the unknown LSBs are upper bounded by $d_p^{(L)}, d_q^{(L)} \leq N^\delta$, where*

$$\delta < \frac{1}{2} - 2\alpha,$$

*then we can compute $(k, \ell)$ in time $\mathcal{O}(\log^2 N)$.*

*Proof.* We rewrite equations (3) and (4) as

$$\begin{aligned} kp &= k - 1 + ed_p, \\ \ell q &= \ell - 1 + ed_q. \end{aligned}$$

Multiplying $kp$ with $\ell q$, we obtain the identity

$$k\ell N = (k-1)(\ell-1) + ed_p(\ell-1) + ed_q(k-1) + e^2 d_p d_q. \tag{5}$$

Let

$$\widetilde{A} := \frac{2^{2i} e^2 d_p^{(M)} d_q^{(M)}}{N}.$$

Since we are given the MSBs $d_p^{(M)} d_q^{(M)}$, $\widetilde{A}$ can efficiently be computed. Using $\delta < \frac{1}{2} - 2\alpha$, we now show $\lceil \widetilde{A} \rceil = k\ell - o(1)$. Hence, for sufficiently large $N$ (which already holds for standard RSA moduli) we have $\lceil \widetilde{A} \rceil = k\ell$.

Let us rewrite (5) as

$$\begin{aligned} k\ell N - \widetilde{A}N =& (k-1)(\ell-1) + ed_p(\ell-1) + ed_q(k-1) + e^2 d_p d_q - \widetilde{A}N \\ =& (k-1)(\ell-1) + ed_p(\ell-1) + ed_q(k-1) \\ &+ 2^i e^2 (d_p^{(M)} d_q^{(L)} + d_p^{(L)} d_q^{(M)}) + e^2 d_p^{(L)} d_q^{(L)}. \end{aligned}$$

Using

$$d_p, d_q = \Theta(N^{1/2}), \quad d_p^{(M)}, d_q^{(M)} = \Theta(N^{1/2-\delta}), \quad 2^i = \Theta(N^{\delta}),$$

and

$$\begin{aligned} k &= \frac{ed_p - 1}{p-1} < \frac{ed_p}{p-1} < \frac{e(p-1)}{p-1} = e, \\ \ell &= \frac{ed_q - 1}{q-1} < \frac{ed_q}{q-1} < \frac{e(q-1)}{q-1} = e, \end{aligned} \tag{6}$$

we obtain

$$\begin{aligned} k\ell N - \widetilde{A}N &= \mathcal{O}(N^{2\alpha}) + \mathcal{O}(N^{2\alpha+1/2}) + \mathcal{O}(N^{2\alpha+1/2+\delta}) + \mathcal{O}(N^{2\alpha+2\delta}) \\ &= \mathcal{O}(N^{2\alpha+1/2+\delta}), \end{aligned}$$

and therefore

$$k\ell - \widetilde{A} = \mathcal{O}(N^{\delta+2\alpha-1/2}) = o(1).$$

Thus, given the MSBs, we can compute $k\ell$ in time $\mathcal{O}(\log^2 N)$.

It remains to show that knowledge of $k\ell$ yields $k$ and $\ell$. Using Equation (5) we have

$$k + \ell = 1 - k\ell(N-1) \bmod e, \tag{7}$$

where the right-hand side is known. By (6) we know that the left-hand side satifies $0 \le k + \ell < 2e$. Thus, either $0 \le k + \ell < e$ or $0 \le k + \ell - e < e$.

Assume for a moment that $0 \leq k + \ell < e$. Then Equation (7) holds over the integers. Thus, $k, \ell$ are the two solutions of the quadratic polynomial equation

$$0 = (x - k)(x - \ell) = x^2 - (k + \ell)x + k\ell = x^2 + (1 - k\ell(N - 1))x + k\ell.$$

We check whether the product of the solutions equals $k\ell$. If it does, we have recovered $k$ and $\ell$. If not, we are in the case $0 \leq k + \ell - e < e$. We then recover $k$ and $\ell$ as the integer solutions of

$$0 = x^2 + (1 - k\ell(N - 1) + e)x + k\ell.$$

This can again be done in time $\mathcal{O}(\log^2 N)$. □

### 3.2 Step 1: Computing $(k, \ell)$, Given LSBs

In Section 3.1, the computation of $k, \ell$ from known MSBs of $d_p, d_q$ within the bound of Lemma 1 was elementary, efficient and provable. Although we achieve in the LSB case the same bound as in Lemma 1, the approach is quite different. We use Coppersmith's lattice-based method in combination with Assumption 1, which makes the recovery of $k, \ell$ heuristic and a bit less efficient, but we still come close to the bound in a matter of seconds, see Section 5.

**Lemma 2** **($(k, \ell)$ from LSB).** *Let $N$ be sufficiently large. Suppose we are given the public key $(N, e)$ with $e = N^\alpha$ and the LSBs $d_p^{(L)}, d_q^{(L)}$. If the unknown MSBs are upper bounded by $d_p^{(M)}, d_q^{(M)} \leq N^\delta$, where*

$$\delta < \frac{1}{2} - 2\alpha,$$

*then we can compute $(k, \ell)$ in polynomial time (under Assumption 1).*

*Proof.* Let us recall Equation (5):

$$k\ell N = (k - 1)(\ell - 1) + ed_p(\ell - 1) + ed_q(k - 1) + e^2 d_p d_q.$$

Plugging in $d_p = d_p^{(M)} 2^i + d_p^{(L)}$ and $d_q = d_q^{(M)} 2^i + d_q^{(L)}$ we obtain

$$k\ell N \equiv (k - 1)(\ell - 1) + ed_p^{(L)}(\ell - 1) + ed_q^{(L)}(k - 1) + e^2 d_p^{(L)} d_q^{(L)} \mod 2^i e,$$

and equivalently

$$k\ell(N - 1) - k(ed_q^{(L)} - 1) - \ell(ed_p^{(L)} - 1) + A \equiv 0 \mod 2^i e, \tag{8}$$

where

$$A := -e^2 d_p^{(L)} d_q^{(L)} + ed_p^{(L)} + ed_q^{(L)} - 1.$$

We derive from (8) a polynomial

$$f(x, y) := (N - 1)xy - (ed_q^{(L)} - 1)x - (ed_p^{(L)} - 1)y + A,$$

which has the root $(k, \ell)$ modulo $2^i e$. Notice that all coefficients of $f$ are known.

