Minkowski sum based lattice construction for multivariate simultaneous Coppersmith's technique and applications to RSA

Yoshinori Aono
aono@nict.go.jp

National Institute of Information and Communications Technology

Abstract. We investigate a lattice construction method for the Coppersmith technique for finding small solutions of a modular equation. We consider its variant for simultaneous equations and propose a method to construct a lattice by combining lattices for solving single equations. As applications, we consider a new RSA cryptanalyses. Our algorithm can factor an RSA modulus from $\ell \geq 2$ pairs of RSA public exponents with the common modulus corresponding to secret exponents smaller than $N^{(9\ell-5)/(12\ell+4)}$, which improves on the previously best known result by Sarkar and Maitra. For partial key exposure situation, we also can factor the modulus if $\beta - \delta/2 + 1/4 < (3\ell-1)(3\ell+1)$, where β and δ are bit-lengths / $\log N$ of the secret exponent and its exposed LSBs, respectively.

1 Introduction

Since the RSA cryptosystem [37] was proposed, its security has been intensively investigated. In particular, polynomial-time algorithms for recovering short secret exponents have been studied [40, 3]. There are two main strategies for recovering a secret exponent in this situation: The continued fraction algorithm was used in this approach [40, 24, 26, 21] and the Coppersmith technique based approach [3, 6, 15, 1]. We consider the latter technique.

Using the Coppersmith technique for finding small roots of a modular equation, Boneh and Durfee [3] proposed an algorithm for recovering a small secret exponent from the corresponding public key pair. Under several acceptable assumptions, the attack is guaranteed to work when the secret exponent is smaller than $N^{0.292}$.

Although the original Coppersmith technique was designed to treat a single modular equation, the method can be extended to multivariate simultaneous equations [14, 38, 39, 19]. Their approaches first construct a single multivariate modular equation whose solutions are also those of the simultaneous equations, and apply the standard Coppersmith technique. This may not be a better strategy from the viewpoint of lattice construction because it does not consider individual equations. May and Ritzenhofen [32] proposed an approach based on the Chinese remainder theorem to solve simultaneous univariate modular equations. In this paper, we study an extension of the Coppersmith technique that directly treats the original simultaneous multivariate equations. We expect that our algorithm will improve several lattice based attacks.

1.1 Contributions of this work

Minkowski sum based lattice construction: We propose a method to construct a lattice for the Coppersmith technique for simultaneous modular equations. We consider simultaneous equations such as $F_1(x_1, y) \equiv 0 \pmod{W_1}$ and $F_2(x_2, y) \equiv 0 \pmod{W_2}$. Assume that we have lattices spanned by the sets of polynomials $\{g_1^{(1)}, \ldots, g_{c_1}^{(1)}\}$ and $\{g_1^{(2)}, \ldots, g_{c_2}^{(2)}\}$ for the equations, respectively. Then, we propose the *Minkowski sum based lattice construction*, which is a method for generating a lattice basis for solving the simultaneous equations, as a set of polynomials of the form $\sum a_\lambda g_\lambda^{(1)} \cdot g_{\lambda'}^{(2)}$. Our method defines the range of suffixes (λ, λ') and the coefficients a_λ of the combination.

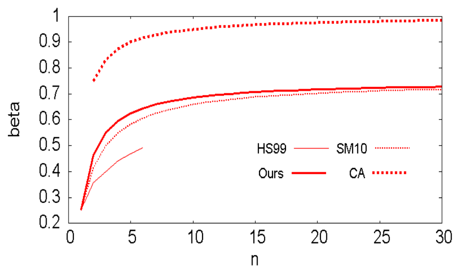
Cryptanalysis of multiple RSA short secret exponents: The above construction method can easily be extended to multivariate and multi-equation situations. By this, we improve the cryptanalysis of RSA with short secret exponents studied in [24, 21, 38, 39]. In this situation, the attacker has ℓ pairs of RSA public keys (e_k, N) with the common modulus, which correspond to secret exponents smaller than N^β for some $\beta \in (0,1)$. Then, we prove that the RSA modulus is efficiently factored if

$$\beta < \frac{9\ell - 5}{12\ell + 4}.$$

Here, we assumed that all e_k's are full-sized i.e., they have the same bit sizes. This improves on the previously known best result by Sarkar and Maitra [39], which achieved $\beta < (3\ell - 1)/(4\ell + 4)$. For large ℓ, both values converge to 3/4. Noting that Howgrave-Graham and Seifert [24] had given an extended version of Wiener's continued fraction attack and obtain the bound

$$\beta < \frac{(2\ell+1)\cdot 2^\ell - (2\ell+1)\binom{\ell}{\ell/2}}{(2\ell-2)\cdot 2^\ell + (4\ell+2)\binom{\ell}{\ell/2}} \text{ if } \ell \text{ is even, and } \beta < \frac{(2\ell+1)\cdot 2^\ell - 4\ell\binom{\ell-1}{(\ell-1)/2}}{(2\ell-2)\cdot 2^\ell + 8\ell\binom{\ell-1}{(\ell-1)/2}} \text{ if } \ell \text{ is odd.}$$

However, Hinek and Lam [21] observed that the attack does not recover the secret exponents if the bound exceeds 0.5, i.e., $\ell > 7$. Hence, our result is the best one. These results are compared in Figure 1. `HS99` is the result by Howgrave-Graham and Seifert [24] for $\ell \le 6$. `SM10` is Sarkar and Maitra [39]. `Ours` shows our result. `CA` indicates the heuristic bound by the counting argument in Section 4.1.



ℓ	2	3	4
HS99: [24]	0.357	0.400	0.441
SM10: [39]	0.416	0.500	0.550
Our result	0.464	0.550	0.596
CA	0.750	0.833	0.875

Fig. 1. Comparison of previous results.

Noting that our lattice can be improved by extracting its sublattice heuristically, whereas its theoretical analysis is not given. For $\ell = 14$, we showed that it achieves the bound at least 0.752, which exceeds 3/4. See Appendix D for detailed argument.

Extension for the partial key exposure situation: We then extend the attack to a situation studied in [4, 15], in which the attacker has ℓ tuples $(e_k, N, \widetilde{d_k})$ where e_k and N are RSA public keys, and each $\widetilde{d_k}$ is δn LSBs (least significant bits) of the corresponding secret exponent smaller than N^β. Then, we prove that the RSA modulus is efficiently factored if

$$\beta - \frac{\delta}{2} + \frac{1}{4} < \frac{3\ell - 1}{3\ell + 1}.$$

Computer experiments for verifying our attack: We perform our computer experiments of the applications for RSA and the partial key exposure situation. Our experiments work well. Interestingly, in the partial key exposure situation, the range of β and δ that we can factor N is slightly larger than that derived by theory.

Related works on automatic lattice construction for the Coppersmith technique: For a single modular equation, it have been widely studied. The first work by Coppersmith [10] gave a good lattice construction for any univariate modular equation. Recently, Aono et al. [2] has proven the optimality of this construction. Blömer and May [7] proposed a construction method for bivariate equations, and Jochemsz and May [25] improved this to a method for treating general multivariate equations. Another viewpoint was given by Kunihiro [28], who proposed a method for converting a lattice for an n-variable equation $f(x_1, \ldots, x_n) \equiv 0 \pmod W$ into a lattice for a new $(n+1)$-variable equation of the form $x_0 f(x_1, \ldots, x_n) + C \equiv 0 \pmod W$ where C is a constant. For simultaneous modular equations, May and Ritzenhofen [32] considered a

Chinese remainder theorem based approach. They proposed a method for constructing a lattice in the univariate case and gave an application to RSA. Recently, Ritzenhofen [36] improved this approach to multivariate simultaneous equations and proposed a lattice construction method for equations with the common modulus. However, the case for coprime moduli was not solved (see [36, Section 5.4]). We consider this problem.

Organization of this paper: Section 2 gives necessary definitions, lemmas, and an outline of the Coppersmith technique. In Section 3, we consider the Coppersmith technique for the simultaneous equations and propose our Minkowski sum based lattice construction. Sections 4 and 5 give applications to cryptanalysis of RSA Section 6 gives experimental results to verify our lattice construction. In Section 7, we suggest and discuss several open problems.

2 Preliminaries

Here we introduce necessary definitions and technical lemmas. For any positive integers a and b, let $[a]$ and $[a:b]$ be the set $\{1,\dots,n\}$ and $\{a,a+1,\dots,b-1,b\}$, respectively. For natural numbers x, A and N, the notation $|x| < A \pmod N$ means that $0 \le x < A$ or $N - A < x < N$ holds.

We use $\prec$ to denote the lexicographic order between integer tuples. For example, consider two 2-tuples, (i_1, i_2) and (i'_1, i'_2), then $(i_1, i_2) \prec (i'_1, i'_2)$ means that $i_1 < i'_1$ or $[i_1 = i'_1$ and $i_2 < i'_2]$ holds. We also use this to order monomials; e.g., $x_1^{i_1} x_2^{i_2} \prec x_1^{i'_1} x_2^{i'_2} \Leftrightarrow (i_1, i_2) \prec (i'_1, i'_2)$. Here, we neglect the coefficients. These notations are used for general n-tuples and n-variable monomials. We use $x_1, x_2, \dots, x_{n-1}$ and y to denote the variables, and fix the priority of variables as $y \prec x_{n-1} \prec \cdots \prec x_1$ to order the n-variable monomials. For example, consider four variables, x_1, x_2, x_3, y, and monomials $3x_2^2x_3$ and $x_1^2x_2^3y$. Then, $3x_2^2x_3 \prec x_1^2x_2^3y$ holds since the corresponding tuples are $(0,2,1,0)$ and $(2,3,0,1)$, respectively. Note that for any integer tuples T_1, T_2, S_1, S_2 of the same dimension, $T_1 \prec S_1$ and $T_2 \prec S_2$ implies that $T_1 + T_2 \prec S_1 + S_2$.

With respect to the above order, we can define the maximum element in a polynomial $f(x_1, \dots, x_\ell, y)$. Let $ax_1^{i_1} \cdots x_\ell^{i_\ell} y^j$ be the non-zero maximum monomial in f. Then, we call it the *head term* of f and denote it by $\mathrm{HT}(f)$. We also call a, $x_1^{i_1} \cdots x_\ell^{i_\ell} y^j$ and $(i_1, \dots, i_\ell, j)$ *head coefficient*, *head monomial* and *head index*, and denote them by $\mathrm{HC}(f)$, $\mathrm{HM}(f)$ and $\mathrm{HI}(f)$, respectively.

Minkowski sum: Let A and B be finite subsets of $\mathbb{Z}^n$, then their Minkowski sum is defined by

$$A \boxplus B = \{(a_1 + b_1, \dots, a_n + b_n) : (a_1, \dots, a_n) \in A, (b_1, \dots, b_n) \in B\}.$$

Note that the sum of three or more sets is similarly defined.

