Provable Recovery from Integer Quotients: Improved Bounds for Nonlinear Hidden Number Problems

Taehun Kang Changmin Lee*

Korea University
Seoul, Republic of Korea

kth9999@korea.ac.kr changminlee@korea.ac.kr

Abstract

The hidden number problem (HNP) asks whether a secret can be recovered from partial information about values linked to it by known algebraic relations. In cryptosystems where partial information about secret values or internal randomness is leaked across various scenarios, HNP techniques provide a foundational framework for security analysis and key-recovery attacks. For some HNPs with nonlinear relations, provable polynomial-time recovery guarantees have long remained unimproved, requiring a substantial fraction of the leading bits.

We extend the provable recovery ranges for the modular inversion HNP (MIHNP), elliptic curve HNP (EC-HNP), and HNP with hidden multipliers (HNPHM), all modulo a prime p. For any sufficiently small constant $\varepsilon > 0$, we increase the tolerable fraction of unrevealed lower bits: for MIHNP, from $1/3 - \varepsilon$ to $1/2 - \varepsilon$; for EC-HNP, from $1/6 - \varepsilon$ to $1/5 - \varepsilon$; and for HNPHM, from $1/5 - \varepsilon$ to $1/3 - \varepsilon$.

Our algorithm constructs polynomials that vanish modulo p at the unrevealed lower bits, and reduces recovery to bounding their possible integer quotients by p over the candidate range. While such a search can branch in general, we exploit the algebraic relations of each problem to prove that the expected search cost remains polynomial in $\log p$ over randomly sampled inputs. On every valid input, the algorithm returns the exact set of consistent secrets, achieving unique recovery with probability $1 - o(1)$ as $p \to \infty$ for every fixed secret.

*Corresponding author.

1 Introduction

The hidden number problem (HNP) asks whether a secret y^* can be recovered from partial information about values $z_1^*, \ldots, z_k^*$ algebraically related to y^*. In cryptosystems where partial information about secret values or internal randomness is leaked across various scenarios, HNP techniques provide a foundational framework for security analysis and key-recovery attacks [BV96, BH19, BHHG01, HGNS03, Sha17]. Originally introduced by [BV96] in the linear setting, the framework has become a central tool for assessing vulnerabilities under partial leakage.

In this work, we study three nonlinear variants of the HNP, aiming to recover their hidden secrets efficiently. We briefly introduce these problems below. Throughout, p denotes a prime, and $\mathbb{Z}_p = \{0, \ldots, p-1\}$ denotes the set of integers modulo p.

- **Modular inversions (MIHNP) [BHHG01].** For a secret $y^* \in \mathbb{Z}_p$ and distinct public shifts $t_1, \ldots, t_k \in \mathbb{Z}_p$ with $y^* + t_i \not\equiv 0 \pmod{p}$, we observe leading bits of $z_i^* = (y^* + t_i)^{-1} \bmod p$ for $1 \le i \le k$. MIHNP arises in the security analysis of pseudorandom generators and message authentication codes.

- **Elliptic curves (EC-HNP) [Sha17].** For a secret point y^* on a public elliptic curve $\mathcal{E}$ defined modulo p and public query points $Q_1, \ldots, Q_k \in \mathcal{E}$, we observe leading bits of the x-coordinate z_i^* of $y^* + Q_i$ whenever $y^* + Q_i$ is finite for $1 \le i \le k$ (and infinity otherwise). An infinity observation directly determines $y^* = -Q_i$. EC-HNP arises in studying the bit security of elliptic-curve Diffie–Hellman keys.

- **Hidden multipliers (HNPHM) [HGNS03].** For a secret $y^* \in \mathbb{Z}_p \setminus \{0\}$ and hidden multipliers $t_1^*, \ldots, t_N^* \in \mathbb{Z}_p \setminus \{0\}$, we observe leading bits of $z_1^* = y^*$, $z_{i+1}^* = t_i^*$, and $z_{N+i+1}^* = y^* t_i^* \bmod p$ for $1 \le i \le N$ (comprising $k = 2N + 1$ values). HNPHM arises in the analysis of timed-release cryptography and noisy exponentiation.

