Improving Bounds on Elliptic Curve Hidden Number Problem for ECDH Key Exchange

Jun Xu[1,2], Santanu Sarkar[3], Huaxiong Wang[4], Lei Hu[1,2]

[1] State Key Laboratory of Information Security, Institute of Information Engineering,Chinese Academy of Sciences, Beijing 100093, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China
[3] Indian Institute of Technology, Sardar Patel Road, Chennai 600036, India
[4] Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University Singapore, Singapore
{xujun,hulei}@iie.ac.cn, santanu@iitm.ac.in, hxwang@ntu.edu.sg

Abstract. Elliptic Curve Hidden Number Problem (EC-HNP) was first introduced by Boneh, Halevi and Howgrave-Graham at Asiacrypt 2001. To rigorously assess the bit security of the Diffie–Hellman key exchange with elliptic curves (ECDH), the Diffie–Hellman variant of EC-HNP, regarded as an elliptic curve analogy of the Hidden Number Problem (HNP), was presented at PKC 2017. This variant can also be used for practical cryptanalysis of ECDH key exchange in the situation of side-channel attacks.

In this paper, we revisit the Coppersmith method for solving the involved modular multivariate polynomials in the Diffie–Hellman variant of EC-HNP and demonstrate that, for any given positive integer d, a given sufficiently large prime p, and a fixed elliptic curve over the prime field $\mathbb{F}_p$, if there is an oracle that outputs about $\frac{1}{d+1}$ of the most (least) significant bits of the x-coordinate of the ECDH key, then one can give a heuristic algorithm to compute all the bits within polynomial time in $\log_2 p$. When $d > 1$, the heuristic result $\frac{1}{d+1}$ significantly outperforms both the rigorous bound $\frac{5}{6}$ and heuristic bound $\frac{1}{2}$. Due to the heuristics involved in the Coppersmith method, we do not get the ECDH bit security on a fixed curve. However, we experimentally verify the effectiveness of the heuristics on NIST curves for small dimension lattices.

Keywords. Hidden number problem, Elliptic curve hidden number problem, Modular inversion hidden number problem, Lattice, Coppersmith method.

1 Introduction

1.1 Background

At CRYPTO 1996, Boneh and Venkatesan [6] first proposed the hidden number problem (HNP) to prove that computing the most significant bits of the Diffie-Hellman (DH) key is as hard as computing the entire key in the DH key exchange for a prime field. It is called the bit security of the DH key exchange. There are a lot of follow-up works, such as [7,1] and [12, Chapter 21.7.1]. HNP has been proven to be an extremely useful tool in many cryptographic areas. One example is its vast use for analysis of DSA and ECDSA in side-channel attacks, such as [15,27]. At USENIX Security 2021, Merget et al. presented the first practical HNP-based attack on the DH key exchange [23]. Albrecht and Heninger presented a new result for solving HNP [2] at Eurocrypt 2021.

The ECDH key exchange is an analog of the DH key exchange, which adopts the group of points on an elliptic curve to enhance performance and security. Roughly speaking, for a given elliptic curve $\mathcal{E}$ over some finite field and a given point $Q \in \mathcal{E}$, two participants with private keys a, b compute $[a]Q, [b]Q$ separately, then send the computed value to each other, and finally, the two participants generate the shared key $[ab]Q$. Naturally, one may want to assess the difficulty of computing partial bits of ECDH key exchange. At ANTS 1998, Boneh [3, Section 5] proposed the open problem: *Does a similar result to the bit security of Diffie-Hellman key exchange [6] hold in the group of points of an elliptic curve?* The issue has been raised for 20 years, but few results have been presented because of the complexity associated with the addition formula of points of an elliptic curve. The reason is also presented in the introduction of papers such as [5,16,28,32].

EC-HNP. In [4, Section 5], Boneh, Halevi and Howgrave-Graham presented the elliptic curve hidden number problem (EC-HNP) to study the bit security of ECDH. The authors stated that EC-HNP can be heuristically solved using the idea from Method II for Modular Inversion Hidden Number Problem (MIHNP). Furthermore, they mentioned that the heuristic approach can be converted into a rigorous one in some cases, which corresponds to the following bit security result. Computing $(1-\epsilon)$ of the most significant bits of the x-coordinate of the ECDH key is as hard as computing the ECDH key itself for a given curve over a prime field, where $\epsilon \approx 0.02$. The detailed proofs were not presented.

Shani [28] demonstrated at PKC 2017 that solving EC-HNP $_x$, which can be viewed as the Diffie-Hellman variant of EC-HNP, is sufficient to demonstrate the bit security of ECDH. The involved strategy is similar to the idea of HNP [6].

Definition 1 (EC-HNP$_x$ [28]). *Fix a prime p, a given elliptic curve $\mathcal{E}$ over $\mathbb{F}_p$, a given point $R \in \mathcal{E}$ and a positive number δ. Let $P \in \mathcal{E}$ be a hidden point. Let $O_{P,R}$ be an oracle that on input m outputs the δ most significant bits of the x-coordinate of $P + [m]R$. That is, $O_{P,R}(m) = \mathrm{MSB}_\delta(x_{P+[m]R})$. The goal is to recover the hidden point P, given query access to the oracle.*

Suppose there is an oracle that outputs some partial information of $[uv]Q$ on input $[u]Q$ and $[v]Q$. For given points Q, $[a]Q$ and $[b]Q$ in the ECDH key exchange, an attacker first selects an integer m, computes $[m]Q$, and then obtains $[a+m]Q$ from $[a]Q+[m]Q = [a+m]Q$. Querying the oracle on input $[a+m]Q$ and $[b]Q$, the attacker can get partial information of $[(a+m)b]Q = [ab]Q + [m][b]Q = P + [m]R$ where $P := [ab]Q$ and $R := [b]Q$. By repeating this process for several m's, the attacker will be able to recover the ECDH key $P = [ab]Q$ if the EC-HNP$_x$ is solved.

In [23, Section 8], Merget et al. mentioned that *this may result in a small timing side-channel information that leaks the MSB of the x-coordinate of the shared point in ECDH. The EC-HNP is related to the HNP and could potentially be applied here.* We contend that the aforementioned attack scenario falls within the scope of EC-HNP$_x$. This attack scenario specifically considers whether the server reuses the same ECDH value $R = [b]Q$ across sessions, where b is the server's static key in TLS-ECDH or a reusable ephemeral key in TLS-ECDHE. A client generates secret a and transmits the value $[a]Q$. Hence, the ECDH key between the server and the client is $P = [ab]Q$. An attacker first chooses some integer m and computes $[a+m]Q$. Then, session's ECDH secret is $[(a+m)b]Q = P + [m]R$. (The above process is very similar to [23, Figure 1]). As a result, if the MSBs of the x-coordinate of $P + [m]R$ are leaked by the small timing side-channel attack for several m, the attacker can obtain the ECDH key P by solving EC-HNP$_x$. EC-HNP$_x$, like HNP, can play an important role in side-channel attacks.

Hardcore bits. Shani rigorously solved EC-HNP$_x$ and then obtained the following bit security result by combining the underlying idea from Method I for MIHNP [4,21]. For a given curve over a prime field, computing about $\frac{5}{6}$ of the most (least) significant bits of the x-coordinate of the ECDH key is as hard as computing the entire ECDH key. Besides, Shani also analyzed the case of extension fields and generalized the result of Jao, Jetchev and Venkatesan [16].

Papers such as [6,28] demonstrated that DH and ECDH have hardcore bits, which are bits that are difficult to compute as the full shared key.

Heuristic algorithm. In [32], Xu et al. used the Coppersmith method to solve EC-HNP$_x$, which was inspired by Method II of MIHNP [4,33]. For a fixed curve over a prime field, if there is an oracle that outputs about $\frac{1}{2}$ of the most (least) significant bits of the x-coordinate of the ECDH key, then there is a heuristic algorithm to compute all the bits in polynomial time.

The Coppersmith method is used to calculate small solutions of polynomials. In 1996, Coppersmith proposed rigorous methods for finding the small roots of a modular univariate polynomial and an integer bivariate polynomial [8,9]. In 2006, Jochemsz and May [18] presented heuristic strategies for finding the small roots of modular (and integer) multivariate polynomials. The Coppersmith method is widely used in the security analysis of cryptosystems, the computational complexity analysis of mathematical problems, and the security proof of cryptosystems; see the survey [22] and recent papers, such as [30,24,10,34].

Since the Coppersmith method for modular multivariate polynomials is heuristic, the result in [32] cannot prove that ECDH has hardcore bits. It is important to note that EC-HNP$_x$ is directly related to the actual cryptanalysis of ECDH key exchange for a fixed curve in the work of side-channel attacks [23]. The problem of solving EC-HNP$_x$ is essentially the problem of finding the desired small root of modular multivariate polynomials. The advantage of the Coppersmith method is that it utilizes algebraic structures of polynomials to improve the ability to find small roots. A natural motivation is that one wants to know the best result if the Coppersmith method is used to deal with EC-HNP$_x$.

Related works. At CRYPTO 2001, Boneh and Shparlinksi [5] showed that if there is an efficient algorithm to predict the least significant bit (LSB) of the ECDH secrets on a non-negligible fraction of a family of curves isomorphic to a curve $\mathcal{E}_0$, then the ECDH key for the curve $\mathcal{E}_0$ can be computed in polynomial time. It does not imply that computing a single LSB of the ECDH key is as hard as computing the entire ECDH key for the same curve $\mathcal{E}_0$. At CRYPTO 2008, Jetchev and Venkatesan [17] utilized isogenies to enlarge the applicability of the method in [5] based on the generalized Riemann hypothesis. However, neither [5] nor [17] provides the hardness of bits for ECDH for a fixed curve. In [5, Section 7], Boneh and Shparlinksi mentioned that they hope their methods will eventually show that a single LSB of ECDH is the hardcore bit for a fixed curve.

1.2 Our Contribution

In this paper, we revisit the Coppersmith method to solve modular multivariate polynomials derived from EC-HNP$_x$ and obtain a new bound.

Result 1. *Let d be any given positive integer. Given a sufficiently large prime $p = 2^{\omega(d^{(2+c)d})}$, and a positive $n = d^{3+c}$ for any constant $c > 0$. For $2n+1$ given calls to the oracle in EC-HNP$_x$, under Assumption 1 (see Page 8), one can recover the hidden point for EC-HNP$_x$ when the number δ of known MSBs (LSBs) satisfies*

$$\frac{\delta}{\log_2 p} > \frac{1}{d+1} + \varepsilon, \tag{1}$$

where $\varepsilon > 0$ and $\varepsilon = o(\frac{1}{d+1})$. The total time complexity is polynomial in $\log_2 p$ for any constant d.

Corresponding to the ECDH case, we have the following result.

Result 2. *Define d, p as in Result 1. Under Assumption 1, one can compute all the bits in polynomial time for a given elliptic curve $\mathcal{E}$ over the prime field $\mathbb{F}_p$ if there is an oracle that outputs about $\frac{1}{d+1}$ of the most (least) significant bits of the x-coordinate of the ECDH key.*

The bound (1) tends to $\delta/\log_2 p > 0$ as d grows large. It means that the ratio of known MSBs or LSBs number can be infinitesimal. When d becomes large, the modulus $p = 2^{\omega(d^{(2+c)d})}$, the involved lattice dimension $w = \mathcal{O}(n^{d+1})$, and the running time of the algorithm become enormous, with the time complexities of the LLL algorithm and the Gröbner basis computation increasing as $d^{\mathcal{O}(d)}$ and $d^{\mathcal{O}(n)}$, respectively.

The heuristic bound (1) for $d > 1$ is better than the rigorous bound $\delta/\log_2 p > \frac{5}{6}$ [28] and the heuristic result $\delta/\log_2 p > \frac{1}{2}$ [32]. Due to the heuristics of the Coppersmith method, the results in this paper and [32] are not rigorous. It should be noted that the $\frac{1}{2}$ bound on $\delta/\log_2 p$ in [32] is asymptotic. That is, the $\frac{1}{2}$ bound can only be reached when the involved lattice dimension and modulus p tend to infinity (see the analysis of Section 1.3). In this work, the smallest dimensions of our lattice to achieve the $\frac{1}{2}$ bound is 2879 for a sufficiently large $p = 2^{\omega(d^{(2+c)d})}$, where $d = 2$. The LLL algorithm terminates within $\mathcal{O}(w^{4+\gamma}b^{1+\gamma})$ bit operations for any $\gamma > 0$ [25], where w is the involved dimension and b is the maximal bit size in the input basis matrix. For our case, $w = 2879$, $w^4 \approx 2^{46}$ and b is bounded by $3d\log_2 p$. Therefore, the LLL algorithm needs a considerable time to get the desired vector. Thus, we do not experimentally show that the $\frac{1}{2}$ barrier is broken.

1.3 Technical overview

As mentioned before, we revisit the Coppersmith method to find the desired root $(e_0, \tilde{e}_1, \cdots, \tilde{e}_n)$ in n given polynomials

$$\mathcal{F}_j(x_0, y_j) := A_j + B_j x_0 + C_j x_0^2 + D_j y_j + E_j x_0 y_j + x_0^2 y_j$$

derived from EC-HNP$_x$, satisfying $\mathcal{F}_j(e_0, \widetilde{e_j}) = 0 \bmod p$ for $1 \leq j \leq n$, where the value X is the upper bound of $|e_0|, |\tilde{e}_1|, \cdots, |\tilde{e}_n|$, i.e., $|e_0| < X, |\tilde{e}_1| < X, \cdots, |\tilde{e}_n| < X$. Since $X = p/2^{\delta}$ for EC-HNP$_x$, where p is the modulus and δ is the number of known MSBs (LSBs), we can see that for a fixed p, X and δ are inversely related. To make δ as small as possible, X must be as large as possible.

For any given positive integer d, we construct w multivariate polynomials $G_1(x_0, y_1, \cdots, y_n)$, $\cdots$, $G_w(x_0, y_1, \cdots, y_n)$ satisfying $G_j(e_0, \tilde{e}_1, \cdots, \tilde{e}_n) = 0 \bmod p^d$ for all $1 \leq j \leq w$. Let $\mathcal{L}$ be a Coppersmith lattice, which is spanned by the coefficient vectors of $G_j(x_0X, y_1X, \cdots, y_nX)$ for all $1 \leq j \leq w$, where w and $\det(\mathcal{L})$ are the dimension and determinant of the lattice $\mathcal{L}$, respectively. The basis matrix of $\mathcal{L}$ can be arranged into a triangular matrix.

After the lattice basis reduction, we expect to get $n+1$ multivariate polynomials $Q_1(x_0, y_1, \cdots, y_n)$, $\cdots$, $Q_{n+1}(x_0, y_1, \cdots, y_n)$ such that $Q_j(e_0, \tilde{e}_1, \cdots, \tilde{e}_n) = 0$ over the integers for all $1 \leq j \leq n$. Under Assumption 1, we can efficiently recover the desired root $(e_0, \tilde{e}_1, \cdots, \tilde{e}_n)$.

In the Coppersmith method, for a sufficiently large modulus p, the condition for finding the target root $(e_0, \tilde{e}_1, \cdots, \tilde{e}_n)$ can be briefly written as

$$(\det(\mathcal{L}))^{\frac{1}{w}} < p^d. \tag{2}$$

As shown in [28,32], the strategy of solving MIHNP can help to solve EC-HNP$_x$. Inspired by the approach for MIHNP [34], we expect to add enough helpful vectors into the lattice of [32].

In [32], a lattice $\mathcal{L}'$ with triangular basis matrix was constructed. For any given positive integer d, take $n = d^3$. Then we can write $\dim(\mathcal{L}') = (2d+1)\binom{n}{d}(1+o(1))$, and $\det(\mathcal{L}') = X^{\overline{\alpha}} p^{\overline{\beta}}$, where $\overline{\alpha} = 2d(2d+1)\binom{n}{d}(1+o(1))$ and $\overline{\beta} = 2d\binom{n}{d}(1+o(1))$. For a sufficiently large $p = 2^{\omega(2^n)}$, the Coppersmith condition (2) states: $|\det(\mathcal{L}')|^{\frac{1}{\dim(\mathcal{L}')}} < p^d$, which reduces to $X < p^{\frac{1}{2} - \frac{1}{2d} - \overline{\varepsilon}}$, where $\overline{\varepsilon} > 0$ and $\overline{\varepsilon} = o(\frac{1}{d})$. Plugging $X = p/2^{\delta}$ into the above relation, we get $\delta/\log_2 p > \frac{1}{2} + \frac{1}{2d} + \overline{\varepsilon}$, which becomes $\delta/\log_2 p > \frac{1}{2}$ when d tends to infinity. It means that, in order to achieve 1/2 bound, the involved lattice dimension $\dim(\mathcal{L}')$ and the size of modulus p tend to infinity.

In this paper, we first consider $\binom{n}{d+1}$ of helpful polynomials. To be specific, we randomly choose $d+1$ different integers from the set $\{1, \cdots, n\}$. Without loss of generality, let $d+1$ integers be $j_1, \cdots, j_{d+1}$, where $1 \le j_1 < \cdots < j_{d+1} \le n$. For any fixed tuple $(j_1, \cdots, j_{d+1})$, we choose a linear combination (with the leading term $y_{j_1} \cdots y_{j_{d+1}}$) of the following polynomials:

$$\sum_{u=1}^{d+1} \sum_{v=0}^{1} K_{u,v} \cdot x_0^v \mathcal{F}_{j_1} \cdots \mathcal{F}_{j_{u-1}} y_{j_u} \mathcal{F}_{j_{u+1}} \cdots \mathcal{F}_{j_{d+1}} \text{ for some } K_{u,v} \in \mathbb{Z}. \tag{3}$$

We then consider the algebraic structure of linear combinations (3) and design a lattice. We construct more compact linear combinations compared to (3) so that all monomials related to x_0^{2d} and x_0^{2d+1} are removed. That is, the monomials $x_0^{i_0} y_1^{i_1} \cdots y_n^{i_n}$ for all $(i_0, i_1, \cdots, i_n) \in \mathcal{I}_3$ are deleted from new linear combinations, where $\mathcal{I}_3 := (\{(i_0, i_1, \cdots, i_n) \mid 2d \le i_0 \le 2d+1, 0 \le i_1, \cdots, i_n \le 1, 0 \le i_1 + \cdots + i_n \le d\}$. Then we get a lattice with triangular basis matrix. In this case, we can deduce that the upper bound $X < p^{1-\frac{1}{d+1}-\varepsilon}$, where $\varepsilon > 0$ and $\varepsilon = o(\frac{1}{d+1})$. Based on $X = p/2^\delta$, we obtain $\delta/\log_2 p > \frac{1}{d+1} + \varepsilon$, which becomes $\delta/\log_2 p > 0$ when d tends to infinity.

The polynomial construction for the lattice in this work looks similar to that in [32]. However, this does not mean that our lattice construction is ordinary. When it comes to the Coppersmith method, small differences in parameter selection can lead to significant differences in efficiency. While dealing with multivariate Coppersmith method, the core point and technical difficulty is constructing as many helpful polynomials as possible. The rest is a conventional technique.

1.4 Organization

The rest of this paper is organized as follows. In Section 2, we review some results on lattice, the Coppersmith method, elliptic curves, the transformation from EC-HNP$_x$ to a class of modular polynomials, and orders of monomials. The existing method is revisited in Section 3. In Section 4, we use algebraic structure of polynomials to design a lattice. In Section 5, we prove that the involved basis matrix is triangular. In Section 6, we compare our result with the existing work. We present our experimental results in Section 7.

2 Preliminaries

Throughout the paper, p is a prime where $p > 3$.

2.1 Lattice

A lattice $\mathcal{L}$ is a discrete subgroup of $\mathbb{R}^m$. An alternative equivalent definition of an integer lattice can be given using a basis. Let $\mathbf{b_1}, \cdots, \mathbf{b_w}$ be linear independent row vectors in $\mathbb{R}^m$, a lattice $\mathcal{L}$ spanned by them is

$$\mathcal{L} = \left\{ \sum_{i=1}^{w} k_i \mathbf{b_i} \mid k_i \in \mathbb{Z} \right\}.$$

The set $\{\mathbf{b_1}, \cdots, \mathbf{b_w}\}$ is called a basis of $\mathcal{L}$ and the matrix $\mathbf{B} = [\mathbf{b_1}^T, \cdots, \mathbf{b_w}^T]^T$ is the corresponding basis matrix. The dimension and determinant of $\mathcal{L}$ are respectively

$$\dim(\mathcal{L}) = w, \det(\mathcal{L}) = \sqrt{\det(\mathbf{B}\mathbf{B}^T)}.$$

When $m = w$, lattice is called full rank. In this paper, the involved lattices are full-rank integer lattices.