2.1 Overview of the Coppersmith technique

We introduce the Coppersmith technique [10, 11] with necessary definitions and technical lemmas. Our formulation is due to Howgrave-Graham [20] and Aono et al. [2].

Fix a polynomial $F(x,y) \in \mathbb{Z}[x,y]$ and $X, Y, W \in \mathbb{N}$. Then consider the problem of finding all integer solutions of

$$F(x,y) \equiv 0 \pmod W \quad (1)$$

within the range of $|x| < X$ and $|y| < Y$. While this problem is generally not easy, the Coppersmith technique efficiently solves it if X and Y are much smaller than W.

The Coppersmith technique first fix an integer $m \ge 2$ and consider a set L of polynomials $g(x,y) \in \mathbb{Z}[x,y]$ satisfying

$$\forall x, y \in \mathbb{Z}\ [F(x,y) \equiv 0 \pmod W \Rightarrow g(x,y) \equiv 0 \pmod{W^m}]. \quad (2)$$

Note that L forms a lattice, i.e., it can easily see that $g_1, g_2 \in L \Rightarrow g_1 - g_2 \in L$. Next, find polynomials $g(x,y) \in L$ satisfying

$$\forall x, y \in \mathbb{Z}, |x| < X, |y| < Y\ [g(x,y) \equiv 0 \pmod{W^m} \Rightarrow g(x,y) = 0]. \quad (3)$$

Suppose two algebraically independent polynomials are found, then the original equation (1) can be converted to simultaneous equations over integers, which are easily solved by the resultant technique [18] or the Gröbner basis technique [9]. As we explain below, a polynomial with small coefficients satisfies (3). Our tasks are to construct a polynomial lattice L, and to find such polynomials in L.

Many algorithms to find small elements in a lattice exist; e.g., the LLL algorithm [29] is widely used. Unfortunately, most of them are designed for treating lattices in Euclidean spaces $\mathbb{R}^n$ w.r.t. the standard Euclidean norms. To use them as a subroutine, a polynomial lattice needs to be converted.

Converting polynomials to vectors: For a polynomial $g(x,y) = \sum_{i,j} a_{i,j}x^i y^j$ and parameters X and Y, define the *vectorization* of the polynomial by

$$\mathcal{V}(g; X, Y) = (a_{0,0}, a_{1,0}X, \ldots, a_{i_w,j_w}X^{i_w}Y^{i_w}).$$

Thus, it maps each term $a_{i,j}x^i y^j$ to each coordinate $a_{i,j}X^iY^j$, respectively. It is a linear mapping with respect to g. Note that the sequence of tuples $\{(i_k, j_k)\}_{k=1}^{w}$ is taken so that it covers all non-zero terms in $g(x,y)$. We define the *polynomial norm* w.r.t. the parameters X, Y by $|\mathcal{V}(g; X, Y)|$. W.r.t. this norm, the following lemma holds.

Lemma 1 (Howgrave-Graham [20], generalized in [25]) *Fix $X, Y, W \in \mathbb{N}$. Let $g(x,y) \in \mathbb{Z}[x,y]$ be a polynomial consisting of w non-zero terms, and $|\mathcal{V}(g; X, Y)| < W/\sqrt{w}$ holds. Then we have*

$$\forall x, y \in \mathbb{Z}, |x| < X, |y| < Y \;\; [g(x,y) \equiv 0 \pmod{W} \Leftrightarrow g(x,y) = 0].$$

Hence, if a polynomial lattice L is given, our task is to find independent polynomials satisfying the above lemma, which is performed by finding short vectors in a Euclidean lattice converted from L using certain parameters. To achieve this, we use a lattice reduction algorithm.

Euclidean Lattices: Consider a sequence of linearly independent vectors $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_c\}$ in $\mathbb{Z}^{\tilde{c}}$ where $\tilde{c} \geq c$. Then, the *Euclidean lattice* spanned by them is defined by

$$L(\mathbf{B}) = \{a_1\mathbf{b}_1 + \cdots + a_c\mathbf{b}_c \;:\; a_k \in \mathbb{Z} \text{ for } k \in [c]\}.$$

We call $\mathbf{b}_1, \ldots, \mathbf{b}_c$ the basis vectors. Following many papers, we assume that a lattice is represented by its basis vectors.

To find short vectors in a lattice, we use the LLL algorithm [29] which computes an LLL-reduced basis from a given basis. The following theorem bounds the lengths of first vectors in such bases.

Theorem 1 *[6] Let L be a Euclidean lattice and $\mathbf{v}_1, \ldots, \mathbf{v}_c$ be its LLL-reduced basis. Then, the following inequality holds for $k \in [c]$.*

$$||\mathbf{v}_k|| \leq 2^{\{(c(c-1)+(k-1)(k-2)\}/4(c-k+1)} |\det(L)|^{1/(c-k+1)} \tag{4}$$

Here, $\det(L)$ is the *lattice determinant* that is defined by using the Gram-Schmidt orthogonal basis $\mathbf{v}_1^*, \ldots, \mathbf{v}_c^*$ as $\det(L) = \prod_{i=1}^{c} ||\mathbf{v}_i^*||$.

Polynomial Lattices: Let $\mathbf{G} = \{g_1, \ldots, g_c\}$ be a sequence of linearly independent polynomials in $\mathbb{Z}[x,y]$. Then, the *polynomial lattice* spanned by them is defined by

$$L(\mathbf{G}) = L(g_1, \ldots, g_c) = \{a_1 g_1 + \cdots + a_c g_c \;:\; a_i \in \mathbb{Z} \text{ for } k \in [c]\}.$$

We also consider the vectorization of polynomial lattices; i.e., for a basis $\mathbf{G} = \{g_1, \ldots, g_c\}$, consider their vectorization $\mathcal{V}(g_1; X, Y), \ldots, \mathcal{V}(g_c; X, Y)$ w.r.t. parameters X and Y. Here, the tuple sequence is assumed to be fixed. Then, define the vectorization of $L(\mathbf{G})$ by the Euclidean lattice spanned by these vectors, and let it be $L(\mathbf{G}; X, Y)$. We use $\det(\mathbf{G}; X, Y)$ to denote the determinant of $L(\mathbf{G}; X, Y)$.

Outline and a working condition for the Coppersmith technique: For fixed X and Y, suppose we have a polynomial lattice $L(\mathbf{G})$ spanned by c polynomials satisfying (2), and it holds that

$$2^{c/4}\det(\mathbf{G};X,Y)^{1/c} < N^m/w. \tag{5}$$

Here, w is the length of tuple sequence used at vectorization, which is equal to the Euclidean dimension of $L(\mathbf{G};X,Y)$, and bounds upper the number of terms of any polynomials in $L(\mathbf{G})$. Then, compute the LLL-reduced basis of $L(\mathbf{G};X,Y)$. By Theorem 1, the first two vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ in the reduced basis are shorter than N^m/w. Hence, the corresponding polynomials, i.e., $h_k(x,y)$ satisfying $\mathbf{v}_k = \mathcal{V}(h_k;X,Y)$ for $k=1,2$, also satisfy $|\mathcal{V}(h_k;X,Y)| \le N^m/w$. Thus, by Lemma 1, these polynomials satisfy

$$\forall x,y \in \mathbb{Z}, |x|<X, |y|<Y\ [F(x,y)\equiv 0 \pmod{W} \Rightarrow h_k(x,y)=0].$$

Finally, finding small integer solutions of $h_1(x,y)=h_2(x,y)=0$, we obtain the desired solutions.

As in many previous works, we regard the following simplified condition as a working condition.

$$\det(\mathbf{G};X,Y)^{1/c} < N^m \tag{6}$$

In many applications, the crucial problem is to construct a lattice $\mathbf{G}$ satisfying (6) for X and Y as large as possible.

The algebraic independence of polynomials $h_k(x,y)$ is necessary to solve the final simultaneous equations over the integers. Unfortunately, this is generally not guaranteed. In this paper, again following previous works, we assume this algebraic independence and justify it by computer experiments.

3 Coppersmith technique for simultaneous equations

We consider a variant of the Coppersmith technique for the simultaneous equations, and propose a new method to construct polynomial lattices. For readability, we consider the following three variable simultaneous equations with two equations having the shared variable y:

$$\begin{matrix} F_1(x_1,y)\equiv 0 \pmod{W_1} \\ F_2(x_2,y)\equiv 0 \pmod{W_2} \end{matrix} \tag{7}$$

Here, if no variable is shared, the simultaneous equations have no meaning.

Our objective is to find all integer solutions within the range of $|x_1|<X_1$, $|x_2|<X_2$ and $|y|<Y$. Fix the above equations, given ranges, and parameters c and m. Then we consider a lattice consisting of three variable polynomials $g_i(x_1,x_2,y)$ such that satisfies

$$\forall x_1,x_2,y\in\mathbb{Z}, \left[\begin{Bmatrix} F_1(x_1,y)\equiv 0 \pmod{W_1} \\ F_2(x_2,y)\equiv 0 \pmod{W_2} \end{Bmatrix} \Rightarrow g_i(x_1,x_2,y)\equiv 0 \pmod{(W_1W_2)^m}\right]. \tag{8}$$

For a lattice $L(\mathbf{G})$ with basis $\mathbf{G}=\{g_1,\ldots,g_c\}$, compute the LLL-reduced basis of $L(\mathbf{G};X_1,X_2,Y)$. By the same argument as that in Section 2.1, we can prove the technique works if

$$\det(\mathbf{G};X_1,X_2,Y)^{1/c} < (W_1W_2)^m.$$

Hence, the problem is also finding the means of constructing better polynomial lattices.

3.1 Minkowski sum based lattice construction

We give a method for constructing a lattice for the simultaneous equations (7), by combining lattices for solving single equations.

For $k=1,2$, let $L(\mathbf{G}_k)$ be a polynomial lattice for solving $F_k(x_k,y)\equiv 0 \pmod{W_k}$ and its basis be $\mathbf{G}_k=\{g_1^{(k)},\ldots,g_{c_k}^{(k)}\}$. Here we assume that the parameter m is fixed. Then for any $\ell_1\in[c_1]$ and $\ell_2\in[c_2]$, the polynomial $g_{\ell_1}^{(1)}\cdot g_{\ell_2}^{(2)}$ satisfies (8). Hence, the set

$$\mathcal{A}=\left\{\sum_{\ell_1\in[c_1],\ell_2\in[c_2]} a_{\ell_1,\ell_2} g_{\ell_1}^{(1)} g_{\ell_2}^{(2)} : a_{\ell_1,\ell_2}\in\mathbb{Z}\right\}$$

forms a polynomial lattice for solving the simultaneous equations. Unfortunately, since the polynomials $\{g^{(1)}_{\ell_1} g^{(2)}_{\ell_2}\}_{\ell_1,\ell_2}$ are not generally independent over the integers, it cannot explicitly obtain the basis of $\mathcal{A}$ and its determinant. Instead, we consider a sublattice of $\mathcal{A}$ and define its basis by using the Minkowski sum of indices.

