Recovering a Sum of Two Squares Decomposition Revisited

Xiaona Zhang[1,2,3], Li-Ping Wang[1,2](✉), Jun Xu[1,2], Lei Hu[1,2],
Liqiang Peng[1,2,3], Zhangjie Huang[1,2], and Zeyi Liu[1,2,3]

[1] State Key Laboratory of Information Security,
Institute of Information Engineering, Chinese Academy of Sciences,
Beijing 100093, China
{zhangxiaona,wangliping,xujun,hulei,pengliqiang,
huangzhangjie,liuzeyi}@iie.ac.cn

[2] Data Assurance and Communications Security Research Center,
Chinese Academy of Sciences, Beijing 100093, China

[3] University of Chinese Academy of Sciences, Beijing, China

Abstract. Recently, in [6] Gomez et al. presented algorithms to recover a decomposition of an integer $N = rA^2 + sB^2$, where N, r, s are positive integers, and A, B are the wanted unknowns. Their first algorithm recovers two addends by directly using rigorous Coppersmith's bivariate integer method when the error bounds of given approximations to A and B are less than $N^{\frac{1}{6}}$. Then by combining with the linearization technique, they improved this theoretical bound to $N^{\frac{1}{4}}$. In this paper, we heuristically reach the bound $N^{\frac{1}{4}}$ with experimental supports by transforming the integer polynomial concerned in their first algorithm into a modular one. Then we describe a better heuristic algorithm, the dimension of the lattice involved in this improved method is much smaller under the same error bounds.

Keywords: Sum of squares · Lattice · LLL algorithm · Coppersmith's method

1 Introduction

Coppersmith's method to solve univariate modular polynomial [5] and bivariate integer polynomial [4] enjoys prevalent cryptographic applications, such as breaking the RSA crypto system as well as many of its variant schemes [1,12,14,16,18–20], cracking the validity of the multi-prime Φ-hiding assumptions [9,21], revealing the secret information of kinds of pseudorandom generators [2,6,10], and analyzing the security of some homomorphic encryption schemes [22]. The essence of this famed algorithm is to find integer linear combinations of polynomials which share a common root modulo a certain integer. These derived polynomials possess small coefficients and can be transformed into ones holding true over integers. Thus one can extract the desired roots using standard root-finding algorithms.

D. Lin et al. (Eds.): Inscrypt 2015, LNCS 9589, pp. 178–192, 2016.
DOI: 10.1007/978-3-319-38898-4_11

A noted theorem of Fermat addresses those integers which can be expressed as the sum of two squares. This property relies on the factorization of the integer, from which a sum of two squares decomposition (if exists) can be efficiently computed [8]. Recently, Gutierrez et al. [7] gave an algorithm to recover a decomposition of an integer $N = rA^2 + sB^2$, where r, s are known integers, and A, B are the wanted unknowns. When approximations A_0, B_0 to A, B are given, their first algorithm can recover the two addends under the condition that the approximation errors $|A - A_0|, |B - B_0|$ are no bigger than $N^{\frac{1}{6}}$.

In this paper, we first illustrate a method to solve a certain bivariate modular polynomial $f_N(x, y) = a_1x^2 + a_2x + a_3y^2 + a_4y + a_0$ based on Coppersmith's method. The trick to solve this kind of polynomial can be directly used to recover the two addends A, B of $N = rA^2 + sB^2$ from their approximations with an error tolerance $N^{\frac{1}{4}}$. The least significant bits exposure attacks on A and B can also be quickly executed by applying the method to solve this certain type polynomial. Next, we present a better method for recovering A, B from its approximations A_0, B_0. This improved approach transforms the problem into seeking the coordinates of a certain vector in our built lattice. The problem of finding these coordinates can be reduced to extracting the small roots of a different bivariate modular polynomial $f'_N(x, y) = b_1x^2 + b_2x + b_3y^2 + b_4y + b_5xy + b_0$. The derived error bound is $N^{\frac{1}{3}}$ in this way.

The rest of this paper is organized as follows. In Sect. 2, we recall some preliminaries. In Sect. 3, we first describe the method to solve $f_N(x, y) = a_1x^2 + a_2x + a_3y^2 + a_4y + a_0$ and then give our deduction on error bound $N^{\frac{1}{4}}$ as well as the least significant bits exposure attacks on A, B, both of which are based on finding the small roots of $f_N(x, y)$. In Sect. 4, we elaborate a better method for recovering the addends of a sum of two squares. The theoretical error bound derived by this approach is $N^{\frac{1}{3}}$. Finally, we give some conclusions in Sect. 5.

