Finding Small Solutions of a Class of Simultaneous Modular Equations and Applications to Modular Inversion Hidden Number Problem and Inversive Congruential Generator

Jun Xu[1,2], Lei Hu[1,2], Zhangjie Huang[1,2], and Liqiang Peng[1,2]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[2] Data Assurance and Communications Security Research Center, Chinese Academy of Sciences, Beijing 100093, China
{jxu,hu,zhjhuang,lqpeng}@is.ac.cn

Abstract. In this paper we revisit the modular inversion hidden number problem and the inversive congruential pseudo random number generator and consider how to more efficiently attack them in terms of fewer samples or outputs. We reduce the attacking problem to finding small solutions of systems of modular polynomial equations of the form $a_i+b_ix_0+c_ix_i+x_0x_i = 0 \pmod p$, and present two strategies to construct lattices in Coppersmith's lattice-based root-finding technique for the solving of the equations. Different from the choosing of the polynomials used for constructing lattices in previous methods, a part of polynomials chosen in our strategies are linear combinations of some polynomials generated in advance and this enables us to achieve a larger upper bound for the desired root. Applying the solving of the above equations to analyze the modular inversion hidden number problem, we put forward an explicit result of Boneh et al. which was the best result so far, and give a further improvement in the involved lattice construction in the sense of requiring fewer samples. Our strategies also give a method of attacking the inversive congruential pseudo random number generator, and the corresponding result is the best up to now.

Keywords: modular inversion hidden number problem, inversion congruential pseudo random number generator, lattice, LLL algorithm, Coppersmith's technique

1 Introduction

In 1996, Boneh and Venkatesan introduced the concept of Hidden Number Problem (HNP) to show the bit security of the Diffie-Hellman key exchange agreement [7]. Later, Nguyen and Shparlinski proposed a lattice-based study on HNP and analyzed the security of the Digital Signature Algorithm (DSA) under the circumstance with the nounce used is partially known [24]. Shparlinski put forward

the generic definition of HNP, and summarized more variants and applications of HNP [27]. Recently, Galindo and Vivek showed that attacking the stateful decryption scheme of the leakage-resilient cryptosystem in [20] can be reduced to a new variant of HNP [14].

In 2001, Boneh et al. proposed the Modular Inversion Hidden Number Problem (ModInv-HNP), a variant of HNP, to construct the efficient algebraic pseudo random number generator (PRNG) and message authentication code (MAC) [6]. An instance of ModInv-HNP is to obtain the unknown $\alpha \in \mathbb{Z}_p$ given $n+1$ samples $(t_i, \mathrm{MSB}_\delta(\alpha_i))$, where the t_i are randomly chosen in $\mathbb{Z}_p \setminus \{-\alpha\}$ and and $\mathrm{MSB}_\delta(\alpha_i)$ is the δ most significant bits of $\alpha_i := (\alpha + t_i)^{-1} \bmod p$, which is treated as an integer in $\{1, \cdots, p-1\}$.

ModInv-HNP can be reduced to finding small solutions of simultaneous modular polynomial equations

$$f_i(x_0, x_i) := a_i + b_i x_0 + c_i x_i + x_0 x_i = 0 \pmod{p},\ 1 \le i \le n,$$

which are obtained from some equations corresponding to the samples $(t_0, \mathrm{MSB}_\delta(\alpha_0))$ and $(t_i, \mathrm{MSB}_\delta(\alpha_i))$. Similar research can be done for the case that the samples are given by $(t_i, \mathrm{LSB}_\delta(\alpha_i))$, where $\mathrm{LSB}_\delta(\alpha_i)$ is the δ least significant bits of α_i.

The Inversive Congruential Generator (ICG) proposed by Eichenauer and Lehn in [13] is an important kind of nonlinear number-theoretic pseudo random number generators, and pseudo random number generators are one of the most fundamental cryptographic primitives. There are extensive applications of ICG in Quasi-Monto Carlo simulation [25, 26] and public key schemes. In the cryptographic setting, nonlinear congruential generator is used to input a secret random seed $v_0 \in \mathbb{Z}_p$ for a given prime p into a nonlinear recurrence function $v_{i+1} = F(v_i) \bmod p$ to generate a sequence $(v_1, v_2, \cdots, v_{n+1})$ and output a random-looking sequence $(\mathrm{MSB}_\delta(v_1), \mathrm{MSB}_\delta(v_2), \cdots, \mathrm{MSB}_\delta(v_{n+1}))$ or $(\mathrm{LSB}_\delta(v_1), \mathrm{LSB}_\delta(v_2), \cdots, \mathrm{LSB}_\delta(v_{n+1}))$, as the desired pseudorandom number. In the case of ICG, the recurrence function is taken as $F(x) = ax^{-1} + b \bmod p$, a very strong goal of attacking ICG is to recover the seed v_0 when given $n+1$ outputs $\mathrm{MSB}_\delta(v_i)$ or $\mathrm{LSB}_\delta(v_i)$ $(1 \le i \le n+1)$. This problem can also be transformed into solving the small solutions of n modular polynomials of the form $a_i + b_i x_0 + c_i x_i + x_0 x_i \pmod{p}$.

1.1 Previous works

Boneh et al. presented a lattice based analysis for the ModInv-HNP problem by directly utilizing the polynomials f_i in the case that the number of samples is sufficiently large, and they can recover the hidden number α with a certain probability when $\delta/\log_2 p > \frac{2}{3}$ [6]. Furthermore, by utilizing the idea of Coppersmith's technique, they stated an improved result when $\delta/\log_2 p > \frac{1}{3}$ but no explicit lattice construction for this case was presented. Moreover, they only focused on sufficiently many samples. Due to these reasons, there are few cryptographic schemes based on ModInv-HNP so far. In 2012, Ling et al. reanalyzed ModInv-HNP and also made a direct use of these polynomials f_i to construct

another lattice [22]. It is interesting that they considered the general case of the number of samples, and also computed the possibility of recovering α. Their analysis requires a condition that $\delta/\log_2 p > (\frac{2}{3}+\varepsilon)$, where the positive real number ε satisfying $n = \lceil \frac{2}{9\varepsilon} \rceil$ and $n+1$ is the number of samples. The asymptotic result in the sense that n is sufficiently large is the same as the first result of Boneh et al. in [6]. Recently, Xu et al. observed that the algorithm of Ling et al. is not ideal when the number of samples is relatively small, and they proposed a heuristic lattice method by combining Coppersmith's lattice technique and the priority queue technique [28]. Their result requires that $\delta/\log_2 p \geq \left(\frac{1}{2} + \frac{1}{(n+2)!} + \varepsilon\right)$, where ε is a real number with very small absolute value depending on the dimension of the underlying lattice. This optimal result for $n = 1$ is the ration $\delta/\log_2 p > \frac{2}{3}$, and $\delta/\log_2 p \to \frac{2}{3}$, which is the same as the first result of Boneh et al. and the result of Ling et al. in the case of sufficiently many samples. However, when n is sufficiently large, the optimal result is $\delta/\log_2 p > \frac{1}{2}$ and $\delta/\log_2 p \to \frac{1}{2}$, that is weaker than the second result of Boneh et al..

For nonlinear pseudo random number generator and ICG, they were cryptanalyzed by researchers in [4, 3, 2]. Blackburn et al. used a lattice method independent of Coppersmith's technique and a linearization technique for pointing out that it can be attacked in polynomial time if sufficiently many bits of some consecutive values v_i are revealed [4, 3]. Later, in the case that the function F is known and $n+1$ outputs $\mathrm{MSB}_\delta(v_i)$ are revealed, Bauer et al. reduced the problem of attacking ICG to solving small solutions of n modular polynomials of the same form as the f_i and obtained the best result by coppersmith's technique for any positive n [2]. Concretely, when $n = 1$, the optimal result of Bauer et al. is $\delta/\log_2 p > \frac{2}{3}$ and $\delta/\log_2 p \to \frac{2}{3}$. When n is sufficiently large, the optimal result is $\delta/\log_2 p > \frac{1}{2}$ and $\delta/\log_2 p \to \frac{1}{2}$.

As an important method of cryptanalysis, Coppersmith's technique is extensively adopted in the field of public key cryptanalysis such as analyzing RSA and its variants [5], implicit integer factorization [23] and nonlinear PRNG [2]. It was firstly used for solving a single univariate modular or a single bivariate integer polynomial equation to attack RSA [9–11]. Then, it was extended to the case of a single multivariate equation in [19, 16]. But the lattice construction based on Coppersmith's technique is difficult to be designed for dealing with the case of simultaneous multivariate equations. Up to now, some concrete lattice constructions have been presented for different systems of multivariate equations, such as [1, 2, 17, 28].