All three nonlinear HNPs can be formulated in terms of bounded integer solutions to polynomial congruences modulo p. For $1 \le j \le k$, let v_j be obtained by setting the unrevealed lower bits of $z_j^* \in \mathbb{Z}_p$ to zero. Then $z_j^* = v_j + x_j^*$, where the error x_j^* is the integer represented by those bits and satisfies $0 \le x_j^* < X$ for a common integer bound $1 \le X < p$. The algebraic relations defining each problem can then be used to derive polynomial congruences in these bounded unknowns. A larger X corresponds to fewer revealed bits.

Recovering hidden secrets from such bounded modular relations has been studied through both heuristic and provable approaches. Heuristic algorithms have achieved improved recovery ranges for these nonlinear HNPs [XSH+19, XSWH26, Sar17], while rigorous polynomial-time recovery guarantees have been established in [LSSW12, Sha17, HGNS03].

In this work, we focus on provable polynomial-time recovery. Existing provable algorithms for these problems rely on lattice-based techniques, using approximate shortest or closest vector algorithms to recover the hidden secret [LSSW12, Sha17, HGNS03]. Their analyses establish recovery guarantees by showing that the lattice output contains sufficient information to identify the correct solution, typically by bounding the probability of competing lattice vectors.

As X grows, ruling out competing lattice vectors becomes more difficult, limiting the provable recovery ranges. However, the lattice output need not be viewed in an all-or-nothing manner: even when it does not directly determine y^*, it may still constrain x_j^*. We use these constraints to guide a complete search for the secrets satisfying the original input constraints. For every fixed y^*, we bound the total expected bit cost of this search over the sampled inputs and any internal randomness. This gives polynomial expected-time recovery for larger X under the stated sampling models.

1.1 Main Results

We improve the provable polynomial-time recovery ranges of MIHNP, EC-HNP, and HNPHM, as summarized in Table 1.

	α		k		m	
Problem	Prior	**This work**	Prior	**This work**	Prior	**This work**
MIHNP [LSSW12]	1/3	**1/2**	$O(\varepsilon^{-1})$	$O(\varepsilon^{-2})$	$O(\varepsilon^{-1})$	$O(\varepsilon^{-4})$
EC-HNP [Sha17]	1/6	**1/5**	$O(\varepsilon^{-1})$	$O(\varepsilon^{-1})$	$O(\varepsilon^{-1})$	$O(\varepsilon^{-2})$
HNPHM [HGNS03]	1/5	**1/3**	$O(\varepsilon^{-1})$	$O(\varepsilon^{-1})$	$O(\varepsilon^{-1})$	$O(\varepsilon^{-1})$

Table 1: Sufficient error bounds $X \le p^{\alpha-\varepsilon}$ under the respective sampling models, for fixed sufficiently small $\varepsilon > 0$ and sufficiently large p. Larger α permits fewer revealed bits; k counts partially known values and m is the lattice rank. For EC-HNP, both bounds apply to query subgroups of size at least $p/\operatorname{polylog} p$; see [Sha17] and Theorem 1.2.

Theorem 1.1 (Main result, informal)**.** *Fix a sufficiently small constant $\varepsilon > 0$. For each of MIHNP, EC-HNP, and HNPHM, there are choices of k and m satisfying the corresponding bounds in Table 1 for which our algorithm has the following guarantees.*

1. *On every valid input, the algorithm terminates and returns exactly the set of secrets satisfying the original input constraints, with probability one over any internal randomness. This set contains y^*.*
2. *For every fixed y^*, under the sampling models and parameter conditions in Sections 4 to 6, the condition $X \le p^{\alpha-\varepsilon}$ with the corresponding "This work" exponent α gives expected bit running time polynomial in $\log p$. The output equals $\{y^*\}$ with probability $1 - o(1)$ as $p \to \infty$.*

The expectation and probability in the second item include both the sampled inputs and any internal reconstruction randomness.