2 Preliminaries

2.1 Lattices

Let $\mathbf{b_1}, \ldots, \mathbf{b}_\omega$ be linear independent row vectors in $\mathbb{R}^n$, and a lattice $\mathcal{L}$ spanned by them is

$$\mathcal{L} = \{\sum_{i=1}^{\omega} k_i \mathbf{b_i} \mid k_i \in \mathbb{Z}\},$$

where $\{\mathbf{b_1}, \ldots, \mathbf{b}_\omega\}$ is a basis of $\mathcal{L}$ and $B = [\mathbf{b_1}^T, \ldots, \mathbf{b}_\omega{}^T]^T$ is the corresponding basis matrix. The dimension and determinant of $\mathcal{L}$ are respectively

$$\dim(\mathcal{L}) = \omega, \det(\mathcal{L}) = \sqrt{\det(BB^T)}.$$

For any two-dimensional lattice $\mathcal{L}$, the Gauss algorithm can find out the reduced basis vectors $\mathbf{v_1}$ and $\mathbf{v_2}$ satisfying

$$\|\mathbf{v_1}\| \leq \|\mathbf{v_2}\| \leq \|\mathbf{v_1} \pm \mathbf{v_2}\|$$

in polynomial time. One can deduce that $\mathbf{v_1}$ is the shortest nonzero vector in $\mathcal{L}$ and $\mathbf{v_2}$ is the shortest vector in $\mathcal{L} \setminus \{k\mathbf{v_1} \mid k \in \mathbb{Z}\}$. Moreover, there are following results, which will be used in Sect. 4.

Lemma 1 (See Gómez et al., 2006 [6], Lemma 3). *Let $\mathbf{v_1}$ and $\mathbf{v_2}$ be the reduced basis vectors of $\mathcal{L}$ by the Gauss algorithm and $\mathbf{x} \in \mathcal{L}$. For the unique pair of integers (α, β) that satisfies $\mathbf{x} = \alpha\mathbf{v_1} + \beta\mathbf{v_2}$, we have*

$$\|\alpha\mathbf{v_1}\| \leq \frac{2}{\sqrt{3}}\|\mathbf{x}\|,\ \|\beta\mathbf{v_2}\| \leq \frac{2}{\sqrt{3}}\|\mathbf{x}\|.$$

Lemma 2 (See Gómez et al., 2006 [6], Lemma 5). *Let $\{\mathbf{u}, \mathbf{v}\}$ be a reduced basis of a 2-rank lattice $\mathcal{L}$ in $\mathbb{R}^r$. Then we have*

$$det(\mathcal{L}) \leq \| \mathbf{u} \| \| \mathbf{v} \| \leq \frac{2}{\sqrt{3}} det(\mathcal{L}).$$

The reduced basis calculation in two-rank lattices is far from being obtained for general lattices. The subsequently proposed reduction definitions all have to make a choice between computational efficiency and good reduction performances. The distinguished LLL algorithm takes a good balance, outputting a basis reduced enough for many applications in polynomial time.

Lemma 3 [17]**.** *Let $\mathcal{L}$ be a lattice. In polynomial time, the LLL algorithm outputs reduced basis vectors $\mathbf{v_1}, \ldots, \mathbf{v}_\omega$ that satisfy*

$$\|\mathbf{v_1}\| \leq \|\mathbf{v_2}\| \leq \cdots \leq \|\mathbf{v_i}\| \leq 2^{\frac{\omega(\omega-1)}{4(\omega+1-i)}} \det(\mathcal{L})^{\frac{1}{\omega+1-i}}, 1 \leq i \leq \omega.$$

2.2 Finding Small Roots

Coppersmith gave rigorous methods for extracting small roots of modular univariate polynomials and bivariate integer polynomials. These methods can be heuristically extended to multivariate cases. Howgrave-Graham's [11] reformulation to Coppersmith' s method is widely adopted by researchers for cryptanalysis.

Lemma 4 [11]**.** *Let $g(x_1, x_2) \in \mathbb{Z}[x_1, x_2]$ be an integer polynomial that consists of at most ω nonzero monomials. Define the norm of $g(x_1, x_2) =: \sum b_{i_1,i_2} x_1^{i_1} x_2^{i_2}$ as the Euclidean norm of its coefficient vector, namely,*

$$\|g(x_1, x_2)\| = \sqrt{\sum b_{i_1,i_2}{}^2}.$$

Suppose that

1. *$g(x_1^{(0)}, x_2^{(0)}) = 0 \pmod{N}$, for $|x_1^{(0)}| < X_1$, $|x_2^{(0)}| < X_2$;*
2. *$\|g(X_1 x_1, X_2 x_2)\| < \frac{N}{\sqrt{\omega}}$.*

Then $g(x_1^{(0)}, x_2^{(0)}) = 0$ holds over integers.

Combining Howgrave-Graham's lemma with the LLL algorithm, one can deduce that if

$$2^{\frac{\omega(\omega-1)}{4(\omega+1-i)}} \det(\mathcal{L})^{\frac{1}{\omega+1-i}} < \frac{N}{\sqrt{\omega}},$$

the polynomials corresponding to the shortest i reduced basis vectors hold over integers. Neglecting the low order terms which are independent on N, the above condition can be simplified as

$$\det(\mathcal{L}) < N^{\omega+1-i}. \tag{1}$$

After obtaining enough equations over integers, one can extract the shared roots by either resultant computation or Gröbner basis technique.

We need the following assumption through our analyses, which is widely adopted in previous works.

Assumption 1. *The Gröbner basis computations for the polynomials corresponding to the first few LLL-reduced basis vectors produce non-zero polynomials.*

3 Recovering the Addends from $N = rA^2 + sB^2$

In this section, we first describe the trick for finding the small roots of polynomial $f_N(x, y) = a_1x^2 + a_2y^2 + a_3x + a_4y + a_0$. Next, we address the problem of recovering the decomposition of a given number $N = rA^2 + sB^2$ only from its approximations to its addends A, B, where N, r, s are public positive integers. Then, we discuss how to achieve A and B when the least significant bits of them are revealed. Both of these two attacks can be transformed into solving the studied polynomial $f_N(x, y)$.

