Minkowski Sum Based Lattice Construction for Multivariate Simultaneous Coppersmith's Technique and Applications to RSA

Yoshinori Aono

National Institute of Information and Communications Technology
aono@nict.go.jp

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 cryptanalysis. 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. Due to the spacing limit, some arguments are omitted; see the full-version [1].

1 Introduction

Since the RSA cryptosystem [31] was proposed, its security has been intensively investigated. In particular, polynomial-time algorithms for recovering short secret exponents have been studied [34,3]. There are two main strategies for recovering a secret exponent in this situation: The continued fraction algorithm was used in this approach [34,19,17] and the Coppersmith technique based approach [3,5,11]. 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 [8,32,33,15]. 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 [26] proposed an approach based on

C. Boyd and L. Simpson (Eds.): ACISP 2013, LNCS 7959, pp. 88–103, 2013.

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.

Related Works on Lattice Construction for the Coppersmith Technique: For a single modular equation, it has been widely studied. The first work by Coppersmith [9] 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 [6] proposed a construction method for bivariate equations, and Jochemsz and May [20] improved this to a method for treating general multivariate equations. Another viewpoint was given by Kunihiro [22], 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 [26] 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 [30] 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 [30, Section 5.4]). We consider this problem.

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 [19,17,32,33]. 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

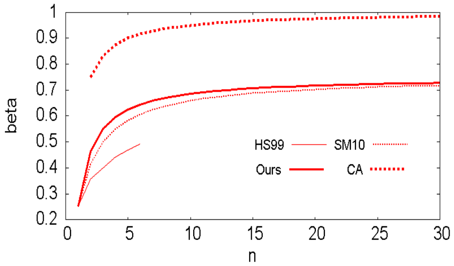
$$\beta < (9\ell - 5)/(12\ell + 4).$$

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 [33], which achieved $\beta < (3\ell - 1)/(4\ell + 4)$. For large ℓ, both values converge to 3/4.

Noting that Howgrave-Graham and Seifert [19] had given an extended version of Wiener's continued fraction attack and obtain the bound

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

However, Hinek and Lam [17] 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 [19] for $\ell \leq 6$. `SM10` is Sarkar and Maitra [33]. `Ours` shows our result. `CA` indicates the heuristic bound by the counting argument in Section 4.1.



ℓ	2	3	4
HS99: [19]	0.357	0.400	0.441
SM10: [33]	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

Extension for the Partial Key Exposure Situation: We then extend the attack to a situation studied in [4,11], 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

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

Computer Experiments: 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.

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 $\{1, \ldots, a\}$ and $\{a, a+1, \ldots, b-1, b\}$, respectively. For natural numbers x, A and N, the notation $|x| < A \pmod{N}$ means that $0 \leq x < A$ 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 $x_1, x_2, \ldots, x_{n-1}$ 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 $f(x_1, \ldots, x_\ell, y)$. 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 $(i_1, \ldots, i_\ell, j)$ *head coefficient*, *head monomial* and *head index*, and denote them by $\mathrm{HC}(f)$, $\mathrm{HM}(f)$ and $\mathrm{HI}(f)$, respectively.

Let A and B be finite subsets of $\mathbb{Z}^n$, then their Minkowski sum is defined by $A \boxplus B = \{(a_1 + b_1, \ldots, a_n + b_n) : (a_1, \ldots, a_n) \in A, (b_1, \ldots, b_n) \in B\}$. Note that the sum of three or more sets is defined recursively.

2.1 Overview of the Coppersmith Technique

We introduce the Coppersmith technique [9,10] with necessary definitions and lemmas. Our formulation is due to Howgrave-Graham [16] 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

$$F(x, y) \equiv 0 \pmod{W} \tag{1}$$

satisfying $|x| < X$ and $|y| < Y$. While this 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 $m \geq 2$ and consider a set L of polynomials $g(x, y) \in \mathbb{Z}[x, y]$ satisfying

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

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

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

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 [14] or the Gröbner basis technique [7]. 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 [23] 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 $g(x,y) = \sum_{i,j} a_{i,j}x^iy^j$ 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 $a_{i,j}x^iy^j$ 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 [16], generalized in [20]) *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 [23] which computes an LLL-reduced basis from a given basis. The following theorem bounds the lengths of first vectors in such bases.