Comparison with prior results. Prior algorithms run in polynomial time on every execution and recover y^* with probability $1 - o(1)$. Our algorithm returns the exact set of consistent secrets on every valid input, with probability one. Its bit cost is polynomial in expectation under our stated sampling and parameter conditions.

To bound the running time of every execution, we stop after $\log p$ times the proved polynomial upper bound on expected bit cost, reporting failure if this limit is reached. Markov's inequality bounds this additional failure probability by $1/\log p$. Thus every execution takes time polynomial in $\log p$, and the output equals $\{y^*\}$ with probability $1 - o(1)$.

For MIHNP, we use the independent uniform shift model of [LSSW12], conditioned on distinct shifts. For HNPHM, we sample nonzero multipliers independently and uniformly, as in the prime-modulus case of [HGNS03]. For EC-HNP, [Sha17] uses a central query and opposite-shift pairs. Using the same query oracle, we instead sample queries independently and uniformly from the public subgroup, rejecting tuples that violate the public conditions. Our EC-HNP expected-cost and uniqueness bounds apply to the resulting distribution of accepted tuples.

1.2 Technical Overview

Our main technical contribution is the problem-specific counting that proves polynomial expected bit cost for the complete search in the enlarged ranges of X. Our approach has three components.

1. **A Common Recovery Framework (Sections 3.1 and 3.2).** Across all three problems, knowing x_1^* gives at most two candidates for the hidden secret y^*. We express x_1^* as an integer linear combination of integer quotients of lattice-reduced polynomial values by p. Bounding these quotients on progressively smaller regions of the error space guides a complete search.

2. **Dual lattice cost analysis (Section 3.3).** We use the dual lattice to bound the total expected bit cost of the search for each fixed input. The analysis controls the variation of the reduced polynomials needed to recover x_1^*, thereby bounding the total number of regions visited by the search.

3. **Problem-specific counting (Sections 4 to 6).** We use each problem's algebraic relations and sampling model to bound how often this search-cost bound is large. The resulting tail bounds yield polynomial expected bit cost in the enlarged ranges of X.

1.2.1 A Common Recovery Framework

In this part, we show the common recovery framework applies to all three problems. We explain the framework using MIHNP as a running example. The subsequent analyses use each problem's algebraic relations and sampling model to bound the expected search cost.

For MIHNP with N public shifts, put $k = N$ and write $z_i^* = (y^* + t_i)^{-1} \bmod p = v_i + x_i^*$ and $0 \le x_i^* < X$ for $1 \le i \le N$. Thus $\mathbf{x}^* = (x_1^*, \ldots, x_N^*) \in \mathcal{B} = \prod_{i=1}^{N} \mathcal{B}_i$, where $\mathcal{B}_i = \{0, \ldots, X-1\}$. More generally, the common search uses a known integer box $\mathcal{B} = \prod_{i=1}^{k} \mathcal{B}_i \subseteq [-X, X]^k \cap \mathbb{Z}^k$, allowing negative errors for more general bounded approximations. We write $\mathbf{v} = (v_1, \ldots, v_k)$ for the approximation vector and $\mathbf{x} = (x_1, \ldots, x_k)$ for a candidate error vector in $\mathcal{B}$; $\mathbf{x}^*$ denotes the true error vector.

Recovering y^* from x_1^*. In MIHNP, knowing x_1^* determines $z_1^* = v_1 + x_1^*$ and hence $y^* \equiv (v_1 + x_1^*)^{-1} - t_1 \pmod p$. For a proposed value $a \in \mathcal{B}_1$ with $v_1 + a \not\equiv 0 \pmod p$, we substitute a for x_1^* in the displayed formula and check the resulting candidate secret against all original input constraints. If $v_1 + a \equiv 0 \pmod p$, we reject a. Thus we can test a without first recovering the other errors.