Now we apply Coppersmith's method to $f$ to compute $k$ and $\ell$. To this end, we need to transform $f$ into a polynomial $g$, which also has the root $(k, \ell)$ modulo $2^i e$, but additionally has at least one *small* coefficient (in the sense that it does not grow as function of $N$). This ensures that for sufficiently large $N$ the coefficients of the polynomial do not affect the enabling condition from Equation (2) in Coppersmith's method.

It is not hard to see that $\gcd(ed_q^{(L)} - 1, 2^i e) = \gcd(ed_q^{(L)} - 1, 2^i)$ is (for randomly chosen RSA keys) a small power of 2. Hence, we may define $g$ by replacing the coefficient of $x$ in $f$ by $\gcd(ed_q^{(L)} - 1, 2^i)$ and multiplying all other coefficients by the multiplicative inverse of

$$\frac{ed_q^{(L)} - 1}{\gcd(ed_q^{(L)} - 1, 2^i)}$$

modulo $2^i e$.

As shown in (6), both $k$ and $\ell$ can be upper bounded by $k, \ell < e$. It is known (see, for instance, the *generalized rectangle* construction in [12, Appendix A]) that under Assumption 1 we can compute *all* roots $(x_0, y_0)$ of $g$ modulo $2^i e$, that satisfy $|x_0|, |y_0| < e$, in polynomial time, provided that

$$e^2 < (2^i e)^{\frac{2}{3}}. \tag{9}$$

Plugging in $e = N^\alpha$ and $2^i = \Theta(N^{1/2-\delta})$, we find that (9) is asymptotically equivalent to

$$\delta < \frac{1}{2} - 2\alpha,$$

which concludes the proof. □

### 3.3 Step 2: Factoring $N$, Given $k$

In Sections 3.1 and 3.2 we described Step 1, the computation of $k, \ell$ from either known MSBs or LSBs of the CRT-exponents. Step 2 now finishes the factorization of $N$ in polynomial time.

To this end, we recall a result of Howgrave-Graham [11] for computing small roots of linear polynomials modulo unknown divisors, which can be seen as a generalization of Coppersmith's *factoring with a hint* result.

**Theorem 2 (Howgrave-Graham).** *Suppose we are given a polynomial $f(x) := x + a$ and an integer $N \in \mathbb{N}$ of unknown factorization. Let $p \geq N^\beta \in \mathbb{N}, \beta \in [0, 1]$ be an unknown divisor of $N$. In time polynomial in $\log N$ and $\log a$, we can compute all integers $x_0$, satisfying*

$$f(x_0) \equiv 0 \mod p \quad \textit{and} \quad |x_0| \leq N^{\beta^2}.$$

*Factoring with a hint* follows when we set $\beta = \frac{1}{2}$ and the coefficient $a$ to either the LSBs or MSBs of $p$. Then we can efficiently recover the unknown part of size $N^{\frac{1}{4}}$, i.e., half the of $p$'s bits.

We prove the following generalization of Theorem 2.

**Theorem 3.** *Suppose we are given a polynomial $f(x) := x + a$ and integers $k, N \in \mathbb{N}$, where $k = N^{\mu}$ for some $\mu \geq 0$. Let $p \geq N^{\beta} \in \mathbb{N}, \beta \in [0, 1]$ be an unknown divisor of $N$. In time polynomial in $\log N$, $\log k$ and $\log a$, we can compute all integers $x_0$, satisfying*

$$f(x_0) \equiv 0 \mod kp \quad \textit{and} \quad |x_0| \leq N^{\beta^2 + \mu}.$$

*Remark 1.* Blömer, May[3, Theorem 13] already showed a similar generalization with $|x_0| \leq N^{\frac{(\beta+\mu)^2}{1+\mu}}$. However, their bound is for $\mu > 0$ strictly weaker than ours, since

$$\frac{(\beta + \mu)^2}{1 + \mu} = \beta^2 + \mu - \mu \cdot \frac{(\beta - 1)^2}{1 + \mu} < \beta^2 + \mu.$$

*Proof.* For integers $m, t \in \mathbb{N}$ and $i = 0, \ldots, m$, we define a collection of polynomials

$$g_i(x) := f^i(x) k^{m-i} N^{\max\{0, t-i\}}.$$

Notice that for every root $x_0 \in \mathbb{Z}$ of $f(x)$ modulo $kp$ we have for any $i$ that

$$g_i(x_0) \equiv 0 \mod k^m p^t.$$

Let us set the bound for the size of the root $x_0$ as

$$X = N^{\beta^2 + \mu}. \tag{10}$$

We construct an $(m + 1) \times (m + 1)$ lattice basis matrix $\mathbf{B}$, where the $i$-th row corresponds to the coefficient vector of $g_i(xX)$ and the $i$-th row corresponds to the monomial $x^i$, see Figure 2 for an example.