3.1 Solving Polynomial $f_N(x, y)$

Without loss of generosity, we assume $a_1 = 1$ since we can make it by multiplying f_N with a_1^{-1} *mod* N. If this inverse does not exist, one can factorize N. Set

$$f(x, y) = a_1^{-1} f_N(x, y) \ mod \ N.$$

Next, we find the small roots of $f(x, y)$ by Coppersmith's method. Build shifting polynomials

$$g_{k,i,j}(x, y) = x^i y^j f^k(x, y) N^{m-k},$$

where $i = 0, 1; k = 0, ..., m - i; j = 0, ..., 2(m - k - i)$. Obviously,

$$g_{k,i,j}(x, y) \equiv 0 \ mod \ N^m.$$

Construct a lattice $\mathcal{L}$ using the coefficient vectors of $g_{k,i,j}(xX, yY)$ as basis vectors. We sort the polynomials $g_{k,i,j}(xX, yY)$ and $g_{k',i',j'}(xX, yY)$ according to the lexicographical order of vectors (k, i, j) and (k', i', j'). In this way, we can

Table 1. Example of the lattice formed by vectors $g_{k,i,j}(xX, yY)$ when $m = 2$. The upper triangular part of this matrix is all zero, so omitted here, and the non-zero items below the diagonal are marked by $*$.

	1	y	y^2	y^3	y^4	x	xy	xy^2	x^2	x^2y	x^2y^2	x^3	x^4
$g_{0,0,0}$	N^2												
$g_{0,0,1}$		YN^2											
$g_{0,0,2}$			Y^2N^2										
$g_{0,0,3}$				Y^3N^2									
$g_{0,0,4}$					Y^4N^2								
$g_{0,1,0}$						XN^2							
$g_{0,1,1}$							XYN^2						
$g_{0,1,2}$								XY^2N^2					
$g_{1,0,0}$	$*$	$*$	$*$			$*$			X^2N				
$g_{1,0,1}$		$*$	$*$	$*$			$*$			X^2YN			
$g_{1,0,2}$			$*$	$*$	$*$			$*$	$*$		X^2Y^2N		
$g_{1,1,0}$						$*$	$*$	$*$	$*$			X^3N	
$g_{2,0,0}$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	X^4

ensure that each of our shifting polynomials introduces one and only one new monomial, which gives a lower triangular structure for $\mathcal{L}$. We give an example for $m = 2$ in the following Table 1.

Then its determinant can be easily calculated as products of the entries on the diagonal as $det(\mathcal{L}) = X^{S_X} Y^{S_Y} N^{S_N}$ as well as its dimension ω where

$$\omega = \sum_{i=0}^{1} \sum_{k=0}^{m-i} \sum_{j=0}^{2(m-k-i)} 1 = 2m^2 + 2m + 1 = 2m^2 + o(m^2).$$

$$S_x = \sum_{i=0}^{1} \sum_{k=0}^{m-i} \sum_{j=0}^{2(m-k-i)} (2k + i) = \frac{1}{3} m(4m^2 + 3m + 2) = \frac{4}{3} m^3 + o(m^3).$$

$$S_y = \sum_{i=0}^{1} \sum_{k=0}^{m-i} \sum_{j=0}^{2(m-k-i)} j = \frac{1}{3} m(4m^2 + 3m + 2) = \frac{4}{3} m^3 + o(m^3).$$

$$S_N = \sum_{i=0}^{1} \sum_{k=0}^{m-i} \sum_{j=0}^{2(m-k-i)} (m - k) = \frac{2}{3} m(2m^2 + 3m + 1) = \frac{4}{3} m^3 + o(m^3).$$

Put these relevant values into inequality $\det(\mathcal{L}) < N^{m\omega}$. After some basic calculations, we gain the bound

$$XY < N^{\frac{1}{2}}.$$

When $X = Y$, which means the two unknowns are balanced, the above result is

$$X = Y < N^{\frac{1}{4}}.$$

We summarize our result in the following theorem.

Theorem 1. *Let N be a sufficiently large composite integer of unknown factorization. Given a bivariate polynomial $f_N(x, y) = a_1x^2 + a_2x + a_3y^2 + a_4y + a_0$ mod N, where $|x| \leq X$, $|y| \leq Y$. Under Assumption 1, if*

$$XY < N^{\frac{1}{2}},$$

one can extract all the solutions (x, y) of equation $f_N(x, y) \equiv\ 0 \pmod N$ in polynomial time.

3.2 Recovering a Decomposition from Approximations

In this subsection, we describe the method to recover A, B of $N = rA^2 + sB^2$ from their approximations.

Supposing that positive integers r and s are given. Set $N = rA^2 + sB^2$, where A, B are balanced addends, and A_0, B_0 are the approximations to A, B, that is $A = A_0 + x$ and $B = B_0 + y$, where x, y are bounded by Δ. Then, one can recover A and B according to Theorem 1 when

$$\Delta < N^{\frac{1}{4}}.$$

The concrete analysis is as follows. Note that

$$N = r(A_0 + x)^2 + s(B_0 + y)^2, \tag{2}$$

which gives rise to a bivariate modular polynomial

$$f_1(x, y) = rx^2 + sy^2 + 2A_0rx + 2B_0sy + rA_0^2 + sB_0^2 \equiv 0 \; mod \; N,$$

this is exactly the same type of the polynomial we discussed in Sect. 3.1. So we gain the result $\Delta < N^{\frac{1}{4}}$ simply by substituting both X and Y appeared in Theorem 1 to Δ.