1.2 Our Contribution

We give two new lattice methods for solving small roots of n multivariate modular polynomial equations $f_i(x_0, x_i) = a_i + b_i x_0 + c_i x_i + x_0 x_i = 0 \pmod{p}$ $(1 \leq i \leq n)$. Unlike the lattice constructions of the previous methods, we first define a new order of monomials and construct the corresponding monomial sets according to two strategies, then we generate suitable polynomials such that their leading monomial are in the predefined monomial sets, and finally, we construct the lattice using these generated polynomials.

In our basic strategy, a part of generated polynomials are linear combinations of several polynomials constructed in advance. For $n+1$ given samples in ModInv-HNP or $n+1$ outputs $\mathrm{MSB}_\delta(v_i)$ in ICG, the hidden number α or the secret seed v_0 can be recovered when

$$\delta/\log_2 p \geq \left(\frac{d(d+1)\sum_{s=0}^{d}\binom{n}{s} + 2\sum_{s=0}^{d} s\binom{n}{s}}{d(d+1)\sum_{s=0}^{d}\binom{n}{s} + 2(d+1)\sum_{s=0}^{d} s\binom{n}{s}} \right),$$

where the parameter d is a positive integer with $1 \leq d \leq n$. For any above-mentioned integers n and d, there is always $\delta/\log_2 p > \frac{1}{3}$. When n is sufficiently large, we can choose suitable d to let the result cover all previous ones. For example, taking $d = 1$, we get the first result of Boneh et al. in [6] and the result of Ling et al. in [22], taking $d = \lfloor \frac{n}{2} \rfloor$ we obtain the second result of Boneh et al, and taking $d = n$ we also get results in [2, 28]. For the general case of n, our result is better than the corresponding result in [22], but it is weaker than the optimal results in [2, 28].

In the extended strategy, we further improve the lattice construction. For a fixed monomial $x_0^{i_0} x_1^{i_1} \cdots x_n^{i_n}$ in the extended monomial set, we firstly regard it as the product of some monomials, which are divided according to values $i_0, i_1, \cdots, i_n$. Then, we construct a polynomial that is the product of several polynomials generated by utilizing the basic strategy for the corresponding monomial. We get the following result that the hidden number α or the secret seed v_0 can be recovered when

$$\delta/\log_2 p \geq \begin{cases} \dfrac{(dk+1)dk\sum_{s=0}^{dk}\binom{n}{s}_{k+1} + \sum_{s=0}^{dk} s\binom{n}{s}_{k+1} + n\sum_{i=0}^{k}\sum_{s=0}^{dk-i} i^2\binom{n-1}{s}_{k+1}}{(dk+1)dk\sum_{s=0}^{dk}\binom{n}{s}_{k+1} + 2(dk+1)\sum_{s=0}^{dk} s\binom{n}{s}_{k+1}}, & \text{if } 1 \leq d \leq n-1, \\ \dfrac{3nk+k+5}{6nk+6}, & \text{if } d = n, \end{cases}$$

where the parameter k is a positive integer and the $\binom{n}{s}_{k+1}$ is a polynomial-coefficient which will be explained in Section 2.1. When $k = 1$, this result is the same as the result in the basic strategy. When $k > 1$, it is better and the ration $\delta/\log_2 p$ is closer to $\frac{1}{3}$. Compared to previous known results, our result in the extended strategy is also preponderant. For $n = 1$, the asymptotic result ($k = +\infty$) is the same as the optimal results in [2, 28]. For $n \geq 4$, our asymptotic result can deduce that the ration $\delta/\log_2 p < \frac{1}{2}$, which is superior to the optimal results even for sufficiently many samples given in [2, 28].

1.3 Organization of the Paper and Notation

The rest of this paper is organized as follows. In Section 2, we recall some terminology and preliminary knowledge. In Section 3, we introduce ModInv-HNP and ICG. In Section 4 and 5, we present two strategies respectively for analyzing ModInv-HNP and attacking ICG. Section 6 is the conclusion.

Throughout the paper, the set $\{0, 1, \cdots, p-1\}$ is donoted as $\mathbb{Z}_p$, in case of need, the elements of $\mathbb{Z}_p$ are also treated as the corresponding integers. The symbol δ is denoted as the number of the known most (least) significant bits of some unknown numbers.

2 Preliminaries

Let $I_n = (i_1, \cdots, i_n)$, $J_n = (j_1, \cdots, j_n) \in \mathbb{N}^n$, where $I_n - J_n = (i_1 - j_1, \cdots, i_n - j_n)$. Below we first describe lexicographic reverse order and graded lexicographic reverse order respectively. Then, we define a new order of monomials.

Lexicographic Reverse Order:

$$I_n \prec_{revlex} J_n \Leftrightarrow \text{ the rightmost nonzero entry in } I_n - J_n \text{ is negative.}$$

For example, for $u_1 = (0, 4, 0, 2)$ and $u_2 = (3, 1, 2, 1)$, we have $u_2 \prec_{revlex} u_1$.

Graded Reverse Lexicographic Order:

$$I_n \prec_{grevlex} J_n \Leftrightarrow \sum_{m=1}^{n} i_m < \sum_{m=1}^{n} j_m \text{ or } (\sum_{m=1}^{n} i_m = \sum_{m=1}^{n} j_m \text{ and } I_n \prec_{revlex} J_n).$$

For the above u_1 and u_2, we have $u_1 \prec_{grevlex} u_2$. For more details about the orders of monomials, please refer to [12].

In this paper we denote the priority of the variables $x_0, x_1, \cdots, x_n$ in a monomial as $x_0, x_1, \cdots, x_{n-1}, x_n$ from high to low respectively. Let $i_0, j_0 \in \mathbb{N}$, we define an order of monomials $\prod_{m=0}^{n} x_m^{i_m}$, $\prod_{m=0}^{n} x_m^{j_m}$ as follows.

New defined order:

$$\prod_{m=0}^{n} x_m^{i_m} \prec \prod_{m=0}^{n} x_m^{j_m} \Leftrightarrow I_n \prec_{grlex} J_n \text{ or } (I_n = J_n \text{ and } i_0 < j_0). \tag{1}$$

For example, we have $x_0 x_2^4 x_4^2 \prec x_1^3 x_2 x_3^2 x_4 \prec x_0^2 x_1^3 x_2 x_3^2 x_4$ from the defined order (1) .

We can also define the leading monomial of a polynomial in terms of the order (1). In our following construction, we first determine the corresponding monomial set, then we find appropriate polynomials such that their leading monomials are in the defined monomial set.

2.1 Polynomial Coefficients

For positive integers k and n, the coefficient of x^s in the expansion of the polynomial $(1 + x + \cdots + x^k)^n$ is called the polynomial coefficient $\binom{n}{s}_{k+1}$, $0 \le s \le nk$. Namely, we have

$$(1 + x + \cdots + x^k)^n = \sum_{s=0}^{nk} \binom{n}{s}_{k+1} x^s,$$

where $\binom{n}{s}_{k+1} = \sum_{n_1+\cdots+kn_k=s} \binom{n}{n-n_1-\cdots-n_k, n_1, \cdots, n_k}$. Obviously, when $k = 1$, the polynomial coefficient $\binom{n}{s}_{k+1}$ is the binomial coefficient $\binom{n}{s}$. When $m(k+1) \leq s \leq (m+1)(k+1) - 1$, we have

$$\binom{n}{s}_{k+1} = \sum_{i=0}^{m} (-1)^i \binom{n+s-i(k+1)-1}{s-i(k+1)} \binom{n}{i},$$

where $m \in \mathbb{N}$. Euler firstly studied this expansion. For more details on polynomial coefficients, please refer to [8].

2.2 Lattice

Let the vectors $b_1, \ldots, b_\omega$ be linearly independent in $\mathbb{R}^n$, the set

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

is called a lattice with basis vectors $b_1, \cdots, b_\omega$ and basis matrix $B = [b_1^T, \cdots, b_\omega^T]^T$. The dimension and determinant of L are

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

If B is a square matrix, then $\det(L) = |\det(B)|$. In this paper all lattice basis matrices are square.

The well known LLL algorithm [21] can find a reduced basis of the lattice satisfying the following lemma.

Lemma 1 ([21]). *Let L be a lattice. Within polynomial time, the LLL algorithm outputs reduced basis vectors $v_1, \ldots, v_\omega$ that satisfy*

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

2.3 Coppersmith's Technique

We use Coppersmith's technique to find small integer roots of a modular polynomial equation. It first generates some polynomials with these small integers as roots for constructing the lattice, then uses any lattice reduction algorithm to get some integer equations with these desired roots. In this process, the following Lemma reformulated by Howgrave-Graham is needed.