$$\begin{array}{c|ccccc}
 & 1 & x & x^2 & x^3 & x^4 \\
\hline
g_0 & k^4N^2 & & & & \\
g_1 & ak^3N & k^3NX & & & \\
g_2 & a^2k^2 & 2ak^2X & k^2X^2 & & \\
g_3 & a^3k & 3a^2kX & 3akX^2 & kX^3 & \\
g_4 & a^4 & 4a^3X & 6a^2X^2 & 4aX^3 & X^4
\end{array}$$

**Fig. 2.** Example of our basis matrix $\mathbf{B}$ with $m = 4, t = 2$. Empty entries are zero.

Using Equation (2), we can compute a univariate polynomial $h(x)$ with all the roots $x_0, |x_0| \leq X$ over the integers, provided that the following enabling condition holds

$$|\det \mathbf{B}| \leq \left(k^m p^t\right)^{m+1}. \tag{11}$$

From $h(x)$ we derive the roots in polynomial time using standard root finding algorithms. (For instance, as noted by Coppersmith [7], the Sturm sequence will suffice, see [14].)

It is not hard to see that

$$\det \mathbf{B} = (kX)^{\frac{m^2+m}{2}} N^{\frac{t^2+t}{2}}.$$

We plug $\det \mathbf{B}$ into (11), set $t = \beta m$, and take the $m^2$-th root on both sides. Ignoring low order terms, the condition becomes

$$(kX)^{\frac{1}{2}} N^{\frac{\beta^2}{2}} \leq p^\beta k.$$

Solving for $X$ yields

$$X \leq p^{2\beta} N^{-\beta^2} k.$$

Using $p \geq N^\beta$, we obtain the more restrictive condition $X \leq N^{\beta^2+\mu}$, which is satisfied by our definition of $X$ in Equation (10). □

Theorem 3 shows that in the case of *approximate divisor multiples* for some known multiple $k$ the size of the efficiently recoverable root grows linearly in $k$. This might seem quite surprising at first sight, but let us also look at the *known part* $a$. In Theorem 2 the coefficient $a$ has to be of size at least $N^{\beta-\beta^2}$, whereas in Theorem 3 it has to be of size at least $N^{(\beta+\gamma)-(\beta^2+\gamma)} = N^{\beta-\beta^2}$, i.e., the amount of required known bits stays constant for every $k$.

Since in Partial Key Exposure attacks for CRT-exponents we get an approximation of $kp$ for some known $k$ (by Step 1), we profit from the *approximate divisor multiple* setting. This is shown in the following Lemma 3, that is a direct application of Theorem 3, and completes the proof of our main Theorem 1.

**Lemma 3.** *Let $N$ be sufficiently large. Suppose we are given the public key $(N, e)$ with $e = N^\alpha$, the value $k$ and additionally the MSBs $d_p^{(M)}, d_q^{(M)}$ or the LSBs $d_p^{(L)}, d_q^{(L)}$. If the unknown parts of $d_p$ and $d_q$ are upper bounded by $N^\delta$, where*

$$\delta < \frac{1}{4} + \alpha,$$

*then we can factor $N$ in polynomial time.*

*Proof.* Let us first prove the MSB case. Using

$$ed_p = 1 + k(p-1) \quad \text{and} \quad d_p = d_p^{(L)} + d_p^{(M)} 2^i$$

with unknown $d_p^{(L)}$, we obtain

$$ed_p^{(L)} + ed_p^{(M)}2^i + k - 1 = kp.$$

This equation yields a polynomial $f_{\mathsf{MSB}}(x) = x + a \bmod kp$ with known coefficient

$$a = \left(ed_p^{(M)}2^i + k - 1\right) \cdot \left(e^{-1} \mod kN\right) \text{ and root } x_0 = d_p^{(L)}.$$

Using $k = \Theta(N^\alpha)$ and $p = \Theta(N^{1/2})$, we conclude from Theorem 3 that we can compute $d_p^{(L)}$ in polynomial time, provided that

$$d_p^{(L)} < N^{\frac{1}{4}+\alpha},$$

which is satisfied, since $d_p^{(L)} \leq N^\delta$. Eventually, the prime factorization follows from $d_p^{(L)}$ via $\gcd(f_{\mathsf{MSB}}(d_p^{(L)}), N) = p$.

The LSB case follow completely analogous using the polynomial

$$f_{\mathsf{LSB}}(x) = x + \left(ed_p^{(L)} + k - 1\right) \cdot \left((2^i e)^{-1} \mod kN\right).$$

□

Notice that for $e \geq N^{\frac{1}{4}}$, the unknown part in Lemma 3 of $d_p, d_q$ can be as large as $N^{\frac{1}{4}+\alpha} \geq N^{\frac{1}{2}}$, i.e., we can factor without knowing any bits of $d_p, d_q$. Thus, our two-step approach cannot work for $e \geq N^{\frac{1}{4}}$, unless factoring is easy, coinciding with Galbraith, Heneghan, McKee [10, Theorem 1].

**Corollary 1.** *For $e \geq N^{\frac{1}{4}}$ computation of $k, \ell$ is as hard as factoring.*

### 3.4 On the Limits of Improving Our Attack

Our two step approach first computes $k$ and $\ell$, but the second step only requires $k$ to factor $N$. One may ask whether the computation of $k$ alone in Step 1 leads to an improved Partial Key Exposure attack. The following lemma answers this in the negative, since the computation of $k, \ell$ is no more difficult than the computation of $k$ alone.

**Lemma 4.** *Suppose there exists a polynomial time algorithm, which on input $(N, e)$ outputs $k$. Then there also exists a polynomial time algorithm, which on input $(N, e)$ outputs $(k, \ell)$.*

*Proof.* We rewrite the congruence

$$k(p-1) + 1 \equiv 0 \mod e, \tag{12}$$

as

$$p \equiv (k-1)k^{-1} \mod e. \tag{13}$$

Notice that from (12) it follows that $k$ is indeed invertible modulo $e$, as its inverse is $(1-p)$.