The experimental results to support the above analysis is displayed in Table 2, which matches well with the derived theoretical bound.

Table 2. Experimental results for error bound $\Delta = \frac{1}{4}$ with 512 bit N

N (bits)	m	dim	$log_N \Delta$	LLL (seconds)	Gröbner (seconds)
512	5	61	0.227	12.901	15.631
	6	85	0.230	49.172	606.360
	7	113	0.233	187.076	517.549
	8	145	0.235	566.471	3204.339
	9	181	0.236	1512.586	5538.002
	10	221	0.237	3430.463	out of memory

Table 3. Experimental results for Remark 1 with 512 bit N

N (bits)	m	dim	$log_N \Delta$	LLL (seconds)	Gröbner (seconds)
512	4	28	0.130	0.842	0.265
	5	36	0.132	3.806	0.842
	6	45	0.133	14.914	1.420
	8	66	0.135	143.349	11.532

Table 4. Experimental results for different modulus with 1024 bit N

N (bits)	M	m	dim	$log_N \Delta$	LLL (seconds)	Gröbner (seconds)
1024	N − 1	6	85	0.23	582.258	144.005
	2N − 1	6	85	0.23	587.046	145.440
	$N^2 - 1$	6	85	0.23	5917.165	1159.431

Remark 1. Gutierrez et al. discussed the same problem in [7]. They arranged Eq. (2) to a bivariate integer polynomial as follows,

$$f_1'(x, y) = rx^2 + sy^2 + 2A_0rx + 2B_0sy + rA_0^2 + sB_0^2 - N. \tag{3}$$

By directly applying Coppersmith's theorem [3], their derived error bound is only $N^{1/6}$. We do experiments for their method, part of the results are displayed in Table 3. The experimental results show that our method works much better.

Coppersmith's original method [3] for solving bivariate integer polynomial is difficult to understand. Coron [13] first reformulated Coppersmith's work and the key idea of which can be described as follows, choosing a proper integer R, and transforming the situation into finding a small root modulo R. Then, by applying LLL algorithm, a polynomial with small coefficients can be found out, which is proved to be algebraically independent with the original equation.

Our approach described above also transforms the integer equation into a modular polynomial. The difference between our method and Coppersmith's theorem [3] lies in the construction of shifting polynomials. We take use of the information of the power of the original polynomial. Although we didn't prove that the obtained polynomial with small coefficients is algebraically independent with the original polynomial, which is true in most cases during the experiments.

Remark 2. We studied different situations to transform Eq. (3) into modular ones as the modulus varies. For instance $q(x, y) = f_1(x, y) + M \equiv 0 \; mod \; (N + M)$. The experimental results for different M are shown in Table 4.

Specifically, we also consider non-constant modular polynomial

$$f_2(x, y) = rx^2 + sy^2 + 2A_0rx + 2B_0sy \equiv 0 \; mod \; (N - rA_0^2 - sB_0^2). \tag{4}$$

In this way, the corresponding theoretical error bound for recovering the addends from their approximations is $N^{1/6}$(please refer to Appendix A for the detailed analyses). However, the experimental results show a much better performance, which is displayed in Table 5.

3.3 Recovering a Decomposition from Non-approximations

Actually, the most significant bits exposure attack of A and B can be viewed as a special case of the above problem (recovering a a sum of two squares from its approximations). In this subsection, we consider the case when the least significant bits of A, B are leaked.

Given r, s are positive integers, set $N = rA^2 + sB^2$, where A, B are balanced addends. When half bits of A and B in the LSBs are intercepted, one can recover A, B according to Theorem 1.

Suppose $A = xM + A_0$, $B = yM + B_0$, where M, A_0 and B_0 are the gained integers, and x, y refers to the unknown parts. Then we have the following relation

$$N = r(xM + A_0)^2 + s(yM + B_0)^2,$$

which can be expanded to a bivariate modular polynomial

$$f_3(x, y) = rM^2x^2 + sM^2y^2 + 2rA_0Mx + 2sB_0My + rA_0^2 + sB_0^2 \equiv \ 0 \ mod \ N.$$

Set the upper bound for x and y as Δ_1 and put it into Theorem 1, we get $\Delta_1 < N^{\frac{1}{4}}$. Since

$$M = \frac{A - A_0}{x} > \frac{A - A_0}{N^{\frac{1}{4}}} \approx \frac{A}{N^{\frac{1}{4}}} \approx \frac{N^{\frac{1}{2}}}{N^{\frac{1}{4}}} = N^{\frac{1}{4}},$$

From these analyses, we get that half information from A and B can reveal the whole knowledge of both addends, no matter the leaked bits are LSBs or MSBs.