For any of the three problems, let $\mathcal{R}(a)$ be the set of secrets $\widehat{y}$ reconstructed from $x_1 = a$ that satisfy all original input constraints, including any supplied leading-bit intervals. For each problem and fixed k, $\mathcal{R}(a)$ can be computed in expected bit time polynomial in $\log p$. In MIHNP this verification is deterministic and returns at most one secret. In HNPHM, x_1 likewise determines the secret directly. In EC-HNP, $x_1 = a$ fixes the first coordinate of $\widehat{y} + Q_1$, giving at most two curve points and hence at most two candidate secrets. A direct scan makes $|\mathcal{B}_1|$ calls to $\mathcal{R}$. For MIHNP, this means X calls, exponentially many in $\log p$ when X is a fixed positive power of p. We use polynomial relations to restrict the values of a that need to be tested.

From polynomial relations to integer quotients. We call $\mathbf{x} \in \mathcal{B}$ consistent with the input if $(y, \mathbf{x})$ satisfies the original input constraints for some secret y. For MIHNP, eliminating y from the inverse constraints at indices i and j gives the quadratic relations [XSH+19]

$$P_{ij}(\mathbf{x}) = (v_i + x_i)(v_j + x_j) + a_{ij}(v_i + x_i - v_j - x_j) \equiv 0 \pmod p, \qquad 1 \le i < j \le N,$$

where $a_{ij} \in (-p/2, p/2] \cap \mathbb{Z}$ satisfies $a_{ij}(t_i - t_j) \equiv 1 \pmod{p}$. These inverses exist because the shifts are distinct. List these P_{ij}, followed by $px_2, \ldots, px_N, px_1$, as $P_1, \ldots, P_m$, with $m = N(N+1)/2$. All vanish modulo p at every consistent $\mathbf{x}$. The appended polynomials impose no additional modular restriction; including $P_m = px_1$ lets us express px_1 through any integer basis obtained from the P_i. The other problems use their corresponding quadratic polynomial families, also including px_1.

Let $\Phi(f)$ record the nonconstant coefficients of f in a fixed monomial order, with each degree-t coefficient multiplied by X^t. The weights reflect the possible sizes of the monomials: for $\mathbf{x} \in \mathcal{B}$, $|f(\mathbf{x}) - f(\mathbf{0})| \le \|\Phi(f)\|_1$. Thus a short coefficient vector gives a small range of polynomial values on $\mathcal{B}$. In the MIHNP family, each P_{ij} has a distinct quadratic monomial $x_i x_j$, and the appended px_i have independent linear coefficient vectors. Hence $\Phi(P_1), \ldots, \Phi(P_m)$ are linearly independent and form a basis of the lattice Λ generated by their integer linear combinations. The corresponding families for EC-HNP and HNPHM also have independent scaled coefficient vectors.

LLL [LLL82], a polynomial-time lattice basis reduction algorithm, replaces this basis with one of controlled vector lengths using invertible integer row operations. Applying the same operations to the full polynomials gives $h_1, \ldots, h_m$, which still vanish modulo p at every consistent $\mathbf{x}$. Since $P_m = px_1$, the inverse basis change gives computable integers α_j such that

$$px_1 = \sum_{j=1}^{m} \alpha_j h_j, \qquad q_j^* = \frac{h_j(\mathbf{x}^*)}{p} \in \mathbb{Z}, \qquad x_1^* = \sum_{j=1}^{m} \alpha_j q_j^*.$$

The integer quotients q_j^* need not be zero. Bounding their possible values therefore restricts x_1^* even when the reduced polynomial values are nonzero. The quotient bounds on $\mathcal{B}$ may still permit many integers; we tighten them by subdividing $\mathcal{B}$.

Narrowing the quotients on common cells. All quotients q_j^* are evaluated at the same unknown vector $\mathbf{x}^*$. We therefore subdivide the k-dimensional error box rather than enumerate m quotients independently. On each smaller box $C = \prod_i C_i$, called a cell, we bound each h_j with $\alpha_j \neq 0$. We compute an interval enclosure by bounding each monomial term on C and adding the resulting intervals. The constant term is retained as an offset, although it does not affect the interval width. At any consistent $\mathbf{x}$, $h_j(\mathbf{x})$ is a multiple of p. An interval bounding its values on C with width less than p contains at most one such multiple. Dividing that interval by p therefore leaves at most one possible integer quotient. For example, if $p = 7$ and $10 \le h_j(\mathbf{x}) \le 16$ on C, the only possible multiple is 14, so every consistent point in C has quotient 2.