Theorem 1. *[5] 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 $\mathbf{G} = \{g_1, \dots, g_c\}$ be a sequence of linearly independent polynomials in $\mathbb{Z}[x, y]$. Then, the *polynomial lattice* spanned by them is defined by $L(\mathbf{G}) = L(g_1, \dots, g_c) = \{a_1 g_1 + \cdots + a_c g_c \ : \ a_i \in \mathbb{Z} \text{ for } k \in [c]\}$. We also consider the vectorization of polynomial lattices; i.e., for a basis $\mathbf{G} = \{g_1, \dots, g_c\}$, consider their vectorization $\mathcal{V}(g_1; X, Y), \dots, \mathcal{V}(g_c; X, Y)$ 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 upper bounds 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:

$$F_1(x_1, y) \equiv 0 \pmod{W_1} \text{ and } F_2(x_2, y) \equiv 0 \pmod{W_2} \tag{7}$$

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

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

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 $\det(\mathbf{G}; X_1, X_2, Y)^{1/c} < (W_1W_2)^m$. 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 $\{g_{\ell_1}^{(1)} g_{\ell_2}^{(2)}\}_{\ell_1, \ell_2}$ 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., $\mathrm{HM}(g_1^{(k)}) \prec \cdots \prec \mathrm{HM}(g_{c_k}^{(k)})$ holds for $k = 1, 2$. If this is not true, an equivalent basis having this property can be efficiently computed by multiplying a unimodular matrix; the computation is performed by a Gaussian elimination-like algorithm, see [2]. Then, for each k, consider the set of indices $I_k = \{\mathrm{HI}(g_\ell^{(k)}) : \ell \in [c_k]\} \subset \mathbb{Z}^3$ 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

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

Here, the range of sum $(*)$ is over all suffix pairs (λ, λ') satisfying $\mathrm{HM}(g_\lambda^{(1)} g_{\lambda'}^{(2)}) = x_1^{i_1} x_2^{i_2} y^j$ and the coefficients a_λ are defined so that

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

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.

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

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

where $a_{\ell,\ell} \neq 0$ 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 can show that the Minkowski sum lattice of them is also lower triangular; for the proof, see the full-version. Note that the situation of three or more lattices, which is considered in our applications, can be proven by induction.

Theorem 2. *For $k = 1, 2$, assume that the polynomial lattice basis $\mathbf{G}_k = \{g^{(k)}_1, \dots, g^{(k)}_{c_k}\}$ has a strictly increasing degree order, and that they are lower triangular. Then the Minkowski sum lattice $L(\mathbf{G}_+)$ is also lower triangular.*

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 $p + q < 3N^{0.5}$.

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 [32,33] (see Boneh and Durfee [3] for deriving single equation), it can prove that the simultaneous equations

$$F_i(x_i, y) = -1 + x_i(y + N) \equiv 0 \pmod{e_i} \text{ for } i = 1, \ldots, \ell \tag{11}$$

have a small solution $(x_1, \ldots, x_\ell, y)$ 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. By a counting argument, we set the following heuristic assumption of bounding β; the detailed argument is given in the full version.

Heuristic Assumption: For a natural number ℓ, assume that

$$\beta < (\ell - 0.5)/\ell. \tag{13}$$

Then, within the range of (12), the equation (11) has only one solution by which we 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 that achieves the bound $\beta < 0.25$. While they achieved $\beta < 0.292$ by their improved lattice, we did not use in this paper.

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 2 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 $I_k = \{(0, \dots, 0, i, \dots, 0, j) : (i, j) \in \mathbb{Z},\ 0 \le j \le i \le m\}$ and their Minkowski sum is

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

For each $(i_1, \dots, i_\ell, j) \in I_+$, a polynomial is written as by

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

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

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

Note that $\mathrm{HC}(g^{(1)}_{i_1,j_1} g^{(2)}_{i_2,j_2} \cdots g^{(\ell)}_{i_\ell,j_\ell}) = e_1^{m-j_1} \cdots e_\ell^{m-j_\ell}$. 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 $a_{j_1,\dots,j_\ell}$ so that the head coefficient of $g_{i_1,\dots,i_\ell,j}$ is this value.

Then, we set the Minkowski sum lattice by $\mathbf{G}_+ = \{g_{i_1,\dots,i_\ell,j} : (i_1, \dots, i_\ell, j) \in I_+\}$ and the order is the lexicographic order of suffixes. By Theorem 2 (and its generalization), the converted lattice $L(\mathbf{G}_+; X_1, \dots, X_\ell, Y)$ is lower triangular. The diagonal element corresponding to $(i_1, \dots, i_\ell, j)$ is

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

Therefore, the determinant is

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

As with the same argument in Section 2.1, the Coppersmith technique works if $\det(\mathbf{G}_+; X_1, \dots, X_\ell, Y)^{1/|I_+|} < (e_1 \cdots e_\ell)^m$, where $|I_+|$ 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

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

By computing the left-hand side, 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

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

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; detailed construction and deriving numerical bounds are given in the full-version.