The well-known LLL lattice reduction algorithm [20] can produce a reduced basis that has the following property.

Lemma 1 (LLL). *Let $\mathcal{L}$ be a w-dimensional integer lattice. Within polynomial time, the LLL algorithm outputs reduced basis vectors $\mathbf{v}_1, \ldots, \mathbf{v}_w$ that satisfy*

$$\|\mathbf{v}_i\| \leq 2^{\frac{w(w-1)}{4(w+1-i)}} (\det(\mathcal{L}))^{\frac{1}{w+1-i}}, 1 \leq i \leq w.$$

2.2 The Coppersmith method

We briefly review how to use the Coppersmith method to solve multivariate modular polynomials.

Problem definition. Let $f_1(x_0, x_1, \cdots, x_n), \cdots, f_m(x_0, x_1, \cdots, x_n)$ be original polynomials, which are irreducible multivariate polynomials defined over $\mathbb{Z}$, with a common root $(\widetilde{x}_0, \widetilde{x}_1, \cdots, \widetilde{x}_n)$ modulo a known integer p such that $|\widetilde{x}_0| < X_0$, $\cdots$, $|\widetilde{x}_n| < X_n$. The goal is to recover the desired root $(\widetilde{x}_0, \cdots, \widetilde{x}_n)$ in polynomial time. To ensure recovery of the desired root, the size of values $X_0, \cdots, X_n$ must be bound.

Polynomials collection. One chooses polynomials,

$$g_1(x_0, x_1, \cdots, x_n), \cdots, g_w(x_0, x_1, \cdots, x_n)$$

such that $(\widetilde{x}_0, \widetilde{x}_1, \cdots, \widetilde{x}_n)$ is a common root modulo a power of p. Generally, multiples of lifting polynomials are selected, where a lifting polynomial is defined as the product of some powers of original polynomials and variables. For example,

$$g_j(x_0, x_1, \cdots, x_n) := p^{d-(\beta_1^j + \cdots + \beta_m^j)} x_0^{\alpha_0^j} x_1^{\alpha_1^j} \cdots x_n^{\alpha_n^j} f_1^{\beta_1^j} \cdots f_m^{\beta_m^j},$$

where $j \in \{1, \cdots, w\}$, $d \in \mathbb{Z}^+$, and $\alpha_0^j, \alpha_1^j, \cdots, \alpha_n^j, \beta_1^j, \cdots, \beta_m^j \in \mathbb{Z}^+ \cup \{0\}$ satisfying $0 \leq \beta_1^j + \cdots + \beta_m^j \leq d$. It is not hard to see that $g_j(\widetilde{x}_0, \widetilde{x}_1, \cdots, \widetilde{x}_n) \equiv 0 \bmod p^d$ for every $j \in \{1, \cdots, w\}$. For the Coppersmith method, the most complex step is the selection of polynomials $g_1, \cdots, g_w$ when dealing with multiple original polynomials. The difference between this paper's polynomial selection and the above strategy is that linear combinations of lifting polynomials are considered.

Lattice construction. Let the vector $\mathbf{b}_j$ $(1 \le j \le w)$ be the coefficient vector of the polynomial $g_j(x_0X_0, x_1X_1, \ldots, x_nX_n)$ with variables $x_0, x_1, \ldots, x_n$. Then one constructs the lattice $\mathcal{L} = \left\{ \sum_{j=1}^{w} k_j \mathbf{b}_j \;\middle|\; k_j \in \mathbb{Z} \right\}$.

Reduced basis. One runs the LLL algorithm and obtains the w reduced basis vectors $\mathbf{v}_1, \ldots, \mathbf{v}_w$, where $\mathbf{v}_j$ is the coefficient vector of the polynomial $h_j(x_0X_0,\ x_1X_1, \ldots, x_nX_n)$ for $j \in \{1, \cdots, w\}$. Note that the LLL algorithm performs linear operations. Hence, $\mathbf{v}_j$ is a linear combination of the vectors $\mathbf{b}_1, \cdots, \mathbf{b}_w$. That is, $h_j(x_0, x_1, \ldots, x_n)$ is a linear combination of $g_1(x_0, x_1, \ldots, x_n), \cdots, g_w(x_0, x_1, \ldots, x_n)$. Then, $h_j(\widetilde{x}_0, \widetilde{x}_1, \cdots, \widetilde{x}_n) = 0 \pmod{p^d}$ for every $j \in [1, \cdots, w]$. In order to get $h_j(\widetilde{x}_0, \widetilde{x}_1, \cdots, \widetilde{x}_n) = 0$ over $\mathbb{Z}$ for some $j \in \{1, \cdots, w\}$, we need the following lemma in this process.

Lemma 2 ([14]). *Let $h(x_0, x_1, \ldots, x_n)$ be an integer polynomial that consists of at most w monomials. Let d be a positive integer and the integers X_i be the upper bound of $|\widetilde{x}_i|$ for $i = 0, 1, \cdots, n$. Let $\|h(x_0X_0, x_1X_1, \ldots, x_nX_n)\|$ be the Euclidean length of the coefficient vector of $h(x_0X_0, x_1X_1, \ldots, x_nX_n)$ with variables $x_0, x_1, \ldots, x_n$. Suppose that*

1. $h(\widetilde{x}_0, \widetilde{x}_1, \cdots, \widetilde{x}_n) = 0 \pmod{p^d}$,
2. $\|h(x_0X_0, x_1X_1, \ldots, x_nX_n)\| < \frac{p^d}{\sqrt{w}}$,

then $h(\widetilde{x}_0, \widetilde{x}_1, \cdots, \widetilde{x}_n) = 0$ holds over $\mathbb{Z}$.

To make $h_j(\widetilde{x}_0, \widetilde{x}_1, \cdots, \widetilde{x}_n) = 0$ for all $1 \le j \le n+1$ hold, from Lemma 1 and Lemma 2, we need the Euclidean lengths of the $n+1$ reduced basis vectors $\mathbf{v}_1, \ldots, \mathbf{v}_{n+1}$ satisfy the condition

$$2^{\frac{w(w-1)}{4(w-n)}} \cdot \big(\det(\mathcal{L})\big)^{\frac{1}{w-n}} < \frac{p^d}{\sqrt{w}},\ w = \dim(\mathcal{L}). \tag{4}$$

Based on Condition (4), one can determine the size of bounds $X_0, \cdots, X_n$.

Desired root recovery. We have no assurance that the $n+1$ obtained polynomials $h_1, \cdots, h_{n+1}$ are algebraically independent. Under Assumption 1, the corresponding equations can be solved using elimination techniques such as the Gröbner basis computation, and then the desired root $(\widetilde{x}_0, \widetilde{x}_1, \cdots, \widetilde{x}_n)$ is recovered. In this paper, we use computer experiments to show that our heuristic approach works.

Assumption 1 ([19]). *Let $h_1, \cdots, h_{n+1} \in \mathbb{Z}[x_0, x_1, \cdots, x_n]$ be the polynomials that are found by the Coppersmith method. Then the ideal generated by the polynomial equations $h_1(x_0, x_1, \cdots, x_n) = 0$, $\cdots$, $h_{n+1}(x_0, x_1, \cdots, x_n) = 0$ has dimension zero.*

The involved Assumption 1 is called the zero-dimensional ideal assumption, which is a relaxation of algebraically independent assumption, first appeared in [19]. We consider a zero-dimensional ideal, namely, an ideal I such that the number of common zeros of the polynomials in I is finite in the algebraic closure of the field of coefficients [11]. It seems very difficult to verify whether there are finite number of common zeros or not.

Helpful polynomials. An important strategy of choosing the above polynomials $g_1(x_0, x_1, \cdots, x_n)$, $\cdots$, $g_w(x_0, x_1, \cdots, x_n)$ is to choose as many *helpful polynomials* as possible.

Definition 2 ([22,29]). *Define d and $\mathcal{L}$ as above. A vector in the triangular basis matrix, which is the coefficient vector of $g(x_0X, x_1X, \cdots, x_nX)$, is called a helpful vector if the absolute value of its diagonal component*[5] *is greater than 0 and less than p^d. That is, $g(x_0, x_1, \cdots, x_n)$ is called a helpful polynomial*[6]*. Else, $g(x_0, x_1, \cdots, x_n)$ is called a non-helpful polynomial.*

Next, we show why helpful polynomials can work. We obtain the simplified condition $(\det(\mathcal{L}))^{\frac{1}{w}} < p^d$ by ignoring low-order terms in Condition (4). For a triangular basis matrix, the left side of the simplified condition is regarded as the geometric mean of all diagonals of the basis matrix. A helpful polynomial contributes to the determinant with a factor greater than 0 and less than p^d. The more helpful polynomials in the lattice, the easier the condition for solving modular equations is to be satisfied. It implies that the Coppersmith method becomes more and more effective, and the above bounds X_i become larger and larger. Therefore, one should choose as many helpful polynomials as possible.

2.3 Elliptic curves

For a prime field $\mathbb{F}_p$, consider an elliptic curve $\mathcal{E}$ over $\mathbb{F}_p$, given in a Weierstrass form $\mathcal{E} : y^2 = x^3 + ax + b$ over $\mathbb{F}_p$ with $a, b \in \mathbb{F}_p$ and $4a^3 + 27b^2 \neq 0$. Let $P = (x_P, y_P) \in \mathbb{F}_p^2$ be a point on the curve $\mathcal{E}$, where x_P (resp. y_P) is called the x-coordinate (resp. y-coordinate) of point P. The set of points on $\mathcal{E}$, together with the point at infinity O, forms an additive abelian group. Hasse's theorem shows that the number of points $\#\mathcal{E}$ on the curve $\mathcal{E}(\mathbb{F}_p)$ satisfies the relation: $|\#\mathcal{E} - p - 1| \leq 2\sqrt{p}$. The additive inverse of point P is $-P = (x_P, -y_P)$. For an integer m, $[m]P$ denotes successive m-time addition of the point P, and $[-m]P = m[-P]$. Given two points $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ on $\mathcal{E}$, where

[5] The diagonal component of the coefficient vector of $g(x_0X, x_1X, \cdots, x_nX)$ corresponds to the leading term of $g(x_0, x_1, \cdots, x_n)$. Specifically, the diagonal component is equal to the leading coefficient of $g(x_0X, x_1X, \cdots, x_nX)$.

[6] There is a one-to-one correspondence between helpful polynomials and helpful vectors. The coefficient vector of $g(x_0X, x_1X, \cdots, x_nX)$ is a helpful vector if and only if $g(x_0, x_1, \cdots, x_n)$ is a helpful polynomial.

$P \neq \pm Q$, consider the addition $P+Q = (x_{P+Q}, y_{P+Q})$. Let $s_{P+Q} = \frac{y_P - y_Q}{x_P - x_Q}$. The x-coordinate and y-coordinate of $P+Q$ are respectively

$$x_{P+Q} = s_{P+Q}^2 - x_P - x_Q,\ y_{P+Q} = s_{P+Q}(x_P - x_{P+Q}) - y_P. \tag{5}$$

2.4 From EC-HNP$_x$ to modular polynomials

We present the transformation in [28] from the problem of recovering x_P in EC-HNP$_x$ (see Definition 1), the x-coordinate of the hidden point $P = (x_P, y_P)$, to the problem of finding small solutions of modular polynomials. In brief, our target is to find the desired small root $(e_0, \tilde{e}_i)$ of the following modular polynomial

$$\mathcal{F}_i(x_0, y_i) := A_i + B_i x_0 + C_i x_0^2 + D_i y_i + E_i x_0 y_i + x_0^2 y_i = 0 \pmod{p},\ 1 \leq i \leq n. \tag{6}$$

Here coefficients A_i, B_i, C_i, D_i, E_i are known, and unknown integers $e_0, \tilde{e}_1, \cdots, \tilde{e}_n$ are all bounded by the value $X := p/2^\delta$. The specific analysis is as follows.
Eliminating y_P. For a given point R in an elliptic curve $\mathcal{E}$ over $\mathbb{F}_p$, we produce $Q = [m]R = (x_Q, y_Q)$ and $-Q = [-m]R = (x_Q, -y_Q)$, where m is a positive integer. According to $y_P^2 = x_P^3 + ax_P + b$, $y_Q^2 = x_Q^3 + ax_Q + b$ and (5), we obtain

$$\begin{aligned}
&x_{P+Q} + x_{P-Q} = (s_{P+Q}^2 - x_P - x_Q) + (s_{P-Q}^2 - x_P - x_Q) \\
&= \left(\frac{y_P - y_Q}{x_P - x_Q}\right)^2 + \left(\frac{y_P + y_Q}{x_P - x_Q}\right)^2 - 2x_P - 2x_Q \\
&= 2\left(\frac{y_P^2 + y_Q^2}{(x_P - x_Q)^2} - x_P - x_Q\right) \\
&= 2\left(\frac{x_Q x_P^2 + (a + x_Q^2)x_P + ax_Q + 2b}{(x_P - x_Q)^2}\right).
\end{aligned} \tag{7}$$

Constructing modular polynomials. Query the oracle $O_{P,R}$ in EC-HNP$_x$ on $2n+1$ different inputs 0 and $\pm m_i$ for $i = 1, \cdots, n$. Then we obtain $O_{P,R}(0)$ and $O_{P,R}(\pm m_i)$. We write $h_i = O_{P,R}(m_i) = \mathrm{MSB}_\delta(x_{P+Q_i}) = x_{P+Q_i} - e_i$ and $h_i' = O_{P,R}(-m_i) = \mathrm{MSB}_\delta(x_{P-Q_i}) = x_{P-Q_i} - e_i'$, where $|e_i| < p/2^{\delta+1}$ and $|e_i'| < p/2^{\delta+1}$ for all $1 \leq i \leq n$. Let $\tilde{h}_i = h_i + h_i'$ and $\tilde{e}_i = e_i + e_i'$, we have $\tilde{h}_i + \tilde{e}_i = x_{P+Q_i} + x_{P-Q_i}$, where $|\tilde{e}_i| < p/2^\delta$ for $i = 1, \cdots, n$. According to (7), we get

$$\tilde{h}_i + \tilde{e}_i = 2\left(\frac{x_{Q_i} x_P^2 + (a + x_{Q_i}^2)x_P + ax_{Q_i} + 2b}{(x_P - x_{Q_i})^2}\right),\ 1 \leq i \leq n. \tag{8}$$

Moreover, we write $h_0 = O_{P,R}(0) = \mathrm{MSB}_\delta(x_P) = x_P - e_0$, where $|e_0| < p/2^{\delta+1}$. Hence, $\tilde{h}_i + \tilde{e}_i = 2(\frac{x_{Q_i}(h_0+e_0)^2 + (a+x_{Q_i}^2)(h_0+e_0) + ax_{Q_i} + 2b}{(h_0+e_0-x_{Q_i})^2})$. After multiplying by $(h_0 + e_0 - x_{Q_i})^2$, we get $A_i + B_i e_0 + C_i e_0^2 + D_i \tilde{e}_i + E_i e_0 \tilde{e}_i + e_0^2 \tilde{e}_i = 0 \bmod p$, $1 \leq i \leq n$, where known coefficients A_i, B_i, C_i, D_i, E_i satisfy (in the field $\mathbb{F}_p$)

$$\begin{aligned}
&A_i = \left(\tilde{h}_i(h_0 - x_{Q_i})^2 - 2h_0^2 x_{Q_i} - 2(a + x_{Q_i}^2)h_0 - 2ax_{Q_i} - 4b\right), \\
&B_i = 2(\tilde{h}_i(h_0 - x_{Q_i}) - 2h_0 x_{Q_i} - a - x_{Q_i}^2), C_i = (\tilde{h}_i - 2x_{Q_i}), \\
&D_i = (h_0 - x_{Q_i})^2, E_i = 2(h_0 - x_{Q_i}).
\end{aligned} \tag{9}$$

Therefore, $(e_0, \tilde{e}_i)$ is a small root of the polynomial

$$\mathcal{F}_i(x_0, y_i) = A_i + B_i x_0 + C_i x_0^2 + D_i y_i + E_i x_0 y_i + x_0^2 y_i = 0 \pmod{p},$$

where $1 \le i \le n$ and $e_0, \tilde{e}_1, \cdots, \tilde{e}_n$ are all bounded by $X := p/2^\delta$. Once the desired vector $(e_0, \tilde{e}_1, \cdots, \tilde{e}_n)$ is obtained, x_P can be recovered based on $x_P = e_0 + h_0$. After x_P is recovered, y_P will be extracted due to $y_P^2 = x_P^3 + a x_P + b \bmod p$.

2.5 Order of monomials

We first recall reverse lexicographic order and graded lexicographic reverse order respectively. For more details, please refer to [31, Section 21.2]. Let i_0, i_1, $\cdots$, i_n, i'_0, i'_1, $\cdots$, i'_n be nonnegative integers.

Reverse lexicographic order: $(i'_1, \cdots, i'_n) \prec_{revlex} (i_1, \cdots, i_n) \Leftrightarrow$ the rightmost nonzero entry in $(i'_1 - i_1, \cdots, i'_n - i_n)$ is negative.

Graded reverse lexicographic order: $(i'_1, \cdots, i'_n) \prec_{grevlex} (i_1, \cdots, i_n) \Leftrightarrow \sum_{m=1}^{n} i'_m < \sum_{m=1}^{n} i_m$ or $\big(\sum_{m=1}^{n} i'_m = \sum_{m=1}^{n} i_m$ and $(i'_1, \cdots, i'_n) \prec_{revlex} (i_1, \cdots, i_n)\big)$.

In this paper, we utilize the following order of monomials, which is also used in [34].

$$\begin{aligned} &x_0^{i'_0} y_1^{i'_1} \cdots y_n^{i'_n} \prec x_0^{i_0} y_1^{i_1} \cdots y_n^{i_n} \Leftrightarrow \\ &(i'_1, \cdots, i'_n) \prec_{grevlex} (i_1, \cdots, i_n) \text{ or } \big((i'_1, \cdots, i'_n) = (i_1, \cdots, i_n) \text{ and } i'_0 < i_0\big). \end{aligned} \tag{10}$$

It is noteworthy that i_0 and i'_0 are treated differently than $i_1, \cdots, i_n$ and $i'_1, \cdots, i'_n$ respectively. According to (10), we can determine the leading term of a multivariate polynomial. For example, for $\mathcal{F}_j = A_j + B_j x_0 + C_j x_0^2 + D_j y_j + E_j x_0 y_j + x_0^2 y_j$ for $1 \le j \le n$ in (6), we have

$$1 \prec x_0 \prec x_0^2 \prec y_j \prec x_0 y_j \prec x_0^2 y_j. \tag{11}$$

Hence, the leading monomial of $\mathcal{F}_j$ is $x_0^2 y_j$. Further, the leading coefficient of $\mathcal{F}_j$ is 1, and the leading term of $\mathcal{F}_j$ is $x_0^2 y_j$.

3 Existing Lattice

In this section, we review the lattice in [32] for solving (6). Here we provide a different description of the lattice, closer to the lattice we introduce later. First, we recall the index set

$$\begin{aligned} \mathcal{I}_{[\mathrm{XHS20}]}(n, d) = \{(i_0, i_1, \cdots, i_n) \mid\ & 0 \le i_0 \le 2d, \\ & 0 \le i_1, \cdots, i_n \le 1, 0 \le l \le d\}, \end{aligned} \tag{12}$$

where integers n, d satisfying $1 \le d \le n$, and $l := i_1 + \cdots + i_n$ satisfying $0 \le l \le d$.

3.1 Lattice $\mathcal{L}_{[\mathrm{XHS20}]}(n, d)$

For any fixed tuple $(i_0, i_1, \cdots, i_n) \in \mathcal{I}_{[\mathrm{XHS20}]}(n, d)$, we construct polynomial $f_{i_0,i_1,\ldots,i_n}(x_0, y_1, \cdots, y_n)$ as follows.