Arguing analogously for $\ell$, it follows that the inverse of $\ell$ modulo $e$ is $(1-q)$. Combining this observation with (13) and assuming that $p$ is w.l.o.g. invertible modulo $e$ yields

$$\ell \equiv \Big(1-q\Big)^{-1} \equiv \Big(1-Np^{-1}\Big)^{-1} \equiv \Big(1-Nk(k-1)^{-1}\Big)^{-1} \mod e.$$

Hence $\ell$ can be efficiently computed from $(N, e, k)$. □

## 4 Limits of the Takayasu-Kunihiro PKE for Small $e$

In [19, Theorem 7] Takayasu and Kunihiro claim the following result for CRT-exponents $d_p, d_q \approx N^{1/2}$.

**Claim 1 (Takayasu, Kunihiro)** *Let $N$ be sufficiently large. Suppose we are given the public key $(N,e)$ with $e = N^\alpha$ and additionally the MSBs $d_p^{(M)}, d_q^{(M)}$ or the LSBs $d_p^{(M)}, d_q^{(M)}$ of the CRT-exponents $d_p, d_q$. If the unknown parts of $d_p$ and $d_q$ are upper bounded by $N^\delta$, where*

1. $\delta < \frac{(12-12\alpha)\tau^2+(12-16\alpha)\tau+3-4\alpha}{24\tau^3+54\tau^2+40\tau+10}$, $\frac{7}{16} < \alpha < 1$, $\tau > 0$ *or,*
2. $\delta < \frac{3-4\alpha}{10}$, $0 < \alpha \leq \frac{3}{4}$ *or,*
3. $\delta < \frac{-24\alpha\tau^3+(12-30\alpha)\tau^2+(12-16\alpha)\tau+3-4\alpha}{36\tau^2+40\tau+10}$, $0 < \alpha < \frac{1}{13}$, $\tau > 0$,

*then we can factor $N$ in polynomial time.*

When numerically optimizing the value of $\tau$, we obtain the results shown in Table 1. We see that for $\alpha$ approaching 0, the value of $\delta$ converges to $\frac{1}{3}$. Hence, for constant $e$, the TK attack claims to succeed for unknown bits of size $N^{1/3}$, or equivalently for known bits of size $N^{1/2-1/3} = N^{1/6}$, i.e., only a third of the bits of the CRT-exponents $d_p, d_q \approx N^{1/2}$ have to be known.

| $\alpha$ | $10^{-8}$ | $10^{-4}$ | 0.05 | 0.10 | 0.15 | 0.20 | 0.25 | 0.30 | 0.35 | 0.40 | 0.45 | 0.50 | 0.55 | 0.60 | 0.65 | 0.70 | 0.75 | 0.80 | 0.85 | 0.90 | 0.95 | 1.00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\delta$ | 0.333 | 0.330 | 0.280 | 0.260 | 0.240 | 0.220 | 0.200 | 0.180 | 0.160 | 0.140 | 0.120 | 0.100 | 0.082 | 0.065 | 0.049 | 0.036 | 0.025 | 0.016 | 0.009 | 0.004 | 0.001 | 0.000 |

**Table 1.** Values of $\alpha$ and $\delta$, for which the TK attack claims to succeed.

This is caused by the fact that for $\alpha \to 0$ the third condition of Claim 1 converges to

$$\delta < \frac{12\tau^2+12\tau+3}{36\tau^2+40\tau+10},$$

whose right-hand side gets arbitrarily close to $\frac{12}{36} = \frac{1}{3}$, when choosing $\tau$ sufficiently large.

We show in the following that any PKE attack, that succeeds in the very small $e$ regime with only a third of the CRT-exponent bits, leads to an improvement of *factoring with a hint* with also only a third of the prime factor bits. This already casts severe doubts on the validity of the asymptotics of the Takayasu-Kunihiro attack in the small $e$ regime. In the subsequent Section 4.1, we give further arguments, why the TK attack fails for $e \leq N^{1/8}$.

We confirm this observation experimentally in Section 5. TK fails in the $e \leq N^{1/8}$ regime, but works perfectly for $e > N^{1/8}$.

**Theorem 4.** *Suppose there exists a polynomial time algorithm* $\mathsf{A}$*, which on input* $(N, e, \widetilde{d_p}, \widetilde{d_q})$ *with* $e = \mathcal{O}(\log N)$ *outputs the prime factors of* $N$*, where* $\widetilde{d_p}, \widetilde{d_q}$ *are* $\mu$*-fractions of the MSBs or LSBs of the CRT-exponents for some* $\mu \in [0, 1]$*. Then there also exists a polynomial time algorithm* $\mathsf{B}$*, which on input* $(N, \widetilde{p})$ *outputs the prime factors of* $N$*, where* $\widetilde{p}$ *is a* $\mu$*-fraction of the MSBs or LSBs of* $p$*.*

*Proof.* Let $p = p^{(M)}2^i + p^{(L)}$ for MSBs $p^{(M)}$ and LSBs $p^{(L)}$. We first prove the result for the case $\widetilde{p} = p^{(L)}$.