For each j with $\alpha_j \neq 0$, we intersect the quotient interval with $\mathbb{Z}$. If any intersection is empty, we discard C. Otherwise, taking the linear combination of these integer quotient ranges with coefficients α_j gives an interval containing x_1 for every consistent point in C. We intersect this interval with C_1. If the intersection is empty, we discard C. If it is $\{a\}$, we compute $\mathcal{R}(a)$. Otherwise, we subdivide further. After $|\mathcal{B}_1|$ cell visits, any unfinished search is replaced by a direct scan of the untested values of $a \in \mathcal{B}_1$.

The interval tests preserve every consistent error vector until its first coordinate is tested. For each tested a, $\mathcal{R}(a)$ checks the original input constraints. Thus the search returns exactly the secrets satisfying the input, with probability one over reconstruction randomness.

1.2.2 Dual lattice cost analysis

We use the dual lattice to bound the search cost on each fixed input and reduce the average-cost analysis to a counting problem. We continue with MIHNP, where $k = N$ and $|\mathcal{B}_1| = X$, keeping N fixed as p grows.

Bounding the subdivision tree by coefficient norms. Fix an input I. Only the polynomials h_j with $\alpha_j \neq 0$ enter the identity for x_1, so their quotient ranges determine how far the search needs to subdivide. For an integer $b \geq 1$, partition each $\mathcal{B}_i$ into $\min\{b, |\mathcal{B}_i|\}$ consecutive integer intervals whose cardinalities differ by at most one. On each resulting cell, h_j has an interval enclosure of width at most $4\|\Phi(h_j)\|_1/b$. This follows by bounding the variation of each linear or quadratic monomial on the cell. Consequently, choosing an integer

$$b > \frac{4}{p} \max_{\alpha_j \neq 0} \|\Phi(h_j)\|_1$$

makes each required quotient range contain at most one integer. At this resolution, every remaining cell is either discarded or gives a single value of x_1 to test. The search uses binary splits along boundaries of this final partition. The fully expanded tree has at most b^N leaves and hence at most $2b^N - 1$ nodes. Discarding a cell or stopping its subdivision early can only reduce the number of visits. Thus the relevant coefficient norms control all cell visits, including intermediate and discarded cells.

Bounding the relevant coefficient norms via the dual lattice. Write $\mathbf{b}_j = \Phi(h_j)$ for the reduced basis of Λ. The dual lattice Λ^* consists of vectors in $\mathrm{span}_{\mathbb{R}}(\Lambda)$ whose inner products with all vectors of Λ are integers. Let $(\mathbf{b}_j^\vee)$ be the basis of Λ^* dual to $(\mathbf{b}_i)$, so that $\langle \mathbf{b}_i, \mathbf{b}_j^\vee \rangle = 1$ when $i = j$ and 0 otherwise. For the target vector $\mathbf{v}_{\mathrm{tar}} = \Phi(px_1) = \sum_i \alpha_i \mathbf{b}_i$, we therefore have $\langle \mathbf{v}_{\mathrm{tar}}, \mathbf{b}_j^\vee \rangle = \alpha_j$. This identifies the dual basis vectors corresponding to the polynomials used in recovering x_1.

A standard consequence of LLL reduction gives a constant $C_m > 0$, depending only on the lattice rank m, such that $\|\mathbf{b}_j\|_2 \|\mathbf{b}_j^\vee\|_2 \leq C_m$ [LLL82]. To apply this bound to the relevant indices, define

$$\lambda_{\mathrm{target}} = \min\left\{\|\mathbf{w}\|_2 : \mathbf{w} \in \Lambda^*, \quad \langle \mathbf{v}_{\mathrm{tar}}, \mathbf{w} \rangle \neq 0\right\}, \qquad Z_{\mathrm{target}} = \max\{1, (p\lambda_{\mathrm{target}})^{-1}\}.$$