Case a: When $l = 0$ and $0 \le i_0 \le 2d$, define

$$f_{i_0,i_1,\cdots,i_n}(x_0, y_1, \cdots, y_n) := x_0^{i_0}.$$

Case b: When $l = 1$ and $0 \le i_0 \le 1$, define

$$f_{i_0,i_1,\cdots,i_n}(x_0, y_1, \cdots, y_n) := x_0^{i_0} y_1^{i_1} \cdots y_n^{i_n}.$$

Case c: When $1 \le l \le d$ and $2l \le i_0 \le 2d$, define

$$f_{i_0,i_1,\cdots,i_n}(x_0, y_1, \cdots, y_n) := x_0^{i_0-2l} \mathcal{F}_1^{i_1} \cdots \mathcal{F}_n^{i_n}.$$

Case d: When $2 \le l \le d$ and $0 \le i_0 \le 2l - 1$, define

$$f_{i_0,i_1,\cdots,i_n}(x_0, y_1, \cdots, y_n) := \sum_{u=1}^{l} \sum_{v=0}^{1} w_{i_0+1,u+lv} \cdot x_0^v \mathcal{F}_{j_1} \cdots \mathcal{F}_{j_{u-1}} y_{j_u} \mathcal{F}_{j_{u+1}} \cdots \mathcal{F}_{j_l}, \tag{13}$$

where $\mathcal{F}_i(x_0, y_i) = A_i + B_i x_0 + C_i x_0^2 + D_i y_i + E_i x_0 y_i + x_0^2 y_i = 0 \pmod p$ for $1 \le i \le n$ defined in (6), integers $j_1, \cdots, j_l$ are defined in Lemma 3, and $w_{i_0+1,u+lv}$ is element of the (i_0+1)-th row and the $(u+lv)$-th column of the matrix $\mathbf{W}_{j_1,\cdots,j_l}$, which is also defined in Lemma 3.

Lemma 3 ([32]). *Let $i_1, \cdots, i_n$ be integers satisfying $0 \le i_1, \cdots, i_n \le 1$. Denote $l = i_1 + \cdots + i_n$, where $2 \le l \le n$. Let $j_1, \cdots, j_l$ be integers satisfying $1 \le j_1 < \cdots < j_l \le n$ and $y_{j_1} \cdots y_{j_l} = y_1^{i_1} \cdots y_n^{i_n}$. Let a $2l \times 2l$ integer matrix $\mathbf{M}_{j_1,\cdots,j_l}$ be the following coefficient matrix:*

$$\begin{pmatrix} \prod_{u\ne 1} (x_0^2 + E_{j_u} x_0 + D_{j_u}) \\ \vdots \\ \prod_{u\ne l} (x_0^2 + E_{j_u} x_0 + D_{j_u}) \\ x_0 \prod_{u\ne 1} (x_0^2 + E_{j_u} x_0 + D_{j_u}) \\ \ddots \\ x_0 \prod_{u\ne l} (x_0^2 + E_{j_u} x_0 + D_{j_u}) \end{pmatrix} = \mathbf{M}_{j_1,\cdots,j_l} \begin{pmatrix} 1 \\ \vdots \\ x_0^{l-1} \\ x_0^l \\ \vdots \\ x_0^{2l-1} \end{pmatrix} \bmod p^{l-1}, \tag{14}$$

where integers D_{j_u} and E_{j_u} are the coefficients in the polynomial $\mathcal{F}_{j_u} = A_{j_u} + B_{j_u} x_0 + C_{j_u} x_0^2 + D_{j_u} y_{j_u} + E_{j_u} x_0 y_{j_u} + x_0^2 y_{j_u}$ for $1 \le u \le l$. Then the matrix $\mathbf{M}_{j_1,\cdots,j_l}$ is invertible over $\mathbb{Z}_{p^{l-1}}$. Denote $\mathbf{W}_{j_1,\cdots,j_l}$ as its inverse matrix. Hence,

$$\mathbf{W}_{j_1,\cdots,j_l} \cdot \mathbf{M}_{j_1,\cdots,j_l} = I_{2l} \bmod p^{l-1}, \tag{15}$$

where I_{2l} is the $2l \times 2l$ identity matrix.

Next, we give an example to understand **Case d**. Consider $n = d = 2$, then we get that $l = 2$, $0 \le i_0 \le 3$ and $i_1 = i_2 = 1$. Based on $1 \le j_1 < j_2 \le 2$ and $y_{j_1} y_{j_2} = y_1^{i_1} y_2^{i_2}$, we have $j_1 = 1, j_2 = 2$. Hence, the corresponding polynomials are $\mathcal{F}_1 = \mathcal{F}_{j_1}, \mathcal{F}_2 = \mathcal{F}_{j_2}$. We first focus on the following relations:

$$\begin{pmatrix} y_1\mathcal{F}_2 \\ y_2\mathcal{F}_1 \\ x_0y_1\mathcal{F}_2 \\ x_0y_2\mathcal{F}_1 \end{pmatrix} = \begin{pmatrix} A_2y_1 + B_2x_0y_1 + C_2x_0^2y_1 \\ A_1y_2 + B_1x_0y_2 + C_1x_0^2y_2 \\ A_2x_0y_1 + B_2x_0^2y_1 + C_2x_0^3y_1 \\ A_1x_0y_2 + B_1x_0^2y_2 + C_1x_0^3y_2 \end{pmatrix} + \begin{pmatrix} D_2 & E_2 & 1 & 0 \\ D_1 & E_1 & 1 & 0 \\ 0 & D_2 & E_2 & 1 \\ 0 & D_1 & E_1 & 1 \end{pmatrix} \begin{pmatrix} y_1y_2 \\ x_0y_1y_2 \\ x_0^2y_1y_2 \\ x_0^3y_1y_2 \end{pmatrix} \bmod p,$$

where

$$\mathbf{M}_{j_1,j_2} = \mathbf{M}_{12} = \begin{pmatrix} D_2 & E_2 & 1 & 0 \\ D_1 & E_1 & 1 & 0 \\ 0 & D_2 & E_2 & 1 \\ 0 & D_1 & E_1 & 1 \end{pmatrix}.$$

We compute the determinant of $\mathbf{M}_{12}$ and get $\det(\mathbf{M}_{12}) = (E_2 - E_1)(D_2E_1 - D_1E_2) - (D_2 - D_1)^2 \equiv -(x_{Q_2} - x_{Q_1})^4 \mod p$, where x_{Q_1}, x_{Q_2} are respectively the x-coordinates of points Q_1, Q_2 in the curve $\mathcal{E}$. According to Section 2.4, $x_{Q_1} \ne x_{Q_2}$ modulo p. Thus, $\mathbf{M}_{12}$ is invertible over $\mathbb{F}_p$. Note that $\mathbf{W}_{12}$ is the inverse matrix. After left multiplying matrix $\mathbf{W}_{12}$ on both sides of the above formula, we have

$$\mathbf{W}_{12} \begin{pmatrix} y_1\mathcal{F}_2 \\ y_2\mathcal{F}_1 \\ x_0y_1\mathcal{F}_2 \\ x_0y_2\mathcal{F}_1 \end{pmatrix} = \mathbf{W}_{12} \begin{pmatrix} A_2y_1 + B_2x_0y_1 + C_2x_0^2y_1 \\ A_1y_2 + B_1x_0y_2 + C_1x_0^2y_2 \\ A_2x_0y_1 + B_2x_0^2y_1 + C_2x_0^3y_1 \\ A_1x_0y_2 + B_1x_0^2y_2 + C_1x_0^3y_2 \end{pmatrix} + \begin{pmatrix} y_1y_2 \\ x_0y_1y_2 \\ x_0^2y_1y_2 \\ x_0^3y_1y_2 \end{pmatrix} \bmod p.$$

Let $(w_{i_0+1,1}, w_{i_0+1,2}, w_{i_0+1,3}, w_{i_0+1,4})$ be the (i_0+1)-th row vector of $\mathbf{W}_{12}$, where $0 \le i_0 \le 3$. We get that

$$\begin{aligned} & w_{i_0+1,1}y_1\mathcal{F}_2 + w_{i_0+1,2}y_2\mathcal{F}_1 + w_{i_0+1,3}x_0y_1\mathcal{F}_2 + w_{i_0+1,4}x_0y_2\mathcal{F}_1 \\ = {} & x_0^{i_0}y_1y_2 + w_{i_0+1,1}\mathcal{H}_{1,0} + w_{i_0+1,2}\mathcal{H}_{2,0} + w_{i_0+1,3}\mathcal{H}_{1,1} + w_{i_0+1,4}\mathcal{H}_{2,1} \end{aligned} \tag{16}$$

in the sense of modulo p, where

$$\begin{pmatrix} \mathcal{H}_{1,0} \\ \mathcal{H}_{2,0} \\ \mathcal{H}_{1,1} \\ \mathcal{H}_{2,1} \end{pmatrix} = \begin{pmatrix} A_2y_1 + B_2x_0y_1 + C_2x_0^2y_1 \\ A_1y_2 + B_1x_0y_2 + C_1x_0^2y_2 \\ A_2x_0y_1 + B_2x_0^2y_1 + C_2x_0^3y_1 \\ A_1x_0y_2 + B_1x_0^2y_2 + C_1x_0^3y_2 \end{pmatrix}.$$

Note that $y_1, x_0y_1, x_0^2y_1, x_0^3y_1, y_2, x_0y_2, x_0^2y_2, x_0^3y_2, x_0^{i_0}y_1y_2$ are monomials of the polynomial in (16). According to the order (10), we have

$$y_1 \prec x_0y_1 \prec x_0^2y_1 \prec x_0^3y_1 \prec y_2 \prec x_0y_2 \prec x_0^2y_2 \prec x_0^3y_2 \prec x_0^{i_0}y_1y_2.$$

According to (13), we obtain that

$$f_{i_0,1,1}(x_0, y_1, y_2) = w_{i_0+1,1}y_1\mathcal{F}_2 + w_{i_0+1,2}y_2\mathcal{F}_1 + w_{i_0+1,3}x_0y_1\mathcal{F}_2 + w_{i_0+1,4}x_0y_2\mathcal{F}_1.$$

Hence, $x_0^{i_0}y_1y_2$ is the leading term of $f_{i_0,1,1}(x_0, y_1, y_2)$ from (16).

Lemma 4 ([32]). *Based on the order (10), the monomial* $x_0^{i_0}y_1^{i_1}\cdots y_n^{i_n}$ *is the leading term of the polynomial* $f_{i_0,i_1,\cdots,i_n}(x_0,y_1,\cdots,y_n)$ *for* $(i_0,i_1,\cdots,i_n)\in\mathcal{I}_{[\mathrm{XHS20}]}(n,d)$. *Let*

$$F_{i_0,i_1,\cdots,i_n}(x_0,y_1,\cdots,y_n):=\begin{cases}p^{d+1-l}f_{i_0,i_1,\cdots,i_n} & \text{for } 1\le l\le d, 0\le i_0\le 2l-1,\\ p^{d-l}f_{i_0,i_1,\cdots,i_n} & \text{for } 0\le l\le d, 2l\le i_0\le 2d.\end{cases}\tag{17}$$

Let $\mathcal{L}_{[\mathrm{XHS20}]}(n,d)$ *be the lattice which is spanned by the coefficient vectors of polynomials*

$$F_{i_0,i_1,\cdots,i_n}(x_0X,y_1X,\cdots,y_nX) \text{ for all } (i_0,i_1,\cdots,i_n)\in\mathcal{I}_{[\mathrm{XHS20}]}(n,d),$$

where the value X *is the upper bound of* $|e_0|,|\widetilde{e}_1|,\cdots,|\widetilde{e}_n|$. *The diagonal elements in triangular basis matrix of lattice* $\mathcal{L}_{[\mathrm{XHS20}]}(n,d)$ *are as follows:*

$$\begin{cases}p^{d+1-l}X^{i_0+l} & \text{for } 1\le l\le d, 0\le i_0\le 2l-1,\\ p^{d-l}X^{i_0+l} & \text{for } 0\le l\le d, 2l\le i_0\le 2d.\end{cases}$$

According to Lemma 4, the dimension and determinant of $\mathcal{L}_{[\mathrm{XHS20}]}(n,d)$ are respectively

$$\dim(\mathcal{L}_{[\mathrm{XHS20}]}(n,d))=(2d+1)\sum_{l=0}^{d}\tbinom{n}{l} \text{ and } \det(\mathcal{L}_{[\mathrm{XHS20}]}(n,d))=X^{\overline{\alpha}}p^{\overline{\beta}},$$

where

$$\overline{\alpha}=d(2d+1)\sum_{l=0}^{d}\tbinom{n}{l}+(2d+1)\sum_{l=0}^{d}l\tbinom{n}{l},\ \overline{\beta}=d(2d+1)\sum_{l=0}^{d}\tbinom{n}{l}-(2d-1)\sum_{l=0}^{d}l\tbinom{n}{l}.$$

For a sufficiently large modulus p, one can use the simplified Coppersmith condition (2), which does not affect the asymptotic bound. Based on (2), we get the condition $\left(\det(\mathcal{L}_{[\mathrm{XHS20}]}(n,d))\right)^{\frac{1}{\overline{w}}}<p^d$, where $\overline{w}=\dim(\mathcal{L}_{[\mathrm{XHS20}]}(n,d))$, which is equivalent to

$$X<p^{\frac{d\overline{w}-\overline{\beta}}{\overline{\alpha}}}.\tag{18}$$

We omit the tedious calculation and give the following results directly. For any $1\le d\le n$, $\frac{d\overline{w}-\overline{\beta}}{\overline{\alpha}}<\frac{1}{2}$. For any given positive integer d, take $n=d^3$. Then we have $\overline{w}=(2d+1)\binom{n}{d}(1+o(1))$, $\overline{\alpha}=2d(2d+1)\binom{n}{d}(1+o(1))$ and $\overline{\beta}=2d\binom{n}{d}(1+o(1))$. For a sufficiently large $p=2^{\omega(2^n)}$, the condition (18) becomes $X<p^{\frac{1}{2}-\frac{1}{2d}-\overline{\varepsilon}}$, where $\overline{\varepsilon}>0$ and $\overline{\varepsilon}=o(\frac{1}{d})$. Plugging $X=p/2^{\delta}$ for EC-HNP$_x$ into the above inequality, we have $\delta/\log_2 p>\frac{1}{2}+\frac{1}{2d}+\overline{\varepsilon}$. When d tends to infinity, this condition reduces to

$$\frac{\delta}{\log_2 p}>\frac{1}{2}.\tag{19}$$

4 New lattice

In this section, we design a new lattice by mining the algebraic structure. We present an example in Appendix B to help understand lattice $\mathcal{L}(n,d,t)$.

4.1 Lattice $\mathcal{L}(n,d,t)$

Let $\mathcal{I}(n,d,t)$ be an index set which is equal to $\mathcal{I}(n,d,t)=\mathcal{I}_1\cup\mathcal{I}_2$, where

$$\begin{aligned}\mathcal{I}_1 &:= \{(i_0,i_1,\cdots,i_n) \mid 0\le i_0\le 2d-1, 0\le i_1,\cdots,i_n\le 1, 0\le l\le d\},\\ \mathcal{I}_2 &:= \{(i_0,i_1,\cdots,i_n) \mid 0\le i_0\le t, 0\le i_1,\cdots,i_n\le 1, l=d+1\}.\end{aligned}$$

Here, $1\le d<n$, $0\le t\le 2d-1$ and $l=i_1+\cdots+i_n$ satisfying $0\le l\le d+1$.

Remark 1. According to (12), we get that the index set $\mathcal{I}_{[\text{XHS20}]}(n,d)$ equals

$$\{(i_0,i_1,\cdots,i_n) \mid 0\le i_0\le 2d, 0\le i_1,\cdots,i_n\le 1, 0\le l\le d\}.$$

It is obvious that $\mathcal{I}_1$ is a subset of $\mathcal{I}_{[\text{XHS20}]}(n,d)$, whereas $\mathcal{I}_2$ is not.

Based on $F_{i_0,i_1,\cdots,i_n}(x_0,y_1,\cdots,y_n)$ in Lemma 4, we construct the polynomial $G_{i_0,i_1,\cdots,i_n}(x_0,y_1,\cdots,y_n)$ as follows.

Case A: For any given $(i_0,i_1,\cdots,i_n)\in\mathcal{I}_1$, we define

$$G_{i_0,i_1,\cdots,i_n}(x_0,y_1,\cdots,y_n)=F_{i_0,i_1,\cdots,i_n}(x_0,y_1,\cdots,y_n).$$

Since $F_{i_0,i_1,\cdots,i_n}(e_0,\widetilde{e}_1,\cdots,\widetilde{e}_n)=0 \bmod p^d$, we have $G_{i_0,i_1,\cdots,i_n}(e_0,\widetilde{e}_1,\cdots,\widetilde{e}_n)=0 \bmod p^d$.

Case B: For any given $(i_0,i_1,\cdots,i_n)\in\mathcal{I}_2$, we define

$$G_{i_0,i_1,\cdots,i_n}(x_0,y_1,\cdots,y_n)=\left(H_{i_0,i_1,\cdots,i_n}+J_{i_0,i_1,\cdots,i_n}+K_{i_0,i_1,\cdots,i_n}\right) \bmod p^d,$$

which is considered to be the corresponding polynomial over $\mathbb{Z}$. Without loss of generality, we let $j_1,\cdots,j_{d+1}$ be integers satisfying $1\le j_1\le\cdots\le j_{d+1}\le n$ and $y_{j_1}y_{j_2}\cdots y_{j_{d+1}}=y_1^{i_1}y_2^{i_2}\cdots y_n^{i_n}$, and

$$\begin{aligned}H_{i_0,i_1,\cdots,i_n} &= \sum_{u=1}^{d+1}\sum_{v=0}^{1} w_{i_0+1,u+v(d+1)}\cdot x_0^v\mathcal{F}_{j_1}\cdots\mathcal{F}_{j_{u-1}}y_{j_u}\mathcal{F}_{j_{u+1}}\cdots\mathcal{F}_{j_{d+1}},\\ J_{i_0,i_1,\cdots,i_n} &= \sum_{u=1}^{d+1}\sum_{v=0}^{1} w_{i_0+1,u+v(d+1)}\cdot x_0^v\mathcal{F}_{j_1}\cdots\mathcal{F}_{j_{u-1}}C_{j_u}\mathcal{F}_{j_{u+1}}\cdots\mathcal{F}_{j_{d+1}},\\ K_{i_0,i_1,\cdots,i_n} &= \sum_{u=1}^{d+1} w_{i_0+1,u+(d+1)}\cdot\mathcal{F}_{j_1}\cdots\mathcal{F}_{j_{u-1}}(B_{j_u}-C_{j_u}E_{j_u})\mathcal{F}_{j_{u+1}}\cdots\mathcal{F}_{j_{d+1}},\end{aligned}$$

where the integers B_{j_u}, C_{j_u} and E_{j_u} are the coefficients in the polynomial $\mathcal{F}_{j_u}=A_{j_u}+B_{j_u}x_0+C_{j_u}x_0^2+D_{j_u}y_{j_u}+E_{j_u}x_0y_{j_u}+x_0^2y_{j_u}$ for $1\le u\le d+1$, and the integer $w_{i_0+1,m}(1\le m\le 2d+2)$ is the m-th component of the (i_0+1)-th row vector in the inverse matrix $\mathbf{W}_{j_1,\cdots,j_{d+1}}$, which is defined in Lemma 3.

For Case B, the desired vector $(e_0,\widetilde{e}_1,\cdots,\widetilde{e}_n)$ is common root of $H_{i_0,i_1,\cdots,i_n}$, $J_{i_0,i_1,\cdots,i_n}$ and $K_{i_0,i_1,\cdots,i_n}$ modulo p^d. Hence, $G_{i_0,i_1,\cdots,i_n}(e_0,\widetilde{e}_1,\cdots,\widetilde{e}_n)=0 \bmod p^d$.