We can assume that each basis $\mathbf{G}_k$ has a strictly increasing degree order, i.e., $\mathrm{HM}(g^{(k)}_1) \prec \cdots \prec \mathrm{HM}(g^{(k)}_{c_k})$ holds for $k = 1, 2$. If this is not true, an equivalent basis having this property can be computed by Gaussian elimination; see [2]. Then, for each k, consider the set of indices $I_k = \{\mathrm{HI}(g^{(k)}_\ell) : \ell \in [c_k]\} \subset \mathbb{Z}^3$ and let their Minkowski sum be I_+. Noting that the elements of I_1 and I_2 have the form $(i_1, 0, j)$ and $(0, i_2, j)$, respectively. For every $(i_1, i_2, j) \in I_+$, define the polynomial $g^+_{i_1,i_2,j}$ to be

$$g^+_{i_1,i_2,j} = \sum_{(*)} a_\lambda g^{(1)}_\lambda g^{(2)}_{\lambda'}. \tag{9}$$

Here, the range of sum $(*)$ is over all suffix pairs (λ, λ') satisfying

$$\mathrm{HM}(g^{(1)}_\lambda g^{(2)}_{\lambda'}) = x_1^{i_1} x_2^{i_2} y^j$$

and the coefficients a_λ are defined so that

$$\mathrm{HC}(g^+_{i_1,i_2,j}) = \mathrm{GCD}_{(*)}\,(\mathrm{HC}(g^{(1)}_\lambda g^{(2)}_{\lambda'})), \tag{10}$$

that is, the greatest common divisor of all head coefficients within the range. It is easy to see that the polynomial satisfies (8). We define the polynomial basis by $\mathbf{G}_+ = \{g^+_{(i_1,i_2,j)} : (i_1, i_2, j) \in I_+\}$. Here, it is clear that the basis polynomials are linearly independent since the head monomials are distinct. We call the polynomial lattice $L(\mathbf{G}_+)$ the *Minkowski sum lattice* of $L(\mathbf{G}_1)$ and $L(\mathbf{G}_2)$. Clearly, $L(\mathbf{G}_+) \subset \mathcal{A}$ holds.

The basic strategy of this construction is to minimize the head coefficient of $g^+_{i_1,i_2,j}$ over all the possible integer combinations. It can be expected that the determinant of the combined lattice is reduced. Note that a combination of a_λ that attains (10) is generally not unique. Hence, care needs to be taken regarding the determinant if the lattice is not triangular. If the lattice is lower triangular, the determinant, which is computed by $\prod |\mathrm{HC}(g^+_{i_1,i_2,j})| X_1^{i_1} X_2^{i_2} Y^j$, is not changed for any allowed combination of a_λ.

Comparison with previous strategy: Many previous works first consider the single equation by multiplying both sides in (7) as

$$F_1(x_1, y) F_2(x_2, y) \equiv 0 \pmod{W_1 W_2}.$$

Then consider a lattice of polynomials $g_i(x_1, x_2, y)$ satisfying

$$\forall x_1, x_2, y \in \mathbb{Z}, [F_1(x_1, y) F_2(x_2, y) \equiv 0 \pmod{W_1 W_2} \Rightarrow g_i(x_1, x_2, y) \equiv 0 \pmod{(W_1 W_2)^m}]$$

using the lattice construction strategy by Jochemsz and May [25], that only considers polynomials of the form $x_1^{i_1} x_2^{i_2} y^j (F_1(x_1, y) F_2(x_2, y))^t (W_1 W_2)^{m-t}$, whose variety of polynomial selection is clearly smaller than our Minkowski sum construction. We provide a small example for our attack on RSA with the short secret exponents in Appendix B.

3.2 Minkowski sum of lower triangular lattices

Suppose the lattices for single equations are lower triangular, that is, there exist sequences of tuples $\{(i_1(\ell), j_1(\ell))\}_{\ell=1}^{c_1}$ and $\{(i_2(\ell), j_2(\ell))\}_{\ell=1}^{c_2}$, the polynomials in bases $\mathbf{G}_k$ can be written as

$$g^{(1)}_\ell = \sum_{\ell'=1}^{\ell} a_{\ell,\ell'} x_1^{i_1(\ell')} y^{j_1(\ell')} \text{ and } g^{(2)}_\ell = \sum_{\ell'=1}^{\ell} b_{\ell,\ell'} x_2^{i_2(\ell')} y^{j_2(\ell')}, \text{ where } a_{\ell,\ell} \neq 0 \text{ and } b_{\ell,\ell} \neq 0.$$

In this case, w.r.t. the above sequences of tuples, the Euclidean lattices $L(\mathbf{G}_k; X_k, Y)$ are lower triangular. We show that the Minkowski sum lattice of them is also lower triangular.

Theorem 2 *For $k = 1, 2$, assume that the polynomial lattice basis $\mathbf{G}_k = \{g_1^{(k)}, \dots, g_{c_k}^{(k)}\}$ has a strictly increasing degree order, and that they are lower triangular. Then the Minkowski sum lattice $L(\mathbf{G}_+)$ is also lower triangular.*

Proof. Recall that $I_k = \{\mathrm{HI}(g_\ell^{(k)}) : \ell \in [c_k]\} \subset \mathbb{Z}^3$ and $I_+ = I_1 \boxplus I_2$. Fix any $(i_1, i_2, j) \in I_+$. We prove that every $g^+_{i_1,i_2,j}$ can be written as the form of

$$g^+_{i_1,i_2,j} = a x_1^{i_1} x_2^{i_2} y^j + (\text{terms whose indices are in } I_+ \text{ and smaller than } x_1^{i_1} x_2^{i_2} y^j).$$

By construction, it suffices to show that any $g_\lambda^{(1)} g_{\lambda'}^{(2)}$ in (9) can be written in this form. Fix a suffix pair (λ, λ') and consider the monomial expansion of $g_\lambda^{(1)} g_{\lambda'}^{(2)}$. Then, any term in this polynomial is a sum of the product of terms in $g_\lambda^{(1)}$ and $g_{\lambda'}^{(2)}$. Let them be $A x_1^{\bar{i}_1} y^{\bar{j}_1}$ and $B x_2^{\bar{i}_2} y^{\bar{j}_2}$; i.e., $g_\lambda^{(1)} g_{\lambda'}^{(2)}$ can be written as a sum of $AB x_1^{\bar{i}_1} x_2^{\bar{i}_2} y^{\bar{j}_1 + \bar{j}_2}$.

Since $L(\mathbf{G}_k)$ are lower triangular, $(\bar{i}_1, 0, \bar{j}_1) \in I_1$ and $(0, \bar{i}_2, \bar{j}_2) \in I_2$ hold. Thus, $(\bar{i}_1, \bar{i}_2, \bar{j}_1 + \bar{j}_2) \in I_+$. By construction, $(\bar{i}_1, 0, \bar{j}_1) \preceq \mathrm{HI}(g_\lambda^{(1)})$ and $(0, \bar{i}_2, \bar{j}_2) \preceq \mathrm{HI}(g_{\lambda'}^{(2)})$. Hence, we have $(\bar{i}_1, \bar{i}_2, \bar{j}_1 + \bar{j}_2) \preceq \mathrm{HI}(g_\lambda^{(1)}) + \mathrm{HI}(g_{\lambda'}^{(2)}) = \mathrm{HI}(g^+_{i_1,i_2,j})$. □

Minkowski sum of three of more lattices: As with the situations of two lattices, the Minkowski sum construction can be extended to three or more lattices. Moreover, we can show that the Minkowski sum lattice of lower triangular lattices is also lower triangular. The detailed construction, theorem, and its proof are left to Appendix A.

4 Cryptanalysis of RSA with short secret exponents

As an application of our Minkowski sum lattice construction, we analyze the RSA with multiple short secret exponents with a common modulus.

Notations: We use the standard notations for the RSA cryptography. That is, p and q are large primes, and let their product be the RSA modulus N. e and d are used to denote the public exponent and secret exponent, respectively. The basic relation $ed \equiv 1 \pmod{\varphi(N)}$ holds. Following [3], we assume that $e \approx N$ and $p + q < 3N^{0.5}$.

In this section, we consider the situation in which the attacker has ℓ pairs of public keys with a common modulus, let them be $(e_1, N), \dots, (e_\ell, N)$, which correspond to small secret exponents satisfying $d_1, \dots, d_\ell < N^\beta$ for some $\beta \in (0, 1)$. For simplicity, we assume that e_i and e_j are coprime to each other for $i \neq j$.

4.1 RSA equation and its limit by a counting argument

Following the work of Sarkar and Maitra [38, 39] (see Boneh and Durfee [3] for deriving single equation), it can prove that the simultaneous equations

$$\begin{aligned} F_1(x_1, y) &= -1 + x_1(y + N) \equiv 0 \pmod{e_1} \\ &\vdots \\ F_\ell(x_\ell, y) &= -1 + x_\ell(y + N) \equiv 0 \pmod{e_\ell} \end{aligned} \tag{11}$$

have a small solution $(x_1, \dots, x_\ell, y)$ satisfying

$$|x_k| < N^\beta, \text{ for } k \in [\ell] \text{ and } |y| < 3N^{0.5}, \tag{12}$$

by which we can recover the secret exponents. Hence, our objective here is to find this solution by the Coppersmith technique.

On the other hand, if β is not small, the solution within the range is not unique. In this situation, the number of solutions becomes exponential in $\log N$; thus, no polynomial-time algorithm exists. We derive a heuristic condition in which a polynomial-time algorithm exists.

For any y' such that $y'+N$ is coprime to all e_k, setting $x'_k = (y'+N)^{-1} \pmod{e_l}$, the tuple $(x'_1, \ldots, x'_\ell, y')$ is a solution of (11). Unfortunately, this solution cannot be used to recover the secret keys since it does not generally satisfy (12).

Following the argument in [1], we consider a heuristic condition for β so that a found solution satisfying (12) is expected to be usable for recovering the secret exponents. Assume that the solutions of (11) are random numbers on $\{1, \ldots, N\}^{\ell+1}$. Since the number of solution tuples is smaller than N, we expect the number of solutions within the range (12) to be smaller than $N \cdot (2N^\beta)^\ell \times (6N^{0.5})/N^{\ell+1} \approx N^{0.5+\ell(\beta-1)}$. Thus, if this is smaller than one, the solution within the range is expected to be the desired one. From this observation, we set the following heuristic assumption.