Lemma 2 ([18]). *Let $f(x_0, x_1, \ldots, x_n) = \sum_{i_0, i_1, \ldots, i_n} a_{i_0, i_1, \ldots, i_n} x_0^{i_0} x_1^{i_1} \cdots x_n^{i_n}$ be an integer polynomial that consists of at most ω monomials. Let m be a positive integer and the X_i be upper bounds of $|\tilde{x}_i|$ for $0 \leq i \leq n$. Suppose that*

1. *$f(\tilde{x}_0, \tilde{x}_1, \ldots, \tilde{x}_n) = 0 \pmod{p^m}$;*

2. $\|f(x_0X_0, x_1X_2, \ldots, x_nX_n)\| < \frac{p^m}{\sqrt{\omega}}$,

then $f(\tilde{x}_0, \tilde{x}_1, \ldots, \tilde{x}_n) = 0$ *holds over* $\mathbb{Z}$.

In Lemma 2, $\|f(x_0X_0, x_1X_2, \ldots, x_nX_n)\|$ is the Euclidean norm of the coefficient vector of the polynomial $f(x_0X_0, x_1X_2, \ldots, x_nX_n)$, i.e.,

$$\|f(x_0X_0, x_1X_2, \ldots, x_nX_n)\| = \sqrt{\sum_{i_0,i_1,\ldots,i_n} (a_{i_0,i_1,\ldots,i_n} X_0^{i_0} X_1^{i_1} \cdots X_n^{i_n})^2}.$$

For computing the small root $(\tilde{x}_0, \tilde{x}_1, \ldots, \tilde{x}_n)$, we expect to find at least $n+1$ algebraically independent polynomials $f_i(x_0, x_1, \cdots, x_n)$ with $f_i(\tilde{x}_0, \tilde{x}_1, \ldots, \tilde{x}_n) = 0$ for $0 \leq i \leq n$. Then, we give the following heuristic assumption.

Assumption 1. *The polynomials corresponding to the first few LLL-reduced vectors are algebraically independent.*

To apply Lemma 1 and Lemma 2, we need

$$\omega^{\frac{1}{2}} 2^{\frac{\omega(\omega-1)}{4(\omega-n)}} (\det(L))^{\frac{1}{\omega-n}} < p^m, \tag{2}$$

where $\omega = \dim(L)$. Note that ω is greater than n and p is sufficiently larger than the term $\omega^{\frac{1}{2}} 2^{\frac{\omega(\omega-1)}{4(\omega-n)}}$ in general, we neglect these terms in (2) and simply use the condition

$$(\det(L))^{1/\dim(L)} < p^m. \tag{3}$$

Finally, we compute their common root $(\tilde{x}_0, \tilde{x}_1, \ldots, \tilde{x}_n)$ using numerical or symbolic methods such as the resultant [15] and Gröbner basis methods [12].

3 ModInv-HNP and ICG

Firstly, we restate the definition of ModInv-HNP as follows.

ModInv-HNP. Our goal is to recover the hidden number $\alpha \in \mathbb{Z}_p$, where p is a known prime. For $n+1$ integers t_i randomly chosen from $\mathbb{Z}_p \setminus \{-\alpha\}$, $n+1$ samples

$$(t_i, \mathrm{MSB}_\delta(\alpha_i)), \quad 0 \leq i \leq n$$

are exposed, where $\alpha_i = (\alpha + t_i)^{-1} \bmod p$.

We transform ModInv-HNP into solving small roots of some modular polynomials. Donote $\mathrm{MSB}_\delta(\alpha_i)$ as u_i and α_i as $u_i + \tilde{x}_i$, we have

$$(\alpha + t_i)(u_i + \tilde{x}_i) = 1 \pmod p$$

with $|\tilde{x}_i| \leq p/2^l$ for $0 \leq i \leq n$. Eliminate α from the above equations, we obtain n following equations

$$a_j + b_j\tilde{x}_0 + c_j\tilde{x}_j + \tilde{x}_0\tilde{x}_j = 0 \pmod p, 1 \leq j \leq n,$$

where

$$a_j = u_0u_j + (u_0 - u_j)(t_0 - t_j)^{-1} \bmod p,$$
$$b_j = u_j + (t_0 - t_j)^{-1} \bmod p,$$
$$c_j = u_0 - (t_0 - t_j)^{-1} \bmod p.$$

Thus, the vector $(\tilde{x}_0, \tilde{x}_1, \cdots, \tilde{x}_n)$ are the small solution of n modular polynomials

$$f_j(x_0, x_j) = a_j + b_j x_0 + c_j x_j + x_0 x_j \pmod p, 1 \le j \le n. \tag{4}$$

Remark 1. If $n+1$ samples $(t_i, \mathrm{LSB}_\delta(\alpha_i))$ are given in ModInv-HNP, we can still reduce it to finding small roots of n simultaneous modular polynomials like (4). Thus, our following strategies are also fit for the least significant bit case.

Next, we introduce the inversive congruential generator.

ICG. For a given prime p, let $F(x) = ax^{-1} + b \bmod p$, where $a, b \in \mathbb{Z}_p$. Input a secret seed $v_0 \in \mathbb{Z}_p$ to the recursive relation $v_{i+1} = F(v_i)$. Then, output the δ consecutive bits of the v_i at each iteration, $\mathrm{MSB}_\delta(v_1), \cdots, \mathrm{MSB}_\delta(v_{n+1})$, as the pseudorandom sequence.

We are concerned with the problem of recovering the seed v_0 for given $n+1$ outputs $\mathrm{MSB}_\delta(v_i)$. When the a and b in $F(x)$ are known, let $\overline{x}_{i-1} = v_i - \mathrm{MSB}_l(v_i)$ with $|\overline{x}_{i-1}| \le p/2^\delta$ for $1 \le i \le n+1$, we can also deduce that the polynomial

$$\overline{a}_j + \overline{b}_j x_0 + \overline{c}_j x_j + x_0 x_j \pmod p$$

has the small solution $(\overline{x}_0, \overline{x}_j)$ for $1 \le j \le n$, where $\overline{a}_{0,j}, \overline{b}_{0,j}, \overline{c}_{0,j}$ can be publicly computed. Once we get the desired small root of the above modular polynomials, whose form are the same as the f_j in (4), we can also recover the secret seed v_0.

Thus, our goal in the following sequel is to find the root $(\tilde{x}_0, \tilde{x}_1, \cdots, \tilde{x}_n)$ in (4), where $|\tilde{x}_i| < X_i$. We take $X_i = X = p/2^\delta$ for $0 \le i \le n$ in the following strategies.

4 Solving Desired Roots in (4): Basic Strategy

Let d be a positive integer with $1 \le d \le n$ and the monomial set $\mathrm{MS}(n, d)$ be

$$\{\prod_{m=0}^{n} x_m^{i_m}, 0 \le i_0 \le d, 0 \le i_1, \cdots, i_n \le 1, 0 \le i_1 + \cdots + i_n \le d\}.$$

Fix i_0 and $I_n = (i_1, \cdots, i_n)$, construct a polynomial $f_{i_0;I_n}$ such that its leading monomial is $\prod_{m=0}^{n} x_m^{i_m}$ according to the monomial order (1). Moreover,

$$f_{i_0;I_n}(\widetilde{x}_0, \widetilde{x}_1, \widetilde{x}_2, \cdots, \widetilde{x}_n) = 0 \pmod{p^{\mathrm{WR}(i_0;I_n)}}. \tag{5}$$

where $\mathrm{WR}(i_0; I_n)$ called as the weight of the relation (5) is the maximum integer satisfied (5).

Let M be $\max\{i_1, \cdots, i_n\}$, obviously, $M = 0$ or 1 in basic strategy. Next, we discuss $f_{i_0;I_n}$ and $\mathrm{WR}(i_0; I_n)$ according to the case of M.

Case 1. If $M = 0$, we have $\prod_{m=0}^{n} x_m^{i_m} = x_0^{i_0}$, then we choose that $f_{i_0;I_n} = x_0^{i_0}$ and the corresponding $\mathrm{WR}(i_0; I_n) = 0$.

Case 2. If $M = 1$, let $\prod_{m=0}^{n} x_m^{i_m} = x_0^{i_0} \prod_{t=1}^{s} x_{j_t}$, where $1 \le s \le n$, and $1 \le j_1 < \cdots < j_s \le n$. Clearly, $\sum_{m=1}^{n} i_m = s$. For the sake of discussion, denote the set S as $\{x_{j_1}, \cdots, x_{j_s}\}$.