Lemma 5. *Define* $G_{i_0,i_1,\cdots,i_n}(x_0, y_1, \cdots, y_n)$ *and* $\mathcal{I}(n, d, t)$ *as above. Let* $\mathcal{L}(n, d, t)$ *be a lattice spanned by the coefficient vectors of* $G_{i_0,i_1,\cdots,i_n}(x_0X, y_1X, \cdots, y_nX)$ *for all* $(i_0, i_1, \cdots, i_n) \in \mathcal{I}(n, d, t)$*, where the value* X *is the upper bound of* $|e_0|, |\widetilde{e}_1|, \cdots, |\widetilde{e}_n|$*. Then the basis matrix is triangular if the coefficient vectors of* $G_{i_0,i_1,\cdots,i_n}(x_0X, y_1X, \cdots, y_nX)$ *are arranged based on the order of the corresponding* $x_0^{i_0} y_1^{i_1} \cdots y_n^{i_n}$ *from low to high. The diagonal elements in the triangular basis matrix of* $\mathcal{L}(n, d, t)$ *are as follows:*

$$\begin{cases} p^{d+1-l} X^{i_0+l} & \text{for } 0 \le l \le d, 0 \le i_0 \le 2l-1, \\ p^{d-l} X^{i_0+l} & \text{for } 0 \le l < d, 2l \le i_0 \le 2d-1, \\ X^{i_0+d+1} & \text{for } l = d+1, 0 \le i_0 \le t. \end{cases} \tag{20}$$

The dimension of $\mathcal{L}(n, d, t)$ is equal to the number of $\mathcal{I}(n, d, t)$. Namely,

$$\dim(\mathcal{L}(n, d, t)) = (t+1)\binom{n}{d+1} + 2d \sum_{l=0}^{d} \binom{n}{l}. \tag{21}$$

The determinant of $\mathcal{L}(n, d, t)$ is equal to

$$\det(\mathcal{L}(n, d, t)) =: X^{\alpha} p^{\beta}, \tag{22}$$

where

$$\begin{aligned} \alpha &= \tfrac{(2d+t+2)(t+1)}{2} \tbinom{n}{d+1} + d \sum_{l=0}^{d} (2d-1+2l) \tbinom{n}{l}, \\ \beta &= 2d^2 \sum_{l=0}^{d} \tbinom{n}{l} - (2d-2) \sum_{l=0}^{d} l \tbinom{n}{l}. \end{aligned}$$

4.2 Improved Bound

According to the steps in the Coppersmith method in Section 2.2, the Coppersmith condition (4) must be satisfied for the polynomials $h_i(x_0, y_1, \ldots, y_n)$ for all $1 \le i \le n+1$, corresponding to the first $n+1$ LLL reduced basis vectors, to contain the desired root $(e_0, \tilde{e}_1, \ldots, \tilde{e}_n)$ over integers. That is,

$$2^{\frac{w(w-1)}{4(w-n)}} \det(\mathcal{L}(n, d, t))^{\frac{1}{w-n}} < \frac{p^d}{\sqrt{w}}, \tag{23}$$

where $w = \dim(\mathcal{L}(n, d, t))$. Once we get the above $n+1$ polynomials h_i's, under Assumption 1, we can compute the wanted root $(e_0, \tilde{e}_1, \ldots, \tilde{e}_n)$ using the Gröbner basis.

Plugging (21) and (22) into (23), we obtain

$$X < \left(2^{-\frac{w(w-1)}{4\alpha}} \cdot w^{-\frac{w-n}{2\alpha}}\right) \cdot p^{S(n,d,t)}, \tag{24}$$

where

$$S(n,d,t) := \frac{d(w-n)-\beta}{\alpha} = \frac{d(t+1)\binom{n}{d+1}+(2d-2)\sum\limits_{l=0}^{d} l\binom{n}{l}-dn}{\frac{(2d+t+2)(t+1)}{2}\binom{n}{d+1}+d\sum\limits_{l=0}^{d}(2d-1+2l)\binom{n}{l}}.$$

For a given sufficiently large $p = 2^{\omega(d^{(2+c)d})}$ for any positive integer d and any constant $c > 0$, the condition (24) can be simplified as

$$X < p^{S(n,d,t)}.$$

By taking integers $t = 0$ and $n = d^{3+c}$, the condition becomes

$$X < p^{1-\frac{1}{d+1}-\varepsilon}. \tag{25}$$

Here, $\varepsilon = o(\frac{1}{d+1}) = \frac{d^2(2d-1)\sum\limits_{l=0}^{d}\binom{n}{l}+2\sum\limits_{l=0}^{d} l\binom{n}{l}+d(d+1)n}{(d+1)^2\binom{n}{d+1}+d(d+1)(2d-1)\sum\limits_{l=0}^{d}\binom{n}{l}+2d(d+1)\sum\limits_{l=0}^{d} l\binom{n}{l}} > 0$. The detailed analysis is presented in Appendix C.

The running time of the LLL algorithm depends on the dimension and the maximal bit size of the input triangular basis matrix. For $t = 0$ and $n = d^{3+c}$, the dimension of $\mathcal{L}(n,d,t)$ is equal to $\binom{n}{d+1} + 2d\sum_{l=0}^{d}\binom{n}{l} = \mathcal{O}(n^{d+1}) = \mathcal{O}(d^{(3+c)d})$, and the bit size of the entries in the triangular basis matrix is bounded by $3d\log_2 p$ from (20). Based on [25], the time complexity of the LLL algorithm is

$$\text{poly}\big(3d\log_2 p, \mathcal{O}(d^{(3+c)d})\big) = \mathcal{O}((\log_2 p)^{\mathcal{O}(1)} d^{\mathcal{O}(d)}) \tag{26}$$

which is polynomial in $\log_2 p$ for any constant d.

The running time of the Gröbner basis computation relies on the degrees and number of variables of input polynomials as well as the size of input polynomials. Based on [13], the time complexity of the Gröbner basis computation for a zero-dimensional system is polynomial in $\max\{S, D^N\} < Nh(eD)^N$, where N is the number of variables, and S is the size of the input polynomials in dense representation, h is the maximal size of the coefficients of the input polynomials, D is arithmetic mean value of the degrees of input polynomials and e is Euler constant. For our lattice $\mathcal{L}(n,d,t)$, when $t = 0$ and $n = d^{3+c}$, the number of variables is $n+1$, the degree of input polynomials h_i's ($1 \le i \le n+1$) is $3d-1$ according to (39), and the maximal size h is less than $d\log_2 p$ based on Lemma 2. That is, $N = n+1$, $D = N(3d-1)/N = 3d-1$, and $h < d\log_2 p$. Hence, the time complexity of the Gröbner basis computation is bounded by

$$\text{poly}(Nh(eD)^N) = \mathcal{O}((\log_2 p)^{\mathcal{O}(1)} d^{\mathcal{O}(n)}) \tag{27}$$

which is polynomial in $\log_2 p$ for any constant d. From (26) and (27), the overall complexity is polynomial in $\log_2 p$ for any constant d.

Finally, if any vector $(x_0, \tilde{y}_1, \cdots, \tilde{y}_n) \in \mathbb{Z}^{n+1}$ such that $\mathcal{F}_j(x_0, \tilde{y}_j) = 0 \bmod p$ for all $1 \le j \le n$ in (6), where the upper bound of $|x_0|$, $|\tilde{y}_1|$, $\cdots$, $|\tilde{y}_n|$ satisfies

(25), then $(x_0, \tilde{y}_1, \cdots, \tilde{y}_n)$ is also a common root over $\mathbb{Z}$ of the input polynomials $h_1, \cdots, h_{n+1}$ of Gröbner basis computation. The following result shows that the number of these roots is not only limited, but also only one with an overwhelming probability.

Lemma 6. *For a given sufficiently large prime $p = 2^{\omega(d^{(2+c)d})}$ for any positive integer d and any constant $c > 0$, given $n = d^{3+c}$ polynomials $\mathcal{F}_j(x_0, y_j)$ satisfying $\mathcal{F}_j(e_0, \widetilde{e}_j) = 0 \bmod p$ for $1 \le j \le n$ in (6), the probability that there is an integer vector $(e'_0, \tilde{e}'_1, \cdots, \tilde{e}'_n) \neq (e_0, \tilde{e}_1, \cdots, \tilde{e}_n)$, such that $\mathcal{F}_i(e'_0, \tilde{e}'_i) = 0 \pmod p$ for all $1 \le i \le n$, where the upper bound of $|e'_0|, |\tilde{e}'_1|, \cdots, |\tilde{e}'_n|$ satisfies (25), does not exceed $\mathcal{O}(\frac{1}{p})$.*

We present the detailed proof in Appendix D, which is inspired by the idea of [21,28].

According to the above analysis, we get the following result.

Theorem 1. *For a given sufficiently large prime $p = 2^{\omega(d^{(2+c)d})}$ for any positive integer d and any constant $c > 0$, given $n = d^{3+c}$ polynomials $\mathcal{F}_j(x_0, y_j)$ satisfying $\mathcal{F}_j(e_0, \widetilde{e}_j) = 0 \bmod p$ for $1 \le j \le n$ in (6), under Assumption 1, one can compute the desired root $(e_0, \tilde{e}_1, \cdots, \tilde{e}_n)$, if the bound X of $|e_0|, |\tilde{e}_1|, \cdots, |\tilde{e}_n|$ satisfies*

$$X < p^{1 - \frac{1}{d+1} - \varepsilon},$$

where $\varepsilon = o(\frac{1}{d+1}) > 0$. The overall time complexity is polynomial in $\log_2 p$ for any constant d.

Since $X = p/2^{\delta}$ for the case of EC-HNP$_x$, we get a new bound for EC-HNP$_x$ from Theorem 1.

Theorem 2. *Define d, n, p, ε as in Theorem 1. For $2n + 1$ given calls to the oracle $O_{P,R}(m)$ in EC-HNP$_x$, under Assumption 1, one can recover the hidden point P when the number δ of known MSBs satisfies*

$$\frac{\delta}{\log_2 p} > \frac{1}{d+1} + \varepsilon.$$

For the least significant bits (LSBs) case, the problem of solving the corresponding EC-HNP$_x$ can be converted into finding the desired root $(e_0, \widetilde{e}_1, \cdots, \widetilde{e}_n)$ of the involved polynomials based on [28, Section 6.1]. Note that the forms of these polynomials as well as the size of the desired root are the same as those in (6). Therefore, we obtain the same bound as in the MSBs case.

For the case of ECDH, we get the following result from Theorem 2.

Theorem 3. *Define d, p as in Theorem 1. For a given elliptic curve $\mathcal{E}$ over the prime field $\mathbb{F}_p$, if there is an oracle that outputs about $\frac{1}{d+1}$ of the most (least) significant bits of the x-coordinate of the ECDH key, under Assumption 1, one can compute all the bits in polynomial time.*

5 Proof of triangular basis matrix

First, we present the following relation, which can be utilized to construct triangular basis matrix.

Lemma 7. *Define the matrices* $\mathbf{M}_{j_1,\cdots,j_{d+1}}$ *and* $\mathbf{W}_{j_1,\cdots,j_{d+1}}$ *as in Lemma 3, where* $1 \le j_1 < \cdots < j_{d+1} \le n$. *Let* $w_{i_0+1,m}$ *be the entry of the* (i_0+1)*-th row and the* m*-th column of* $\mathbf{W}_{j_1,\cdots,j_{d+1}}$, *where* $0 \le i_0 \le 2d+1, 1 \le m \le 2d+2$. *Then we have*

$$\begin{cases} \sum\limits_{u=1}^{d+1} w_{i_0+1,d+1+u} = 0 \bmod p^d, & \text{for } 0 \le i_0 \le 2d, \\ \sum\limits_{u=1}^{d+1} \big(w_{i_0+1,u} + w_{i_0+1,d+1+u} \sum\limits_{m \ne u} E_{j_m}\big) = 0 \bmod p^d, & \text{for } 0 \le i_0 \le 2d-1, \end{cases} \tag{28}$$

where E_{j_m} *is the coefficient of the polynomial* $\mathcal{F}_{j_m} = A_{j_m} + B_{j_m}x_0 + C_{j_m}x_0^2 + D_{j_m}y_{j_m} + E_{j_m}x_0y_{j_m} + x_0^2y_{j_m}$ *for* $1 \le m \le d+1$.

Proof. According to (14), we get that the $(2d+2)\times(2d+2)$ matrix $\mathbf{M}_{j_1,\cdots,j_{d+1}}$ is the following coefficient matrix:

$$\begin{pmatrix} \prod\limits_{u\ne 1}(x_0^2 + E_{j_u}x_0 + D_{j_u}) \\ \vdots \\ \prod\limits_{u\ne d+1}(x_0^2 + E_{j_u}x_0 + D_{j_u}) \\ x_0\prod\limits_{u\ne 1}(x_0^2 + E_{j_u}x_0 + D_{j_u}) \\ \ddots \\ x_0\prod\limits_{u\ne d+1}(x_0^2 + E_{j_u}x_0 + D_{j_u}) \end{pmatrix} = \mathbf{M}_{j_1,\cdots,j_{d+1}} \begin{pmatrix} 1 \\ \vdots \\ x_0^d \\ x_0^{d+1} \\ \vdots \\ x_0^{2d+1} \end{pmatrix} \bmod p^d. \tag{29}$$

For the sake of discussion, let $\widetilde{F}_{j_m} = \prod_{u\ne m}(x_0^2 + E_{j_u}x_0 + D_{j_u})$ for all $1 \le m \le d+1$. The last column of $\mathbf{M}_{j_1,\cdots,j_{d+1}}$ corresponds to the vector whose elements are respectively the coefficients of x_0^{2d+1} in the following polynomials

$$\widetilde{F}_{j_1}, \cdots, \widetilde{F}_{j_{d+1}}, x_0\cdot\widetilde{F}_{j_1}, \cdots, x_0\cdot\widetilde{F}_{j_{d+1}}.$$

Note that the coefficient of x_0^{2d+1} in the polynomial $\widetilde{F}_{j_m}$ is 0 for all $1 \le m \le d+1$, and the coefficient of x_0^{2d+1} in the polynomial $x_0\widetilde{F}_{j_m}$ is 1 for all $1 \le m \le d+1$. That is, the last column of $\mathbf{M}_{j_1,\cdots,j_{d+1}}$ is $(0,\cdots,0,1,\cdots,1)^T$, where the number of components 1 is $d+1$. Since $(w_{i_0+1,1},\cdots,w_{i_0+1,2d+2})$ is the (i_0+1)-th row of the inverse matrix $\mathbf{W}_{j_1,\cdots,j_{d+1}}$ modulo p^d, for $0 \le i_0 \le 2d$, we get that

$$(w_{i_0+1,1},\cdots,w_{i_0+1,2d+2})\cdot(0,\cdots,0,1,\cdots,1)^T = 0 \bmod p^d,$$

i.e. $\sum\limits_{u=1}^{d+1} w_{i_0+1,d+1+u} = 0 \bmod p^d$.

The penultimate column of $\mathbf{M}_{j_1,\cdots,j_{d+1}}$ corresponds to the vector whose elements are respectively the coefficients of x_0^{2d} in the following polynomials

$$\widetilde{F}_{j_1},\cdots,\widetilde{F}_{j_{d+1}},x_0\cdot\widetilde{F}_{j_1},\cdots,x_0\cdot\widetilde{F}_{j_{d+1}}.$$

Note that the coefficient of x_0^{2d} in $\widetilde{F}_{j_m}$ is 1 for all $1\le m\le d+1$, and the coefficient of x_0^{2d} in $x_0\widetilde{F}_{j_m}$ is $E_{j_1}+\cdots+E_{j_{m-1}}+E_{j_{m+1}}+\cdots+E_{j_{d+1}}$ for $1\le m\le d+1$. It implies that the penultimate column of $\mathbf{M}_{j_1,\cdots,j_{d+1}}$ is $(1,\cdots,1,\sum_{m\neq 1}E_{j_m},\cdots,\sum_{m\neq d+1}E_{j_m})^T$, where the number of components 1 is $d+1$. Based on $(w_{i_0+1,1},\cdots,w_{i_0+1,2d+2})$ is the (i_0+1)-th row of $\mathbf{W}_{j_1,\cdots,j_{d+1}}$ modulo p^d, for $0\le i_0\le 2d-1$, we obtain that

$$(w_{i_0+1,1},\cdots,w_{i_0+1,2d+2})\cdot(1,\cdots,1,\sum_{m\neq 1}E_{j_m},\cdots,\sum_{m\neq d+1}E_{j_m})^T=0 \bmod p^d.$$

That is, $\sum_{u=1}^{d+1}\left(w_{i_0+1,u}+w_{i_0+1,d+1+u}\sum_{m\neq u}E_{j_m}\right)=0 \bmod p^d$. $\square$

The above lemma is now used to show the form of $G_{i_0,i_1,\cdots,i_n}(x_0,y_1,\cdots,y_n)$ for $(i_0,i_1,\cdots,i_n)\in\mathcal{I}_2$.

Lemma 8. *Define $G_{i_0,i_1,\cdots,i_n}(x_0,y_1,\cdots,y_n)$ and $\mathcal{I}_1,\mathcal{I}_2$ as in Section 4. If the tuple $(i_0,i_1,\cdots,i_n)\in\mathcal{I}_2$, then we have*

$$G_{i_0,i_1,\cdots,i_n}=x_0^{i_0}y_1^{i_1}\cdots y_n^{i_n}+\sum_{(i_0',i_1',\cdots,i_n')\in\mathcal{I}_1}a_{i_0',i_1',\cdots,i_n'}x_0^{i_0'}y_1^{i_1'}\cdots y_n^{i_n'},$$

where $a_{i_0',i_1',\cdots,i_n'}\in\mathbb{Z}$.

Proof. First, we present that the leading term of $G_{i_0,i_1,\cdots,i_n}(x_0,y_1,\cdots,y_n)$ is $x_0^{i_0}y_1^{i_1}\cdots y_n^{i_n}$ for $(i_0,i_1,\cdots,i_n)\in\mathcal{I}_2$. In this case,

$$G_{i_0,i_1,\cdots,i_n}=H_{i_0,i_1,\cdots,i_n}+J_{i_0,i_1,\cdots,i_n}+K_{i_0,i_1,\cdots,i_n}$$

in the sense of modulo p^d. Here,

$$\begin{aligned}
H_{i_0,i_1,\cdots,i_n}&=\sum_{u=1}^{d+1}\sum_{v=0}^{1}w_{i_0+1,u+v(d+1)}\cdot x_0^v\mathcal{F}_{j_1}\cdots\mathcal{F}_{j_{u-1}}y_{j_u}\mathcal{F}_{j_{u+1}}\cdots\mathcal{F}_{j_{d+1}},\\
J_{i_0,i_1,\cdots,i_n}&=\sum_{u=1}^{d+1}\sum_{v=0}^{1}w_{i_0+1,u+v(d+1)}\cdot x_0^v\mathcal{F}_{j_1}\cdots\mathcal{F}_{j_{u-1}}C_{j_u}\mathcal{F}_{j_{u+1}}\cdots\mathcal{F}_{j_{d+1}},\\
K_{i_0,i_1,\cdots,i_n}&=\sum_{u=1}^{d+1}w_{i_0+1,u}\cdot\mathcal{F}_{j_1}\cdots\mathcal{F}_{j_{u-1}}(B_{j_u}-C_{j_u}E_{j_u})\mathcal{F}_{j_{u+1}}\cdots\mathcal{F}_{j_{d+1}},
\end{aligned}$$

where integers $1\le j_1<\cdots<j_{d+1}\le n$ satisfy $y_{j_1}\cdots y_{j_{d+1}}=y_1^{i_1}\cdots y_n^{i_n}$.