Heuristic assumption: For a natural number ℓ, assume that

$$\beta < \frac{\ell - 0.5}{\ell}. \tag{13}$$

Then, within the range of (12), the equation (11) has only one solution $(x'_1, \ldots, x'_\ell, y')$ by which can recover the corresponding secret keys d_k.

4.2 Our lattice construction and bound

Here we give our polynomial lattice to solve the simultaneous equations (11) and a new security analysis of RSA. As mentioned in Section 3.1, assume that lattices for solving single equation $F_k(x_k, y) = -1 + x_k(y+N) \equiv 0 \pmod{e_k}$ are given. We follow the work of Boneh and Durfee [3], and employ their simple lower triangular lattice: fix an integer $m \geq 2$ and set

$$g_{i,j}^{(k)}(x_k, y) = x_k^{i-j} F_k(x_k, y) e_k^{m-j} \text{ and } \mathbf{G}_k = \{g_{i,j}^{(k)} : (i,j) \in \mathbb{Z}^2, 0 \leq j \leq i \leq m\} \tag{14}$$

for $k = 1, \ldots, \ell$. It is clear that $g_{i,j}^{(k)}(x_k, y)$ satisfies (2) w.r.t. $F_k(x_k, y) \equiv 0 \pmod{e_k}$ and m.

For each k, ordering its basis in the lexicographic order in suffixes (i,j), the polynomial sequence has strictly increasing order since $\mathrm{HM}(g_{i,j}^{(k)}) = x_k^i y^j$ and $\mathrm{HI}(g_{i,j}^{(k)}) = (0, \ldots, 0, i, 0, \ldots, 0, j) \in \mathbb{Z}^{\ell+1}$ (the k-th and $\ell+1$-th coordinates are i and j, respectively). As shown in [3], the lattice $L(\mathbf{G}_k; X_k, Y)$ is lower triangular. Thus these bases satisfy the assumption of Theorem 3 in Appendix A, and the Minkowski sum lattice $L(\mathbf{G}_+)$ is also lower triangular.

We explicitly give the Minkowski sum lattice. The index set corresponding to $\mathbf{G}_k$ is given by $I_k = \{(0, \ldots, 0, i, \ldots, 0, j) : (i,j) \in \mathbb{Z},\ 0 \leq j \leq i \leq m\}$ and their Minkowski sum is

$$I_+ = I_1 \boxplus \cdots \boxplus I_\ell = \{(i_1, \ldots, i_\ell, j) : 0 \leq i_1, \ldots, i_\ell \leq m \text{ and } 0 \leq j \leq i_1 + \cdots + i_\ell\}.$$

Following Appendix A, for each $(i_1, \ldots, i_\ell, j) \in I_+$, construct a polynomial by

$$g_{i_1,\ldots,i_\ell,j} = \sum_{j_1,\ldots,j_\ell} a_{j_1,\ldots,j_\ell} \cdot g_{i_1,j_1}^{(1)} g_{i_2,j_2}^{(2)} \cdots g_{i_\ell,j_\ell}^{(\ell)}.$$

Following (9), the sum is over indices such that $\mathrm{HM}(g_{i_1,j_1}^{(1)} g_{i_2,j_2}^{(2)} \cdots g_{i_\ell,j_\ell}^{(\ell)}) = x_1^{i_1} \cdots x_\ell^{i_\ell} y^j$. In this situation, i_k are fixed, and $(j_1, \ldots, j_\ell)$ moves over all integer tuples subject to $0 \leq j_k \leq i_k$ and $j_1 + \cdots + j_\ell = j$. Next we consider the coefficients; again as mentioned in Section 3.1, the coefficients $a_{j_1,\ldots,j_\ell}$ are selected so that

$$\mathrm{HC}(g_{i_1,\ldots,i_\ell,j}) = \mathop{\mathrm{GCD}}_{j_1,\ldots,j_\ell} \left(\mathrm{HC}(g_{i_1,j_1}^{(1)} g_{i_2,j_2}^{(2)} \cdots g_{i_\ell,j_\ell}^{(\ell)}) \right).$$

Note that $\mathrm{HC}(g_{i_1,j_1}^{(1)} g_{i_2,j_2}^{(2)} \cdots g_{i_\ell,j_\ell}^{(\ell)}) = e_1^{m-j_1} \cdots e_\ell^{m-j_\ell}$. Since j_k can move from zero to $\min(i_k, j)$, the greatest common divisor is $e_1^{m-\min(i_1,j)} \cdots e_\ell^{m-\min(i_\ell,j)}$. Thus, we can take $a_{j_1,\ldots,j_\ell}$ so that the head coefficient of $g_{i_1,\ldots,i_\ell,j}$ is this value.

Then, we set the Minkowski sum lattice by $\mathbf{G}_+ = \{g_{i_1,\ldots,i_\ell,j} : (i_1,\ldots,i_\ell,j) \in I_+\}$ and the order is the lexicographic order of suffixes. By Theorem 3, the converted lattice $L(\mathbf{G}_+; X_1,\ldots,X_\ell,Y)$ is lower triangular. The diagonal element corresponding to $(i_1,\ldots,i_\ell,j)$ is

$$\mathrm{HC}(g_{i_1,\ldots,i_\ell,j}) \times X_1^{i_1}\cdots X_\ell^{i_\ell} Y^j = e_1^{m-\min(i_1,j)}\cdots e_\ell^{m-\min(i_\ell,j)} X_1^{i_1}\cdots X_\ell^{i_\ell} Y^j.$$

Therefore, the determinant is

$$\det(\mathbf{G}_+; X_1,\ldots,X_\ell,Y) = \prod_{(i_1,\ldots,i_\ell,j)\in I_+} \left[e_1^{m-\min(i_1,j)}\cdots e_\ell^{m-\min(i_\ell,j)} X_1^{i_1}\cdots X_\ell^{i_\ell} Y^j\right].$$

As with the same argument in Section 2.1, the Coppersmith technique works if

$$\det(\mathbf{G}_+; X_1,\ldots,X_\ell,Y)^{1/|I_+|} < (e_1\cdots e_\ell)^m.$$

Here, $|I_+|$ denotes the number of elements in I_+. Using approximations $e_k \approx N$ for $k \in [\ell]$, $X_1 = \cdots = X_\ell = N^\beta$ and $Y \approx N^{0.5}$, the condition can be rewritten as

$$\sum_{(i_1,\ldots,i_\ell,j)\in I_+} \left[0.5j + (i_1+\cdots+i_\ell)\beta - \sum_{k=1}^{\ell}\min(i_k,j)\right] < 0. \tag{15}$$

By computing the left-hand side (see Appendix C), we derive the condition

$$\left(-\frac{3}{16}\ell^2 + \frac{5}{48}\ell + \left(\frac{\ell^2}{4} + \frac{\ell}{12}\right)\beta\right) m^{\ell+2} + o(m^{\ell+2}) < 0.$$

Thus, when m is sufficiently large, this condition is

$$\beta < \frac{9\ell-5}{12\ell+4}. \tag{16}$$

Appendix B gives a small example of this construction for $\ell = 2$ and $m = 1$.

Heuristic improvement of lattice: Suppose $\beta > 0.5$. We can construct a new lattice by removing polynomials whose indexes satisfy both $j > \max\{i_1,\ldots,i_\ell\}$ and $0.5j + (i_1+\cdots+i_\ell)\beta - \sum_{k=1}^{\ell}\min(i_k,j) > 0$, which have negative contributions in the sigma (15). It can be shown that the new lattice is also lower triangular. However, we have never derived an explicit formula of the working condition. Appendix D provides the detailed argument.

5 Application to partial key exposure attack on RSA

Next, we consider the partial key exposure attack on RSA. Assume that the attacker has ℓ pairs of RSA public keys $(e_1,N),\ldots,(e_\ell,N)$, and δn LSBs of the corresponding d_k. Moreover, each d_k is assumed to be smaller than N^β.

Let $M = 2^{\lfloor \delta n\rfloor}$ and the exposed parts be $\widetilde{d_k}$ for $k \in [\ell]$. Then, following the derivation of the single equation for the situation that single $(e, N, \widetilde{d})$ is given [15], we consider the simultaneous equations

$$\begin{aligned} F_1(x_1,y) &= e_1\widetilde{d}_1 - 1 + x_1(y+N) \equiv 0 \pmod{e_1 M}\\ &\vdots\\ F_\ell(x_\ell,y) &= e_\ell\widetilde{d}_\ell - 1 + x_\ell(y+N) \equiv 0 \pmod{e_\ell M}. \end{aligned} \tag{17}$$

By the counting argument in Section 4.1, we can assume that if $\beta - \delta < (\ell - 0.5)/\ell$, then the solution satisfying $|x_1|,\ldots,|x_\ell| < N^\beta$ and $|y| < 3N^{0.5}$ is unique, and it can be used to factor N.

The basic lattice construction is the same as in the above section; i.e., we let

$$g_{i,j}^{(k)} = x_k^{i-j}(F_k(x_k,y))^j (e_k M)^{m-j} \text{ and } \mathbf{G}_k = \{g_{i,j}^{(k)} : (i,j)\in\mathbb{Z}^2, 0 \le j \le i \le m\}.$$

Note that only the constant terms and moduli differ between (11) and (17). Thus, $L(\mathbf{G}_k)$ for $k \in [\ell]$ and their Minkowski sum $L(\mathbf{G}_+)$ are also lower triangular. Moreover, the set of indices $I_1, \ldots, I_\ell$ and their Minkowski sum I_+ are also the same as in Section 4.2.

For each $(i_1, \ldots, i_\ell, j) \in I_+$, we give the polynomial $g_{i_1,\ldots,i_\ell,j}$. First note that

$$\mathrm{HC}(g^{(1)}_{i_1,j_1} g^{(2)}_{i_2,j_2} \cdots g^{(\ell)}_{i_\ell,j_\ell}) = e_1^{m-j_1} \cdots e_\ell^{m-j_\ell} M^{\ell m - j_1 - \cdots - j_\ell}.$$

Thus, as Section 4.2, each j_k can move from zero to $\min(i_k, j)$, and we can take the coefficients in (9) so that

$$\mathrm{HT}(g_{i_1,\ldots,i_\ell,j}) = e_1^{m-\min(i_1,j)} \cdots e_\ell^{m-\min(i_\ell,j)} M^{\ell m - j} x_1^{i_1} \cdots x_\ell^{i_\ell} y^j.$$

Hence, we have

$$\det(\mathbf{G}_+; X_1, \ldots, X_\ell, Y) = \prod_{(i_1,\ldots,i_\ell,j)\in I_+} \left[e_1^{m-\min(i_1,j)} \cdots e_\ell^{m-\min(i_\ell,j)} \times M^{\ell m - j} X_1^{i_1} \cdots X_\ell^{i_\ell} Y^j \right].$$

By the approximations $e_k \approx N$, $X_k = N^\beta$, $Y \approx N^{0.5}$ and $M \approx N^\delta$, the attack works if

$$\sum_{(i_1,\ldots,i_\ell,j)\in I} \left[(0.5-\delta)j + (i_1 + \cdots + i_\ell)\beta - \sum_{k=1}^{\ell} \min(i_k, j) \right] < 0. \tag{18}$$

When m becomes large, the condition is

$$\beta - \frac{\delta}{2} + \frac{1}{4} < \frac{3\ell - 1}{3\ell + 1}. \tag{19}$$

Here, the detailed computation to derive (19) from (18) is given in Appendix C.