For every j with $\alpha_j \neq 0$, $\|\mathbf{b}_j^\vee\|_2 \geq \lambda_{\mathrm{target}}$. The LLL bound and the comparison of norms in fixed dimensions then give

$$\frac{\|\Phi(h_j)\|_1}{p} \leq \frac{O_N(1)}{p\lambda_{\mathrm{target}}} = O_N(Z_{\mathrm{target}}) \qquad (\alpha_j \neq 0).$$

The basis-length choice of b above can therefore be made with $b = O_N(Z_{\mathrm{target}})$. The algorithm uses the computed basis lengths to choose its subdivision; $\lambda_{\mathrm{target}}$ and Z_{target} are used only in the analysis. The unrestricted dual minimum can be small because of vectors orthogonal to $\mathbf{v}_{\mathrm{tar}}$; our bound does not depend on their lengths.

Each cell visit, including any verification call, takes polynomial bit time for fixed N. Combining the cell bound with the direct-scan cap, Theorem 3.4 gives the following bound on the total bit cost $T(I)$:

$$T(I) \leq \mathrm{poly}_N(\log p)\left[1 + \min\{X, Z_{\mathrm{target}}(I)^N\}\right].$$

Here I is fixed, and the MIHNP search and verification are deterministic.

From per-input bounds to average cost. For the average-cost analysis, fix y^* and sample the shifts uniformly without replacement from the residues t satisfying $y^* + t \not\equiv 0 \pmod{p}$. Set $v_i = X\lfloor z_i^*/X \rfloor$; when X is a power of two, these are the leading-bit approximations. The sampled shifts thus determine the input I. Through this sampled input, Z_{target} becomes a random variable. It remains to bound the expectation of $\min\{X, Z_{\mathrm{target}}^N\}$. The standard tail-integral identity gives

$$\mathbb{E}\left[\min\{X, Z_{\mathrm{target}}^N\}\right] = 1 + N \int_1^{X^{1/N}} u^{N-1} \Pr[Z_{\mathrm{target}} > u]\, du.$$

The expectation and probability are over the sampled inputs, with y^* fixed. The upper limit comes from the direct-scan cap: once Z_{target}^N exceeds X, the minimum no longer increases. Thus only the tail for $1 \le u < X^{1/N}$ is needed. For example, a bound $\Pr[Z_{\text{target}} > u] \le A(p)u^{-N}$ on this range gives an expectation at most $1 + A(p)\log X$. A fixed tail exponent larger than N instead gives a bounded integral after factoring out its prefactor. Either estimate suffices when the prefactor is polynomial in $\log p$.

For the other problems, the same argument uses k in place of N and the direct-scan cap $|\mathcal{B}_1| \le 2X + 1$. When reconstruction is randomized, the fixed-input cost bound is an expectation over that randomness, which is then averaged over the sampled inputs.

From short dual vectors to algebraic constraints. For $u \ge 1$, the definition of Z_{target} gives

$$Z_{\text{target}} > u \quad \Longleftrightarrow \quad \exists\, \mathbf{w} \in \Lambda^* :\ \|\mathbf{w}\|_2 < \frac{1}{pu}, \quad \langle \mathbf{v}_{\text{tar}}, \mathbf{w} \rangle \neq 0.$$

We bound this tail by counting sampled inputs whose dual lattice contains such a vector. In MIHNP, $\Phi(px_i) \in \Lambda$ for every $1 \le i \le N$, so such a vector gives

$$s_i = \langle \Phi(px_i), \mathbf{w} \rangle \in \mathbb{Z}, \qquad |s_i| < X/u, \qquad s_1 \neq 0.$$

The dual lattice condition gives integrality, while $\|\Phi(px_i)\|_2 = pX$ gives the size bound. The nonzero pairing with $\mathbf{v}_{\text{tar}}$ ensures $s_1 \neq 0$. For a monomial M recorded by Φ, the same norm estimate gives $|p\langle \Phi(M), \mathbf{w} \rangle| < X^{\deg M}/u$. The MIHNP analysis below combines these pairings with the all-pairs quadratic relations to obtain congruences with bounded integer remainders. For each fixed choice of (s_i) and the bounded integer remainders in these congruences, we count the permitted inverse values. Summing over the integer choices then bounds the tail probability.