We define $\mathsf{B}$ as follows. Given $(N, \widetilde{p})$, we iterate over all odd candidate public exponents $e = 3, 5, 7, \ldots$ and for every $e$ we test all tuples $(k, \ell) \in \{1, 2, \ldots e-1\}^2$. For every $(e, k, \ell)$, we compute

$$\begin{aligned} \widetilde{d_p} &:= (k(\widetilde{p} - 1) + 1)e^{-1} \mod 2^i, \\ \widetilde{d_q} &:= (\ell(N\widetilde{p}^{-1} - 1) + 1)e^{-1} \mod 2^i, \end{aligned}$$

and then run $\mathsf{A}$ on input $(N, e, \widetilde{d_p}, \widetilde{d_q})$. Notice, if $(N, e)$ is a valid RSA public key and furthermore $k$ and $\ell$ are the corresponding parameters in the sense of the CRT key equations (3) and (4), then from (3) and (4) it easily follows that $\widetilde{d_p}$ and $\widetilde{d_q}$ are indeed $\mu$-fractions of the LSBs of the CRT-exponents. Hence, whenever $(N, e)$ is valid, $\mathsf{A}$ outputs the prime factors of $N$ and we terminate $\mathsf{B}$.

As any $n \in \mathbb{N}$ has at most $\log_2 n$ prime factors, it follows that after $\mathcal{O}(\log \phi(N)) = \mathcal{O}(\log N)$ iterations, we choose an $e$ coprime to $\phi(N)$. This implies that $(N, e)$ is a valid public key. Thus, $\mathsf{B}$ terminates after at most $\mathcal{O}(\log N)$ choices of $e$, from which we conclude that $\mathsf{B}$ calls $\mathsf{A}$ at most $\mathcal{O}(\log^3 N)$ times.

It remains to prove the MSB case, i.e., $\widetilde{p} = p^{(M)}$. We first note that given $\widetilde{p}$, we immediately obtain from the MSBs of $\frac{N}{\widetilde{p}2^i}$ a $\mu$-fraction of the MSBs of the other prime factor $q$. Let us denote these MSBs by $\widetilde{q}$.

We iterate, analogous to the LSB case, over all tuples $(e, k, \ell)$. As before, we are guaranteed to obtain a valid tuple in time $\mathcal{O}(\log^3 N)$. For a valid tuple $(e, k, \ell)$, we rewrite (3) as

$$d_p^{(M)} = \frac{k(\widetilde{p}2^i - 1) + 1}{2^i e} + \frac{kp^{(L)} - ed_p^{(L)}}{2^i e},$$

where we may bound

$$\left|\frac{kp^{(L)} - ed_p^{(L)}}{2^i e}\right| \leq \frac{kp^{(L)}}{2^i e} + \frac{ed_p^{(L)}}{2^i e} < \frac{2^i e}{2^i e} + \frac{2^i e}{2^i e} = 2.$$

It follows that given $\widetilde{p}$ and a valid tuple $(e, k, \ell)$, we compute the MSBs of the corresponding $d_p$ as

$$\widetilde{d_p} = \left\lceil \frac{k(\widetilde{p}2^i - 1) + 1}{2^i e} \right\rceil \pm \epsilon_p,$$

where $\epsilon_p \in \{0, 1, 2\}$. Analogous, we compute the MSBs of $d_q$ as

$$\widetilde{d_q} = \left\lceil \frac{\ell(\widetilde{q}2^i - 1) + 1}{2^i e} \right\rceil \pm \epsilon_q,$$

where $\epsilon_q \in \{0, 1, 2\}$.

Analogous to the LSB case we now simply run for all candidates algorithm $\mathsf{A}$ on input $(N, e, \widetilde{d_p}, \widetilde{d_q})$. □

### 4.1 Why TK Fails for $e \leq N^{1/8}$

In the MSB case, the TK attack uses Coppersmith's method to compute the integer root $(d_p^{(L)}, d_q^{(L)}, k, \ell)$ of the polynomial

$$\begin{aligned} f_{\mathsf{MSB}}(x_1, x_2, y_1, y_2) := {} & e^2 x_1 x_2 + (e^2 d_q^{(M)} 2^i - e)x_1 + (e^2 d_p^{(M)} 2^i - e)x_2 \\ & + ex_1 y_2 + ex_2 y_1 + (ed_q^{(M)} 2^i - 1)y_1 + (ed_p^{(M)} 2^i - 1)y_2 \\ & - (N-1)y_1 y_2 + c_{\mathsf{MSB}}, \end{aligned}$$

where $c_{\mathsf{MSB}} \in \mathbb{Z}$ is some constant. Similarly, in the LSB case, the TK attack uses Coppersmith's method to compute the integer root $(d_p^{(M)}, d_q^{(M)}, k, \ell)$ of the polynomial

$$\begin{aligned} f_{\mathsf{LSB}}(x_1, x_2, y_1, y_2) := {} & e^2 2^{2i} x_1 x_2 + (e^2 d_q^{(L)} - e)2^i x_1 + (e^2 d_p^{(L)} - e)2^i x_2 \\ & + e2^i x_1 y_2 + e2^i x_2 y_1 + (ed_q^{(L)} - 1)y_1 + (ed_p^{(L)} - 1)y_2 \\ & - (N-1)y_1 y_2 + c_{\mathsf{LSB}}, \end{aligned}$$

where $c_{\mathsf{LSB}} \in \mathbb{Z}$ is some constant.