In order to show the case of $H_{i_0,i_1,\cdots,i_n}(x_0, y_1, \cdots, y_n)$, we first consider the following equations:

$$\begin{pmatrix} y_{j_1}\cdot \mathcal{F}_{j_2}\cdots\mathcal{F}_{j_{d+1}} \\ \ddots \\ \mathcal{F}_{j_1}\cdots\mathcal{F}_{j_d}y_{j_{d+1}} \\ x_0\cdot y_{j_1}\mathcal{F}_{j_2}\cdots\mathcal{F}_{j_{d+1}} \\ \ddots \\ x_0\cdot\mathcal{F}_{j_1}\cdots\mathcal{F}_{j_d}y_{j_{d+1}} \end{pmatrix} = \begin{pmatrix} \mathcal{H}_{1,0} \\ \vdots \\ \mathcal{H}_{d+1,0} \\ \mathcal{H}_{1,1} \\ \vdots \\ \mathcal{H}_{d+1,1} \end{pmatrix} + \mathbf{M}_{j_1,\cdots,j_{d+1}} \begin{pmatrix} y_{j_1}\cdot y_{j_2}\cdots y_{j_{d+1}} \\ x_0\cdot y_{j_1}y_{j_2}\cdots y_{j_{d+1}} \\ \vdots \\ x_0^{2d+1}\cdot y_{j_1}y_{j_2}\cdots y_{j_{d+1}} \end{pmatrix} \bmod p^d. \tag{30}$$

Here, the matrix $\mathbf{M}_{j_1,\cdots,j_{d+1}}$ is defined in (29), and the polynomial $\mathcal{H}_{u,v}$ ($1 \le u \le d+1$, $0 \le v \le 1$) is composed of the terms in $x_0^v\mathcal{F}_{j_1}\cdots\mathcal{F}_{j_{u-1}}y_{j_u}\mathcal{F}_{j_{u+1}}\cdots\mathcal{F}_{j_{d+1}}$ except the terms of monomials

$$y_{j_1}\cdots y_{j_{d+1}}, x_0y_{j_1}\cdots y_{j_{d+1}}, \cdots, x_0^{2d+1}y_{j_1}\cdots y_{j_{d+1}}.$$

It implies that the leading monomial in $\mathcal{H}_{u,v}$ is $x_0^{i_0'}y_{k_1}\cdots y_{k_m}$, where $0 \le i_0' \le 2d+1$ and $\{k_1, \cdots, k_m\} \subsetneqq \{j_1, \cdots, j_{d+1}\}$. Hence, $m < d+1$. According to the order (10), we get

$$x_0^{i_0'}y_{k_1}\cdots y_{k_m} \prec y_{j_1}\cdots y_{j_{d+1}} \prec x_0y_{j_1}\cdots y_{j_{d+1}} \prec \cdots \prec x_0^{2d+1}y_{j_1}\cdots y_{j_{d+1}}. \tag{31}$$

Note that $\mathbf{W}_{j_1,\cdots,j_{d+1}}$ is the inverse matrix of $\mathbf{M}_{j_1,\cdots,j_{d+1}}$ modulo p^d. Multiplying the two sides of Equation (30) by $\mathbf{W}_{j_1,\cdots,j_{d+1}}$ to the left, we get

$$\mathbf{W}_{j_1,\cdots,j_{d+1}}\begin{pmatrix} y_{j_1}\cdot \mathcal{F}_{j_2}\cdots\mathcal{F}_{j_{d+1}} \\ \ddots \\ \mathcal{F}_{j_1}\cdots\mathcal{F}_{j_d}y_{j_{d+1}} \\ x_0\cdot y_{j_1}\mathcal{F}_{j_2}\cdots\mathcal{F}_{j_{d+1}} \\ \ddots \\ x_0\cdot\mathcal{F}_{j_1}\cdots\mathcal{F}_{j_d}y_{j_{d+1}} \end{pmatrix} = \mathbf{W}_{j_1,\cdots,j_{d+1}}\begin{pmatrix} \mathcal{H}_{1,0} \\ \vdots \\ \mathcal{H}_{d+1,0} \\ \mathcal{H}_{1,1} \\ \vdots \\ \mathcal{H}_{d+1,1} \end{pmatrix} + \begin{pmatrix} y_{j_1}\cdot y_{j_2}\cdots y_{j_{d+1}} \\ x_0\cdot y_{j_1}y_{j_2}\cdots y_{j_{d+1}} \\ \vdots \\ x_0^{2d+1}\cdot y_{j_1}y_{j_2}\cdots y_{j_{d+1}} \end{pmatrix} \tag{32}$$

(in the sense of modulo p^d). Since $(w_{i_0+1,1}, \cdots, w_{i_0+1,2d+2}$ is the (i_0+1)-th row of $\mathbf{W}_{j_1,\cdots,j_{d+1}}$, where $0 \le i_0 \le t$, from (32), we have

$$\begin{aligned} H_{i_0,i_1,\cdots,i_n} &= \sum_{u=1}^{d+1}\sum_{v=0}^{1} w_{i_0+1,u+(d+1)v}\cdot x_0^v\mathcal{F}_{j_1}\cdots\mathcal{F}_{j_{u-1}}y_{j_u}\mathcal{F}_{j_{u+1}}\cdots\mathcal{F}_{j_{d+1}} \\ &= x_0^{i_0}y_{j_1}y_{j_2}\cdots y_{j_{d+1}} + \sum_{u=1}^{d+1}\sum_{v=0}^{1} w_{i_0+1,u+(d+1)v}\mathcal{H}_{u,v} \bmod p^d. \end{aligned} \tag{33}$$

Based on $x_0^{i_0}y_{j_1}y_{j_2}\cdots y_{j_{d+1}} = x_0^{i_0}y_1^{i_1}\cdots y_n^{i_n}$ and (31), we obtain that $x_0^{i_0}y_1^{i_1}\cdots y_n^{i_n}$ is the leading term of $H_{i_0,i_1,\cdots,i_n}$. Moreover, all monomials except $x_0^{i_0}y_1^{i_1}\cdots y_n^{i_n}$ in $H_{i_0,i_1,\cdots,i_n}$ belong to the set

$$\{x_0^{i_0'}y_1^{i_1'}\cdots y_n^{i_n'} \mid 0 \le i_0' \le 2d+1, 0 \le i_1', \cdots, i_n' \le 1, 0 \le i_1' + \cdots + i_n' \le d\}. \tag{34}$$

For the case of $J_{i_0,i_1,\cdots,i_n}$, let $x_0^{r_0}y_{s_1}\cdots y_{s_m}$ be the leading monomial of $J_{i_0,i_1,\cdots,i_n}$, where $0 \le r_0 \le 2d+1$ and $\{s_1, \cdots, s_m\} \subsetneqq \{j_1, \cdots, j_{d+1}\}$. Thus,

$m < d+1$. Based on the order (10), we get $x_0^{r_0} y_{s_1} \cdots y_{s_m} \prec x_0^{i_0} y_{j_1} \cdots y_{j_{d+1}}$. That is, $x_0^{r_0} y_{s_1} \cdots y_{s_m} \prec x_0^{i_0} y_1^{i_1} \cdots y_n^{i_n}$.

Similarly, we can also prove that the order of the leading monomial of $K_{i_0,i_1,\cdots,i_n}$ is less than the order of $x_0^{i_0} y_1^{i_1} \cdots y_n^{i_n}$.

To sum up, we get that $x_0^{i_0} y_1^{i_1} \cdots y_n^{i_n}$ is the leading term of $G_{i_0,i_1,\cdots,i_n}(x_0, y_1, \cdots, y_n)$. In addition, all monomials except the leading monomial $x_0^{i_0} y_1^{i_1} \cdots y_n^{i_n}$ in $G_{i_0,i_1,\cdots,i_n}$ lie in the set (34).

Then, we prove that $G_{i_0,i_1,\cdots,i_n}(x_0, y_1, \cdots, y_n)$ does not contain any term related to x_0^{2d+1} and x_0^{2d}. It means that all monomials except $x_0^{i_0} y_1^{i_1} \cdots y_n^{i_n}$ in $G_{i_0,i_1,\cdots,i_n}$ lie in $\{x_0^{i'_0} y_1^{i'_1} \cdots y_n^{i'_n} \mid (i'_1, \cdots, i'_n) \in \mathcal{I}_1\}$. That is, we can rewrite $G_{i_0,i_1,\cdots,i_n}$ as

$$x_0^{i_0} y_1^{i_1} \cdots y_n^{i_n} + \sum_{(i'_0,i'_1,\cdots,i'_n) \in \mathcal{I}_1} a_{i'_0,i'_1,\cdots,i'_n} x_0^{i'_0} y_1^{i'_1} \cdots y_n^{i'_n},$$

where $a_{i'_0,i'_1,\cdots,i'_n} \in \mathbb{Z}$, and $\mathcal{I}_1 = \{(i'_0, i'_1, \cdots, i'_n) \mid 0 \le i'_0 \le 2d-1, 0 \le i'_1, \cdots, i'_n \le 1, 0 \le i'_1 + \cdots + i'_n \le d\}$.

For the convenience of subsequent analysis, we rewrite $\mathcal{F}_{j_u} = A_{j_u} + B_{j_u} x_0 + C_{j_u} x_0^2 + D_{j_u} y_{j_u} + E_{j_u} x_0 y_{j_u} + x_0^2 y_{j_u}$ as

$$x_0^2(y_{j_u} + C_{j_u}) + x_0(E_{j_u} y_{j_u} + B_{j_u}) + (D_{j_u} y_{j_u} + A_{j_u}), 1 \le u \le d+1.$$

We rewrite $G_{i_0,i_1,\cdots,i_n}$ for $(i_0, i_1, \cdots, i_n) \in \mathcal{I}_2$ as

$$G_{i_0,i_1,\cdots,i_n} = \mathcal{T}_1 + \mathcal{T}_2 + \mathcal{T}_3$$

in the sense of modulo p^d, where

$$\begin{aligned}
\mathcal{T}_1 &:= \sum_{u=1}^{d+1} w_{i_0+1,u+(d+1)} \cdot x_0 \mathcal{F}_{j_1} \cdots \mathcal{F}_{j_{u-1}} (y_{j_u} + C_{j_u}) \mathcal{F}_{j_{u+1}} \cdots \mathcal{F}_{j_{d+1}} \\
\mathcal{T}_2 &:= \sum_{u=1}^{d+1} w_{i_0+1,u} \cdot \mathcal{F}_{j_1} \cdots \mathcal{F}_{j_{u-1}} (y_{j_u} + C_{j_u}) \mathcal{F}_{j_{u+1}} \cdots \mathcal{F}_{j_{d+1}} \\
\mathcal{T}_3 &:= \sum_{u=1}^{d+1} w_{i_0+1,u+(d+1)} \cdot \mathcal{F}_{j_1} \cdots \mathcal{F}_{j_{u-1}} (B_{j_u} - C_{j_u} E_{j_u}) \mathcal{F}_{j_{u+1}} \cdots \mathcal{F}_{j_{d+1}}.
\end{aligned}$$

Since $\deg(x_0) = 2$ in $\mathcal{F}_{j_u}$ for $1 \le u \le d+1$, we have that $\deg(x_0) \le 2d+1$ for $\mathcal{T}_1$, and $\deg(x_0) \le 2d$ for $\mathcal{T}_2$ and $\mathcal{T}_3$.

We can deduce that the x_0^{2d+1}-related term in $G_{i_0,i_1,\cdots,i_n}$ only appears in $\mathcal{T}_1$. Specifically, the x_0^{2d+1}-related term is

$$\sum_{u=1}^{d+1} w_{i_0+1,u+(d+1)} \cdot x_0^{2d+1} (y_{j_1} + C_{j_1}) \cdots (y_{j_{d+1}} + C_{j_{d+1}})$$

in sense of modulo p^d. According to (28), we have $\sum_{u=1}^{d+1} w_{i_0+1,u+d+1} = 0 \bmod p^d$, where $0 \le i_0 \le 2d-1$. Therefore, $G_{i_0,i_1,\cdots,i_n}$ does not have any term related to x_0^{2d+1}.

We can deduce that the x_0^{2d}-related term in $G_{i_0,i_1,\cdots,i_n}$ appears in $\mathcal{T}_1$, $\mathcal{T}_2$ and $\mathcal{T}_3$.

For the case of $\mathcal{T}_1 = \sum_{u=1}^{d+1} w_{i_0+1,u+(d+1)} \cdot x_0 \mathcal{F}_{j_1} \cdots \mathcal{F}_{j_{u-1}} (y_{j_u} + C_{j_u}) \mathcal{F}_{j_{u+1}} \cdots \mathcal{F}_{j_{d+1}}$, based on $\mathcal{F}_{j_u} = x_0^2(y_{j_u} + C_{j_u}) + x_0(E_{j_u}(y_{j_u} + C_{j_u}) + (B_{j_u} - C_{j_u}E_{j_u})) + (A_{j_u} + D_{j_u} y_{j_u})$ for $1 \le u \le d+1$, the x_0^{2d}-related term of $\mathcal{T}_1$ is

$$\begin{aligned} &\sum_{u=1}^{d+1} w_{i_0+1,u+(d+1)} (\sum_{m \neq u} E_{j_m}) \cdot x_0^{2d} (y_{j_1} + C_{j_1}) \cdots (y_{j_{d+1}} + C_{j_{d+1}}) \\ &+ \sum_{u=1}^{d+1} (\sum_{m \neq u} w_{i_0+1,m+(d+1)}) \cdot x_0^{2d} (B_{j_u} - C_{j_u} E_{j_u}) \prod_{m \neq u} (y_{j_m} + C_{j_m}). \end{aligned} \tag{35}$$

For the case of $\mathcal{T}_2 = \sum_{u=1}^{d+1} w_{i_0+1,u} \cdot \mathcal{F}_{j_1} \cdots \mathcal{F}_{j_{u-1}} (y_{j_u} + C_{j_u}) \mathcal{F}_{j_{u+1}} \cdots \mathcal{F}_{j_{d+1}}$, the x_0^{2d}-related term of $\mathcal{T}_2$ is

$$\sum_{u=1}^{d+1} w_{i_0+1,u} \cdot x_0^{2d} (y_{j_1} + C_{j_1}) \cdots (y_{j_{d+1}} + C_{j_{d+1}}). \tag{36}$$

For the case of $\mathcal{T}_3 = \sum_{u=1}^{d+1} w_{i_0+1,u+(d+1)} \cdot \mathcal{F}_{j_1} \cdots \mathcal{F}_{j_{u-1}} (B_{j_u} - C_{j_u} E_{j_u}) \mathcal{F}_{j_{u+1}} \cdots \mathcal{F}_{j_{d+1}}$, the x_0^{2d}-related term of $\mathcal{T}_3$ is

$$\sum_{u=1}^{d+1} w_{i_0+1,u+(d+1)} \cdot x_0^{2d} (B_{j_u} - C_{j_u} E_{j_u}) \prod_{m \neq u} (y_{j_m} + C_{j_m}). \tag{37}$$

According to (35), (36) and (37), we get that the x_0^{2d}-related term in $G_{i_0,i_1,\cdots,i_n}$ is equal to

$$\begin{aligned} &\sum_{u=1}^{d+1} (\sum_{u=1}^{d+1} w_{i_0+1,d+1+u}) \cdot x_0^{2d} (B_{j_u} - C_{j_u} E_{j_u}) \prod_{m \neq u} (y_{j_m} + C_{j_m}) \\ &+ \sum_{u=1}^{d+1} (w_{i_0+1,u} + w_{i_0+1,d+1+u} \sum_{m \neq u} E_{j_m}) \cdot x_0^{2d} (y_{j_1} + C_{j_1}) \cdots (y_{j_{d+1}} + C_{j_{d+1}}) \end{aligned} \tag{38}$$

in sense of modulo p^d. According to (28), we have that $\sum_{u=1}^{d+1} w_{i_0+1,d+1+u} = 0 \pmod{p^d}$ and $\sum_{u=1}^{d+1} \left(w_{i_0+1,u} + w_{i_0+1,d+1+u} \sum_{m \neq u} E_{j_m} \right) = 0 \pmod{p^d}$ for $0 \le i_0 \le 2d-1$. Hence, $G_{i_0,i_1,\cdots,i_n}$ does not have any term related to x_0^{2d}, where $(i_0, i_1, \cdots, i_n) \in \mathcal{I}_2$. $\square$

Finally, we show that the involved basis matrix of $\mathcal{L}(n,d,t)$ is triangular. That is, we provide proof for Lemma 5.

Proof. First, we present that the leading term of $G_{i_0,i_1,\cdots,i_n}(x_0, y_1, \cdots, y_n)$ is $x_0^{i_0} y_1^{i_1} \cdots y_n^{i_n}$ for $(i_0, i_1, \cdots, i_n) \in \mathcal{I}(n,d,t)$. We respectively consider **Case A** and **Case B**.

For **Case A**, the corresponding $(i_0, i_1, \cdots, i_n) \in \mathcal{I}_1$. We define

$$G_{i_0,i_1,\cdots,i_n}(x_0, y_1, \cdots, y_n) = F_{i_0,i_1,\cdots,i_n}(x_0, y_1, \cdots, y_n)$$

From Lemma 4, and $\mathcal{I}_1 \subset \mathcal{I}_{[\text{XHS20}]}(n,d)$, we obtain that the leading term of $G_{i_0,i_1,\cdots,i_n}(x_0, y_1, \cdots, y_n)$ is as follows:

$$\begin{cases} p^{d+1-l} x_0^{i_0} y_1^{i_1} \cdots y_n^{i_n} & \text{for } 1 \le l \le d \text{ and } 0 \le i_0 \le 2l-1, \\ p^{d-l} x_0^{i_0} y_1^{i_1} \cdots y_n^{i_n} & \text{for } 0 \le l < d \text{ and } 2l \le i_0 \le 2d-1. \end{cases}$$

For **Case B**, the corresponding $(i_0, i_1, \cdots, i_n) \in \mathcal{I}_2$. From Lemma 8, we get that the leading term of $G_{i_0,i_1,\cdots,i_n}(x_0, y_1, \cdots, y_n)$ is $x_0^{i_0} y_1^{i_1} \cdots y_n^{i_n}$, where $l = i_1 + \cdots + i_n = d+1$ and $0 \le i_0 \le t$.

To sum up, the leading term of $G_{i_0,i_1,\cdots,i_n}(x_0, y_1, \cdots, y_n)$ is equal to

$$\begin{cases} p^{d+1-l} x_0^{i_0} y_1^{i_1} \cdots y_n^{i_n} & \text{for } 1 \le l \le d \text{ and } 0 \le i_0 \le 2l-1, \\ p^{d-l} x_0^{i_0} y_1^{i_1} \cdots y_n^{i_n} & \text{for } 0 \le l < d \text{ and } 2l \le i_0 \le 2d-1, \\ x_0^{i_0} y_1^{i_1} \cdots y_n^{i_n} & \text{for } l = d+1 \text{ and } 0 \le i_0 \le t. \end{cases} \tag{39}$$

Next, we prove that the basis matrix of $\mathcal{L}(n, d, t)$ can be arranged into a triangular matrix. Since the basis matrix of $\mathcal{L}(n, d, t)$ is made up of the coefficient vectors of polynomials $G_{i_0,i_1,\cdots,i_n}(x_0X,\ y_1X, \cdots, y_nX)$ for all $(i_0, i_1, \cdots, i_n) \in \mathcal{I}(n, d, t)$, and there is a one-to-one correspondence between the polynomial $G_{i_0,i_1,\cdots,i_n}(x_0, y_1, \cdots, y_n)$ and the corresponding polynomial $G_{i_0,i_1,\cdots,i_n}(x_0X, y_1X,\ \cdots, y_nX)$, our goal translates to show that $G_{i_0,i_1,\cdots,i_n}(x_0, y_1, \cdots, y_n)$ for all $(i_0, i_1,\ \cdots, i_n) \in \mathcal{I}(n, d, t)$ form a triangular matrix.

For the level $l = 0$, the corresponding polynomial $G_{i_0,i_1,\cdots,i_n}(x_0, y_1, \cdots, y_n)$ is equal to $p^d x_0^{i_0}$ for $i_0 = 0, 1, \cdots, 2d-1$. From the order (10), we have $p^d \prec p^d x_0 \prec \cdots \prec p^d x_0^{2d-1}$. It implies that all $G_{i_0,i_1,\cdots,i_n}(x_0, y_1, \cdots, y_n)$ for $l = 0$ generate a triangular matrix. The remaining proof is inductive. For any fixed tuple $(i_0, i_1, \cdots,\ i_n) \in \mathcal{I}(n, d, t)$, suppose that all polynomials $G_{i'_0,i'_1,\cdots,i'_n}(x_0, y_1, \cdots, y_n)$, satisfying $x_0^{i'_0} y_1^{i'_1} \cdots y_n^{i'_n} \prec x_0^{i_0} y_1^{i_1} \cdots y_n^{i_n}$, have produced a triangular matrix as stated in Lemma 5. Then we prove that all polynomials added after the polynomial $G_{i_0,i_1,\cdots,i_n}(x_0, y_1, \cdots, y_n)$ still form a triangular matrix. Based on the above analysis, $x_0^{i_0} y_1^{i_1} \cdots y_n^{i_n}$ is the leading monomial of the polynomial $G_{i_0,i_1,\cdots,i_n}(x_0,\ y_1, \cdots, y_n)$. Let $x_0^{k_0} y_1^{k_1} \cdots y_n^{k_n}$ be any given monomial of $G_{i_0,i_1,\cdots,i_n}(x_0, y_1, \cdots, y_n)$ other than the leading monomial $x_0^{i_0} y_1^{i_1} \cdots y_n^{i_n}$. Obviously, we have $x_0^{k_0} y_1^{k_1} \cdots y_n^{k_n} \prec x_0^{i_0} y_1^{i_1} \cdots y_n^{i_n}$. Since $x_0^{k_0} y_1^{k_1} \cdots y_n^{k_n}$ is the leading monomial of polynomial $G_{k_0,k_1,\cdots,k_n}(x_0, y_1,\ \cdots, y_n)$, we get that all monomials except $x_0^{i_0} y_1^{i_1} \cdots y_n^{i_n}$ already appeared in the diagonals of a triangular matrix. Thus, all polynomials after $G_{i_0,i_1,\cdots,i_n}(x_0, y_1, \cdots, y_n)$ is added still produce a triangular matrix. To summarize, the basis matrix of $\mathcal{L}(n, d, t)$ is triangular according to the order of $x_0^{i_0} y_1^{i_1} \cdots y_n^{i_n}$ for all $(i_0, i_1,\ \cdots, i_n) \in \mathcal{I}(n, d, t)$ from low to high.