1.2.3 MIHNP

Fix $0 < \varepsilon < 1/2$ and suppose $X \le p^{1/2-\varepsilon}$. For sufficiently large p, the all-pairs inverse relations give $\Pr[Z_{\text{target}} > 1] \ll_N 1/X$ for a choice $N = O(\varepsilon^{-2})$. The direct-scan cap will then give polynomial expected bit cost. Under the sampling model above, the inverse values z_i^* are uniform over ordered tuples of distinct nonzero residues: $t \mapsto (y^* + t)^{-1}$ is a bijection from the valid shifts onto $\mathbb{Z}_p \setminus \{0\}$.

Dual constraints from inverse relations. Consider a vector $\mathbf{w}$ witnessing $Z_{\text{target}} > 1$, and use the integers s_i from the preceding analysis, so $|s_i| < X$ and $s_1 \neq 0$. The inverse relations give $a_{ij} \equiv z_i^* z_j^*/(z_j^* - z_i^*) \pmod p$. Moreover, multiplying P_{ij} by p and subtracting linear lattice vectors shows that $\Phi(px_ix_j) \in \Lambda$. Thus $p\langle \Phi(x_ix_j), \mathbf{w} \rangle$ is an integer of magnitude less than X^2. Substituting $v_i = z_i^* - x_i^*$ in the pairings with $\Phi(P_{ij})$ gives

$$\frac{s_i(z_j^*)^2 - s_j(z_i^*)^2}{z_j^* - z_i^*} \equiv r_{ij} \pmod p, \qquad r_{ij} \in \mathbb{Z}, \quad |r_{ij}| \le 3X^2 \quad (i < j).$$

Here r_{ij} combines the quadratic pairing with the two integer products $s_ix_j^*$ and $s_jx_i^*$. The denominators are nonzero because the inverse values are distinct. For fixed (s_i), these congruences restrict rational functions of each inverse value to short intervals of residues.

Simultaneous restrictions on the sampled values. Fix the entire integer tuple (s_i) before exposing the sampled inverse values. Suppose j inverse values with distinct nonzero coefficients s_i have already been fixed. For the next value $z = z_\ell^*$, each fixed index i gives a condition on the rational function

$$f_i(z) = \frac{s_i z^2 - s_\ell (z_i^*)^2}{z - z_i^*} = s_i(z + z_i^*) + \frac{(s_i - s_\ell)(z_i^*)^2}{z - z_i^*}.$$

To apply the exponential-sum bound, we verify that every nonzero linear combination of these functions is nonconstant. The selected coefficients remain distinct and nonzero modulo p, since $2X < p$ for sufficiently large p. If $s_i \neq s_\ell$, the displayed decomposition shows a simple pole at z_i^* that the other functions cannot cancel. At most one selected coefficient equals s_ℓ; its function is $s_i(z + z_i^*)$, which is nonconstant because $s_i \neq 0$ modulo p. Thus every nonzero linear combination is nonconstant.

Using the rational-function exponential-sum bound of [CP06], the simultaneous counting estimate in Section 4 bounds the conditional probability of all j conditions by $O_N(E_j)$, where

$$E_j = \left(\frac{X^2}{p}\right)^j + p^{-1/2}.$$

The first term is the product of the allowed residue fractions; the $p^{-1/2}$ term is the error term in the counting estimate. This estimate already includes all permitted r_{ij} and is uniform over the previously fixed inverse values. We can therefore multiply the conditional bounds without assuming independence of the conditions.

Repeated and zero coefficients. We now sum these conditional bounds over the admissible coefficient tuples. If the s_i have κ distinct nonzero values, each fixed pattern of equal and zero entries permits $O_N(X^\kappa)$ coefficient tuples. Choose one representative index for each distinct nonzero coefficient value. Expose the inverse values at these κ indices first. In this order, they face $0, 1, \ldots, \kappa - 1$ conditions, respectively. Each of the remaining $N - \kappa$ inverse values then faces κ conditions. Summing over coefficient tuples and patterns gives

$$\Pr[Z_{\text{target}} > 1] \ll_N \sum_{\kappa=1}^{N} X^\kappa \left(\prod_{j=1}^{\kappa-1} E_j\right) E_\kappa^{N-\kappa}.$$

This count accounts for the loss of conditions when coefficients repeat or vanish through the smaller number of distinct integers that must be chosen.