From Figure 1, we see that our attacks for recovering $k$ and $\ell$ from Lemmas 1 and 2 require for $e \leq N^{1/8}$ less known bits than the TK attack. Thus, in the very small $e$ regime, it does not seem useful to treat $k$ and $\ell$ as unknowns in the TK attack. Instead, we should use in a first step our attacks from Lemmas 1 and 2 to obtain $k$ and $\ell$ and after that plug them into $f_{\mathsf{MSB}}(x_1, x_2, y_1, y_2)$ and $f_{\mathsf{LSB}}(x_1, x_2, y_1, y_2)$ to eliminate the variables $y_1$ and $y_2$. By that, we obtain two new polynomials

$$\begin{aligned} f_{\mathsf{MSB}}(x_1,x_2) &:= e^2x_1x_2 + (e^2d_q^{(M)}2^i - e)x_1 + (e^2d_p^{(M)}2^i - e)x_2 \\ &\quad + e\ell x_1 + ekx_2 + c^*_{\mathsf{MSB}}, \\ f_{\mathsf{LSB}}(x_1,x_2) &:= e^2 2^{2i}x_1x_2 + (e^2d_q^{(L)} - e)2^i x_1 + (e^2d_p^{(L)} - e)2^i x_2 \\ &\quad + e2^i\ell x_1 + e2^ikx_2 + c^*_{\mathsf{LSB}}, \end{aligned}$$

for some constants $c^*_{\mathsf{MSB}}, c^*_{\mathsf{LSB}} \in \mathbb{Z}$. Intuitively, this should only improve the TK attack, because then we have to recover only two unknowns instead of four.

Unfortunately, the bi-variate polynomials $f_{\mathsf{MSB}}(x_1,x_2)$ and $f_{\mathsf{LSB}}(x_1,x_2)$ are (unlike their four-variate counterparts) not irreducible over the integers. Indeed, all coefficients of $f_{\mathsf{MSB}}(x_1,x_2)$ are divisble by $e$ and all coefficients of $f_{\mathsf{LSB}}(x_1,x_2)$ are divisible by $e2^i$. Hence, we can not directly apply Coppersmith's method here, as Coppersmith's method only works with irreducible polynomials. Instead, we first have to divide $f_{\mathsf{MSB}}(x_1,x_2)$ by $e$ and $f_{\mathsf{LSB}}(x_1,x_2)$ by $e2^i$.

After that, we can apply the following standard result by Coppersmith [6].

**Theorem 5 (Coppersmith).** *Let $f(x,y) \in \mathbb{Z}[x,y]$ be an irreducible polynomial of degree one in each variable separately. Let $X, Y \in \mathbb{N}$ and let $W \in \mathbb{N}$ denote the largest coefficient of $f(Xx, Yy)$. Given $f$, $X$ and $Y$, we can compute all integer pairs $(x_0,y_0) \in \mathbb{Z}$ satisfying*

$$f(x_0,y_0) = 0 \quad \textit{and} \quad |x_0| \le X, |y_0| \le Y$$

*in time polynomial in $\log W$, provided that*

$$XY < W^{2/3}.$$

A standard computation shows that Theorem 5 yields for both $f_{\mathsf{MSB}}(x_1,x_2)/e$ and $f_{\mathsf{LSB}}(x_1,x_2)/(e2^i)$ the bound

$$\delta < \frac{1}{4} + \frac{\alpha}{2}.$$

Notice that this bound is for $\alpha \to 0$ inferior to the claimed TK result (Claim 1), and for every $\alpha > 0$ inferior to our result from Lemma 3.

## 5 Experimental Results

Since it is crucial to verify the validity of Assumption 1, we present in this section some experimental data for our PKE. Additionally, we present experimental evidence that the Takayasu-Kunihiro PKE [19] fails in the small $e$ regime.

We implemented our experiments in SAGE 9.3 using Linux Ubuntu 18.04.4 with an Intel® Core$^{\text{TM}}$ i7-7920HQ CPU 3.67 GHz. Our source codes are publicly available on GitHub.[3]

[3] `https://github.com/juliannowakowski/crtrsa-small-e-pke`

| Bit-size of $e$ | #Unknown MSBs | Lattice Dim. | LLL time |
|---|---|---|---|
| 16 | 256 | 9 | $< 1$s |
| 32 | 275 | 9 | $< 1$s |
| 48 | 290 | 9 | $< 1$s |
| 64 | 302 | 9 | $< 1$s |
| 85 | 315 | 64 | 5s |
| 96 | 294 | 64 | 6s |
| 112 | 258 | 64 | 6s |
| 128 | 226 | 64 | 7s |
| 144 | 186 | 64 | 9s |
| 160 | 155 | 64 | 12s |
| 176 | 118 | 64 | 14s |
| 192 | 82 | 64 | 16s |

**Table 2.** Experimental results for Step 1 of our PKE: Recovering $k$ and $\ell$, given the LSBs of the CRT-exponents for 1024 bit $N$.

| Bit-size of $e$ | #Unknown MSBs | Lattice Dim. | LLL time |
|---|---|---|---|
| 16 | 256 | 41 | 34s |
| 32 | 275 | 41 | 28s |
| 48 | 290 | 41 | 39s |
| 64 | 302 | 41 | 49s |
| 85 | 315 | 15 | $< 1$s |
| 96 | 294 | 7 | $< 1$s |
| 112 | 258 | 7 | $< 1$s |
| 128 | 226 | 7 | $< 1$s |
| 144 | 186 | 5 | $< 1$s |
| 160 | 155 | 5 | $< 1$s |
| 176 | 118 | 5 | $< 1$s |
| 192 | 82 | 5 | $< 1$s |

**Table 3.** Experimental results for Step 2 of our PKE: Factoring $N$, given $k$ and the LSBs of the CRT-exponents for 1024 bit $N$.

*Our PKE.* While Step 2 of our PKE is in both the MSB and LSB case rigorous, Step 1 is in the case LSB case heuristic. For every instance in Table 2 we ran 100 experiments. Assumption 1 was valid in every run, i.e., we could always extract the root $(k, \ell)$ using a Grbner basis in a matter of seconds.