The diagonal elements in the triangular basis matrix of $\mathcal{L}(n, d, t)$ are all from the leading coefficients of $G_{i_0,i_1,\cdots,i_n}(x_0X, y_1X, \cdots, y_nX)$ for $(i_0, i_1, \cdots, i_n) \in \mathcal{I}(n, d, t)$. Based on (39), the diagonal elements of triangular basis matrix are as follows:

$$\begin{cases} p^{d+1-l} X^{i_0+l} & \text{for } 1 \le l \le d \text{ and } 0 \le i_0 \le 2l-1, \\ p^{d-l} X^{i_0+l} & \text{for } 0 \le l < d \text{ and } 2l \le i_0 \le 2d-1, \\ X^{i_0+d+1} & \text{for } l = d+1 \text{ and } 0 \le i_0 \le t. \end{cases}$$

$\square$

6 Comparison with the existing work

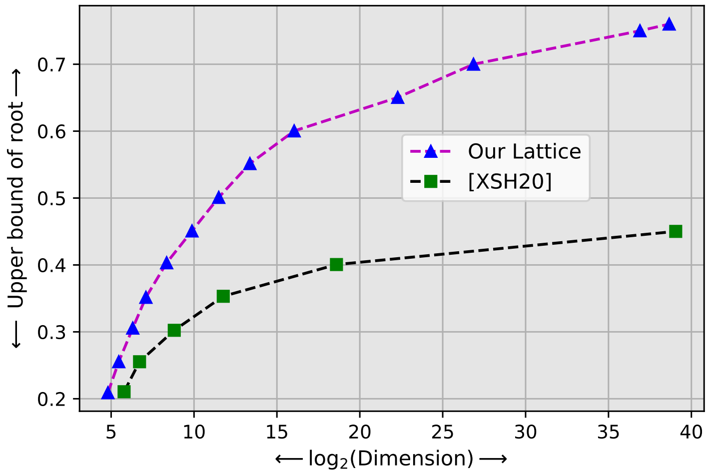

Fig. 1. Comparison of the theoretical upper bound of the root for different dimensions.

Figure 1 compares the theoretical upper bound X for the lattice in Section 4.1 and that in [32]. We can see that our lattice is significantly better than that in [32]. In Figure 1, we take the smallest lattice dimension among different n, d, t for the fixed upper bound. For example, to cross the bound 0.45, the minimum lattice is 940 ($n = 13, d = 2, t = 1$) whereas the minimum dimension in [32] is $2^{39.06}$ ($n = 40, d = 13$).

In Table 1, we present a theoretical comparison of the smallest lattice dimension on the fixed percentage $\delta/\log_2 p$ for a sufficiently large $p = 2^{\omega(d^{(2+c)d})}$. The symbol " $-$ " means that even with a huge lattice dimension, the corresponding $\delta/\log_2 p \leq 0.50$ can not be obtained.

From the second row of Table 1, we can see that in order to reach the 0.60 bound of $\delta/\log_2 p$, the smallest dimension of [32] is 394995 ($n = 16, d = 7$), and the smallest dimensions of our lattice is 326 ($n = 24, d = 1, t = 0$). Therefore, our lattice is practical, while the lattice in [32] is not practical.

Based on the fourth row of Table 1, the smallest lattice dimension is 2879 ($n = 23, d = 2, t = 0$) to obtain the 0.50 bound of $\delta/\log_2 p$. The LLL algorithm terminates within $\mathcal{O}(w^{4+\gamma}b^{1+\gamma})$ bit operations for any $\gamma > 0$ [25], where w is the lattice dimension, and b is the maximal bit-size in the input basis matrix. For $w = 2879$, $w^4 \approx 2^{46}$. The bit-size b for our lattice is bounded by $3d\log_2 p$ (see (20) in Lemma 5). Hence, for a sufficiently large p, it takes a considerable amount of time for the LLL algorithm to output the desired short vector.

Table 1. Comparison of the smallest dimensions for known bit percentages.

$\delta/\log_2 p$	Our		[32]	
	Lattice in Section 4.1		Lattice	
	(n, d, t)	Dimension	(n, d)	Dimension
0.65	(15,1,0)	137	(10,4)	3474
0.60	(24,1,0)	326	(16,7)	394995
0.55	(13,2,1)	940	(40,13)	$2^{39.06}$
0.50	(23,2,0)	2879	-	-
0.45	(37,2,0)	10586	-	-
0.40	(71,2,0)	67383	-	-

7 Experiments

We have implemented our experiments in SAGE 9.3 using Linux Ubuntu with Intel® Core™ i7-7920HQ CPU 3.67 GHz. We have used the L^2 algorithm [26] for lattice reduction. We tested the algorithm up to lattice dimension 298. In our experiments, the zero-dimensional ideal assumption, i.e. Assumption 1 is always valid. Our experimental results are shown in Table 2. We run 100 experiments for each parameter.

We always get more than $\frac{w}{2}$ polynomials that satisfy the desired root over $\mathbb{Z}$ after lattice reduction, where w is the dimension of the lattice. Intermediate coefficient swell is a well-known difficulty for computing Gröbner bases over integers. To overcome this problem, we compute Gröbner basis over small prime fields GF(q) such that the product of these primes is larger than the size of unknown values. Then we use the Chinese Remainder Theorem to find the desired root. Using this method, we can find the root after lattice reduction in a few seconds for all parameters. If X is the upper bound of root, we need to consider primes up to N such that $\prod_{\text{prime } q \le N} q > X$. Since $\prod_{\text{prime } q \le N} q = e^{\theta(N)}$, we need $e^{\theta(N)} > X$, where $\theta(N) = \sum_{\text{prime } q \le N} \log q$ is the first Chebyshev function. Since $\theta(N)$ asymptotically approaches to N for large values of N, considering first $\log_e X$ many prime fields will be sufficient for large N for our attack.

After Gröbner basis computation, we get polynomials of the form $x_0 - e_0, y_1 - \tilde{e}_1, y_2 - \tilde{e}_2, \ldots, y_n - \tilde{e}_n$ in GF(q). Let $T = \prod_{q \le N} q$. Hence using Chinese Remainder Theorem we get $\hat{e_i} \equiv e_i \bmod T$ for $i \in [0, n]$. Thus $e_i = \hat{e_i}$ or $e_i = \hat{e_i} - T$. Hence we can easily collect secrets. We always collect the root for our theoretical values. In fact, experimentally we are able to cross these bounds. In these situations also, success rate is close to 100 percent in all cases.

One can see from Table 2 that it is possible to find the hidden point P by querying the oracle $2n + 1 = 2 \cdot 21 + 1 = 43$ times for the case of NIST-521 and $(n, d, t) = (21, 1, 0)$. Theoretically, knowing 318 MSBs/LSBs of the x-coordinate of $P + [m]R$ in each query should be sufficient for our attack, where the x-coordinate has 521 bits in total. In practice, we are getting better results. Experimentally, knowledge of 301 bits is sufficient to find the hidden point.

Table 2. Experimental results of Section 4.1 on NIST curves. From Equation (24), the required bounds is $X < p^{S(n,d,t)}$ for the lattice $\mathcal{L}(n,d,t)$. Thus the number of known bits should be lower bounded by $(1 - S(n,d,t)) \log_2 p$. The column of Theo. represents this value. The column of Exp. gives corresponding experimental values.

Curve	n d t	Dim.	Theo.	Exp.	Given	Known MSBs			Known LSBs		
						Suc.	LLL (sec.)	GB (sec.)	Suc.	LLL (sec.)	GB (sec.)
NIST-192			143	132	69%	94%	0.51	0.04	96%	0.53	0.04
NIST-224			167	154	69%	99%	0.51	0.04	95%	0.64	0.05
NIST-256	6 1 1	44	191	176	69%	100%	0.57	0.05	100%	0.65	0.05
NIST-384			286	263	68%	100%	0.74	0.07	100%	0.99	0.07
NIST-521			388	357	69%	100%	1.06	0.08	100%	1.23	0.11
NIST-192			137	125	65%	100%	2.26	0.11	100%	2.41	0.11
NIST-224			160	145	65%	100%	2.44	0.15	100%	2.92	0.12
NIST-256	10 1 0	67	182	165	64%	100%	2.79	0.13	100%	3.13	0.13
NIST-384			272	245	64%	100%	4.06	0.17	100%	4.97	0.19
NIST-521			371	330	63%	100%	6.49	0.23	100%	6.60	0.23
NIST-192			135	129	67%	100%	10.64	0.17	100%	10.08	0.18
NIST-224			157	150	67%	100%	13.86	0.18	100%	13.54	0.21
NIST-256	5 2 1	84	180	172	67%	100%	18.78	0.21	100%	18.92	0.23
NIST-384			269	256	67%	100%	32.69	0.28	100%	31.92	0.36
NIST-521			365	347	67%	100%	38.43	0.34	100%	38.67	0.37
NIST-192			129	120	63%	100%	14.44	0.40	100%	11.90	0.33
NIST-224			150	139	62%	100%	17.17	0.49	100%	14.12	0.39
NIST-256	13 1 0	106	172	159	62%	100%	18.17	0.56	100%	17.09	0.43
NIST-384			257	235	61%	100%	26.69	0.76	100%	27.20	0.58
NIST-521			349	320	61%	100%	41.83	0.92	100%	42.51	0.78
NIST-192			135	130	68%	100%	19.12	0.34	100%	22.64	0.36
NIST-224			158	152	68%	100%	25.70	0.42	100%	26.76	0.41
NIST-256	6 2 0	108	180	174	68%	100%	29.42	0.48	100%	31.77	0.45
NIST-384			270	263	68%	100%	49.65	0.65	100%	52.67	0.59
NIST-521			366	360	69%	100%	78.84	0.82	100%	80.13	0.73
NIST-192			123	116	60%	99%	47.61	1.27	98%	48.77	1.00
NIST-224			144	135	60%	100%	54.27	1.39	100%	55.35	1.12
NIST-256	16 1 0	154	164	155	61%	100%	66.70	1.45	100%	67.10	1.21
NIST-384			246	230	60%	100%	119.05	2.13	100%	118.08	1.79
NIST-521			334	310	60%	100%	164.07	2.73	100%	166.56	2.03
NIST-192			130	126	66%	99%	111.52	1.27	99%	114.83	0.98
NIST-224			152	148	66%	100%	133.61	1.29	100%	138.78	1.17
NIST-256	7 2 0	151	174	168	66%	100%	145.50	1.52	100%	147.39	1.25
NIST-384			260	253	66%	100%	264.65	1.97	100%	262.15	1.65
NIST-521			353	340	65%	100%	357.88	2.53	100%	363.22	2.07
NIST-192			135	128	67%	100%	59.41	0.27	100%	64.74	0.22
NIST-224			158	150	67%	100%	64.67	0.29	100%	67.65	0.24
NIST-256	5 3 0	161	180	170	66%	100%	73.62	0.33	100%	71.92	0.27
NIST-384			270	255	66%	100%	120.58	0.43	100%	124.39	0.37
NIST-521			367	345	66%	100%	175.77	0.51	100%	176.14	0.46
NIST-192			134	125	65%	100%	82.25	0.21	100%	84.92	0.20
NIST-224			156	145	65%	100%	88.77	0.27	100%	89.34	0.23
NIST-256	5 3 1	166	178	166	65%	100%	100.87	0.29	100%	104.57	0.25
NIST-384			267	250	65%	100%	144.94	0.41	100%	140.31	0.34
NIST-521			361	339	65%	100%	211.27	0.51	100%	214.37	0.41
NIST-192			132	124	65%	100%	94.37	0.21	99%	98.16	0.20
NIST-224			154	144	64%	95%	106.45	0.22	95%	107.29	0.22
NIST-256	5 3 2	171	176	165	64%	100%	106.31	0.25	100%	103.60	0.24
NIST-384			264	247	64%	100%	175.18	0.34	100%	170.94	0.34
NIST-521			358	335	64%	100%	260.96	0.42	100%	263.96	0.42
NIST-192			118	114	59%	97%	320.58	4.30	95%	313.52	4.19
NIST-224			137	132	59%	94%	444.92	4.78	94%	452.65	4.79
NIST-256	21 1 0	254	157	152	59%	100%	524.03	5.21	100%	544.92	5.22
NIST-384			235	225	59%	100%	864.33	7.11	100%	880.24	6.82
NIST-521			318	301	58%	100%	1272.32	9.37	100%	1280.23	9.50

Xu et al. [32] used a dimension 294 lattice to recover the hidden point when the number of exposed bits is 333 (see the last row of [32, Table 1], where $333 \approx 0.64 \cdot 521$). Here using a 254-dimension lattice, we can recover the hidden point when the number of exposed bits is 301.

Acknowledgments: The authors would like to thank anonymous reviewers for their helpful comments and suggestions. Jun Xu and Lei Hu was supported the National Natural Science Foundation of China (Grants 61732021, 62272454). Huaxiong Wang was supported by the National Research Foundation, Singapore under its Strategic Capability Research Centres Funding Initiative and Singapore Ministry of Education under Research Grant MOE2019-T2-2-083.

References

1. Adi Akavia. Solving hidden number problem with one bit oracle and advice. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*, pages 337–354. Springer, 2009.
2. Martin R. Albrecht and Nadia Heninger. On bounded distance decoding with predicate: Breaking the "lattice barrier" for the hidden number problem. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021*, pages 528–558, Cham, 2021. Springer International Publishing.
3. Dan Boneh. The decision Diffie-Hellman problem. In *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings*, pages 48–63, 1998.
4. Dan Boneh, Shai Halevi, and Nick Howgrave-Graham. The modular inversion hidden number problem. In *ASIACRYPT 2001*, pages 36–51. https://www.iacr.org/archive/asiacrypt2001/22480036.pdf. Springer, 2001.
5. Dan Boneh and Igor E. Shparlinski. On the unpredictability of bits of the elliptic curve Diffie–Hellman scheme. In *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, pages 201–212, 2001.
6. Dan Boneh and Ramarathnam Venkatesan. Hardness of computing the most significant bits of secret keys in Diffie-Hellman and related schemes. In *CRYPTO 1996*, pages 129–142. Springer, 1996.
7. Dan Boneh and Ramarathnam Venkatesan. Rounding in lattices and its cryptographic applications. In Michael E. Saks, editor, *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, 5-7 January 1997, New Orleans, Louisiana, USA*, pages 675–681. ACM/SIAM, 1997.
8. Don Coppersmith. Finding a small root of a bivariate integer equation; factoring with high bits known. In *EUROCRYPT 1996*, pages 178–189. Springer, 1996.
9. Don Coppersmith. Finding a small root of a univariate modular equation. In *EUROCRYPT 1996*, pages 155–165. Springer, 1996.
10. Jean-Sébastien Coron and Rina Zeitoun. Improved factorization of $n=p^r q^s$. In Nigel P. Smart, editor, *Topics in Cryptology - CT-RSA 2018 - The Cryptographers' Track at the RSA Conference 2018, San Francisco, CA, USA, April 16-20, 2018, Proceedings*, volume 10808 of *Lecture Notes in Computer Science*, pages 65–79. Springer, 2018.

11. Jean-Charles Faugère, Patrizia M. Gianni, Daniel Lazard, and Teo Mora. Efficient computation of zero-dimensional Gröbner Bases by change of ordering. *J. Symb. Comput.*, 16(4):329–344, 1993.
12. Steven D. Galbraith. *Mathematics of Public Key Cryptography.* Cambridge University Press, 2012.
13. Amir Hashemi and Daniel Lazard. Sharper complexity bounds for zero-dimensional Gröbner bases and polynomial system solving. *Int. J. Algebra Comput.*, 21(5):703–713, 2011.
14. Nicholas Howgrave-Graham. Finding small roots of univariate modular equations revisited. In *Crytography and Coding*, pages 131–142. Springer, 1997.
15. Jan Jancar, Vladimir Sedlacek, Petr Svenda, and Marek Sýs. Minerva: The curse of ECDSA nonces systematic analysis of lattice attacks on noisy leakage of bit-length of ECDSA nonces. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(4):281–308, 2020.
16. David Jao, Dimitar Jetchev, and Ramarathnam Venkatesan. On the bits of elliptic curve Diffie-Hellman keys. In *Progress in Cryptology - INDOCRYPT 2007, 8th International Conference on Cryptology in India, Chennai, India, December 9-13, 2007, Proceedings*, pages 33–47, 2007.
17. Dimitar Jetchev and Ramarathnam Venkatesan. Bits security of the elliptic curve Diffie-Hellman secret keys. In *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*, pages 75–92, 2008.
18. Ellen Jochemsz and Alexander May. A strategy for finding roots of multivariate polynomials with new applications in attacking RSA variants. In *ASIACRYPT 2006*, pages 267–282. Springer, 2006.
19. Ellen Jochemsz and Alexander May. A polynomial time attack on RSA with private CRT-exponents smaller than $N^{0.073}$. In Alfred Menezes, editor, *Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007, Proceedings*, volume 4622 of *Lecture Notes in Computer Science*, pages 395–411. Springer, 2007.
20. Arjen Klaas Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.
21. San Ling, Igor E Shparlinski, Ron Steinfeld, and Huaxiong Wang. On the modular inversion hidden number problem. *Journal of Symbolic Computation*, 47(4):358–367, 2012.
22. Alexander May. Using LLL-reduction for solving RSA and factorization problems. In *The LLL algorithm*, pages 315–348. Springer, 2010.
23. Robert Merget, Marcus Brinkmann, Nimrod Aviram, Juraj Somorovsky, Johannes Mittmann, and Jörg Schwenk. Raccoon attack: Finding and exploiting most-significant-bit-oracles in TLS-DH(E). In *30th USENIX Security Symposium (USENIX Security 21)*, Vancouver, B.C., August 2021. USENIX Association.
24. Matús Nemec, Marek Sýs, Petr Svenda, Dusan Klinec, and Vashek Matyas. The return of Coppersmith's attack: Practical factorization of widely used RSA moduli. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 1631–1648, 2017.
25. Arnold Neumaier and Damien Stehlé. Faster LLL-type reduction of lattice bases. In Sergei A. Abramov, Eugene V. Zima, and Xiao-Shan Gao, editors, *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC 2016, Waterloo, ON, Canada, July 19-22, 2016*, pages 373–380. ACM, 2016.

26. Phong Q. Nguyen and Damien Stehlé. An LLL algorithm with quadratic complexity. *SIAM J. Comput.*, 39(3):874–903, 2009.
27. Keegan Ryan. Return of the hidden number problem. A widespread and novel key extraction attack on ECDSA and DSA. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(1):146–168, 2019.
28. Barak Shani. On the bit security of elliptic curve Diffie-Hellman. In *Public-Key Cryptography - PKC 2017 - 20th IACR International Conference on Practice and Theory in Public-Key Cryptography, Amsterdam, The Netherlands, March 28-31, 2017, Proceedings, Part I*, pages 361–387, 2017.
29. Atsushi Takayasu and Noboru Kunihiro. Better lattice constructions for solving multivariate linear equations modulo unknown divisors. In *Information Security and Privacy - 18th Australasian Conference, ACISP 2013, Brisbane, Australia, July 1-3, 2013. Proceedings*, pages 118–135, 2013.
30. Atsushi Takayasu, Yao Lu, and Liqiang Peng. Small CRT-exponent RSA revisited. In *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part II*, pages 130–159, 2017.
31. Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra (3. ed.)*. Cambridge University Press, 2013.
32. Jun Xu, Lei Hu, and Santanu Sarkar. Cryptanalysis of elliptic curve hidden number problem from PKC 2017. *Des. Codes Cryptogr.*, 88(2):341–361, 2020.
33. Jun Xu, Santanu Sarkar, Lei Hu, Zhangjie Huang, and Liqiang Peng. Solving a class of modular polynomial equations and its relation to modular inversion hidden number problem and inversive congruential generator. *Des. Codes Cryptogr.*, 86(9):1997–2033, 2018.
34. Jun Xu, Santanu Sarkar, Lei Hu, Huaxiong Wang, and Yanbin Pan. New results on modular inversion hidden number problem and inversive congruential generator. In *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part I*, pages 297–321, 2019.