Table 5. Experimental results for Remark 2 with 512 bit N

N (bits)	m	dim	$log_N \Delta$	LLL (seconds)	Gröbner (seconds)
512	2	12	0.16	0.001	0.001
	3	24	0.19	0.016	0.14
	4	40	0.20	0.406	1.888
	5	60	0.21	2.558	45.490
	7	112	0.22	57.954	2028.294

4 A Better Method for Recovering the Addends

In this section, we reduce the problem of recovering a sum of two squares decomposition to seeking the coordinates of a desired vector in a certain lattice. Then we can find these coordinates by applying Coppersmith's method to solve a type of modular polynomials where the concerned monomials are x^2, y^2, xy, x, y and 1. Dealt this way, the theoretical error tolerance can be improved to $N^{1/3}$, and the involved lattices in this approach possess much smaller dimensions compared to the ones in Sect. 3.

4.1 The Reduction of Recovering the Addends

From the initial key relation $N = r(A_0 + x)^2 + s(B_0 + y)^2$ we have

$$2rA_0x + 2sB_0y + rx^2 + sy^2 = N - rA_0^2 - sB_0^2. \tag{5}$$

Hence, the recovery of vector

$$\mathbf{e} := (X_1, X_2, X_3) = ((r+s)\mathit{\Delta} x, (r+s)\mathit{\Delta} y, rx^2 + sy^2)$$

solves the problem. Here $\mathit{\Delta}$ represents the upper bound for x and y. It is not hard to see that vector $\mathbf{e}$ is in a shifted lattice $\mathbf{c} + \mathcal{S}$, $\mathbf{c} = (c_1, c_2, c_3) \in \mathbb{Z}^3$, where $(\frac{c_1}{(r+s)\mathit{\Delta}}, \frac{c_2}{(r+s)\mathit{\Delta}}, c_3)$ is a particular solution of (5) and $\mathcal{S}$ is a two-dimensional lattice

$$\begin{pmatrix} (r+s)\mathit{\Delta} & 0 & -2A_0r \\ 0 & (r+s)\mathit{\Delta} & -2B_0s \end{pmatrix}.$$

According to Minkowski's theorem [15], when $||\mathbf{e}|| < \sqrt{2}\sqrt{det(\mathcal{S})}$, one can recover $\mathbf{e}$ by solving the closet vector problem. Further, the norm of $\mathbf{e}$ satisfies $||\mathbf{e}|| \leq \sqrt{3}(r+s)\mathit{\Delta}^2$, and $det(\mathcal{S}) \geq 2(r+s)\mathit{\Delta}\sqrt{\frac{min(r,s)*N}{2}}$ with condition $min(r,s) * N \geq 4\sqrt{N}\mathit{\Delta}(r^{3/2} + s^{3/2})$. These constraints give rise to the error bound $\mathit{\Delta} < N^{1/6}$, as discussed in [7].

Next, we present our analysis for the case when $\mathit{\Delta} > N^{1/6}$. Here, we tag $\mathbf{f} = ((r+s)\mathit{\Delta} f_1, (r+s)\mathit{\Delta} f_2, f_3)$ as the output of the CVP algorithm on $\mathcal{S}$, and use $\{\mathbf{u} = ((r+s)\mathit{\Delta} u_1, (r+s)\mathit{\Delta} u_2, u_3), \mathbf{v} = ((r+s)\mathit{\Delta} v_1, (r+s)\mathit{\Delta} v_2, v_3)\}$ to denote the Gauss reduced basis for $\mathcal{S}$. Then $\mathbf{e} = \mathbf{f} + \alpha\mathbf{u} + \beta\mathbf{v}$, where α, β represent the corresponding coordinates of vector $\mathbf{e} - \mathbf{f}$ in lattice $\mathcal{S}$. Thus, the problem is converted to finding the parameters α and β, which satisfy equation

$$\begin{aligned} &2A_0r(f_1 + \alpha u_1 + \beta v_1) + 2B_0s(f_2 + \alpha u_2 + \beta v_2) \\ &+ r(f_1 + \alpha u_1 + \beta v_1)^2 + s(f_2 + \alpha u_2 + \beta v_2)^2 + rA_0^2 + sB_0^2 - N = 0. \end{aligned} \tag{6}$$

We first derive the upper bounds for the unknowns α, β. Since $\mathbf{e} - \mathbf{f} = \alpha\mathbf{u} + \beta\mathbf{v}$, from Lemma 1, we get

$$||\alpha\mathbf{u}||||\beta\mathbf{v}|| \leq \frac{2}{\sqrt{3}}||\mathbf{e} - \mathbf{f}|| \leq 4(r+s)\mathit{\Delta}^2.$$

Thus, $|\alpha| \leq \frac{4(r+s)\Delta^2}{||\mathbf{u}||}$, $|\beta| \leq \frac{4(r+s)\Delta^2}{||\mathbf{v}||}$. Further, according to Lemma 2, there is $det(\mathcal{S}) \leq ||\mathbf{u}||||\mathbf{v}|| \leq \frac{2}{\sqrt{3}}det(\mathcal{S})$. Then we have

$$|\alpha||\beta| \leq \frac{4(r+s)\Delta^2}{det(\mathcal{S})} \leq c_1\Delta^{3/2}N^{-1/4},$$

where $c_1 = 2^{7/4}(r+s)^{1/2}min(r,s)^{-1/4}$ is a constant.