5 Application to 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 [11], we consider the simultaneous equations

$$F_i(x_i, y) = e_i \widetilde{d_i} - 1 + x_i(y + N) \equiv 0 \pmod{e_i M} \text{ for } i = 1, \ldots, \ell. \tag{17}$$

By the counting argument, 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

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

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

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

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, the determinant $\det(\mathbf{G}_+; X_1, \ldots, X_\ell, Y)$ is

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

Params: 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 $d_1, \dots, d_\ell$ 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 $\bar{y} = 1 - p - q$.

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 $L(\mathbf{G}_+; X_1, \dots, X_\ell, Y)$. Then, apply the LLL algorithm to $L(\mathbf{G}_+; X_1, \dots, X_\ell, Y)$.

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

Step 4: First check $h_i(\bar{x_1}, \dots, \bar{x_\ell}, \bar{y}) = 0$ 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

Plugging the approximations $e_k \approx N$, $X_k = N^\beta$, $Y \approx N^{0.5}$ and $M \approx N^\delta$, the working condition is

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

Calculating the left-hand side, when m becomes large, the condition is

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

6 Computer Experiments of our RSA Cryptanalysis

Experimental Environment: The experiments were conducted on a workstation with 16GB of RAM and two Intel Xeon X5675@3.07GHz. We wrote our experimental code in the C++ language using the following libraries. To compute the LLL reduced basis, we used Shoup's NTL library [28] version 5.5.2 compiled with the GMP library [13] version 5.0.4. The polynomial computation was performed using the GiNaC library [12] 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 in $\mathbb{Z}_p$ 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

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

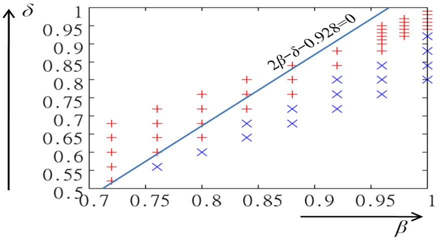
ℓ	m	n	β_{thm}	dim	β_{exp}	LLL-time
2	2	512	0.386	27	0.386	3.2 sec
		1024			0.386	10.55 sec.
2	3	512	0.405	64	0.406	5 min. 33 sec
		1024			0.406	30 min. 44 sec.
2	4	512	0.414	125	0.416	3 hrs. 50 min.
		1024			0.414	20hrs. 26min.
3	2	512	0.464	108	0.464	41 min. 25 sec.
		1024			0.464	3 hrs. 17 min.

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 $r_k = \text{Res}_{x_1}(h_1, h_k) \bmod P$ for $k = 2, \ldots, \ell+1$, and next we compute $\text{Res}_{x_2}(r_2, r_k) \bmod P$ 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 pseudoprime 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).

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 $\text{Res}_{x_3}(r_1, r_2)$, 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



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

time is permitted [1]. Hence, we regard the experiment as a success. From the observation, we conclude our method works well.

6.2 Experiments for Partial Key Exposure Situation

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.

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 in the full-verstion achieves this goal, though this is not proven.

[1] An ACISP reviewer proposed to use the Gröbner basis instead, and use more polynomials since the LLL algorithm usually finds more small vectors than required.

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 [11], RSA-CRT [20], Takagi's RSA [21], small e [4,5,24], unbalanced p and q situation [25], and special settings of e [27]. For more information, see [29, Chap. 10].

Acknowledgments. I would like to thank Noboru Kunihiro and Shiho Moriai for helpful discussion and comments on an earlier version of this paper. I also would like to thank the anonymous reviewers for insightful comments.

References

1. Aono, Y.: Minkowski sum based lattice construction for multivariate simultaneous Coppersmith's technique and applications to RSA, Cryptology ePrint Archive, 2012/675 (2012)
2. Aono, Y., Agrawal, M., Satoh, T., Watanabe, O.: On the Optimality of Lattices for the Coppersmith Technique. In: Susilo, W., Mu, Y., Seberry, J. (eds.) ACISP 2012. LNCS, vol. 7372, pp. 376–389. Springer, Heidelberg (2012); The full-version is available online at Cryptology ePrint Archive, 2012/134
3. Boneh, D., Durfee, G.: Cryptanalysis of RSA with private key d less than $N^{0.292}$. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 1–11. Springer, Heidelberg (1999)
4. 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.) ASIACRYPT 1998. LNCS, vol. 1514, pp. 25–34. Springer, Heidelberg (1998)
5. Blömer, J., May, A.: New partial key exposure attacks on RSA. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 27–43. Springer, Heidelberg (2003)
6. Blömer, J., May, A.: A tool kit for finding small roots of bivariate polynomials over the integers. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 251–267. Springer, Heidelberg (2005)
7. Cox, D., Little, J., O'Shea, D.: Ideals, varieties, and algorithms: An introduction to computational algebraic geometry and commutative algebra. Springer, New York (2007)
8. Coron, J.-S., Naccache, D., Tibouchi, M.: Fault Attacks Against EMV Signatures. In: Pieprzyk, J. (ed.) CT-RSA 2010. LNCS, vol. 5985, pp. 208–220. Springer, Heidelberg (2010)
9. Coppersmith, D.: Finding a small root of a univariate modular equation. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 155–165. Springer, Heidelberg (1996)
10. Coppersmith, D.: Finding a small root of a bivariate integer equation; factoring with high bits known. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 178–189. Springer, Heidelberg (1996)
11. Ernst, M., Jochemsz, E., May, A., de Weger, B.: Partial key exposure attacks on RSA up to full size exponents. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 371–386. Springer, Heidelberg (2005)
12. GiNaC is Not a CAS, `http://www.ginac.de/`
13. The GNU MP Bignum Library, `http://gmplib.org/`
14. Healy, A.D.: Resultants, Resolvents and the Computation of Galois Groups, `http://www.alexhealy.net/papers/math250a.pdf`