Supplementary Material

A Proof of Lemma 3

Proof. Since p is a prime, our goal is to show that $\mathbf{M}_{j_1,\cdots,j_l}$ is invertible over prime field $\mathbb{F}_p$. Denote $\widetilde{F}_{j_m} = \prod_{u=1,u\neq m}^{l}(x_0^2 + E_{j_u}x_0 + D_{j_u})$ for all $1 \le m \le l$. According to (14), the rows of $2l \times 2l$ matrix $\mathbf{M}_{j_1,\cdots,j_l}$ respectively correspond to the coefficient vectors of the following $2l$ polynomials on a basis $(1, x_0, \cdots, x_0^{2l-1})$ over $\mathbb{Z}_{p^{l-1}}$:

$$\widetilde{F}_{j_1}, \cdots, \widetilde{F}_{j_l}, x_0 \cdot \widetilde{F}_{j_1}, \cdots, x_0 \cdot \widetilde{F}_{j_l}.$$

Note that $E_{j_u} := 2(h_0 - x_{Q_{j_u}}) \bmod p$ and $D_{j_u} := (h_0 - x_{Q_{j_u}})^2 \bmod p$, where $x_{Q_{j_u}}$ is the x-coordinate of the point Q_{j_u}. Hence, $x_0^2 + E_{j_u}x_0 + D_{j_u} = (x_0 + h_0 - x_{Q_{j_u}})^2 \bmod p$ for $1 \le u \le l$. For the sake of discussion, let $\gamma_{j_u} := h_0 - x_{Q_{j_u}}$. Hence, $\widetilde{F}_{j_m} = \prod_{u=1,u\neq m}^{l}(x_0 + \gamma_{j_u})^2 \bmod p^{l-1}$ for all $1 \le m \le l$.

The matrix $\mathbf{M}_{j_1,\cdots,j_l}$ is invertible over $\mathbb{F}_p$ if and only if $\widetilde{F}_{j_1}, \cdots, \widetilde{F}_{j_l}, x_0 \cdot \widetilde{F}_{j_1}, \cdots, x_0 \cdot \widetilde{F}_{j_l}$ are linearly independent polynomials over $\mathbb{F}_p$. Suppose that there exist $r_1, \cdots, r_l, s_1, \cdots, s_l \in \mathbb{F}_p$ such that $r_1\widetilde{F}_{j_1} + \cdots + r_l\widetilde{F}_{j_l} + s_1x_0 \cdot \widetilde{F}_{j_1} + s_lx_0 \cdot \widetilde{F}_{j_1} = 0$, i.e., $(r_1 + x_0s_1) \cdot \widetilde{F}_{j_1} + \cdots + (r_l + x_0s_l) \cdot \widetilde{F}_{j_l} = 0$. Taking modulo $(x_0 + \gamma_{j_u})^2$ on both sides of the above relation, we get

$$(r_u + x_0s_u) \cdot \widetilde{F}_{j_u}^2 = 0 \bmod (x_0 + \gamma_{j_u})^2 \text{ for all } u = 1, \cdots, l. \tag{40}$$

Note that $\gamma_{j_1} = h_0 - x_{Q_{j_1}}, \cdots, \gamma_{j_l} = h_0 - x_{Q_{j_l}}$ are different in $\mathbb{F}_p$. We have that $x_0 + \gamma_{j_1}, \cdots, x_0 + \gamma_{j_l}$ are pairwise coprime over $\mathbb{F}_p$, further, $\gcd(\widetilde{F}_{j_u}, x_0 + \gamma_{j_u}) = 1$ for all $1 \le u \le l$. Thus, we obtain $r_u + x_0s_u = 0 \bmod (x_0 + \gamma_{j_u})^2$ for $u \in [1, \cdots, l]$ from (40). Since $\deg((x_0 + \gamma_{j_u})^2) = 2$ and $\deg(r_u + x_0s_u) \le 1$, we get $r_u + x_0s_u = 0$ for all $1 \le u \le l$, i.e., $r_1 = s_1 = \cdots = r_l = s_l = 0$. It implies that $r_1\widetilde{F}_{j_1} + \cdots + r_l\widetilde{F}_{j_l} + s_1x_0 \cdot \widetilde{F}_{j_1} + s_lx_0 \cdot \widetilde{F}_{j_l} = 0 \Longleftrightarrow r_1 = s_1 = \cdots = r_l = s_l = 0$. Hence, $\widetilde{F}_{j_1}, \cdots, \widetilde{F}_{j_l}, x_0 \cdot \widetilde{F}_{j_1}, \cdots, x_0 \cdot \widetilde{F}_{j_l}$ are linearly independent over $\mathbb{F}_p$, that is, $\mathbf{M}_{j_1,\cdots,j_l}$ is invertible over $\mathbb{F}_p$. □

B An example of $\mathcal{L}(2, 1, t)$

We present a toy example to understand lattice $\mathcal{L}(n, d, t)$. Take $n = 2$, $d = 1$ and $0 \le t \le 1$ (the optimal t is determined later). The index set $\mathcal{I}(2, 1, t) = \mathcal{I}_1 \cup \mathcal{I}_2$, where

$$\mathcal{I}_1 = \{(i_0, i_1, i_2) \mid 0 \le i_0 \le 1, 0 \le i_1, i_2 \le 1, 0 \le i_1 + i_2 \le 1\},$$
$$\mathcal{I}_2 = \{(i_0, i_1, i_2) \mid 0 \le i_0 \le t, 0 \le i_1, i_2 \le 1, i_1 + i_2 = 2\}.$$

where $0 \le l = i_1 + i_2 \le 2$. From the order (10), the monomials $x_0^{i_0}y_1^{i_1}y_2^{i_2}$ for $(i_0, i_1, i_2) \in \mathcal{I}(2, 1, t)$ are arranged as follows:

$$1 \prec x_0 \prec y_1 \prec x_0y_1 \prec y_2 \prec x_0y_2 \prec y_1y_2 \prec \cdots \prec x_0^t y_1y_2. \tag{41}$$

For given polynomials $\mathcal{F}_j$ satisfying $\mathcal{F}_j(e_0, \widetilde{e}_j) = 0 \pmod p$ for $j = 1, 2$, we generate polynomials $G_{i_0,i_1,i_2}(x_0, y_1, y_2)$ as follows.

Case A: The tuples (i_0, i_1, i_2)'s are respectively equal to

$$(0, 0, 0), (1, 0, 0), (0, 1, 0), (1, 1, 0), (0, 0, 1), (1, 0, 1).$$

We define $G_{i_0,i_1,i_2} = F_{i_0,i_1,i_2}$. Based on Section 3.1, we have $G_{0,0,0} = p, G_{1,0,0} = px_0, G_{0,1,0} = py_1, G_{1,1,0} = px_0y_1$, $G_{0,0,1} = py_2, G_{1,0,1} = px_0y_2$.

Case B: The tuples $(i_0, i_1, i_2) = (i_0, 1, 1)$, where $i_0 = 0, \cdots, t$ and $0 \leq t \leq 1$ Then, $j_1 = 1, j_2 = 2$ based on $y_{j_1}y_{j_2} = y_1^{i_1}y_2^{i_2}$. We define

$$G_{i_0,1,1} = H_{i_0,1,1} + J_{i_0,1,1} + K_{i_0,1,1}$$

in sense of modulo p, which is considered to be the corresponding polynomial over $\mathbb{Z}$. Here,

$$\begin{aligned}
&H_{i_0,1,1} := w_{i_0+1,1}y_1\mathcal{F}_2 + w_{i_0+1,2}y_2\mathcal{F}_1 + w_{i_0+1,3}x_0y_1\mathcal{F}_2 + w_{i_0+1,4}x_0y_2\mathcal{F}_1,\\
&J_{i_0,1,1} = w_{i_0+1,1}C_1\mathcal{F}_2 + w_{i_0+1,2}C_2\mathcal{F}_1 + w_{i_0+1,3}x_0C_1\mathcal{F}_2 + w_{i_0+1,4}x_0C_2\mathcal{F}_1,\\
&K_{i_0,1,1} = w_{i_0+1,3}(B_1 - C_1E_1)\mathcal{F}_2 + w_{i_0+1,4}(B_2 - C_2E_2)\mathcal{F}_1,
\end{aligned}$$

where the vector $(w_{i_0+1,1}, w_{i_0+1,2}, w_{i_0+1,3}, w_{i_0+1,4})$ is the (i_0+1)-th row in the matrix $\mathbf{W}_{12}$, which is the inverse (modulo p) of the following matrix

$$\mathbf{M}_{12} = \begin{pmatrix} D_2 & E_2 & 1 & 0\\ D_1 & E_1 & 1 & 0\\ 0 & D_2 & E_2 & 1\\ 0 & D_1 & E_1 & 1 \end{pmatrix}$$

which is defined in Section 3.1, and B_j, C_j, D_j, E_j are the coefficients of $\mathcal{F}_j = A_j + B_jx_0 + C_jx_0^2 + D_jy_j + E_jx_0y_j + x_0^2y_j$ for $1 \leq j \leq 2$.

According to $\mathbf{W}_{12} \cdot \mathbf{M}_{12} = \mathbf{I}_4 \bmod p$, where $\mathbf{I}_4$ is the 4×4 identity matrix. Note that $0 \leq i_0 \leq 1$, we have

$$\begin{aligned}
&(w_{i_0+1,1}, w_{i_0+1,2}, w_{i_0+1,3}, w_{i_0+1,4}) \cdot (0, 0, 1, 1)^T = 0 \bmod p,\\
&(w_{i_0+1,1}, w_{i_0+1,2}, w_{i_0+1,3}, w_{i_0+1,4}) \cdot (1, 1, E_2, E_1)^T = 0 \bmod p.
\end{aligned}$$

That is,

$$\begin{aligned}
&w_{i_0+1,3} + w_{i_0+1,4} = 0 \bmod p,\\
&w_{i_0+1,1} + w_{i_0+1,2} + E_2w_{i_0+1,3} + E_1w_{i_0+1,4} = 0 \bmod p.
\end{aligned} \tag{42}$$

Next, we will present that $G_{i_0,1,1}$ $(i_0 = 0, 1)$ can be written as

$$G_{i_0,1,1} = \Delta_{i_0,1} + \Delta_{i_0,2}x_0 + \Delta_{i_0,3}y_1 + \Delta_{i_0,4}x_0y_1 + \Delta_{i_0,5}y_2 + \Delta_{i_0,6}x_0y_2 + x_0^{i_0}y_1y_2,$$

where the coefficients $\Delta_{i_0,j}$'s are known integers for $0 \leq i_0 \leq 1$ and $1 \leq j \leq 6$. It means that the leading term of $G_{i_0,1,1}$ is $x_0^{i_0}y_1y_2$. Moreover, the monomials except $x_0^{i_0}y_1y_2$ in $G_{i_0,1,1}$ lie in $\{1, x_0, y_1, x_0y_1, y_2, x_0y_2\}$. That is, $G_{i_0,1,1}$ does not contain monomials $x_0^3, x_0^3y_1, x_0^3y_2, x_0^2, x_0^2y_1, x_0^2y_2$. Now, we give the detailed analysis as follows.

Note that $G_{i_0,1,1} = H_{i_0,1,1} + J_{i_0,1,1} + K_{i_0,1,1}$, and

$$\begin{array}{l} H_{i_0,1,1} = w_{i_0+1,1}y_1\mathcal{F}_2 + w_{i_0+1,2}y_2\mathcal{F}_1 + w_{i_0+1,3}x_0y_1\mathcal{F}_2 + w_{i_0+1,4}x_0y_2\mathcal{F}_1 \\ J_{i_0,1,1} = w_{i_0+1,1}C_1\mathcal{F}_2 + w_{i_0+1,2}C_2\mathcal{F}_1 + w_{i_0+1,3}x_0C_1\mathcal{F}_2 + w_{i_0+1,4}x_0C_2\mathcal{F}_1. \\ K_{i_0,1,1} = w_{i_0+1,3}(B_1 - C_1E_1)\mathcal{F}_2 + w_{i_0+1,4}(B_2 - C_2E_2)\mathcal{F}_1. \end{array}$$

According to (16) in Section 3.1, we get that all monomials in the polynomial $H_{i_0,1,1}$ lie in $\{y_1, x_0y_1, x_0^2y_1, x_0^3y_1, y_2, x_0y_2, x_0^2y_2, x_0^3y_2, x_0^{i_0}y_1y_2\}$. Moreover, $x_0^{i_0}y_1y_2$ is the leading term of $H_{i_0,1,1}$ based on the order (10).

Note that polynomials $J_{i_0,1,1}$ and $K_{i_0,1,1}$ can be regarded as some linear combinations of $\mathcal{F}_1, \mathcal{F}_2, x_0\mathcal{F}_1, x_0\mathcal{F}_2$. It means that monomials in these two polynomials belong to $\{1, x_0, x_0^2, x_0^3, y_1, x_0y_1, x_0^2y_1, x_0^3y_1, y_2, x_0y_2, x_0^2y_2, x_0^3y_2\}$. To sum up, all monomials in $G_{i_0,1,1}$ lie in

$$\{1, x_0, x_0^2, x_0^3, y_1, x_0y_1, x_0^2y_1, x_0^3y_1, y_2, x_0y_2, x_0^2y_2, x_0^3y_2, x_0^{i_0}y_1y_2\}.$$

Based on the order (10), we have $1 \prec x_0 \prec x_0^2 \prec x_0^3 \prec y_1 \prec x_0y_1 \prec x_0^2y_1 \prec x_0^3y_1 \prec y_2 \prec x_0y_2 \prec x_0^2y_2 \prec x_0^3y_2 \prec x_0^{i_0}y_1y_2$. Hence, $x_0^{i_0}y_1y_2$ is the leading term of the polynomial $G_{i_0,1,1}$.

Next, we show that $G_{i_0,1,1}$ does not contain monomials $x_0^3, x_0^3y_1, x_0^3y_2, x_0^2, x_0^2y_1, x_0^2y_2$. For the convenience of subsequent analysis, we rewrite $\mathcal{F}_j$ as

$$x_0^2(y_j + C_j) + x_0(E_jy_j + B_j) + (D_jy_j + A_j), 1 \le j \le 2.$$

Moreover, we rewrite $G_{i_0,1,1}$ as

$$\begin{array}{l} w_{i_0+1,1}(y_1 + C_1)\mathcal{F}_2 + w_{i_0+1,2}(y_2 + C_2)\mathcal{F}_1 \\ +w_{i_0+1,3}(B_1 - C_1E_1)\mathcal{F}_2 + w_{i_0+1,4}(B_2 - C_2E_2)\mathcal{F}_1 \\ +w_{i_0+1,3}(y_1 + C_1)x_0\mathcal{F}_2 + w_{i_0+1,4}(y_2 + C_2)x_0\mathcal{F}_1 \end{array}$$

in the sense of modulo p.

First, let us focus on the x_0^3-term in $G_{i_0,1,1}$, which only appears in

$$w_{i_0+1,3}(y_1 + C_1)x_0\mathcal{F}_2 + w_{i_0+1,4}(y_2 + C_2)x_0\mathcal{F}_1.$$

Hence, the x_0^3-term is equal to $(w_{i_0+1,3} + w_{i_0+1,4})x_0^3(y_1 + C_1)(y_2 + C_2) \bmod p$. From (42), we have $(w_{i_0+1,3} + w_{i_0+1,4}) \bmod p = 0$. Hence, $G_{i_0,1,1}$ does not contain monomials $x_0^3, x_0^3y_1, x_0^3y_2$.

Next, let us focus on the x_0^2-term in $G_{i_0,1,1}$. The x_0^2-term in $w_{i_0+1,1}(y_1 + C_1)\mathcal{F}_2 + w_{i_0+1,2}(y_2 + C_2)\mathcal{F}_1$ is

$$(w_{i_0+1,1} + w_{i_0+1,2})x_0^2(y_1 + C_1)(y_2 + C_2). \tag{43}$$

The x_0^2-term in $w_{i_0+1,3}(B_1 - C_1E_1)\mathcal{F}_2 + w_{i_0+1,4}(B_2 - C_2E_2)\mathcal{F}_1$ is

$$w_{i_0+1,3}(B_1 - E_1C_1)x_0^2(y_2 + C_2) + w_{i_0+1,4}(B_2 - E_2C_2)x_0^2(y_1 + C_1). \tag{44}$$

The x_0^2-term in $w_{i_0+1,3}(y_1+C_1)x_0\mathcal{F}_2 + w_{i_0+1,4}(y_2+C_2)x_0\mathcal{F}_1$ is

$$\begin{aligned}
&w_{i_0+1,3}x_0^2(y_1+C_1)(E_2y_2+B_2) + w_{i_0+1,4}x_0^2(y_2+C_2)(E_1y_1+B_1) \\
&= w_{i_0+1,3}E_2x_0^2(y_1+C_1)(y_2+C_2) + w_{i_0+1,4}E_1x_0^2(y_1+C_1)(y_2+C_2) \\
&+w_{i_0+1,3}(B_2-E_2C_2)x_0^2(y_1+C_1) + w_{i_0+1,4}(B_1-E_1C_1)x_0^2(y_2+C_2).
\end{aligned} \tag{45}$$

Based on (43), (44) and (45), we deduce that the x_0^2-term in $G_{i_0,1,1}$ is

$$\begin{aligned}
&(w_{i_0+1,3}+w_{i_0+1,4})(B_2-E_2C_2)x_0^2(y_1+C_1) \\
&+(w_{i_0+1,3}+w_{i_0+1,4})(B_1-E_1C_1)x_0^2(y_2+C_2) \\
&+(w_{i_0+1,1}+w_{i_0+1,2}+w_{i_0+1,3}E_2+w_{i_0+1,4}E_1)x_0^2(y_1+C_1)(y_2+C_2)
\end{aligned}$$

in the sense of modulo p. According to (42), we get $(w_{i_0+1,3}+w_{i_0+1,4}) \bmod p = 0$ and $(w_{i_0+1,1}+w_{i_0+1,2}+w_{i_0+1,3}E_2+w_{i_0+1,4}E_1) \bmod p = 0$. Hence, $G_{i_0,1,1}$ does not contain monomials $x_0^2, x_0^2y_1, x_0^2y_2$.

In Table 3, we present the triangular basis matrix of lattice $\mathcal{L}(2,1,t)$. The dimension and determinant of $\mathcal{L}(2,1,t)$ are respectively equal to

$$\dim(\mathcal{L}(2,1,t)) = 7+t, \det(\mathcal{L}(2,1,t)) = X^{\frac{(t+1)(t+4)}{2}+7}p^6,$$

where $0 \le t \le 1$. From the simplified Coppersmith condition (2), we deduce $(\det(\mathcal{L}(2,1,t)))^{\frac{1}{7+t}} < p$, which is equivalent to

$$X < p^{\frac{2t+2}{t^2+5t+18}}.$$

For $t = 0, 1$, the bound becomes $X < p^{\frac{1}{9}}$, and $X < p^{\frac{1}{6}}$ respectively. When $t = 1$, the bound $X < p^{\frac{1}{6}}$ is optimal.

Table 3. The triangular basis matrix of lattice $\mathcal{L}(2,1,t)$.

Poly	1	x_0	y_1	x_0y_1	y_2	x_0y_2	y_1y_2	$\cdots$	$x_0^ty_1y_2$
$G_{0,0,0}$	p								
$G_{1,0,0}$		pX							
$G_{0,1,0}$			pX						
$G_{1,1,0}$				pX^2					
$G_{0,0,1}$					pX				
$G_{1,0,1}$						pX^2			
$G_{0,1,1}$	–	–	–	–	–	–	X^2		
$\ldots$	–	–	–	–	–	–	–	$\ddots$	
$G_{t,1,1}$	–	–	–	–	–	–			X^{t+2}

Off-diagonal entries are denoted by the symbol “ – ” and their values have no effect on the determinant of lattice.

Remark 2. For $(n,d,t) = (2,1,1)$, the leading terms of polynomials $G_{0,1,1}$ and $G_{1,1,1}$ are $y_1y_2, x_0y_1y_2$ respectively. Hence, the leading coefficients of $G_{0,1,1}(x_0X, y_1X, y_2X)$ and $G_{1,1,1}(x_0X, y_1X, y_2X)$ are X^2 and X^3 respectively, which are diagonal elements in the triangular basis matrix of lattice $\mathcal{L}(2,1,1)$. According to the bound $0 < X < p^{\frac{1}{6}}$, we get $0 < X^3 < p^d$, where $d = 1$. Therefore, $G_{0,1,1}$ and $G_{1,1,1}$ are all helpful polynomials.

C Proof of improved bounds

Our target is to obtain a lower bound of $\left(2^{-\frac{w(w-1)}{4\alpha}} w^{-\frac{w-n}{2\alpha}}\right) p^{S(n,d,t)}$ in the right-hand side of Equation (24).