Notice that Eq. (6) can be arranged to

$$\begin{aligned}&(ru_1^2+su_2^2)\alpha^2+(rv_1^2+sv_2^2)\beta^2+2(ru_1v_1+su_2v_2)\alpha\beta+2(A_0ru_1\\ &+B_0su_2+rf_1u_1+sf_2u_2)\alpha+2(A_0rv_1+B_0sv_2+rf_1v_1+sf_2v_2)\beta\\ &+2A_0rf_1+2B_0sf_2+rf_1^2+sf_2^2+rA_0^2+sB_0^2\equiv\ 0\ mod\ N,\end{aligned} \tag{7}$$

which represents a certain type of modular polynomials consisting of monomials x^2, y^2, xy, x, y and 1. Next, we describe our analysis for solving such polynomials.

4.2 Solving a Certain Type of Modular Polynomials

Let $f'_N(x,y) = b_1x^2 + b_2y^2 + b_3xy + b_4x + b_5y + b_0\ mod\ N$. Assume $b_1 = 1$, otherwise, set

$$f'(x,y) = b_1^{-1}f'_N(x,y)\ mod\ N.$$

If the inverse $b_1^{-1}\ mod\ N$ does not exist, one can factorize N. Next, we use Coppersmith's method to find the small roots of this polynomial. Build shifting polynomials $h_{k,i,j}(x,y)$ which possess the same roots modular N^m with $f'(x,y) \equiv 0\ mod\ N$ as follows:

$$h_{k,i,j}(x,y) = x^iy^jf'^k(x,y)N^{m-k},$$

where $i = 0,1; k = 0,...,m-i; j = 0,...,2(m-k)-i$.

Construct a lattice $\mathcal{L}'$ using the coefficient vectors of $h_{k,i,j}(xX,yY)$ as basis vectors. We sort the polynomials $h_{k,i,j}(xX,yY)$ and $h_{k',i',j'}(xX,yY)$ according to lexicographical order of vectors (k,i,j) and (k',i',j'). Therefore, we can ensure that each of our shifting polynomials introduces one and only one new monomial, which gives a triangular structure for $\mathcal{L}'$.

Then the determinant of $\mathcal{L}'$ can be easily calculated as products of the entries on the diagonal as $det(\mathcal{L}') = X^{S_X}Y^{S_Y}N^{S_N}$ as well as its dimension ω where

$$\omega = \sum_{k=0}^{m-i}\sum_{i=0}^{1}\sum_{j=0}^{2(m-k)-i} 1 = 2m^2 + o(m^2),$$

$$S_X = \sum_{k=0}^{m-i}\sum_{i=0}^{1}\sum_{j=0}^{2(m-k)-i} (2k+i) = \frac{4}{3}m^3 + o(m^3),$$

$$S_Y = \sum_{k=0}^{m-i}\sum_{i=0}^{1}\sum_{j=0}^{2(m-k)-i} j = \frac{4}{3}m^3 + o(m^3),$$

$$S_N = \sum_{k=0}^{m-i}\sum_{i=0}^{1}\sum_{j=0}^{2(m-k)-i} (m-k) = \frac{4}{3}m^3 + o(m^3).$$

Put these relevant values into inequality $\det(\mathcal{L}') < N^{m\omega}$. After some basic calculations, we gain the bound

$$XY < N^{\frac{1}{2}}.$$

We summarize our result in the following theorem.

Theorem 2. *Let N be a sufficiently large composite integer of unknown factorization and $f'_N(x,y) = b_1x^2 + b_2x + b_3y^2 + b_4y + b_5xy + b_0$ mod N be a bivariate modular polynomial, where $|x| \leq X$, $|y| \leq Y$. Under Assumption 1, if*

$$XY < N^{\frac{1}{2}},$$

one can extract all the solutions (x,y) of equation $f'_N(x,y) \equiv 0 \pmod N$ in polynomial time.

Next, we use the above method to solve Eq. (7), and then recover the unknown addends.

4.3 Recover the Addends

Notice that Eq. (7) is exactly the same type of polynomial discussed in Sect. 4.2. Put the derived upper bounds for $|\alpha||\beta|$ in Sect. 4.1 into Theorem 2,

$$|\alpha||\beta| \leq c_1 \Delta^{3/2} N^{-1/4} \leq N^{1/2}.$$

Solve this inequality, omit the constant terms, and we obtain the optimized bound for the approximation error terms

$$\Delta < N^{\frac{1}{3}}. \tag{8}$$

Compared to Sect. 3, this method performs much better in practice since the dimensions of the involved lattices are much smaller when the error bounds are the same. We present the comparison results in Table 6, where one can see a remarkable improvement in the performing efficiency.

Remark 3. As in Sect. 3, we also analyzed the case when transforming Eq. (6) into a non-constant modular polynomial. The corresponding error bound is then $N^{1/4}$. Table 7 is the experimental results for this situation. Please refer to Appendix B for the detailed analysis.

5 Conclusions and Discussions

We revisit the problem of recovering the two addends in this paper. Our first algorithm improves Gutierrez et al.'s first result $N^{1/6}$ to $N^{1/4}$ by transforming the derived polynomial into a modular one. Then we improve this bound to $N^{1/3}$ in theory by reducing the problem of recovering a sum of two squares decomposition to seeking the coordinates of a desired vector in a certain lattice.