For the sake of completeness, we give in Table 3 also some experimental data for Step 2 of our attack. As in Step 1, our PKE requires here only small lattice dimensions to succeed, especially for $e$ larger than 80 bit – making our PKE very efficient. Notice that our attack is best, in the sense of allowing a maximum of unknown bits, for 85-bit $e$ which is a $\frac{1}{12}$-fraction of $N$'s bit-size. In this case, we require $512 - 315 = 197$ known bits for lattice dimension 64, being roughly 38% of the bits of $d_p, d_q$.

In Figure 3, we compare our experimental data with our asymptotic result. We see that with our small lattice dimensions we already closely reach the asymptotic bound. The reason is that we use the Coppersmith technique in the MSB case with univariate polynomials only, and in the LSB case with bivariate polynomials.

Compare e.g. in Figure 3 with the experimental data of the TK attack that uses 4-variate polynomials, and is for similar lattice dimensions far off from the asymptotic bounds.

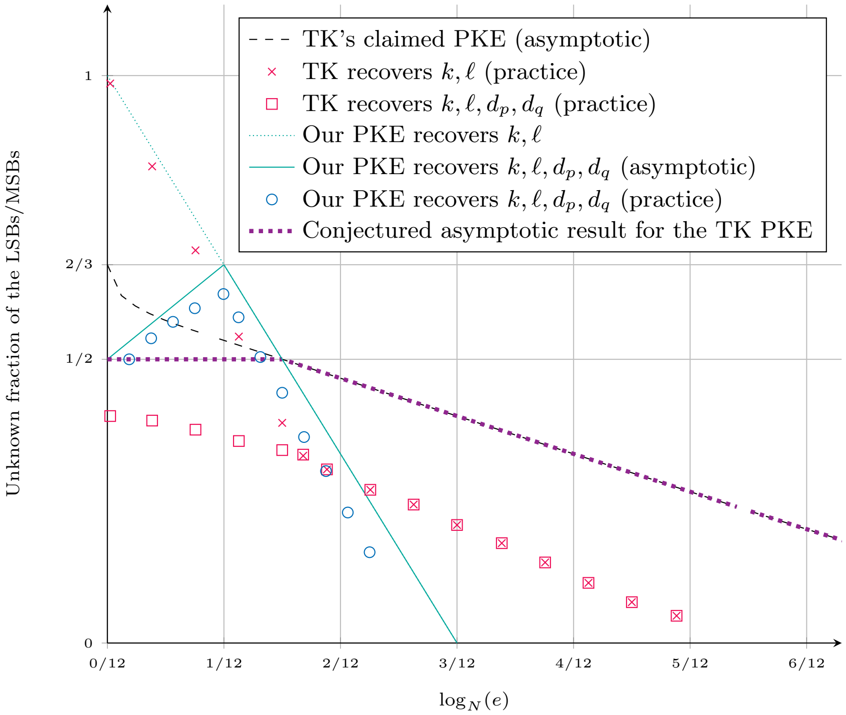

**Fig. 3.** Comparison between the Takayasu-Kunihiro attack and our attack (enlarged version of Figure 1).

*TK PKE [19].* In Tables 4 and 5 we show our experimental results for the TK attack, as also plotted in Figure 3. TK succeeds to find the factorization when it recovers $(k, \ell, d_p, d_q)$. We only ran TK successfully, when providing (significantly) more than half of the bits of $d_p, d_q$, see Table 4. When providing less than half of the bits of $d_p, d_q$ in the small $e$ regime, the Gröbner basis reveals only $k, \ell$, see Table 5. Thus, the polynomials obtained from the TK attack generate an ideal of non-zero variety, as opposed to our heuristic in Assumption 1.

The reason that TK still recovers $k, \ell$ is that the TK lattice contains as a sublattice our construction from Step 1 (at least in the LSB case).

Notice that Figure 3 shows that the graph of TK's asymptotic result becomes the steeper the closer it gets to constant $e$. In contrast, the graph of our experimental results for the TK PKE flattens in the small $e$ regime. We see this as evidence that the TK attack converges for small $e$ actually to a $\frac{1}{2}$-fraction of unknown bits, instead of a $\frac{2}{3}$-fraction. Indeed, we conjecture that for sufficiently large lattice dimensions the TK PKE works until the second intersection with our result, i.e. the point $(\frac{1}{8}, \frac{1}{2})$, for a $\frac{1}{2}$-fraction of unknown bits and after that coincides with the claimed asymptotic result, see also Figure 3.

| Bit-size of $e$ | #Unknown MSBs | Lattice Dim. | LLL time |
|---|---|---|---|
| 2 | 204 | 64 | 19s |
| 16 | 202 | 64 | 21s |
| 32 | 200 | 64 | 23s |
| 64 | 192 | 64 | 30s |
| 96 | 182 | 64 | 39s |
| 128 | 174 | 64 | 47s |
| 144 | 166 | 64 | 53s |
| 160 | 156 | 64 | 70s |
| 192 | 138 | 64 | 141s |
| 224 | 122 | 64 | 197s |
| 256 | 106 | 64 | 253s |
| 288 | 88 | 64 | 290s |
| 320 | 72 | 64 | 320s |
| 352 | 54 | 64 | 358s |
| 384 | 36 | 64 | 375s |
| 416 | 24 | 64 | 402s |

**Table 4.** Experimental results for the TK attack, recovering $k, \ell, d_p, d_q$, given the LSBs of the CRT-exponents for 1024 bit $N$.

| Bit-size of $e$ | #Unknown MSBs | Lattice Dim. | LLL time |
|---|---|---|---|
| 2 | 504 | 64 | 1s |
| 16 | 479 | 64 | 5s |
| 32 | 430 | 64 | 13s |
| 64 | 354 | 64 | 25s |
| 96 | 276 | 64 | 43s |
| 128 | 198 | 64 | 65s |

**Table 5.** Experimental results for the TK attack, recovering *only* $k$ and $\ell$, given the LSBs of the CRT-exponents for 1024 bit $N$.