(1) If $i_0 \ge s$, we choose the polynomial

$$f_{i_0;I_n} = x_0^{(i_0-s)} \prod_{t=1}^{s} f_{0,j_t} =: f(i_0; S)$$

and the corresponding

$$\mathrm{WR}(i_0; I_n) = s.$$

(2) If $i_0 < s$, when $s = 1$, clearly, $i_0 = 0$, then we choose the polynomial

$$f_{i_0;I_n} = x_{j_1} =: f(i_0; S)$$

and the corresponding

$$\mathrm{WR}(i_0; I_n) = 0.$$

When $s > 1$, let the polynomial $g_m(x_0, x_{j_1}, \cdots, x_{j_s}) = (\prod_{t \ne m} f_{0,j_t}) x_{j_m}$ for $1 \le m \le s$, clearly,

$$g_m(\widetilde{x}_0, \widetilde{x}_{j_1}, \cdots, \widetilde{x}_{j_s}) = 0 \pmod{p^{s-1}}.$$

Note that $g_1, \cdots, g_s$ have common monomials $\prod_{t=1}^{s} x_{j_t}, x_0 \prod_{t=1}^{s} x_{j_t}, \cdots, x_0^{s-1} \prod_{t=1}^{s} x_{j_t}$, thus, we rearrange them according to the following way.

$$\begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_s \end{bmatrix} = M(S) \begin{bmatrix} \prod_{t=1}^{s} x_{j_t} \\ x_0 \prod_{t=1}^{s} x_{j_t} \\ \vdots \\ x_0^{s-1} \prod_{t=1}^{s} x_{j_t} \end{bmatrix} + \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_s \end{bmatrix} \mod p^{s-1}, \tag{6}$$

where the matrix $M(S)$ is the coefficient matrix about monomials $\prod_{t=1}^{s} x_{j_t}$, $x_0 \prod_{t=1}^{s} x_{j_t}$, $\cdots$, $x_0^{s-1} \prod_{t=1}^{s} x_{j_t}$ in $h_1, \cdots, h_s$, i.e.,

$$M(S) = \begin{bmatrix} \sigma_{s-1}(\wedge_1) & \cdots & \sigma_2(\wedge_1) & \sigma_1(\wedge_1) & 1 \\ \sigma_{s-1}(\wedge_2) & \cdots & \sigma_2(\wedge_2) & \sigma_1(\wedge_2) & 1 \\ & & \cdots & & \\ \sigma_{s-1}(\wedge_s) & \cdots & \sigma_2(\wedge_s) & \sigma_1(\wedge_s) & 1 \end{bmatrix},$$

here the $\sigma_i(y_1, \cdots, y_{s-1})$ is the i-th elementary symmetric polynomial about $s-1$ variables $y_1, \cdots, y_{s-1}$ and

$$\wedge_m = (c_{0,j_1}, \cdots, c_{0,j_{m-1}}, c_{0,j_{m+1}}, \cdots, c_{0,j_s})$$

for $1 \leq i \leq s-1$, $1 \leq m \leq s$.

We can compute out the determinant of the matrix $M(S)$ by mathematical induction, i.e.,

$$\det(M(S)) = \prod_{1 \leq m < t \leq s} (c_{0,j_m} - c_{0,j_t}).$$

In general, the matrix $M(S)$ is invertible in $\mathbb{Z}_{p^{s-1}}$ since $\gcd(\det(M(S)), p) = 1$ in all almost cases. Thus, according to (6), we have

$$M(S)^{-1} \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_s \end{bmatrix} = \begin{bmatrix} \prod_{t=1}^{s} x_{j_t} \\ x_0 \prod_{t=1}^{s} x_{j_t} \\ \vdots \\ x_0^{s-1} \prod_{t=1}^{s} x_{j_t} \end{bmatrix} + \mathrm{M}(S)^{-1} \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_s \end{bmatrix} \mod p^{s-1}, \tag{7}$$

where $M(S)^{-1}$ is the inverse matrix of $M(S)$ in $\mathbb{Z}_{p^{s-1}}$. For convenience, denote $b_{l,m}$ as the $(l+1)$-th row and the m-th column of the matrix $M(S)^{-1}$, $0 \leq l \leq s-1$, $1 \leq m \leq s$. In the following discussion, the $b_{l,m}$ are regarded as corresponding integers.

Note that the h_m are composed of the remaining terms in g_m except for the corresponding terms of monomials $\prod_{t=1}^{s} x_{j_t}$, $x_0 \prod_{t=1}^{s} x_{j_t}$, $\cdots$, $x_0^{s-1} \prod_{t=1}^{s} x_{j_t}$, then the order of all monomials in h_m is lower than $\prod_{t=1}^{s} x_{j_t}$ according to the defined order (1). Hence, the monomial $x_0^{i_0} \prod_{t=1}^{s} x_{j_t}$ is the leading term of polynomial $\sum_{m=1}^{s} b_{i_0,m} g_m$. Then, we choose

$$f_{i_0;I_n} = \sum_{m=1}^{s} b_{i_0,m} g_m =: f(i_0; S)$$

in this case. Clearly, the corresponding

$$\mathrm{WR}(i_0; I_n) = \sum_{m=1}^{n} i_m - 1.$$

Remark 2. In fact, we only need a matrix $\widetilde{M}(S)$ such that $\widetilde{M}(S)M(S) \bmod p$ is a lower triangular unimodular matrix. Here, we take $M(S)^{-1}$ as $\widetilde{M}(S)$ in (7).

We have concretely generated the polynomial $f_{i_0;I_n}$ according to the above way, and the corresponding $\mathrm{WR}(i_0; I_n)$ was summarized as follows.

$$\mathrm{WR}(i_0; I_n) = \begin{cases} \sum\limits_{m=1}^{n} i_m - 1, \ 0 \le i_0 < \sum\limits_{m=1}^{n} i_m, \\ \\ \sum\limits_{m=1}^{n} i_m, \qquad \sum\limits_{m=1}^{n} i_m \le i_0 \le d. \end{cases} \tag{8}$$

Let $|\tilde{x}_i| < X_i$, $0 \le i \le n$, we construct a lattice $L(n, d)$ using coefficient vectors of following polynomials:

$$p^{(d-\mathrm{WR}(i_0;I_n))} f_{i_0;I_n}(x_0X_0, x_1X_1, \cdots, x_nX_n),$$

We order these vectors such that the basis matrix is a lower triangular. Denote

$$\mathrm{WT}(i_0; I_n) = \sum_{m=0}^{n} i_m, \tag{9}$$

which is called as weight of the term $\prod\limits_{m=0}^{n} x_m^{i_m}$ in $\mathrm{MS}(n, d)$. Then, diagonal elements of the basis matrix are

$$p^{(d-\mathrm{WR}(i_0;I_n))} X^{\mathrm{WT}(i_0;I_n)},$$

where $0 \le i_0 \le d$, $0 \le i_1, \cdots, i_n \le 1$, $0 \le i_1 + \cdots + i_n \le d$.

Clearly, the dimension of the lattice $L(n, d)$ is equal to the number of monomials in $\mathrm{MS}(n, d)$, i.e.,

$$\dim(L(n, d)) = (d + 1) \sum_{s=0}^{d} \binom{n}{s}. \tag{10}$$

We compute out that the determinant of the lattice L is

$$\det(L(n, d)) = p^{(d \cdot \dim(L) - \mathrm{WR}(n,d))} X^{\mathrm{WT}(n,d)}, \tag{11}$$

where

$$\mathrm{WR}(n, d) = \sum_{s=0}^{d} \sum_{i_0=0}^{d} \binom{n}{s} \mathrm{WR}(i_0; I_n),$$

$$\mathrm{WT}(n,d) = \sum_{s=0}^{d}\sum_{i_0=0}^{d}\binom{n}{s}\mathrm{WT}(i_0; I_n).$$

From (8) and (9), we get

$$\mathrm{WR}(n,d) = d\sum_{s=0}^{d} s\binom{n}{s},$$

and

$$\mathrm{WT}(n,d) = \frac{d(d+1)}{2}\sum_{s=0}^{d}\binom{n}{s} + (d+1)\sum_{s=0}^{d} s\binom{n}{s}.$$

Let

$$\mathrm{F}(n,d) = \frac{\mathrm{WR}(n,d)}{\mathrm{WT}(n,d)},$$

According to (3) and (11), there is the following relation

$$X < p^{\mathrm{F}(n,d)},$$

where

$$\mathrm{F}(n,d) = \frac{2d\sum\limits_{s=0}^{d} s\binom{n}{s}}{d(d+1)\sum\limits_{s=0}^{d}\binom{n}{s} + 2(d+1)\sum\limits_{s=0}^{d} s\binom{n}{s}}. \tag{12}$$

Note that $X = p/2^{\delta}$, we give the following result about ModInv-HNP and ICG.