Table 6. A comparison between Sect. 4 (the left part datas) and Sect. 3 (the right part datas)

N (bits)	$log_N \Delta$	m	dim	LLL (seconds)	Gröbner (seconds)	m'	dim'	LLL' (seconds)	Gröbner' (seconds)
1024	0.19	1	6	0.016	0.001	2	13	0.047	0.031
	0.20	2	15	0.187	0.109	3	25	1.248	0.406
	0.21	2	15	0.172	0.109	3	25	1.030	0.967
	0.22	2	15	0.187	0.140	4	41	14.383	3.416
512	0.23	4	45	6.334	11.591	6	85	49.172	606.360
	0.235	5	66	47.612	68.391	8	145	566.471	3204.339
	0.236	6	91	229.789	579.091	9	181	1512.586	5538.002
	0.237	7	120	949.094	3410.151	10	221	3430.463	out of memory
	0.238	7	120	855.868	1696.823	—	—	—	-
	0.239	8	153	2852.619	out of memory	—	—	—	-

Table 7. Experimental results for Remark 3 with 512 bit N

N (bits)	m	dim	$log_N \Delta$	LLL (seconds)	Gröbner (seconds)
512	2	14	0.21	0.031	0.016
	3	27	0.22	0.328	0.187
	6	90	0.23	180.930	188.434

J.Gutierrez et al. did similarly in [7], and their optimized bound is $N^{1/4}$. Our second approach performs much better than the first one since the dimension of the required lattice is much smaller when the same error bounds are considered. The tricks to solve the derived polynomials in Sects. 3 and 4 are similar, both of which transform integer relations to modular polynomials. We study four kinds of modular polynomials in our work (two types are discussed in Remarks 2 and 3). The tricks for solving these polynomials may find other applications in cryptanalysis.

We do experiments to testify the deduced results. The tests are done in Magma on a PC with Intel(R) Core(TM) Quad CPU (3.20 GHz, 4.00 GB RAM, Windows 7). These datas well support our analyses, however, as the error terms go larger, the dimensions of the required lattices are huger. The time, memory costs also increase greatly, which stops our experiment at a not good enough point. Hope people who are interested in this problem can bring us further supports for the experiments.

Acknowledgements. The authors would like to thank anonymous reviewers for their helpful comments and suggestions. The work of this paper was partially supported by National Natural Science Foundation of China (No. 61170289) and the National Key Basic Research Program of China (2013CB834203).

A Analysis for Remark 2

In this part, we give the details to show that when dealing with Eq. (3) as a non-constant modular polynomial (4), the corresponding error bound is $N^{1/6}$.

First, we display the trick for finding the small roots of $f_2(x, y) = rx^2 + sy^2 + 2A_0rx + 2B_0sy \equiv 0 \ mod \ (N - rA_0^2 - sB_0^2)$. Set $M = N - rA_0^2 - sB_0^2$ as the modulus. The shifting polynomials for this equation can be constructed as

$$\begin{cases} g_{k,i}^1(x, y) = y^i M^m, \\ i = 1, ..., 2m; \\ g_{k,i}^2(x, y) = x^j y^i f_3^k(x, y) M^{m-k}, \\ k = 0, ..., m-1; j = 1, 2; i = 0, ..., 2(m-k-1); \end{cases}$$

Suppose $|x| \leq X = N^\delta, |y| \leq Y = N^\delta$, then $M \approx N^{\frac{1}{2}+\delta}$. Similarly, the coefficients of $g^1(xX, yY), g^2(xX, yY)$ can be arranged as a lower triangular lattice $\mathcal{L}_1$, whose determinant can be easily calculated as $det(\mathcal{L}_1) = X^{S_X} Y^{S_Y} M^{S_M}$, where

$$\begin{aligned} \omega &= 2m^2 + 2m = 2m^2 + o(m^2). \\ S_X &= \frac{1}{3}m(4m^2 + 3m + 2) = \frac{4}{3}m^3 + o(m^3). \\ S_Y &= \frac{1}{3}m(4m^2 + 3m + 2) = \frac{4}{3}m^3 + o(m^3). \\ S_M &= \frac{1}{3}m(4m^2 + 9m - 1) = \frac{4}{3}m^3 + o(m^3). \end{aligned}$$

Put these values into inequality $det(\mathcal{L}_1) \leq M^{m\omega}$, we obtain $\delta \leq \frac{1}{6}$, which means that the error bound derived by this method is

$$\Delta \leq N^{\frac{1}{6}},$$

a poorer bound compared to $N^{\frac{1}{4}}$. The experimental results in Table 5 show that this method works much better in practice than in theoretic analysis, although still weaker than the result in Sect. 3.2.

B Analysis for Remark 3

Notice that the problem of finding coordinates for vector $\mathbf{e} - \mathbf{f}$ can also be transformed into solving a non-constant modular equation

$$\begin{aligned} q(\alpha, \beta) = &(ru_1^2 + su_2^2)\alpha^2 + (rv_1^2 + sv_2^2)\beta^2 + 2(ru_1v_1 + su_2v_2)\alpha\beta \\ &+ (2rf_1u_1 + 2sf_2u_2 - u_3)\alpha + (2rf_1v_1 + 2rf_2v_2 - v_3)\beta \\ &\equiv 0 \ mod \ (N - 2rA_0f_1 - 2sB_0f_2 - rf_1^2 - sf_2^2 - rA_0^2 - sB_0^2) \end{aligned}$$