15. Herrmann, M.: Improved cryptanalysis of the multi-prime Φ-hiding assumption. In: Nitaj, A., Pointcheval, D. (eds.) AFRICACRYPT 2011. LNCS, vol. 6737, pp. 92–99. Springer, Heidelberg (2011)
16. Howgrave-Graham, N.: Finding small roots of univariate modular equations revisited. In: Darnell, M. (ed.) Cryptography and Coding 1997. LNCS, vol. 1355, pp. 131–142. Springer, Heidelberg (1997)
17. Hinek, M.J., Lam, C.C.Y.: Common modulus attacks on small private exponent RSA and some fast variants (in practice). Journal of Mathematical Cryptology 4(1), 58–93 (2010)
18. Herrmann, M., May, A.: Attacking power generators using unravelled linearization: When do we output too much? In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 487–504. Springer, Heidelberg (2009)
19. Howgrave-Graham, N., Seifert, J.-P.: Extending Wiener's attack in the presence of many decrypting exponents. In: Baumgart, R. (ed.) CQRE 1999. LNCS, vol. 1740, pp. 153–166. Springer, Heidelberg (1999)
20. 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.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 267–282. Springer, Heidelberg (2006)
21. Kunihiro, N., Kurosawa, K.: Deterministic polynomial time equivalence between factoring and key-recovery attack on Takagi's RSA. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 412–425. Springer, Heidelberg (2007)
22. Kunihiro, N.: Solving generalized small inverse problems. In: Steinfeld, R., Hawkes, P. (eds.) ACISP 2010. LNCS, vol. 6168, pp. 248–263. Springer, Heidelberg (2010)
23. Lenstra, A.K., Lenstra Jr., H.W., Lovász, L.: Factoring polynomials with rational coefficients. Mathematische Annalen 261, 515–534 (1982)
24. Luo, P., Zhou, H.-J., Wang, D.-S., Dai, Y.-Q.: Cryptanalysis of RSA for a special case with $d > e$. Science in China Series F: Information Sciences 52(4), 609–616 (2009)
25. May, A.: Cryptanalysis of unbalanced RSA with small CRT-exponent. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 242–256. Springer, Heidelberg (2002)
26. May, A., Ritzenhofen, M.: Solving systems of modular equations in one variable: How many RSA-encrypted messages does Eve need to know? In: Cramer, R. (ed.) PKC 2008. LNCS, vol. 4939, pp. 37–46. Springer, Heidelberg (2008)
27. Maitra, S., Sarkar, S.: A New Class of Weak Encryption Exponents in RSA. In: Chowdhury, D.R., Rijmen, V., Das, A. (eds.) INDOCRYPT 2008. LNCS, vol. 5365, pp. 337–349. Springer, Heidelberg (2008)
28. Shoup, V.: NTL: A Library for doing Number Theory, `http://www.shoup.net/ntl/index.html`
29. Nguyen, P.Q., Vallée, B.: The LLL algorithm: Survey and applications. Springer, Berlin (2009)
30. Ritzenhofen, M.: 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`
31. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptsystems. Communications of the ACM 21(2), 120–128 (1978)
32. Sarkar, S., Maitra, S.: Cryptanalysis of RSA with two decryption exponents. Information Processing Letter 110, 178–181 (2010)
33. Sarkar, S., Maitra, S.: Cryptanalysis of RSA with more than one decryption exponent. Information Processing Letter 110, 336–340 (2010)
34. Wiener, M.J.: Cryptanalysis of short RSA secret exponents. IEEE Transactions on Information Theory 36(3), 553–558 (1990)