## References

1. Aono, Y.: A new lattice construction for partial key exposure attack for RSA. In: Jarecki, S., Tsudik, G. (eds.) Public Key Cryptography - PKC 2009, 12th International Conference on Practice and Theory in Public Key Cryptography, Irvine, CA, USA, March 18-20, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5443, pp. 34–53. Springer (2009)
2. Blömer, J., May, A.: New partial key exposure attacks on RSA. In: Boneh, D. (ed.) Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings. Lecture Notes in Computer Science, vol. 2729, pp. 27–43. Springer (2003)
3. Blömer, J., May, A.: A tool kit for finding small roots of bivariate polynomials over the integers. In: Cramer, R. (ed.) Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3494, pp. 251–267. Springer (2005)
4. Boneh, D., Durfee, G.: Cryptanalysis of RSA with private key $d$ less than $N^{0.292}$. In: Stern, J. (ed.) Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding. Lecture Notes in Computer Science, vol. 1592, pp. 1–11. Springer (1999)

5. Boneh, D., Durfee, G., Frankel, Y.: An attack on RSA given a small fraction of the private key bits. In: Ohta, K., Pei, D. (eds.) Advances in Cryptology - ASIACRYPT '98, International Conference on the Theory and Applications of Cryptology and Information Security, Beijing, China, October 18-22, 1998, Proceedings. Lecture Notes in Computer Science, vol. 1514, pp. 25–34. Springer (1998)
6. Coppersmith, D.: Finding a small root of a bivariate integer equation; factoring with high bits known. In: Maurer, U.M. (ed.) Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding. Lecture Notes in Computer Science, vol. 1070, pp. 178–189. Springer (1996)
7. Coppersmith, D.: Small solutions to polynomial equations, and low exponent RSA vulnerabilities. J. Cryptol. **10**(4), 233–260 (1997)
8. Coron, J., May, A.: Deterministic polynomial-time equivalence of computing the RSA secret key and factoring. J. Cryptol. **20**(1), 39–50 (2007)
9. Ernst, M., Jochemsz, E., May, A., de Weger, B.: Partial key exposure attacks on RSA up to full size exponents. In: Cramer, R. (ed.) Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3494, pp. 371–386. Springer (2005)
10. Galbraith, S.D., Heneghan, C., McKee, J.F.: Tunable balancing of RSA. In: Boyd, C., Nieto, J.M.G. (eds.) Information Security and Privacy, 10th Australasian Conference, ACISP 2005, Brisbane, Australia, July 4-6, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3574, pp. 280–292. Springer (2005)
11. Howgrave-Graham, N.: Approximate integer common divisors. In: Silverman, J.H. (ed.) Cryptography and Lattices, International Conference, CaLC 2001, Providence, RI, USA, March 29-30, 2001, Revised Papers. Lecture Notes in Computer Science, vol. 2146, pp. 51–66. Springer (2001)
12. Jochemsz, E., May, A.: A strategy for finding roots of multivariate polynomials with new applications in attacking RSA variants. In: Lai, X., Chen, K. (eds.) Advances in Cryptology - ASIACRYPT 2006, 12th International Conference on the Theory and Application of Cryptology and Information Security, Shanghai, China, December 3-7, 2006, Proceedings. Lecture Notes in Computer Science, vol. 4284, pp. 267–282. Springer (2006)
13. Kakvi, S.A., Kiltz, E., May, A.: Certifying RSA. In: Wang, X., Sako, K. (eds.) Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7658, pp. 404–414. Springer (2012)
14. Knuth, D.E.: The Art of Computer Programming, Volume II: Seminumerical Algorithms. Addison-Wesley (1969)
15. May, A.: Computing the RSA secret key is deterministic polynomial time equivalent to factoring. In: Franklin, M.K. (ed.) Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings. Lecture Notes in Computer Science, vol. 3152, pp. 213–219. Springer (2004)
16. May, A.: Using lll-reduction for solving RSA and factorization problems. In: Nguyen, P.Q., Vallée, B. (eds.) The LLL Algorithm - Survey and Applications, pp. 315–348. Information Security and Cryptography, Springer (2010)

17. May, A., Nowakowski, J., Sarkar, S.: Partial key exposure attack on short secret exponent CRT-RSA. In: Tibouchi, M., Wang, H. (eds.) Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part I. Lecture Notes in Computer Science, vol. 13090, pp. 99–129. Springer (2021)
18. Sarkar, S., Maitra, S.: Partial key exposure attack on CRT-RSA. In: Abdalla, M., Pointcheval, D., Fouque, P., Vergnaud, D. (eds.) Applied Cryptography and Network Security, 7th International Conference, ACNS 2009, Paris-Rocquencourt, France, June 2-5, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5536, pp. 473–484 (2009)
19. Takayasu, A., Kunihiro, N.: Partial key exposure attacks on CRT-RSA: better cryptanalysis to full size encryption exponents. In: Malkin, T., Kolesnikov, V., Lewko, A.B., Polychronakis, M. (eds.) Applied Cryptography and Network Security - 13th International Conference, ACNS 2015, New York, NY, USA, June 2-5, 2015, Revised Selected Papers. Lecture Notes in Computer Science, vol. 9092, pp. 518–537. Springer (2015)
20. Takayasu, A., Kunihiro, N.: Partial key exposure attacks on RSA: achieving the boneh-durfee bound. Theor. Comput. Sci. **761**, 51–77 (2019)
21. Takayasu, A., Lu, Y., Peng, L.: Small crt-exponent RSA revisited. J. Cryptol. **32**(4), 1337–1382 (2019)