6 Computer experiments of our RSA cryptanalysis

Experimental environment: The experiments were conducted on a standard workstation with 16GB of RAM and two Intel Xeon X5675 processors running at 3.07GHz. We wrote our experimental program in the C++ language using the following libraries. To compute the LLL reduced basis, we used Shoup's NTL library [34] version 5.5.2 compiled with the GMP library [17] version 5.0.4. The polynomial computation was performed using the GiNaC library [16] version 1.6.2. We compiled our source code using `g++` version 4.5.4 with the `-O3` option. We also used `Maple 15` to compute the resultant under modulo prime in the final step of the experiments. We performed our experiments on the Windows 7 platform and ran our program in a single thread.

6.1 Experiments for short RSA secret exponents

Figure 2 shows the procedure of our computer experiments. In Step 1, "pseudoprime" means an odd integer that passes the Euler-Jacobi primality testing for bases 2, 3, 5 and 7. In Step 2, we use the command `LLL_XD(L,0.99,0,0,1)`. In the second-half of Step 4, we first generate a random $0.5n$ bit prime number P. Then, we erase the variable x_1 by computing $r_k = \mathrm{Res}_{x_1}(h_1, h_k) \bmod P$ for $k = 2, \ldots, \ell+1$, and next we compute $\mathrm{Res}_{x_2}(r_2, r_k) \bmod P$ for $k = 3, \ldots, \ell+1$ modulo P, and repeat this process. Finally, we obtain a univariate polynomial $R(y)$ and check $R(\bar{y}) \equiv 0 \pmod{P}$. We repeat this check for three distinct prime numbers via `Maple 15`.

Parameters and results: Note first that if m and ℓ are fixed, condition (15) is written in a linear function w.r.t. β, and the maximum β satisfying the inequality is easily computed. This β is a theoretical bound when N becomes large along with neglecting several factors as described in Section 2.1. For each m and ℓ we compute the maximum β and the dimension of lattice. They are shown in Table 1. The column "limit" indicates the right-hand side of (16).

Parameters: ℓ: Number of RSA keys; n: RSA bit length; β: ratio of secret keys to n

Step 1: (Generate a sample RSA instance) Randomly choose $\lfloor n/2 \rfloor$-bit pseudoprimes p and q, and let $N = pq$. Randomly choose ℓ $\lfloor \beta n \rfloor$-bit odd integers $d_1, \ldots, d_\ell$ such that $\mathrm{GCD}(d_k, (p-1)(q-1)) = 1$ for all $k \in [\ell]$. Compute the corresponding e_k by $d_k^{-1} \pmod{(p-1)(q-1)}$. For each $k \in [\ell]$, define the RSA polynomial $f_k(x_k, y) = -1 + x_k(N + y)$ and let the solutions $\bar{x_k} = (1 - e_k d_k)/(p-1)(q-1)$ and $\bar{y} = 1 - p - q$ corresponding to the secret keys.

Step 2: Set the bounds $X_k = \lfloor N^\beta \rfloor$ and $Y = \lfloor 3N^{0.5} \rfloor$. Construct the polynomial lattice $L(\mathbf{G})$ in Section 4.2, and compute the Euclidean lattice $L(\mathbf{G}_+; X_1, \ldots, X_\ell, Y)$. Then, apply the LLL algorithm to $L(\mathbf{G}_+; X_1, \ldots, X_\ell, Y)$.

Step 3: From the reduced basis, pick the first $\ell + 1$ vectors $\mathbf{v}_1, \ldots, \mathbf{v}_{\ell+1}$. Then, compute the corresponding polynomials $h_k(x_1, \ldots, x_k, y)$, i.e., take polynomials so that $\mathbf{v}_k = \mathcal{V}(h_k; X_1, \ldots, X_k, Y)$ for $k \in [\ell + 1]$.

Step 4: First check $h_i(\bar{x_1}, \ldots, \bar{x_\ell}, \bar{y}) = 0$ for all $k \in [\ell + 1]$. If it is not true, reject the instance. After the polynomials pass the first check, compute the resultant of polynomials modulo prime to check the algebraic independence. If the instance passes two checks, then we regard the experiment as successful.

Fig. 2. Procedure of our computer experiments

Table 1. Theoretical β bound and lattice dimension for small ℓ and several m

$\ell = 2$								
m	2	3	4	5	6	7	10	limit
β	0.386	0.405	0.416	0.424	0.430	0.434	0.442	0.464
dim	27	64	125	216	343	512	1331	-

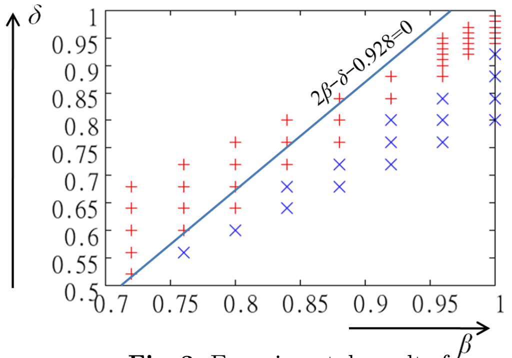
$\ell = 3$								
m	2	3	4	5	6	7	10	limit
β	0.464	0.486	0.500	0.508	0.514	0.519	0.527	0.550
dim	108	352	875	1836	3430	5888	21296	-

Table 2. Experimental results for short secret exponents

ℓ	m	n	β_{thm}	dim	β_{exp}	LLL-time
2	2	512	0.386	27	0.386	3.2 sec
		1024			0.386	10.55 sec.
2	3	512	0.405	64	0.406	5 min. 33 sec
		1024			0.406	30 min. 44 sec.
2	4	512	0.414	125	0.416	3 hrs. 50 min.
		1024			0.414	20hrs. 26min.
3	2	512	0.464	108	0.464	41 min. 25 sec.
		1024			0.464	3 hrs. 17 min.

We carried out our experiments to search for the practical bound of β for several choices of ℓ, m and n. We executed our procedure for each β at intervals of 0.002. Table 2 shows the experimental results. The column "β_{exp}" indicates the experimental bound of β for parameters (l, m, n); that is, the instance passed the final test at that β and failed at $\beta + 0.002$. The columns "β_{thm}" and "dim" are the theoretical bound of β and the lattice dimension, respectively; which are the same as shown in Table 1. The running time of the LLL algorithm for processing $L(\mathbf{G}_+)$ is given in the column "LLL-time."

We note that for $\ell = 3$ and $m = 2$, the second half of Step 4 is not finished due to computational time. More precisely, `Maple` computed two bivariate polynomials, $r_1(x_3, y)$ and $r_2(x_3, y)$ from $h_1, \ldots, h_4$. It took over 120 hours to compute $\mathrm{Res}_{x_3}(r_1, r_2)$, and we stopped the computation. However, we can observe that $h_1, \ldots, h_4$ are algebraically independent since they are reduced to the bivariate polynomials, and can expect that the final resultant will be computed if more time is permitted. Hence, we regard the experiment as a success. From the observation, we conclude our method works well.



β	δ	LLL-time	result
1.00	0.96	18 hrs. 59 min.	+ (passed)
1.00	0.94	16 hrs. 13 min.	+ (passed)
1.00	0.92	15 hrs. 46 min.	× (fault)
0.96	0.90	15 hrs. 42 min.	+ (passed)
0.96	0.88	16 hrs. 9 min.	+ (passed)
0.96	0.84	14 hrs. 46 min.	× (fault)

Fig. 3. Experimental results for partial key exposure situation.

6.2 Experiments for partial key exposure situation

Next we conducted our experiments on the partial key exposure situation. The experimental procedure is similar to in Figure 2. Different points are the definition of $F_k(x_k, y)$, and that $M = 2^{\lfloor \delta n \rfloor}$ and $\widetilde{d_k} = d \bmod M$ are added in Step 1.

We fixed the parameters $\ell = 3$ and $m = 2$ since it could be taken β close to one. Unfortunately, for this ℓ, only the lattice constructed with $m = 2$ can be reduced in reasonable time. The lattice dimension is 108 as in the above subsection. For several choices of β and δ, we generated 1024-bit RSA sample instances and tested them.

Figure 3 shows the result. In the figure, the horizontal and vertical axes are β and δ, respectively. Each mark represents one experiment (β, δ) at the point. The marks "+" and "×" mean that the instance passed and was a fault, respectively. The left table in Figure 3 indicates the running time of the LLL algorithm and experimental results for several β and δ close to $\beta = 1$. Again, note that the final resultant computation was not finished and regard that the experiment is successful if `Maple` computes two bivariate polynomials.

Here, the left/upper area of the diagonal line indicates $\delta > 2\beta - 0.928$, which is derived from (18) for $\ell = 3$ and $m = 2$. Hence, it means that the experimental result is slightly better than the theory. We think this gap is caused by an existence of a better sublattice that improves the original bound (18); similar situations are reported Boneh and Durfee [3]. Unfortunately, we cannot extract this sublattice. For the constructed lattice basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$, write the first vector in the reduced lattice by $\sum_{i=1}^{n} a_i \mathbf{b}_i$. We observed the coefficients are all non-zero in our experiments. Therefore, the sublattice is not spanned by a subset of $\mathbf{B}$. Extracting this sublattice, or finding why the practical result is better than the theory are topics for our future work.

7 Discussion and open problems

Minkowski sum lattice construction: Although our lattice construction works well, it is not optimal. That is, in Section 3.1, $L(\mathbf{G}_+)$ is a sublattice of $\mathcal{A}$ that spanned by all possible combination of polynomials. Providing a method to extract the lattice basis of $\mathcal{A}$, and deriving the condition so that $L(\mathbf{G}_+)$ and $\mathcal{A}$ are equivalent are open problems.

Cryptanalysis of RSA with small secret exponents: Both our bound (16) and that by Sarkar and Maitra converge to $N^{0.75}$ when ℓ becomes large, whereas the limit by the counting argument is N. Filling this gap is an interesting problem. We expect that our heuristic improvement shown in Appendix D achives this goal, though this is not proven.