Theorem 1. *Given $n+1$ samples in ModInv-HNP or $n+1$ outputs $\mathrm{MSB}_{\delta}(v_i)$ in ICG. Choose an integer d such that $1 \le d \le n$. Then, under Assumption 1, we can recover the hidden number α or the secret seed v_0 in polynomial time when*

$$\delta/\log_2 p \ge \frac{d(d+1)\sum\limits_{s=0}^{d}\binom{n}{s} + 2\sum\limits_{s=0}^{d} s\binom{n}{s}}{d(d+1)\sum\limits_{s=0}^{d}\binom{n}{s} + 2(d+1)\sum\limits_{s=0}^{d} s\binom{n}{s}}. \tag{13}$$

Remark 3. Our result is suitable for any positive integer n. For positive integers n and d satisfying $1 \le d \le n$, according to (13) we can get

$$\delta/\log_2 p > \frac{1}{3} + \frac{2}{3n+3}.$$

Furthermore, there is always $\delta/\log_2 p > \frac{1}{3}$ even when n is sufficiently large.

Remark 4. Take $d = 1$, the relation (13) becomes

$$\delta/\log_2 p \geq \frac{2}{3} + \frac{1}{9n+3}.$$

This result is better than the corresponding result at Journal of Symbolic Computation [22]. The asymptotic result $\delta/\log_2 p > \frac{2}{3}$ and $\delta/\log_2 p \to \frac{2}{3}$ in the sense that n is sufficiently large is the same as the first result of Boneh et al. at ASIACRYPT 2001 [7] and the result in [22].

Remark 5. Take $d = n$, the relation (13) becomes

$$\delta/\log_2 p \geq \frac{1}{2} + \frac{1}{2n+2}.$$

When n is sufficiently large, the asymptotic result $\delta/\log_2 p > \frac{1}{2}$ and $\delta/\log_2 p \to \frac{1}{2}$ is the same as the corresponding optimal results in [2, 28].

Remark 6. Take $d = \lfloor \frac{n}{2} \rfloor$, the relation (13) becomes

$$\delta/\log_2 p \geq \frac{n^2 2^n + o(n^2 2^n)}{3n^2 2^n + o(n^2 2^n)}.$$

When n is sufficiently large, the asymptotic result $\delta/\log_2 p > \frac{1}{3}$ and $\delta/\log_2 p \to \frac{1}{3}$ is the same as the second result of Boneh et al. in [7].

For showing the comparison between this result and previous results, we give Table 1 and Table 2. We compare the ration $\delta/\log_2 p$ for the same small n in Table 1. We take the corresponding optimal d for concrete n in our result. In Table 2, we compare the smallest n needed for fixed $\delta/\log_2 p$. For $n \geq 9$, we can deduce that the ration $\delta/\log_2 p < 1/2$, which is better than the result in [2, 22, 28] and the first result in [7]. Interestingly, the ration $\delta/\log_2 p$ in our result is close to $\frac{1}{3}$ when $n \geq 100$. The symbol " $-$ " in the two tables denotes that no concrete result was given by the corresponding authors, and the symbol " $\times$ " denotes the situation that can not be reached in the corresponding paper.

Table 1. The minimum value of $\delta/\log_2 p$ for small n

Result \ n	1	2	3	4	5	6	7	8	9
[7]	-	-	-	-	-	-	-	-	-
[2]	0.6667	0.5714	0.5333	0.5161	0.5079	0.5039	0.5020	0.5010	0.5005
[22]	0.8889	0.7778	0.7407	0.7222	0.7111	0.7037	0.6984	0.6944	0.6914
[28]	0.6667	0.5417	0.5083	0.5014	0.5002	0.5000	0.5000	0.5000	0.5000
This paper	0.7500	0.6667	0.6250	0.5841	0.5611	0.5378	0.5220	0.5073	0.4953
	$d = 1$	$d = 2$	$d = 2, 3$	$d = 3$	$d = 3$	$d = 4$	$d = 4$	$d = 5$	$d = 5$

In order to explain our basic strategy, we give an example of the case $n = 2$ in Appendix B. Below we give an extended strategy for further improvement.

Table 2. The smallest n needed for fixed $\delta/\log_2 p$

Result \ $\delta/\log_2 p$	0.6678	0.5714	0.5005	0.4953	0.4276	0.3782	0.3419
[7] (the first)	–	×	×	×	×	×	×
[7] (the second)	–	–	–	–	–	–	–
[22]	200	×	×	×	×	×	×
[2]	1	2	9	×	×	×	×
[28]	1	2	8	×	×	×	×
This paper	2	3	9	9	20	50	100

5 Solving Desired Root in (4): Extended Strategy

Let k be a positive integer, and define the monomial set $\mathrm{MS}(n, d, k)$ as

$$\{\prod_{m=0}^{n} x_m^{i_m}, 0 \le i_0 \le dk, 0 \le i_1, \cdots, i_n \le k, 0 \le i_1 + \cdots + i_n \le dk\}.$$

Obviously, the monomial set $\mathrm{MS}(n, d, k)$ is equal to $\mathrm{MS}(n, d)$ in the basic strategy when $k = 1$. For a fixed monomial $\prod_{m=0}^{n} x_m^{i_m} \in \mathrm{MS}(n, d, k)$, we generate the polynomial $f_{i_0;I_n}$ such that $\prod_{m=0}^{n} x_m^{i_m}$ is the leading monomial according to the defined order (1) and compute the corresponding $\mathrm{WR}(i_0; I_n)$. Let $\max\{i_1, \cdots, i_n\} = M$, below we discuss $f_{i_0;I_n}$ and $\mathrm{WR}(i_0; I_n)$ in accordance with M.

Case 1. When $M = 0$, we choose $f_{i_0;I_n} = x_0^{i_0}$ and $\mathrm{WR}(i_0; I_n) = 0$.

Case 2. When $M > 0$, we classify variables $x_1, \cdots, x_n$ according to concrete values of $i_1, \cdots, i_n$ and rewrite $\prod_{m=0}^{n} x_m^{i_m}$ as

$$x_0^{i_0} \prod_{l=1}^{M} \prod_{t=1}^{s_l} x_{j_t},$$

where integers $s_1, \cdots, s_M$ are satisfied that

$$\begin{cases} 0 < s_M \le \cdots \le s_2 \le s_1 \le n, \\ 0 < \sum_{l=1}^{M} s_l \le dk, \end{cases}$$

and

$$1 \le j_1 < j_2 < \cdots < j_{s_1} \le n.$$

For the sake of discussion, let the set

$$S_l = \{x_{j_1}, \cdots, x_{j_{s_l}}\}, 1 \le l \le M.$$

It is evident that the positive integer s_l is cardinality of the set S_l and

$$\emptyset \subset S_M \subseteq \cdots \subseteq S_1 \subseteq I_n.$$

(1) If $0 \leq i_0 \leq s_M - 1$, we choose

$$f_{i_0;I_n} = (\prod_{l=1}^{M-1} f(0;S_l)) f(i_0;S_M)$$

and

$$\mathrm{WR}(i_0;I_n) = \sum_{l=1}^{M} s_l - M.$$

(2) If $\sum\limits_{l=j+1}^{M} s_l \leq i_0 \leq \sum\limits_{l=j}^{M} s_l - 1$, we choose

$$f_{i_0;I_n} = (\prod_{l=1}^{j-1} f(0;S_l)) f(i_0 - \sum_{l=j+1}^{M} s_l; S_j)(\prod_{l=j+1}^{M} f(s_l;S_l))$$

and

$$\mathrm{WR}(i_0;I_n) = \sum_{l=1}^{M} s_l - j$$

for $1 \leq j \leq M-1$.