Set $M = |N - 2rA_0f_1 - 2sB_0f_2 - rf_1^2 - sf_2^2 - rA_0^2 - sB_0^2|$ as the modulus. Then the problem reduced to solving

$$q'(x,y) = x^2 + b_2y^2 + b_3xy + b_4x + b_5y \equiv 0 \ mod \ M.$$

Here we assume that $q'(x,y)$ is a monic irreducible polynomial, since we can make it satisfied by multiplying the modular inverse term. We apply Coppersmith's method to solve this polynomial. The shifting polynomials can be constructed as

$$\begin{cases} g_{k,i}^1(x,y) = y^iM^m, \\ i = 1, ..., 2m; \\ g_{k,i}^2(x,y) = y^iq'^k(x,y)M^{m-k}, \\ k = 1, ..., m, i = 0, ..., 2(m-k); \\ g_{k,i}^3(x,y) = xy^iq'^k(x,y)M^{m-k}, \\ k = 0, ..., m-1, i = 0, ..., 2(m-k)-1; \end{cases}$$

From the former analysis, we know that $|x|, |y| \leq \Delta^{3/2}N^{-1/4} = X = Y$, and $M \approx \Delta^2$. Similarly, the coefficients of $g^1(xX, yY), g^2(xX, yY)$ and $g^3(xX, yY)$ can be arranged as a lower triangular lattice $\mathcal{L}_2$, whose determinant can be easily calculated as $det(\mathcal{L}_2) = X^{S_X}Y^{S_Y}M^{S_M}$, where

$$\begin{aligned} \omega &= 2m^2 + 3m = 2m^2 + o(m^2). \\ S_X &= \frac{2}{3}m(2m^2 + 3m + 1) = \frac{4}{3}m^3 + o(m^3). \\ S_Y &= \frac{2}{3}m(2m^2 + 3m + 1) = \frac{4}{3}m^3 + o(m^3). \\ S_M &= \frac{1}{6}m(8m^2 + 15m + 1) = \frac{4}{3}m^3 + o(m^3). \end{aligned}$$

Put these values into inequality $det(\mathcal{L}_2) \leq M^{m\omega}$, we gain the corresponding error bound

$$\Delta \leq N^{\frac{1}{4}}.$$

References

1. Aono, Y.: A new lattice construction for partial key exposure attack for RSA. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 34–53. Springer, Heidelberg (2009)
2. Bauer, A., Vergnaud, D., Zapalowicz, J.-C.: Inferring sequences produced by nonlinear pseudorandom number generators using Coppersmith's methods. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 609–626. Springer, Heidelberg (2012)
3. Coppersmith, D.: Small solutions to polynomial equations, and low exponent RSA vulnerabilities. J. Cryptol. **10**(4), 233–260 (1997)

4. 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)
5. 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)
6. Gomez, D., Gutierrez, J., Ibeas, A.: Attacking the pollard generator. IEEE Trans. Inf. Theor. **52**(12), 5518–5523 (2006)
7. Gutierrez, J., Ibeas, Á., Joux, A.: Recovering a sum of two squares decomposition. J. Symb. Comput. **64**, 16–21 (2014)
8. Hardy, K., Muskat, J.B., Williams, K.S.: A deterministic algorithm for solving $n = fu^2 + gv^2$ in coprime integers u and v. J. Math. Comput. **55**, 327–343 (1990)
9. 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)
10. 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)
11. Howgrave-Graham, N.: Finding small roots of univariate modular equations revisited. In: Darnell, M. (ed.) Crytography and Coding. LNCS, vol. 1355, pp. 131–142. Springer, Heidelberg (1997)
12. Jochemsz, E., May, A.: A strategy for finding roots of multivariate polynomials with new applications in attacking RSA variants. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 267–282. Springer, Heidelberg (2006)
13. Coron, J.-S.: Finding small roots of bivariate integer polynomial equations revisited. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 492–505. Springer, Heidelberg (2004)
14. Kakvi, S.A., Kiltz, E., May, A.: Certifying RSA. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 404–414. Springer, Heidelberg (2012)
15. Kannan, R.: Minkowski's convex body theorem and integer programming. Math. Oper. Res. **12**(3), 415–440 (1987)
16. Kiltz, E., O'Neill, A., Smith, A.: Instantiability of RSA-OAEP under chosen-plaintext attack. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 295–313. Springer, Heidelberg (2010)
17. Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring polynomials with rational coefficients. Math. Ann. **261**(4), 515–534 (1982)
18. May, A.: Using LLL-reduction for solving RSA and factorization problems. In: Nguyen, P.Q., Vallée, B. (eds.) The LLL Algorithm: Survey and Applications. ISC, pp. 315–348. Springer, Heidelberg (2010)
19. Sarkar, S.: Reduction in lossiness of RSA trapdoor permutation. In: Bogdanov, A., Sanadhya, S. (eds.) SPACE 2012. LNCS, vol. 7644, pp. 144–152. Springer, Heidelberg (2012)
20. Sarkar, S., Maitra, S.: Cryptanalysis of RSA with two decryption exponents. Inf. Process. Lett. **110**, 178–181 (2010)
21. Tosu, K., Kunihiro, N.: Optimal bounds for multi-prime ϕ-hiding assumption. In: Mu, Y., Seberry, J., Susilo, W. (eds.) ACISP 2012. LNCS, vol. 7372, pp. 1–14. Springer, Heidelberg (2012)
22. van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully homomorphic encryption over the integers. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 24–43. Springer, Heidelberg (2010)