Cryptanalysis of RSA in other situations: The proposed Minkowski sum based lattice construction can be applied to other situations of cryptanalysis of RSA including revealed MSBs [15], RSA-CRT [25], Takagi's RSA [27], small e [4, 6, 30], unbalanced p and q situation [31], and special settings of e [33]. For more information, see [35, Chap. 10].

References

1. Y. Aono, A new lattice construction for partial key exposure attack for RSA, in *Proceedings of PKC 2009*, Lecture Notes in Computer Science, vol. 5443, pp. 34-53, Springer, Heidelberg, 2009.

2. Y. Aono, M. Agrawal, T. Sato and O. Watanabe, On the optimality of lattices for the Coppersmith technique, in *Proceedings of ACISP 2012*, Lecture Notes in Computer Science, vol. 7372, pp. 376-389, Springer, Heidelberg, 2012. The full-version is available online at *Cryptology ePrint Archive*, 2012/134.
3. D. Boneh and G. Durfee, Cryptanalysis of RSA with private key d less than $N^{0.292}$, in *Proceedings of Eurocrypt 1999*, Lecture Notes in Computer Science, 1592, pp. 389-401, Springer, Heidelberg, 1999.
4. D. Boneh, G. Durfee and Y. Frankel, Exposing an RSA private key given a small fraction of its bits in *Proceedings of Asiacrypt* 1998, Lecture Notes in Computer Science, vol. 1514, pp. 25-34, Springer, Heidelberg, 1998.
5. J. Blömer and A. May, Low secret exponent RSA revisited in *Proceedings of CaLC 2001*, Lecture Notes in Computer Science, vol. 2146, pp. 4-19, Springer, Heidelberg, 2001.
6. J. Blömer and A. May, New partial key exposure attacks on RSA, in *Proceedings of Crypto 2003*, Lecture Notes in Computer Science, vol. 2729, pp. 27-43, Springer, Heidelberg, 2003.
7. J. Blömer and A. May, A tool kit for finding small roots of bivariate polynomials over the integers, *Proceedings of Eurocrypt 2005*, Lecture Notes in Computer Science, vol. 3494, pp. 251-267, Springer, Heidelberg, 2005.
8. E. Brier, D. Naccache, P. Q. Nguyen and M. Tibouchi, Modulus fault attacks against RSA-CRT signatures, in *Proceedings of CHES 2011*, Lecture Notes in Computer Science, vol. 6917, pp. 192-206, Springer, Heidelberg, 2011.
9. D. Cox, J. Little, D. O'Shea, Ideals, varieties, and algorithms: An introduction to computational algebraic geometry and commutative algebra, Springer-Verlag, New York, 2007.
10. D. Coppersmith, Finding a small root of a univariate modular equation, *Proceedings of Eurocrypt 1996*, Lecture Notes in Computer Science, vol. 1070, pp. 155-165, Springer, Heidelberg, 1996.
11. D. Coppersmith, Finding a small root of a bivariate integer equation; factoring with high bits known, *Proceedings of Eurocrypt 1996*, Lecture Notes in Computer Science, vol. 1070, pp. 178-189, Springer, Heidelberg, 1996.
12. D. Coppersmith, Finding small solutions to small degree polynomials, *Proceedings of CaLC 2001*, Lecture Notes in Computer Science, vol. 2146, pp. 20-31, Springer, Heidelberg, 2001.
13. J. S. Coron and A. May, Deterministic polynomial-time equivalence of computing the RSA secret key and factoring, *Journal of Cryptology*, vol. 20,No. 1, pp. 39-50, 2007.
14. J. S. Coron, D. Naccache and M. Tibouchi, Fault attacks against EMV signatures, *Proceedings of CT-RSA 2010*, Lecture Notes in Computer Science, vol. 2010, pp. 208-220, Springer, Heidelberg, 2010.
15. M. Ernst, E. Jochemsz, A. May, and B. Weger, Partial key exposure attacks on RSA up to full size exponents, in *Proceedings of Eurocrypt 2005*, Lecture Notes in Computer Science, vol. 3494, pp. 371-386, Springer, Heidelberg, 2005.
16. GiNaC is Not a CAS, `http://www.ginac.de/`.
17. The GNU MP Bignum Library, `http://gmplib.org/`.
18. A. D. Healy, Resultants, Resolvents and the Computation of Galois Groups, Available online at `http://www.alexhealy.net/papers/math250a.pdf`
19. M. Herrmann, Improved cryptanalysis of the multi-prime Φ-hiding assumption, *Proceedings of Africacrypt 2011*, Lecture Notes in Computer Science, vol. 6737, pp. 92-99, Springer, Heidelberg, 2011.
20. N. Howgrave-Graham, Finding small roots of univariate modular equations revisited, *Proceedings of Cryptography and Coding*, Lecture Notes in Computer Science, vol. 1355, pp. 131-142, Springer, Heidelberg, 1997.
21. M. J. Hinek and C. C. Y. Lam, Common modulus attacks on small private exponent RSA and some fast variants (in practice), in *Journal of Mathematical Cryptology*, vol. 4, Issue 1, pp. 58-93, 2010.
22. M. Herrmann and A. May, Solving linear equations modulo divisors: On factoring given any bits, in *Proceedings of Asiacrypt 2008*, Lecture Notes in Computer Science, vol. 5350, pp. 406-424, Springer, Heidelberg, 2008.
23. M. Herrmann and A. May, Attacking power generators using unravelled linearization: When do we output too much?, in *Proceedings of Asiacrypt 2009*, Lecture Notes in Computer Science, vol. 5912, pp. 487-504, Springer, Heidelberg, 2009.
24. N. Howgrave-Graham and J. P. Seifert, Extending Wiener's attack in the presence of many decrypting exponents, *Proceedings of Secure Networking 1999*, Lecture Notes in Computer Science, vol. 1740, pp. 153-166, Springer, Heidelberg, 1999.

25. E. Jochemsz and A. May, A strategy for finding roots of multivariate polynomials with new applications in attacking RSA variants, in *Proceedings of Asiacrypt 2006*, Lecture Notes in Computer Science, vol. 4284, pp. 267-282, Springer, Heidelberg, 2006.
26. E. Jochemsz and B. Weger, A partial key exposure attack on RSA using a 2-dimensional lattice, in *Proceedings of ISC 2006*, Lecture Notes in Computer Science, vol. 4176, pp. 203-216, Springer, Heidelberg, 2006.
27. N. Kunihiro and K. Kurosawa, Deterministic polynomial time equivalence between factoring and key-recovery attack on Takagi's RSA, in *Proceedings of PKC 2007*, Lecture Notes in Computer Science, vol. 4450, pp. 412-425, Springer, Heidelberg, 2007.
28. N. Kunihiro, Solving generalized small inverse problems, in *Proceedings of ACISP 2010*, Lecture Notes in Computer Science, vol. 6168, pp. 248-263, Springer, Heidelberg, 2010.
29. A. K. Lenstra, H. W. Lenstra, Jr. and L. Lovász Factoring polynomials with rational coefficients, *Mathematische Annalen*, vol. 261, pp. 515-534, 1982.
30. P. Luo, H.-J. Zhou, D.-S. Wang and Y.-Q Dai, Cryptanalysis of RSA for a special case with $d > e$, *Science in China Series F: Information Sciences*, vol. 52, num. 4, pp. 609-616, 2009.
31. A. May, Cryptanalysis of unbalanced RSA with small CRT-exponent, in *Proceedings of Crypto 2002*, Lecture Notes in Computer Science, vol. 2442, pp. 242-256, Springer, Heidelberg, 2002.
32. A. May and M. Ritzenhofen, Solving systems of modular equations in one variable: How many RSA-encrypted messages does Eve need to know?, in *Proceedings of PKC 2008*, Lecture Notes in Computer Science, vol. 4939, pp. 37-46, Springer, Heidelberg, 2008.
33. S. Maitra and S. Sarkar, A New Class of Weak Encryption Exponents in RSA, in *Proceedings of Indocrypt 2008*, Lecture Notes in Computer Science, vol. 5365, pp. 337-349, 2008.
34. V. Shoup, NTL: A Library for doing Number Theory, `http://www.shoup.net/ntl/index.html`
35. P. Q. Nguyen and B. Vallée, The LLL algorithm: Survey and applications, Springer-Verlag, Berlin Heidelberg, 2009.
36. M. Ritzenhofen, On efficiently calculating small solutions of systems of polynomial equations: lattice-based methods and applications to cryptography, Ph.D. thesis, Ruhr University Bochum, `http://www-brs.ub.ruhr-uni-bochum.de/netahtml/HSS/Diss/RitzenhofenMaike/diss.pdf`
37. R. L. Rivest, A. Shamir, and L. Adleman, A method for obtaining digital signatures and public-key cryptsystems, *Communications of the ACM*, vol. 21, No. 2, pp. 120-128, 1978.
38. S. Sarkar and S. Maitra, Cryptanalysis of RSA with two decryption exponents, in *Information Processing Letter*, vol. 110, pp. 178-181, 2010.
39. S. Sarkar and S. Maitra, Cryptanalysis of RSA with more than one decryption exponent, in *Information Processing Letter*, vol. 110, pp. 336-340, 2010.
40. M. J. Wiener, Cryptanalysis of short RSA secret exponents, *IEEE Transactions on Information Theory* vol. 36, no. 3, pp. 553-558, 1990.

A Minkowski sum of three or more lattices

Here we give the detailed construction of the Minkowski sum of three or more lattices. For fixed integer $\ell \geq 3$, consider polynomial lattice bases $\mathbf{G}_k = \{g_1^{(k)}, \ldots, g_{c_k}^{(k)}\}$ for $k \in [\ell]$. Again, assume that they have strictly increasing degree order. For simplicity, we also assume that a polynomial $g_i^{(k)}$ is bivariate of variables x_k and y; thus, only y is shared. Note that extending to a general case is trivial, but it yields extremely complicated expressions.