(3) If $\sum\limits_{l=1}^{M} s_l \leq i_0 \leq dk$, we choose

$$f_{i_0;I_n} = x_0^{(i_0 - \sum_{l=1}^{M} s_l)} \prod_{l=1}^{M} f(s_l;S_l)$$

and

$$\mathrm{WR}(i_0;I_n) = \sum_{l=1}^{M} s_l.$$

Next, we construct a lattice $L(n,d,k)$ using coefficient vectors of polynomials

$$f_{i_0;I_n}(x_0X_0, x_1X_1, \cdots, x_nX_n).$$

The corresponding lattice matrix is lower triangular, and the form of its diagonal elements is

$$p^{(dk - \mathrm{WR}(i_0;I_n))} X^{\mathrm{WT}(i_0;I_n)},$$

where

$$\mathrm{WT}(i_0;I_n) = \sum_{m=0}^{n} i_m.$$

For the sake of the following analysis, let the set $S(n,k,dk)$ be

$$\{(i_1,\cdots,i_n), 0 \le i_1,\cdots,i_n \le k, 0 \le i_1+\cdots+i_n \le dk\}.$$

We know that the dimension of the lattice $L(n,d,k)$ is equal to the cardinality of the monomial set $\mathrm{MS}(n,d,k)$. Thus,

$$\dim(L(n,d,k)) = (dk+1)|S(n,k,dk)|,$$

where the integer $|S(n,k,dk)|$ is the cardinality of $S(n,k,dk)$. Moreover, $|S(n,k,dk)|$ can also be regarded as the sum of coefficients of the x^s in the expansion of the polynomial $(1+x+\cdots+x^k)^n$, $s=0,1,\cdots,dk$. Namely,

$$\dim(L(n,d,k)) = (dk+1)\sum_{s=0}^{dk}\binom{n}{s}_{k+1}. \tag{14}$$

The determinant of $L(n,d,k)$ is

$$\det(L(n,d,k)) = p^{(dk\dim(L(n,d,k))-\mathrm{WR}(n,d,k))}X^{\mathrm{WT}(n,d,k)}, \tag{15}$$

where

$$\mathrm{WR}(n,d,k) = \sum_{i_0=0}^{dk}\sum_{I_n\in S(n,k,dk)}\mathrm{WR}(i_0;I_n),$$

and

$$\mathrm{WT}(n,d,k) = \sum_{i_0=0}^{dk}\sum_{I_n\in S(n,k,dk)}\mathrm{WT}(i_0;I_n).$$

Furthermore, we compute out

$$\mathrm{WR}(n,d,k) = \begin{cases} \frac{2dk+1}{2}\sum\limits_{s=0}^{dk} s\binom{n}{s}_{k+1} - \frac{n}{2}\sum\limits_{i=0}^{k}\sum\limits_{s=0}^{dk-i} i^2\binom{n-1}{s}_{k+1}, & \text{if } 1\le d\le n-1. \\ \frac{1}{6}nk(3nk-k+1)(k+1)^n, & \text{if } d=n. \end{cases}$$

and

$$\mathrm{WT}(n,d,k) = \frac{(dk+1)dk}{2}\sum_{s=0}^{dk}\binom{n}{s}_{k+1} + (dk+1)\sum_{s=0}^{dk}s\binom{n}{s}_{k+1}.$$

We leave computation of $\mathrm{WR}(n,d,k)$ and $\mathrm{WT}(n,d,k)$ in Appendix A.

Let

$$\mathrm{F}(n,d,k) = \frac{\mathrm{WR}(n,d,k)}{\mathrm{WT}(n,d,k)},$$

according to (3) and (15), we have

$$X < p^{\mathrm{F}(n,d,k)},$$

where

$$\mathrm{F}(n,d,k) = \begin{cases} \dfrac{(2dk+1)\sum\limits_{s=0}^{dk} s\binom{n}{s}_{k+1} - n\sum\limits_{i=0}^{k}\sum\limits_{s=0}^{dk-i} i^2\binom{n-1}{s}_{k+1}}{(dk+1)dk\sum\limits_{s=0}^{dk}\binom{n}{s}_{k+1} + 2(dk+1)\sum\limits_{s=0}^{dk} s\binom{n}{s}_{k+1}}, & \text{if } 1 \le d \le n-1. \\ \dfrac{3nk-k+1}{6nk+6}, & \text{if } d = n. \end{cases} \tag{16}$$

From $X = p/2^{\delta}$, we can get the following theorem about ModInv-HNP and ICG.

Theorem 2. *Given $n+1$ samples in ModInv-HNP or $n+1$ outputs $\mathrm{MSB}_{\delta}(v_i)$ in ICG. Choose positive integers d, k such that $1 \le d \le n$. Then, under Assumption 1, we can recover the hidden number α or the secret seed v_0 in polynomial time when*

$$\delta/\log_2 p \ge \begin{cases} \dfrac{(dk+1)dk\sum\limits_{s=0}^{dk}\binom{n}{s}_{k+1} + \sum\limits_{s=0}^{dk} s\binom{n}{s}_{k+1} + n\sum\limits_{i=0}^{k}\sum\limits_{s=0}^{dk-i} i^2\binom{n-1}{s}_{k+1}}{(dk+1)dk\sum\limits_{s=0}^{dk}\binom{n}{s}_{k+1} + 2(dk+1)\sum\limits_{s=0}^{dk} s\binom{n}{s}_{k+1}}, & 1 \le d \le n-1. \\ \dfrac{3nk+k+5}{6nk+6}, & d = n. \end{cases} \tag{17}$$

Remark 7. For any positive integers k, n and d such that $1 \le d \le n$, we can deduce that

$$\mathrm{WR}(n,d,k) < dk\sum_{s=0}^{dk} s\binom{n}{s}_{k+1}$$

and

$$\mathrm{WT}(n,d,k) > \frac{3(dk+1)}{2}\sum_{s=0}^{dk} s\binom{n}{s}_{k+1}.$$

Then, it leads to

$$\delta > \left(\frac{1}{3} + \frac{2}{3nk+3}\right)\log_2 p.$$

Namely, there is always $\delta/\log_2 p > \frac{1}{3}$ even for sufficiently large positive integers n, k.

Remark 8. When $k = 1$, the above relation (17) degenerates into the relation (13) in theorem 1.

We give Table 3 and Table 4 to indicate relations between the ration $\delta/\log_2 p$ and concrete n. The positive integer d chosen in the tables are optimal for the corresponding n. When $k = 1$, the result in the extended strategy is the same as that in the basic strategy. When $k = +\infty$, the result in the extended strategy is more ideal than that in the basic strategy. From Table 1 and Table 3, we find out the result in the extended strategy is the same as the result in [2, 28] when

$n = 1$. When $n = 2$ and 3, the result in the extended strategy is better than the result in [2], but weaker than our recent result in [28]. For $n \geq 4$, the result in the extended strategy can lead to the ration $\delta/\log_2 p < 1/2$, which is always better than results in [2, 28].

Table 3. The minimum value of $\delta/\log_2 p$ for small n in the extended strategy

k \ n	1	2	3	4	5	6	7	8	9
1	0.7500	0.6667	0.6250	0.5841	0.5611	0.5378	0.5220	0.5073	0.4953
	$d = 1$	$d = 2$	$d = 2, 3$	$d = 3$	$d = 3$	$d = 4$	$d = 4$	$d = 5$	$d = 5$
$+\infty$	0.6667	0.5714	0.5085	0.4748	0.4518	0.4369	0.4235	0.4141	0.4066
	$d = 1$	$d = 1$	$d = 2$	$d = 3$	$d = 4$	$d = 3$	$d = 4$	$d = 4$	$d = 5$

Table 4. The smallest n needed for a fixed $\delta/\log_2 p$ in the extended strategy

k \ $\delta/\log_2 p$	0.6678	0.5714	0.5005	0.4953	0.4276	0.3782	0.3419
1	2	3	9	9	20	50	100
$+\infty$	1	2	4	4	7	16	86

To illustrate the extended strategy, we give an example when $(n, k) = (2, 2)$ in Appendix C.

6 Conclusion

We revisited the modular inversion hidden number problem and inversive congruential pseudo random number generator and reduced these two problems to solving small roots of a class of simultaneous modular polynomial equations. We presented two strategies based on Coppersmith's technique to solve such the equation system, our methods of choosing polynomials for constructing lattices can make the upper bound of the desired root better. For analyzing the modular inversion hidden number problem, we gave a concrete lattice for explaining the best result up to now proposed by Boneh et al., and further improved the lattice construction such that the samples required are fewer. Applying to attack the inversive congruential pseudo random number generator, we achieved a best result so far.