First, we consider the term $2^{-\frac{w(w-1)}{4\alpha}} w^{-\frac{w-n}{2\alpha}}$. From $w = (t+1)\binom{n}{d+1} + 2d\sum_{l=0}^{d}\binom{n}{l}$,
$\alpha = \frac{(2d+t+2)(t+1)}{2}\binom{n}{d+1} + d\sum_{l=0}^{d}(2d-1+2l)\binom{n}{l}$, we obtain

$$\frac{\alpha}{dw} = \frac{\frac{(2d+t+2)(t+1)}{2}\binom{n}{d+1} + d\sum\limits_{l=0}^{d}(2d-1+2l)\binom{n}{l}}{d(t+1)\binom{n}{d+1} + 2d^2\sum\limits_{l=0}^{d}\binom{n}{l}} > 1.$$

Based on the relation, we get that $2^{-\frac{w(w-1)}{4\alpha}} > 2^{-\frac{w}{4d}}$ and $w^{-\frac{w-n}{2\alpha}} > w^{-\frac{1}{2d}}$. It implies that

$$\left(2^{-\frac{w(w-1)}{4\alpha}} w^{-\frac{w-n}{2\alpha}}\right) p^{S(n,d,t)} > \left(2^{-\frac{w}{4d}} w^{-\frac{1}{2d}}\right) p^{S(n,d,t)} = p^{S(n,d,t) - \frac{w+2\log_2 w}{4d\log_2 p}}.$$

For a sufficiently large p, $\frac{w+2\log w}{4d}$ is negligible compared to $\log_2 p$ (the size of p will be explicitly given in subsequent analysis). Hence, the right-hand side of condition (24) is simplified as $p^{S(n,d,t)}$ for a sufficiently large p.

Next, we focus on $S(n,d,t)$, and write $S(n,d,t) = \frac{2d}{2d+2+t}(1-\epsilon(n,d,t))$, where

$$\epsilon(n,d,t) := \frac{2(2d-1)d^2\sum\limits_{l=0}^{d}\binom{n}{l} + 2(2+t-dt)\sum\limits_{l=0}^{d} l\binom{n}{l} + nd(2d+t+2)}{(t+1)d(2d+t+2)\binom{n}{d+1} + 2(2d-1)d^2\sum\limits_{l=0}^{d}\binom{n}{l} + 4d^2\sum\limits_{l=0}^{d} l\binom{n}{l}}.$$

Here, $\epsilon(n,d,t) > 0$ which is because that $2(2d-1)d^2\sum\limits_{l=0}^{d}\binom{n}{l} - 2dt\sum\limits_{l=0}^{d} l\binom{n}{l} > 0$ for all $0 \le t \le 2d-1$. Furthermore, we have

$$0 < \epsilon(n,d,t) < \frac{2(2d-1)d^2\sum\limits_{l=0}^{d}\binom{n}{l} + 2(t+2)d\sum\limits_{l=0}^{d}\binom{n}{l} + nd(2d+2+t)}{d(2d+2+t)(t+1)\binom{n}{d+1}}.$$

From $1 \le d \le n-1$ and $0 \le t \le 2d-1$, we can get that

$$\frac{2(2d-1)d^2\sum_{l=0}^{d}\binom{n}{l}+2(t+2)d\sum_{l=0}^{d}\binom{n}{l}}{d(2d+2+t)(t+1)\binom{n}{d+1}}+\frac{nd(2d+2+t)}{d(2d+2+t)(t+1)\binom{n}{d+1}}<2d\frac{\sum_{l=0}^{d}\binom{n}{l}}{\binom{n}{d+1}}+\frac{\binom{n}{1}}{\binom{n}{d+1}}.$$

It means that $0<\epsilon(n,d,t)<2d\frac{\sum_{l=0}^{d}\binom{n}{l}}{\binom{n}{d+1}}+\frac{\binom{n}{1}}{\binom{n}{d+1}}$. From $\frac{\binom{n}{l}}{\binom{n}{d+1}}=\frac{(d+1)!(n-d-1)!}{l!(n-l)!}=\frac{d+1}{n-d}\cdot\frac{d}{n-d+1}\cdots\frac{l+1}{n-l}\le(\frac{d+1}{n-d})^{d-l+1}$ for all $0\le l\le d$, we get $0<\epsilon(n,d,t)<2d\sum_{l=0}^{d}(\frac{d+1}{n-d})^{d-l+1}+(\frac{d+1}{n-d})^{d}$. According to the summation formula of geometric series, we have $\sum_{l=0}^{d}(\frac{d+1}{n-d})^{d-l+1}=\frac{d+1}{n-2d-1}(1-(\frac{d+1}{n-d})^{d+1})$. Hence, the above relation is equivalent to

$$0<\epsilon(n,d,t)<\frac{(d+1)2d}{n-2d-1}\big(1-(\frac{d+1}{n-d})^{d+1}\big)+(\frac{d+1}{n-d})^{d}.$$

By taking $n=d^{3+c}$ for any constant $c>0$, we get that $\lim_{d\to\infty}(\epsilon(n,d,t)/(1/d))=0$. That is, $\epsilon(n,d,t)=o(\frac{1}{d})$. From $S(n,d,t)=\frac{2d}{2d+2+t}(1-\epsilon(n,d,t))$, we obtain $S(n,d,t)>\frac{2d}{2d+2+t}(1-\frac{(d+1)2d}{n-2d-1}\big(1-(\frac{d+1}{n-d})^{d+1}\big)-(\frac{d+1}{n-d})^{d})$. When $t=0$, the term $\frac{2d}{2d+2+t}$ becomes the maximum of $\frac{d}{d+1}$. For $t=0$, the bound $X<p^{S(n,d,t)}$ becomes $X<p^{1-\frac{1}{d+1}-\varepsilon}$. Here

$$\varepsilon:=\frac{d}{d+1}\cdot\epsilon(n,d,0)=\frac{d^2(2d-1)\sum_{l=0}^{d}\binom{n}{l}+2\sum_{l=0}^{d}l\binom{n}{l}+d(d+1)n}{(d+1)^2\binom{n}{d+1}+d(d+1)(2d-1)\sum_{l=0}^{d}\binom{n}{l}+2d(d+1)\sum_{l=0}^{d}l\binom{n}{l}}>0, \quad (46)$$

where $\varepsilon=o(\frac{1}{d+1})$ by taking $n=d^{3+c}$ for any constant $c>0$.

Finally, we explicitly present the size of p such that $\frac{w+2\log w}{4d\log_2 p}$ is negligible, i.e. $\frac{w+2\log w}{4d\log_2 p}=o(\frac{1}{d+1})$. According to $w=\binom{n}{d+1}(1+o(1))=\frac{n^{d+1}}{(d+1)!}(1+o(1))$ and Stirling's approximation $(d+1)!\approx\sqrt{2\pi(d+1)}(\frac{d+1}{e})^{d+1}$, we have $\frac{w+2\log w}{4d\log_2 p}\approx\frac{e^{d+1}\cdot(\frac{n}{d+1})^{d+1}}{4d\sqrt{2\pi(d+1)}\log_2 p}=\frac{d^{(2+c)d}(1+o(1))}{\log_2 p}$ by taking $n=d^{3+c}$. Hence, in order to make $\frac{w+2\log w}{4d\log_2 p}=o(\frac{1}{d+1})$ satisfied, i.e. $\frac{w+2\log w}{4d^2\log_2 p}=o(1)$, we need $p=2^{\omega(d^{(2+c)d})}$.

D Proof of Lemma 6

Proof. Suppose that there exists a vector $(e_0',\tilde{e}_1',\cdots,\tilde{e}_n')$ such that $\mathcal{F}_i(e_0',\tilde{e}_i')=0 \pmod p$ for all $1\le i\le n$. From (6), we have $\mathcal{F}_i(e_0,\tilde{e}_i)=0 \pmod p$. If $e_0'\ne e_0$, then $(e_0',\tilde{e}_1',\cdots,\tilde{e}_n')\ne(e_0,\tilde{e}_1,\cdots,\tilde{e}_n)$. On the other hand, if $e_0'=e_0$, then we deduce $\tilde{e}_i'-\tilde{e}_i=0 \bmod p$ for all $1\le i\le n$, based on $\mathcal{F}_i(e_0',\tilde{e}_i')=0 \pmod p$ and $\mathcal{F}_i(e_0,\tilde{e}_i)=0 \pmod p$. Note that $|\tilde{e}_i'-\tilde{e}_i|<2\cdot p^{1-\frac{1}{d+1}-\varepsilon}<p$ as $p=2^{\omega(d^{(2+c)d})}$.

We get $\tilde{e}'_i = \tilde{e}_i$ for all $1 \le i \le n$. Therefore, $(e'_0, \tilde{e}'_1, \cdots, \tilde{e}'_n) \neq (e_0, \tilde{e}_1, \cdots, \tilde{e}_n)$ if and only if $e'_0 \neq e_0$.

Let $(d_0, d_{0,0}, d_i, d_{0,i}, d_{00,i}) := (e_0 - e'_0, e_0^2 - e_0'^2, \tilde{e}_i - \tilde{e}'_i, e_0\tilde{e}_i - e'_0\tilde{e}'_i, e_0^2\tilde{e}_i - e_0'^2\tilde{e}'_i)$. We can rewrite $\mathcal{F}_i(e_0, \tilde{e}_i) - \mathcal{F}_i(e'_0, \tilde{e}'_i) = 0 \pmod p$ as $B_i d_0 + C_i d_{0,0} + D_i d_i + E_i d_{0,i} + d_{00,i} = 0 \pmod p$. Plugging the expressions for B_i, C_i, D_i, E_i in (9) into the above relation, we obtain $2(\tilde{h}_i(h_0 - x_{Q_i}) - 2h_0 x_{Q_i} - a - x_{Q_i}^2)d_0 + (\tilde{h}_i - 2x_{Q_i})d_{0,0} + (h_0 - x_{Q_i})^2 d_i + (2(h_0 - x_{Q_i}))d_{0,i} + d_{00,i} = 0 \pmod p$. Then plugging the expression of $\tilde{h}_i$ in (8) into the above equation, we get the univariate polynomial in x_{Q_i}:

$$U_i x_{Q_i}^4 + V_i x_{Q_i}^3 + W_i x_{Q_i}^2 + Y_i x_{Q_i} + Z_i = 0 \pmod p, \quad 1 \le i \le n, \tag{47}$$

where

$$\begin{pmatrix} U_i \\ V_i \\ W_i \\ Y_i \\ Z_i \end{pmatrix} = \mathbf{M}_i \begin{pmatrix} d_0 \\ d_{0,0} \\ d_i \\ d_{0,i} \\ d_{00,i} \end{pmatrix} \mod p, \tag{48}$$

i.e., the coefficient vector $(U_i, V_i, W_i, Y_i, Z_i)$ is obtained by a linear transformation of the vector $(d_0, d_{0,0}, d_i, d_{0,i}, d_{00,i})$. Here, the matrix $\mathbf{M}_i$ (in the field $\mathbb{F}_p$) is defined as

$$\mathbf{M}_i = \left[C_0, C_1, C_2, C_3, C_4\right],$$

where column vectors

$$C_0 = \begin{pmatrix} -2 \\ -4x_P + 4e_0 + 2\tilde{e}_i \\ 6x_P^2 - 12e_0x_P - 6\tilde{e}_i x_P + 2e_0\tilde{e}_i - 6a \\ 6\tilde{e}_i x_P^2 - 4e_0\tilde{e}_i x_P + 4ax_P - 4ae_0 - 8b \\ -2\tilde{e}_i x_P^3 + 2e_0\tilde{e}_i x_P^2 + 2ax_P^2 - 4ae_0x_P + 8bx_P - 8be_0 \end{pmatrix},$$

$$C_1 = \begin{pmatrix} 1 \\ -4x_P + 2e_0 \\ 6x_P^2 - 6e_0x_P + e_0^2 \\ -4x_P^3 + 6e_0x_P^2 - 2e_0^2x_P \\ x_P^4 - 2e_0x_P^3 + e_0^2x_P^2 \end{pmatrix}, C_2 = \begin{pmatrix} 0 \\ -2 \\ 6x_P - \tilde{e}_i \\ 2\tilde{e}_i x_P + 2a \\ -\tilde{e}_i x_P^2 + 2ax_P + 4b \end{pmatrix},$$

$$C_3 = \begin{pmatrix} 0 \\ -2 \\ 6x_P - 2e_0 \\ -6x_P^2 + 4e_0x_P \\ 2x_P^3 - 2e_0x_P^2 \end{pmatrix}, \qquad C_4 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ -2x_p \\ x_P^2 \end{pmatrix}.$$

After tedious calculation of determinants of $\mathbf{M}_i$, we get $\det(\mathbf{M}_i) = -64(x_P^6 + 2ax_P^4 + 2bx_P^3 + a^2x_P^2 + 2abx_P + b^2) \mod p$, which is independent of the value i.

Based on $y_P^2 = x_P^3 + ax_P + b \bmod p$, $\det(\mathbf{M}_i) = -64y_P^4 \bmod p$ for all $1 \le i \le n$. Since p is an odd prime, $\mathbf{M}_i$ is invertible over $\mathbb{F}_p$ if and only if $y_P \ne 0 \bmod p$.

Note that $d_0 = e_0 - e_0'$. In order to compute the probability that $(e_0', \tilde{e}_1', \cdots, \tilde{e}_n') \ne (e_0, \tilde{e}_1, \cdots, \tilde{e}_n)$, we define the following two events:

$$(\text{E1}): y_P = 0 \bmod p; \quad (\text{E2}): y_P \ne 0 \bmod p \text{ and } d_0 \ne 0.$$

If (E1) holds,then $y_P = 0 \bmod p$. According to $y_P^2 = x_P^3 + ax_P + b \bmod p$, we have $x_P^3 + ax_P + b = 0 \bmod p$, which has at most 3 values for $x_P \in \mathbb{F}_p$. Note that the hidden point $P \in \mathcal{E}$ is chosen uniformly and randomly. Therefore, the probability that the event (E1) holds is no more than $\frac{3}{\#\mathcal{E}} \le \frac{3}{p+1-2\sqrt{p}}$, where $\#\mathcal{E}$ is the number of points on the curve $\mathcal{E}$ satisfying $p+1-2\sqrt{p} \le \#\mathcal{E} \le p+1+2\sqrt{p}$ by Hasse's theorem.

If (E2) holds, then $y_P \ne 0 \bmod p$. Hence, the matrices $\mathbf{M}_i$ are invertible for all $1 \le i \le n$. For any fixed $d_0 \ne 0$, we get that $(d_0, d_{0,0}, d_i, d_{0,i}, d_{00,i})$ are nonzero vectors for all $1 \le i \le n$. According to (48), all polynomials in (47) are non-constant zero polynomials, where the degree of x_{Q_i} is at most 4. In order to compute the probability that the event (E2) happens, we consider the number of n-tuples $(x_{Q_1}, \cdots, x_{Q_n}) \in (\mathcal{E}_x \setminus \{x_P\})^n$ such that (47) holds for all the cases of $d_0 \ne 0$, where $\mathcal{E}_x := \{z \in \mathbb{F}_p \mid \exists Q \in \mathcal{E}, x_Q = z\}$. It is worth noting that $\#\mathcal{E} - 1 \le 2|\mathcal{E}_x|$, and $x_{Q_1}, \cdots, x_{Q_n}$ are different in $\mathbb{F}_p$ according to Section 2.4. From $d_0 = e_0 - e_0' \ne 0$, where e_0 is a fixed integer, we consider the following situations for e_0'.

1). If $D_i + E_i e_0' + e_0'^2 \ne 0 \bmod p$ for all $1 \le i \le n$, then we get $\widetilde{e}_i' = -(A_i + B_i e_0' + C_i e_0'^2)(D_i + E_i e_0' + e_0'^2)^{-1} \bmod p$ based on $F_i(e_0', \tilde{e}_i') = 0 \pmod p$. It implies that the n-tuple $(\widetilde{e}_1', \cdots, \widetilde{e}_n')$ is uniquely determined by e_0'. In other words, the vectors $(d_0, d_{0,0}, d_i, d_{0,i}, d_{00,i})$ for all $1 \le i \le n$ are uniquely determined by d_0. For the convenience of discussion, let $Y := p^{1-\frac{1}{d+1}-\varepsilon}$. Since $d_0 = e_0 - e_0' \ne 0$ and $-Y < e_0, e_0' < Y$, we have $-2Y < d_0 < 2Y$. Thus d_0 can take at most $4Y - 1$ values. In this situation, there are no more than $(4Y-1) \cdot 4^n$ tuples $(x_{Q_1}, \cdots, x_{Q_n})$ such that the event (E2) happens. Thus the probability that the event (E2) holds does not exceed

$$\frac{(4Y-1)\cdot 4^n}{|\mathcal{E}_x \setminus \{x_P\}| \cdot (|\mathcal{E}_x \setminus \{x_P\}|-1) \cdots (|\mathcal{E}_x \setminus \{x_P\}|-n+1)} < \frac{2^{2n+2}Y}{(\frac{\#\mathcal{E}-1}{2}-n)^n} \le \frac{2^{3n+2} \cdot p^{1-\frac{1}{d+1}-o(\frac{1}{d})}}{(p-2\sqrt{p}-2n)^n}.$$

2). If $D_i + E_i e_0' + e_0'^2 = 0 \bmod p$ for some $1 \le i \le n$, then we deduce $A_i + B_i e_0' + C_i e_0'^2 = 0 \pmod p$ according to $F_i(e_0', \tilde{e}_i') = 0 \pmod p$. After plugging the expressions of A_i, B_i, C_i, D_i, E_i in (9) into $D_i + E_i e_0' + e_0'^2 = 0 \bmod p$ and $A_i + B_i e_0' + C_i e_0'^2 = 0 \pmod p$, we omit the involved calculation process and directly present the following relations:

$$x_{Q_i} = h_0 + e_0' \bmod p \text{ and } x_{Q_i}^3 + ax_{Q_i} + b = 0 \bmod p.$$

Rewriting them as an equation in e_0', we get $(h_0 + e_0')^3 + a(h_0 + e_0') + b = 0 \bmod p$. This equation has at most 3 values for e_0'. According to $x_{Q_i} = h_0 + e_0' \bmod p$, there are no more than 3 values for x_{Q_i}. Note that $x_{Q_1}, \cdots, x_{Q_n}$ are different in $\mathbb{F}_p$. Hence, $x_{Q_j} \neq x_{Q_i} = h_0 + e'_0 \bmod p$ for all $j \neq i$. Further, we deduce that

$$D_j + E_j e'_0 + e'^2_0 = (h_0 - x_{Q_j} + e'_0)^2 \neq 0 \bmod p \text{ for } j = 1, \cdots, i-1, i+1, \cdots, n.$$

Then $\widetilde{e}'_j = -(A_j + B_j e'_0 + C_j e'^2_0)(D_j + E_j e'_0 + e'^2_0)^{-1} \bmod p$ from $F_j(e'_0, \tilde{e}'_j) = 0 \pmod p$. Therefore, the $(n-1)$-tuple $(\widetilde{e}'_1, \cdots, \widetilde{e}'_{i-1}, \widetilde{e}'_{i+1}, \cdots, \widetilde{e}'_n)$ is uniquely determined by e'_0. In other words, the vectors $(d_0, d_{0,0}, d_j, d_{0,j}, d_{00,j})$ for all $j \neq i$ are uniquely determined by d_0. Since $d_0 = e_0 - e'_0$ and e'_0 can take at most 3 values, we get that d_0 takes no more than 3 values. Thus, there are no more than 3 values for x_{Q_i}, and at most $3 \cdot 4^{n-1}$ values for the $(n-1)$-tuple $(x_{Q_1}, \cdots, x_{Q_{i-1}}, x_{Q_{i+1}}, x_{Q_n})$ such that the event (E2) happens. In other words, the probability that the event (E2) holds is at most

$$\frac{3\cdot(3\cdot 4^{n-1})}{|\mathcal{E}_x\backslash\{x_P\}|\cdot(|\mathcal{E}_x\backslash\{x_P\}|-1)\cdots(|\mathcal{E}_x\backslash\{x_P\}|-n+1)} < \frac{3^2 4^{n-1}}{(\frac{\#\mathcal{E}-1}{2}-n)^n} \leq \frac{3^2 2^{3n-2}}{(p-2\sqrt{p}-2n)^n}.$$

So the probability that $(e'_0, \tilde{e}'_1, \cdots, \tilde{e}'_n) \neq (e_0, \tilde{e}_1, \cdots, \tilde{e}_n)$ does not exceed $\frac{3}{p-2\sqrt{p}}$ $+\frac{2^{3n+2}\cdot p^{1-\frac{1}{d+1}-\varepsilon}}{(p-2\sqrt{p}-2n)^n} + \frac{3^2 2^{3n-2}}{(p-2\sqrt{p}-2n)^n}$. It becomes $\mathcal{O}(\frac{1}{p})$ for a sufficiently large $p = 2^{\omega(d^{(2+c)d})}$ and $n = d^{3+c}$ for any constant $c > 0$. $\square$