For each $k \in [\ell]$, let I_k be the set of indices $\{HI(g_i^{(k)}) : i \in [c_k]\} \subset \mathbb{Z}^{\ell+1}$ and define the Minkowski sum $I_+ = I_1 \boxplus \cdots \boxplus I_\ell$. Then, for every $(i_1, \ldots, i_\ell, j) \in I_+$, define the polynomial

$$g^+_{i_1,\ldots,i_\ell,j} = \sum_{(*)} a_{\lambda_1,\ldots,\lambda_\ell} g^{(1)}_{\lambda_1} \cdots g^{(\ell)}_{\lambda_\ell}$$

where the range of sum $(*)$ is over all suffix tuples $(\lambda_1, \ldots, \lambda_\ell)$ satisfying $HM(g^{(1)}_{\lambda_1} \cdots g^{(\ell)}_{\lambda_\ell}) = x_1^{i_1} \cdot \cdots \cdot x_\ell^{i_\ell} y^j$, and the coefficients are defined so that the head coefficient of the polynomial

$g^+_{i_1,\ldots,i_\ell,j}$ is $GCD_{(*)}(HC(g^{(1)}_{\lambda_1}\cdots g^{(\ell)}_{\lambda_\ell}))$. Using the above polynomials, define the Minkowski sum construction $\mathbf{G}^+$ by the lattice spanned by all $g^+_{i_1,\ldots,i_\ell,j}$ for $(i_1,\ldots,i_\ell,j)\in I_+$.

In a similar way to Theorem 2, it can be shown that the Minkowski sum of ℓ lower triangular lattice is also lower triangular.

Theorem 3 *Let $\ell \geq 3$ be an integer. For $k \in [\ell]$, assume that the polynomial lattice basis $\mathbf{G}_k = \{g^{(k)}_1,\ldots,g^{(k)}_{c_k}\} \subset \mathbb{Z}[x_k,y]$ has a strictly increasing degree order, and is lower triangular. Then, the Minkowski sum lattice $L(\mathbf{G}_+)$ is also lower triangular.*

Proof. Since the bases $\mathbf{G}_k$ are all lower triangular, that is, for each $k \in [\ell]$, there exists a sequence of integer pairs $\{(i_k(1),j_k(1)),(i_k(2),j_k(2)),\ldots,(i_k(c_k),j_k(c_k))\}$ and each polynomial $g^{(k)}_i(x_k,y)$ for $i\in[c_k]$ can be written as

$$g^{(k)}_i(x_k,y) = \sum_{i'=1}^{i} d^{(k)}_{i,i'} x_k^{i_k(i')} y^{j_k(i')},\ \text{where } d_{i,i}\neq 0.$$

Fix any $(i_1,\ldots,i_\ell,j)\in I_+$. We show that $g^+_{i_1,\ldots,i_\ell,j}$ can be written as

$$H\cdot x_1^{i_1}\cdots x_\ell^{i_\ell}y^j + (\text{terms whose indices are in } I_+ \text{ and smaller than } x_1^{i_1}\cdots x_\ell^{i_\ell}y^j).$$

for any $(i_1,\ldots,i_k,j)\in I_+$. Here, H is the heading coefficient. Thus, as the proof of Theorem 2, we show any $g^{(1)}_{\lambda_1}\cdots g^{(\ell)}_{\lambda_\ell}$ can be written as the above form for any allowed combination of $(\lambda_1,\ldots,\lambda_\ell)$. Fix a suffix tuple $(\lambda_1,\ldots,\lambda_\ell)$ and consider the monomial expansion of $g^{(1)}_{\lambda_1}\cdots g^{(\ell)}_{\lambda_\ell}$.

Pick any term in the polynomial $g^{(k)}_{\lambda_k}$ for $k\in[\ell]$ and let it be $A_k x_k^{\bar{i}_k} y^{\bar{j}_k}$. The product of these is $A_1\cdots A_k x_1^{\bar{i}_1}\cdots x_\ell^{\bar{i}_\ell} y^{\bar{j}_1+\cdots+\bar{j}_\ell}$. Thus, it is clear that the corresponding index $(\bar{i}_1,\ldots,\bar{i}_\ell,\bar{j}_1+\cdots+\bar{j}_\ell)$ is in I_+. By the relation $(0,\ldots,0,\bar{i}_k,0,\ldots,0,\bar{j}_k)\prec HI(g_k)$, we have

$$\begin{aligned}
&(\bar{i}_1,\ldots,\bar{i}_\ell,\bar{j}_1+\cdots+\bar{j}_\ell)\\
&=(\bar{i}_1,0,\ldots,0,\bar{j}_1)+(0,\bar{i}_2,0,\ldots,0,\bar{j}_2)+\cdots+(0,\ldots,\bar{i}_\ell,\bar{j}_\ell)\\
&\prec HI(g^{(1)}_{\lambda_1})+\cdots+HI(g^{(\ell)}_{\lambda_\ell}) = HI(g^+_{i_1,\ldots,i_\ell,j}).
\end{aligned}$$

Therefore, the lattice $L(\mathbf{G}_+)$ spanned by all $g^+_{i_1,\ldots,i_\ell,j}$ for all $(i_1,\ldots,i_\ell,j)\in I_+$ is lower triangular. □

B Small example of Minkowski sum construction for RSA with short secret exponents

Here we give an example of the lattice construction in Section 4.2 for $m=1$ and $\ell=2$.

For $k=1$ and 2, the polynomials satisfying (2) are $g^{(k)}_1(x_k,y)=e_k$, $g^{(k)}_2(x_k,y)=e_kx_k$ and $g^{(k)}_3(x_k,y)=F_k(x_k,y)=-1+x_k(y+N)$. The index sets are $I_1=\{(0,0,0),(1,0,0),(1,0,1)\}$ and $I_2=\{(0,0,0),(0,1,0),(0,1,1)\}$. Then, the Minkowski sum of these sets is

$$I_+ = I_1 \boxplus I_2 = \{(0,0,0),(0,1,0),(0,1,1),(1,0,0),(1,0,1),(1,1,0),(1,1,1),(1,1,2)\}.$$

For all elements in this set, construct the polynomials satisfying (8) via (9):

$$\begin{aligned}
g^+_{0,0,0} &= e_1e_2, & g^+_{0,1,0} &= e_1e_2x_2,\\
g^+_{0,1,1} &= e_1F_2(x_2,y), & g^+_{1,0,0} &= e_1e_2x_1,\\
g^+_{1,0,1} &= e_2F_1(x_1,y), & g^+_{1,1,0} &= e_1e_2x_1x_2,\\
g^+_{1,1,1} &= a_1e_1x_1F_2(x_2,y)+a_2e_2x_2F_1(x_1,y),\ \text{and} & &\\
g^+_{1,1,2} &= F_1(x_1,y)F_2(x_2,y). & &
\end{aligned}$$

Here, for $(i_1, i_2, j) = (1, 1, 1)$, there are two polynomial pairs $(g^{(1)}_{\lambda}, g^{(2)}_{\lambda'})$ such that $\mathrm{HM}(g^{(1)}_{\lambda} g^{(2)}_{\lambda'}) = x_1 x_2 y$ in (9); that is, both $g^{(1)}_2 g^{(2)}_3$ and $g^{(1)}_3 g^{(2)}_2$ satisfy it. Hence, the coefficients of the integer combination are taken so that $\mathrm{HC}(g^{+}_{1,1,1}) = \mathrm{GCD}(\mathrm{HC}(g^{(1)}_2 g^{(2)}_3), \mathrm{HC}(g^{(1)}_3 g^{(2)}_2)) = \mathrm{GCD}(e_1, e_2) = 1$. Thus, set integers a_1 and a_2 satisfying $a_1e_1 + a_2e_2 = 1$, and $g^{+}_{1,1,1} = a_1 g^{(1)}_2 g^{(2)}_3 + a_2 g^{(1)}_3 g^{(2)}_2 = a_1e_1x_1F_2(x_2, y) + a_2e_2x_2F_1(x_1, y)$.

Therefore, the lattice $L(G^{+}; X_1, X_2, Y)$ is given by

$$\begin{bmatrix} e_1e_2 & & & & & & & & \\ & e_1e_2X_2 & & & & & & & \\ -e_1 & e_1NX_2 & e_1X_2Y & & & & & & \\ & & & e_1e_2X_1 & & & & & \\ -e_2 & & & e_2NX_1 & e_2X_1Y & & & & \\ & & & & & e_1e_2X_1X_2 & & & \\ & -a_2e_2X_2 & & -a_1e_1X_1 & & NX_1X_2 & X_1X_2Y & & \\ 1 & -NX_2 & -X_2Y & -NX_1 & -X_1Y & N^2X_1X_2 & 2NX_1X_2Y & X_1X_2Y^2 \end{bmatrix}. \tag{20}$$

The determinant of this lattice is $e_1^5 e_2^5 X_1^5 X_2^5 Y^5 \approx N^{12.5+10\delta}$. Then, the condition (15) is $12.5 + 10\delta < 16 \Leftrightarrow \delta < 0.35$.

On the other hand, the previous works consider the single equation

$$F_1(x_1, y)F_2(x_2, y) = (-1 + x_1(y + N))(-1 + x_2(y + N)) \equiv 0 \pmod{e_1e_2}$$

and then construct a polynomial lattice using the strategy by Jochemsz and May [25], which selects a polynomial basis as

$$\begin{aligned} &g_{0,0,0} = e_1e_2, && g_{0,1,0} = e_1e_2x_2, \\ &g_{0,1,1} = e_1e_2x_2y, && g_{1,0,0} = e_1e_2x_1, \\ &g_{1,0,1} = e_1e_2x_1y, && g_{1,1,0} = e_1e_2x_1x_2, \\ &g_{1,1,1} = e_1e_2x_1x_2y, \text{and } && g_{1,1,2} = F_1(x_1, y)F_2(x_2, y). \end{aligned}$$

The determinant of the constructed lattice is $e_1^7 e_2^7 X_1^5 X_2^5 Y^5 \approx N^{16.5+10\delta}$, which is slightly higher than our construction, and it does not satisfy the condition (15) for any $\delta > 0$.

C Deriving bounds in Sections 4.2 and 5

We give a detailed explanation for evaluating

$$\sum_{(i_1,\ldots,i_\ell,j)\in I_+} \left[0.5j + (i_1 + \cdots + i_\ell)\beta - \sum_{k=1}^{\ell} \min(i_k, j) \right] < 0 \tag{15}$$

in Section 4.2, and

$$\sum_{(i_1,\ldots,i_\ell,j)\in I_+} \left[(0.5 - \delta)j + (i_1 + \cdots + i_\ell)\beta - \sum_{k=1}^{\ell} \min(i_k, j) \right] < 0. \tag{18}$$

in Section 5.