References

1. Aono, Y.: Minkowski sum based lattice construction for multivariate simultaneous coppersmiths technique and applications to RSA. In: Information Security and Privacy, Springer (2013) 88–103
2. Bauer, A., Vergnaud, D., Zapalowicz, J.C.: Inferring sequences produced by non-linear pseudorandom number generators using coppersmiths methods. In Fischlin, M., Buchmann, J., Manulis, M., eds.: Public Key Cryptography-PKC 2012. Volume 7293 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2012) 609–626
3. Blackburn, S.R., Gomez-perez, D., Gutierrez, J., Shparlinski, I.E.: Predicting non-linear pseudorandom number generators. MATH. COMPUTATION **74** (2004) 2004
4. Blackburn, S., Gomez-Perez, D., Gutierrez, J., Shparlinski, I.: Predicting the inversive generator. In Paterson, K., ed.: Cryptography and Coding. Volume 2898 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2003) 264–275
5. Boneh, D., Durfee, G.: Cryptanalysis of rsa with private key d less than n 0.292. In Stern, J., ed.: Advances in Cryptology EUROCRYPT 99. Volume 1592 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (1999) 1–11
6. Boneh, D., Halevi, S., Howgrave-Graham, N.: The modular inversion hidden number problem. In: ASIACRYPT 2001, Springer (2001) 36–51
7. Boneh, D., Venkatesan, R.: Hardness of computing the most significant bits of secret keys in Diffie-Hellman and related schemes. In: CRYPTO 1996, Springer (1996) 129–142
8. Comtet, L.: Advanced Combinatorics. D. Reidel Publishing Company (1974)
9. Coppersmith, D.: Finding a small root of a bivariate integer equation; factoring with high bits known. In: EUROCRYPT 1996, Springer (1996) 178–189
10. Coppersmith, D.: Finding a small root of a univariate modular equation. In: EUROCRYPT 1996, Springer (1996) 155–165
11. Coppersmith, D.: Small solutions to polynomial equations, and low exponent RSA vulnerabilities. Journal of Cryptology **10**(4) (1997) 233–260
12. Cox, D.A.: Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra. Springer (2007)
13. Eichenauer, J., Lehn, J.: A non-linear congruential pseudo random number generator. Statistische Hefte **27**(1) (1986) 315–326
14. Galindo, D., Vivek, S.: Limits of a conjecture on a leakage-resilient cryptosystem. Information Processing Letters **114**(4) (2014) 192 – 196
15. Gelfand, I., Gelfand, I., Kapranov, M., Zelevinsky, A.: Discriminants, Resultants, and Multidimensional Determinants. Mathematics (Birkhäuser). Birkhäuser Boston (2008)
16. Herrmann, M., May, A.: Solving linear equations modulo divisors: On factoring given any bits. In Pieprzyk, J., ed.: Advances in Cryptology - ASIACRYPT 2008. Volume 5350 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2008) 406–424
17. Herrmann, M., May, A.: Attacking power generators using unravelled linearization: When do we output too much? In: Advances in Cryptology–ASIACRYPT 2009. Springer (2009) 487–504
18. Howgrave-Graham, N.: Finding small roots of univariate modular equations revisited. In: Crytography and Coding. Springer (1997) 131–142

19. Jochemsz, E., May, A.: A strategy for finding roots of multivariate polynomials with new applications in attacking rsa variants. In: ASIACRYPT 2006. Springer (2006) 267–282
20. Kiltz, E., Pietrzak, K.: Leakage resilient elgamal encryption. In Abe, M., ed.: Advances in Cryptology - ASIACRYPT 2010. Volume 6477 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2010) 595–612
21. Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring polynomials with rational coefficients. Mathematische Annalen **261**(4) (1982) 515–534
22. Ling, S., Shparlinski, I.E., Steinfeld, R., Wang, H.: On the modular inversion hidden number problem. Journal of Symbolic Computation **47**(4) (2012) 358–367
23. May, A., Ritzenhofen, M.: Implicit factoring: On polynomial time factoring given only an implicit hint. In: PKC 2009. Springer (2009) 1–14
24. Nguyen, Shparlinski: The insecurity of the digital signature algorithm with partially known nonces. Journal of Cryptology **15**(3) (2002) 151–176
25. Niederreiter, H.: New developments in uniform pseudorandom number and vector generation. In Niederreiter, H., Shiue, P.S., eds.: Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing. Volume 106 of Lecture Notes in Statistics. Springer New York (1995) 87–120
26. Niederreiter, H., Shparlinski, I.: Recent advances in the theory of nonlinear pseudorandom number generators. In Fang, K.T., Niederreiter, H., Hickernell, F., eds.: Monte Carlo and Quasi-Monte Carlo Methods 2000. Springer Berlin Heidelberg (2002) 86–102
27. Shparlinski, I.E.: Playing hide-and-seek with numbers: the hidden number problem, lattices, and exponential sums. In: proceeding of symposia in applied mathematics. Volume 62. (2005) 153–177
28. Xu, J., Hu, L., Huang, Z., Peng, L.: Modular inversion hidden number problem revisited. In: Information Security Practice and Experience. Springer (2014) 537–551

A Computation of $\mathbf{WR}(n, d, k)$ and $\mathbf{WT}(n, d, k)$

First, we compute $\mathrm{WR}(n, d, k)$. Fix a vector $I_n \in S(n, k, dk)$, from the concrete value of $\mathrm{WR}(i_0; I_n)$, we have

$$\sum_{i_0=0}^{dk} \mathrm{WR}(i_0; I_n) = (dk+1)(\sum_{l=1}^{M} s_l) - (\sum_{l=1}^{M} l s_l).$$

We know that $\mathrm{WR}(i_0; I_n) = 0$ when $M = 0$. When $M > 0$, note that

$$\prod_{m=1}^{n} x_m^{i_m} = \prod_{l=1}^{M} \prod_{t=1}^{s_l} x_{j_t},$$

we can deduce that

$$\sum_{l=1}^{M} s_l = \sum_{m=1}^{n} i_m, \sum_{l=1}^{M} l s_l = \sum_{m=1}^{n} \sum_{t=1}^{i_m} t.$$

Thus,

$$\sum_{i_0=0}^{dk} \mathrm{WR}(i_0; I_n) = \frac{2dk+1}{2} \sum_{m=1}^{n} i_m - \frac{1}{2} \sum_{m=1}^{n} i_m^2.$$

According to

$$\mathrm{WR}(n,d,k) = \sum_{i_0=0}^{dk} \sum_{I_n \in S(n,k,dk)} \mathrm{WR}(i_0; I_n),$$

we have

$$\mathrm{WR}(n,d,k) = \frac{2dk+1}{2} \sum_{I_n \in S(n,k,dk)} \sum_{m=1}^{n} i_m - \frac{1}{2} \sum_{I_n \in S(n,k,dk)} \sum_{m=1}^{n} i_m^2.$$

We rewrite $\mathrm{WR}(n,d,k)$ using the concept of polynomial coefficients as

$$\mathrm{WR}(n,d,k) = \frac{2dk+1}{2} \sum_{s=0}^{dk} s \binom{n}{s}_{k+1} - \frac{n}{2} \sum_{i=0}^{k} \sum_{s=0}^{\min\{dk-i,(n-1)k\}} i^2 \binom{n-1}{s}_{k+1}.$$

Furthermore, if $d = n$, we have

$$\mathrm{WR}(n,n,k) = \frac{1}{6} nk(3nk - k + 1)(k+1)^n.$$

If $1 \le d \le n-1$, we get

$$\mathrm{WR}(n,d,k) = \frac{2dk+1}{2} \sum_{s=0}^{dk} s \binom{n}{s}_{k+1} - \frac{n}{2} \sum_{i=0}^{k} \sum_{s=0}^{dk-i} i^2 \binom{n-1}{s}_{k+1}.$$

Finally, we compute $\mathrm{WT}(n,d,k)$. Note that $\mathrm{WT}(i_0; I_n) = \sum_{m=0}^{n} i_m$, we have

$$\sum_{i_0=0}^{dk} \mathrm{WT}(i_0; I_n) = \frac{(dk+1)dk}{2} + (dk+1) \sum_{m=1}^{n} i_m.$$

From

$$\mathrm{WT}(n,d,k) = \sum_{i_0=0}^{dk} \sum_{I_n \in S(n,k,dk)} \mathrm{WT}(i_0; I_n),$$

we get

$$\mathrm{WT}(n,d,k) = \frac{(dk+1)dk}{2} \sum_{I_n \in S(n,k,dk)} 1 + (dk+1) \sum_{I_n \in S(n,k,dk)} \sum_{m=1}^{n} i_m.$$

Namely,

$$\mathrm{WT}(n,d,k) = \frac{(dk+1)dk}{2} \sum_{s=0}^{dk} \binom{n}{s}_{k+1} + (dk+1) \sum_{s=0}^{dk} s \binom{n}{s}_{k+1}.$$

B An Example of the Basic Strategy

We consider $n = 2$, the corresponding monomial set

$$\mathrm{MS}(2, d) = \{x_0^{i_0} x_1^{i_1} x_2^{i_2}, 0 \le i_0 \le d, 0 \le i_1, i_2 \le 1, 0 \le i_1 + i_2 \le d\}$$

for $1 \le d \le 2$.