For simplicity, we use the notation $\dot{\sum}$ to denote the sum $\sum_{i_1=0}^{m} \cdots \sum_{i_\ell=0}^{m}$. Then, the sum over $(i_1, \ldots, i_\ell, j) \in I_+$ can explicitly be written as $\sum_{(i_1,\ldots,i_\ell,j)\in I_+} = \dot{\sum} \sum_{j=0}^{i_1+\cdots+i_\ell}$, and the following formulas hold for any $\ell, m \in \mathbb{N}$ and $a, b \in [\ell]$.

$$\dot{\sum} i_a i_b = \begin{cases} \dfrac{m^{\ell+2}}{3} + o(m^{\ell+2}) & (a = b) \\ \dfrac{m^{\ell+2}}{4} + o(m^{\ell+2}) & (a \neq b) \end{cases} \tag{21}$$

By this, we have

$$\dot{\sum}(i_1 + \cdots + i_n)^2 = \left(\ell^2 \cdot \frac{m^{\ell+2}}{4} + \ell \cdot \frac{m^{\ell+2}}{12}\right) + o(m^{\ell+2}). \tag{22}$$

Thus,

$$\begin{aligned}\sum_{(i_1,\ldots,i_\ell,j)\in I_+} j &= \dot{\sum} \sum_{j=0}^{i_1+\cdots+i_\ell} j = \dot{\sum} \left\{ \frac{(i_1 + \cdots + i_\ell)^2}{2} + o(m) \right\} \\ &= \left(\ell^2 \cdot \frac{m^{\ell+2}}{8} + \ell \cdot \frac{m^{\ell+2}}{24}\right) + o(m^{\ell+2}).\end{aligned}$$

and

$$\begin{aligned}\sum_{(i_1,\ldots,i_\ell,j)\in I_+} (i_1 + \cdots + i_\ell) &= \dot{\sum} \sum_{j=0}^{i_1+\cdots+i_\ell} (i_1 + \cdots + i_\ell) = \dot{\sum} \left\{(i_1 + \cdots + i_\ell)^2 + o(m)\right\} \\ &= \left(\ell^2 \cdot \frac{m^{\ell+2}}{4} + \ell \cdot \frac{m^{\ell+2}}{12}\right) + o(m^{\ell+2}).\end{aligned}$$

Next consider the sum $\sum_{(i_1,\ldots,i_\ell,j)\in I_+} \left[\sum_{k=1}^{\ell} \min(i_k, j)\right]$. By symmetry, it suffices to calculate $\sum_{(i_1,\ldots,i_\ell,j)\in I_+} \min(i_1, j)$. Using the relation

$$\sum_{j=0}^{i_1+\cdots+i_\ell} \min(i_1, j) = \sum_{j=0}^{i_1} j + \sum_{j=i_1+1}^{i_1+\cdots+i_\ell} i_1 = \frac{i_1(i_1+1)}{2} + (i_2 + \cdots + i_\ell)i_1,$$

we have

$$\begin{aligned}\sum_{(i_1,\ldots,i_\ell,j)\in I_+} \sum_{k=1}^{\ell} \min(i_k, j) &= \ell \sum_{(i_1,\ldots,i_\ell,j)\in I_+} \min(i_1, j) \\ &= \ell \dot{\sum} \left[\frac{i_1(i_1+1)}{2} + (i_2 + \cdots + i_\ell)i_1\right] \\ &= \left(\frac{\ell^2}{4} - \frac{\ell}{12}\right) m^{\ell+2} + o(m^{\ell+2}).\end{aligned} \tag{23}$$

C.1 Condition for RSA short secret exponents

Using the above formulas, the left-hand side of (15) is

$$\sum_{(i_1,\ldots,i_\ell,j)\in I_I} \left[0.5j + (i_1 + \cdots + i_\ell)\beta - \sum_{k=1}^{\ell} \min(i_k, j)\right] = \left(-\frac{3}{16}\ell^2 + \frac{5}{48}\ell + \left(\frac{\ell^2}{4} + \frac{\ell}{12}\right)\beta\right) m^{\ell+2} + o(m^{\ell+2}).$$

Thus, the condition is to be

$$\beta < \frac{9\ell - 5}{12\ell + 4}, \tag{16}$$

when m is sufficiently large.

C.2 Condition for RSA partial key exposure situation

The left-hand side of (18) is

$$\sum_{(i_1,\ldots,i_\ell,j)\in I} \left[(0.5-\delta)j + (i_1+\cdots+i_\ell)\beta - \sum_{k=1}^{\ell}\min(i_k,j)\right]$$
$$= \left(\frac{1}{4}-\frac{\delta}{2}+\beta\right)\left(\frac{\ell^2}{4}+\frac{\ell}{12}\right)m^{\ell+2} - \left(\frac{\ell^2}{4}-\frac{\ell}{12}\right)m^{\ell+2}.$$

Hence, the condition is to be

$$\beta - \frac{\delta}{2} + \frac{1}{4} < \frac{3\ell-1}{3\ell+1}, \tag{19}$$

when m is sufficiently large.

D Heuristic improvement of lattices for small short secret exponents

Here we give the detailed argument for our improvement of lattices mentioned in Section 4.2. Fix the parameters $\beta > 0.5$ and m. Consider a subset of I_+:

$$I_{++} = \left\{(i_1,\ldots,i_\ell,j)\in I_+ : j \le \max\{i_1,\ldots,i_\ell\} \text{ or } 0.5j + (i_1+\cdots+i_\ell)\beta - \sum_{k=1}^{\ell}\min(i_k,j) < 0\right\} \tag{24}$$

Thus, it deletes tuples such that j is large and terms in (15) are greater than zero. Then, construct the lattice with basis $\mathbf{G}_{++} = \{g^+_{i_1,\ldots,i_\ell,j} : (i_1,\ldots,i_\ell,j)\in I_{++}\} \subset \mathbf{G}_+$.

Proposition 1 *The lattice $L(\mathbf{G}_{++})$ is lower triangular.*

Proof. As the proof of Theorem 2, it suffices to show that for every $(i_1,\ldots,i_\ell,j)\in I_{++}$,

$$g^+_{i_1,\ldots,i_\ell,j} = a x_1^{i_1}\cdots x_\ell^{i_\ell} y^j + (\text{terms whose indices are in } I_{++} \text{ and smaller than } x_1^{i_1}\cdots x_\ell^{i_\ell}y^j).$$

Recall that $g^{(k)}_{i_k,j_k} = x^{i_k-j_k}(-1+x_k(y+N))^{j_k}e^{m-j_k}$. Thus, it is easy to see that the polynomial is a linear combination of $x^{i'_k}y^{j'_k}$ such that the following three inequalities hold.

$$0 \le j'_k,\ i'_k \le i_k \text{ and } i'_k - j'_k \ge i_k - j_k \tag{25}$$

Since each term in $g^+_{i_1,\ldots,i_\ell,j}$ is a sum of the product of terms in $g^{(1)}_{i_1,j_1},\ldots,g^{(\ell)}_{i_\ell,j_\ell}$, it can be written as $A\cdot x_1^{i'_1}\cdots x_\ell^{i'_\ell}y^{j'_1+\cdots+j'_\ell}$. Thus, it suffices to show $(i'_1,\ldots,i'_\ell,j')\in I_{++}$, where we let $j' = j'_1+\cdots+j'_\ell$.

First, we consider the situation that $(i_1,\ldots,i_\ell,j)$ satisfies the first inequality in the definition of I_{++}; that is, $j \le \max\{i_1,\ldots,i_\ell\}$. In this situation, we have

$$\begin{aligned}
j' &= j'_1+\cdots+j'_\ell \\
&\le \sum_k (i'_k - i_k + j_k) \le \sum_k (i'_k - i_k) + \max\{i_1,\ldots,i_\ell\} \\
&= \max\left\{i_1 + \sum_k (i'_k - i_k),\ldots, i_\ell + \sum_k (i'_k - i_k)\right\} \\
&\le \max\{i_1 + (i'_1 - i_1),\cdots, i_\ell + (i'_\ell - i_\ell)\} \quad (\text{since } i'_k \le i_k) \\
&= \max\{i'_1,\ldots,i'_\ell\}.
\end{aligned}$$

Thus, $(i'_1, \ldots, i'_\ell, j') \in I_{++}$.

Next we consider the situation that $(i_1, \ldots, i_\ell, j)$ satisfies $0.5j + (i_1 + \cdots + i_\ell)\beta - \sum_{k=1}^{\ell} \min(i_k, j) < 0$. By Theorem 2, $(i'_1, \ldots, i'_\ell, j') \in I_+$ is already shown; thus, for the case $j' \leq \max\{i'_1, \ldots, i'_\ell\}$, the claim is proved. Then consider another case. First note that $j' > \max\{i'_1, \ldots, i'_\ell\}$ implies $\min(i'_k, j') = i'_k$ for all $k \in [\ell]$. Thus, we have,

$$0.5j' + (i'_1 + \cdots + i'_\ell)\beta - \sum_{k=1}^{\ell} \min(i'_k, j') = 0.5(j'_1 + \cdots + j'_\ell) + (i'_1 + \cdots + i'_\ell)(\beta - 1).$$

Then, by inequality (25),

$$\begin{aligned}
&0.5(j'_1 + \cdots + j'_\ell) + (i'_1 + \cdots + i'_\ell)(\beta - 1) \\
&\leq 0.5(j_1 + \cdots + j_\ell) - 0.5(i_1 + \cdots + i_\ell) + (\beta - 0.5)(i'_1 + \cdots + i'_\ell) \\
&\leq 0.5(j_1 + \cdots + j_\ell) - 0.5(i_1 + \cdots + i_\ell) + (\beta - 0.5)(i_1 + \cdots + i_\ell) \\
&= 0.5(j_1 + \cdots + j_\ell) + (\beta - 1)(i_1 + \cdots + i_\ell) < 0.
\end{aligned}$$

The last equality holds from $(i_1, \ldots, i_\ell, j) \in I_{++}$ Therefore, for both situations the tuple $(i'_1, \ldots, i'_\ell, j')$ is in I_{++} and $L(\mathbf{G}_{++})$ is lower triangular. □

As the same argument in Section 4.2, the Coppersmith technique works if

$$\sum_{(i_1, \ldots, i_\ell, j) \in I_{++}} \left[0.5j + (i_1 + \cdots + i_\ell)\beta - \sum_{k=1}^{\ell} \min(i_k, j) \right] < 0. \tag{26}$$

The explicit formula has never been given. In Table 3, we show the comparison between the bounds of β from (26) and (15), for several choices of m and ℓ. Interestingly, for $\ell = 14$, $m = 5$ and $\beta = 0.752$, it can see that (26) holds by a naive computation. Thus, it exceeds the limit of 3/4 given by our Minkowski sum construction.

Table 3. Comparison of β bounds between the Minkowski sum construction and heuristic improvement

$\ell = 5$					$\ell = 9$					$\ell = 14$			
m	3	4	5	6	m	3	4	5	6	m	3	4	5
β_{improve}	0.582	0.596	0.604	0.611	β_{improve}	0.677	0.688	0.697	0.702	β_{improve}	0.735	0.745	0.752
β_{original}	0.571	0.583	0.590	0.596	β_{original}	0.641	0.649	0.655	0.659	β_{original}	0.678	0.683	0.686