When $d = 1$, we arrange the order of all monomials in $\mathrm{MS}(2, 1)$ according to the order (1), i.e.,

$$1 \prec x_0 \prec x_1 \prec x_0 x_1 \prec x_2 \prec x_0 x_2.$$

Then, we generate following polynomials $f_{i_0;I_2}$ as follows.

$$f_{0;(0,0)} = 1, f_{1;(0,0)} = x_0, f_{0;(1,0)} = x_1, f_{1;(1,0)} = f_{01}, f_{0;(0,1)} = x_2, f_{1;(0,1)} = f_{02}.$$

Next, we construct the lattice $L(2, 1)$ using the coefficients vector of polynomials

$$p^{1-\mathrm{WT}(i_0;I_n)} f_{i_0;I_n}(x_0 X_0, x_1 X_1, x_2 X_2),$$

where $X_i = p/2^\delta$, $0 \le i \le 2$. Finally, we compute out

$$\dim(L(2, 1)) = 6, F(2, 1) = 2/7.$$

Namely, we can recover the hidden number α when $\delta/\log_2 p \ge \frac{5}{7}$.

When $d = 2$, we arrange the order of all monomials in $\mathrm{MS}(2, 2)$ as follows.

$$1 \prec x_0 \prec x_0^2 \prec x_1 \prec x_0 x_1 \prec x_0^2 x_1 \prec x_2 \prec x_0 x_2 \prec x_0^2 x_2 \prec x_1 x_2 \prec x_0 x_1 x_2 \prec x_0^2 x_1 x_2.$$

Then, we generate remaining polynomials $f_{i_0;I_2}$ respectively.

$$f_{2;(0,0)} = x_0^2, f_{2;(1,0)} = x_0 f_{01}, f_{2;(0,1)} = x_0 f_{02},$$

$$f_{0;(1,1)} = (b_{02} - b_{01})^{-1}(x_2 f_{01} - x_1 f_{02}) \bmod p.$$

$$f_{1;(1,1)} = x_2 f_{01}, f_{2;(1,1)} = f_{01} f_{02}.$$

Next, we construct the lattice $L(2, 2)$ using the coefficients vector of polynomials

$$p^{2-\mathrm{WT}(i_0;I_n)} f_{i_0;I_n}(x_0 X_0, x_1 X_1, x_2 X_2).$$

Finally, we compute out

$$\dim(L(2, 2)) = 12, F(2, 2) = 1/3.$$

Thus, we can recover the hidden number α when $\delta/\log_2 p \ge \frac{2}{3}$.

C An Example of the Extended Strategy

We consider the case that $(n,k)=(2,2)$, when $d=1$, the corresponding monomial set $\mathrm{MS}(2,1,2)$ is

$$\{x_0^{i_0}x_1^{i_1}x_2^{i_2}, 0\le i_0\le 2, 0\le i_1+i_2\le 2\}.$$

We arrange all monomials in $\mathrm{MS}(2,1,2)$ according to the order (1) as follows.

$$1\prec x_0\prec x_0^2\prec x_1\prec x_0x_1\prec x_0^2x_1\prec x_2\prec x_0x_2\prec x_0^2x_2\prec$$

$$x_1^2\prec x_0x_1^2\prec x_0^2x_1^2\prec x_1x_2\prec x_0x_1x_2\prec x_0^2x_1x_2\prec x_2^2\prec x_0x_2^2\prec x_0^2x_2^2.$$

We have obtained some $f_{i_0;I_2}$ in Appendix B. Then, we generate the remaining polynomials

$$f_{0;(2,0)}=x_1^2, f_{1;(2,0)}=x_1f_{01}, f_{2;(2,0)}=f_{01}^2,$$
$$f_{0;(0,2)}=x_2^2, f_{1;(0,2)}=x_2f_{01}, f_{2;(0,2)}=f_{02}^2.$$

Next, we construct the lattice $L(2,1,2)$ using the coefficients vector of polynomials

$$p^{2-\mathrm{WT}(i_0;I_n)}f_{i_0;I_n}(x_0X_0,x_1X_1,x_2X_2).$$

Finally, we compute out

$$\dim(L(2,1,2))=18, F(2,1,2)=1/3.$$

Namely, we can recover the hidden number α when $\delta/\log_2 p\ge\frac{2}{3}$.

When $d=2$, the monomial set $\mathrm{MS}(2,2,2)$ is

$$\{x_0^{i_0}x_1^{i_1}x_2^{i_2}, 0\le i_0\le 4, 0\le i_1,i_2\le 2, 0\le i_1+i_2\le 4\}.$$

All monomials in $\mathrm{MS}(2,2,2)$ are ordered according to the following way:

$$1\prec x_0\prec x_0^2\prec x_0^3\prec x_0^4\prec$$
$$x_1\prec x_0x_1\prec x_0^2x_1\prec x_0^3x_1\prec x_0^4x_1\prec$$
$$x_2\prec x_0x_2\prec x_0^2x_2\prec x_0^3x_2\prec x_0^4x_2\prec$$
$$x_1^2\prec x_0x_1^2\prec x_0^2x_1^2\prec x_0^3x_1^2\prec x_0^4x_1^2\prec$$
$$x_1x_2\prec x_0x_1x_2\prec x_0^2x_1x_2\prec x_0^3x_1x_2\prec x_0^4x_1x_2\prec$$
$$x_2^2\prec x_0x_2^2\prec x_0^2x_2^2\prec x_0^3x_2^2\prec x_0^4x_2^2\prec$$
$$x_1^2x_2\prec x_0x_1^2x_2\prec x_0^2x_1^2x_2\prec x_0^3x_1^2x_2\prec x_0^4x_1^2x_2\prec$$
$$x_1x_2^2\prec x_0x_1x_2^2\prec x_0^2x_1x_2^2\prec x_0^3x_1x_2^2\prec x_0^4x_1x_2^2\prec$$
$$x_1^2x_2^2\prec x_0x_1^2x_2^2\prec x_0^2x_1^2x_2^2\prec x_0^3x_1^2x_2^2\prec x_0^4x_1^2x_2^2.$$

We generate the remaining polynomials $f_{i_0;I_2}$ as follows.

$$f_{3;(0,0)}=x_0^3,\ f_{4;(0,0)}=x_0^4,\ f_{3;(1,0)}=x_0^2f_{01},\ f_{4;(1,0)}=x_0^3f_{01},$$

$$f_{3;(0,1)} = x_0^2 f_{02},\ f_{4;(0,1)} = x_0^3 f_{02},\ f_{3;(1,1)} = x_0 f_{01} f_{02},\ f_{4;(1,1)} = x_0^2 f_{01} f_{02},$$
$$f_{3;(2,0)} = x_0 f_{01}^2,\ f_{4;(2,0)} = x_0^2 f_{01}^2,\ f_{3;(0,2)} = x_0 f_{02}^2,\ f_{4;(0,2)} = x_0^2 f_{02}^2,$$
$$f_{0;(1,2)} = x_2 f_{0;(1,1)},\ f_{1;(1,2)} = f_{02} f_{0;(1,1)},\ f_{2;(1,2)} = f_{02} f_{1;(1,1)},$$
$$f_{3;(1,2)} = f_{01} f_{02}^2,\ f_{4;(1,2)} = x_0 f_{01} f_{02}^2,\ f_{0;(2,1)} = x_1 f_{0;(1,1)},$$
$$f_{1;(2,1)} = f_{01} f_{0;(1,1)},\ f_{2;(2,1)} = f_{01} f_{1;(1,1)},\ f_{3;(2,1)} = f_{01}^2 f_{02},$$
$$f_{4;(2,1)} = x_0 f_{01}^2 f_{02},\ f_{0;(2,2)} = f_{0;(1,1)}^2,\ f_{1;(2,2)} = f_{0;(1,1)} f_{1;(1,1)},$$
$$f_{2;(2,2)} = f_{01} f_{02} f_{0;(1,1)},\ f_{3;(2,2)} = f_{01} f_{02} f_{1;(1,1)},\ f_{4;(2,2)} = f_{01}^2 f_{02}^2.$$

Similarly, we can construct the lattice $L(2,2,2)$ and compute out

$$\dim(L(2,2,2)) = 45, F(2,2,2) = 11/30.$$

Further, we can recover the hidden number α when $\delta/\log_2 p \geq \frac{19}{30}$.