Since $X^2/p \leq p^{-2\varepsilon}$, after $O(\varepsilon^{-1})$ simultaneous conditions we have $E_j = O(p^{-1/2})$. For $\kappa = N$, the $O(X)$ choices for each subsequent s_i combine with this probability factor to give $O(X/\sqrt{p}) = O(p^{-\varepsilon})$. This explains the limiting exponent $1/2$.

Expected cost and uniqueness. The direct-scan cap gives

$$\mathbb{E}\Big[\min\{X, Z_{\text{target}}^N\}\Big] \leq 1 + (X - 1)\Pr[Z_{\text{target}} > 1].$$

It therefore suffices to show $X \Pr[Z_{\text{target}} > 1] = O_N(1)$. We bound X times each summand of the grouped estimate above. Put $J = \lceil (4\varepsilon)^{-1} \rceil$. When $\kappa < J$, the repeated factor $E_\kappa^{N-\kappa}$ provides the required decay. When $\kappa \geq J$, all but $O(\varepsilon^{-1})$ of the $N - 1$ probability factors are $O(p^{-1/2})$. Writing F_κ for the κth summand above, $E_j \leq 2p^{-\min\{2\varepsilon j, 1/2\}}$ gives $XF_\kappa \ll_N p^{\kappa[1-\varepsilon(2N-J)]}$ for $\kappa < J$, and

$XF_\kappa \ll_N p^{-\varepsilon N+(J+1)/2}$ for $\kappa \geq J$. Taking $N = \lceil \varepsilon^{-2} \rceil$ makes both exponents negative, yielding $X \Pr[Z_{\text{target}} > 1] = O_N(1)$.

Substituting this estimate into the expectation bound above and applying the per-input cost bound gives expected bit cost $\text{poly}_\varepsilon(\log p)$, uniformly for every fixed y^*. The full tail analysis in Section 4 gives more precise recovery bounds for fixed sample counts. Finally, fix a candidate $\widehat{y} \neq y^*$. Consistency allows only $O(X)$ differences between the inverse values $(\widehat{y} + t_i)^{-1} \bmod p$ and z_i^*. Each nonzero difference gives a quadratic equation in the shift, with at most two solutions. Sampling without replacement and taking a union bound over $\widehat{y}$ give an additional-secret probability $O_N(X^N/p^{N-1}) = o(1)$. Thus the search outputs $\{y^*\}$ with probability $1 - o(1)$.

1.2.4 EC-HNP

For EC-HNP, we give the quadratic relations needed by the common framework and a counting argument that bounds the expected search cost.

Setup. Let $p > 3$ be prime and let $\mathcal{E}$ be a public nonsingular curve $w^2 = \varphi(z) = z^3 + a_\mathcal{E} z + b_\mathcal{E}$ modulo p. Write $\mathcal{O}$ for its group identity and π_1 for the first coordinate of a finite point. The public parameters include a subgroup $G = \langle R_0 \rangle$ of known order M and integers $N \geq 5$ and $1 \leq X < p$. The queries $Q_1, \ldots, Q_N \in G$ satisfy $Q_i \neq \mathcal{O}$ and $Q_j - Q_i = (a_{ij}, b_{ij})$ finite with $b_{ij} \neq 0$ for $i \neq j$. For a hidden point $y^* \in \mathcal{E}$, put $W_i = y^* + Q_i$. At each finite W_i, the input gives an approximation v_i with $z_i^* = \pi_1(W_i) \equiv v_i + x_i^* \pmod p$, and $0 \leq x_i^* < X$. Leading-bit inputs also specify the original integer intervals, which are checked during reconstruction.

For fixed N, we assume the public input has bit length bounded by a fixed polynomial in $\log p$. When $W_i = \mathcal{O}$, the input reports infinity, which determines $y^* = -Q_i$; we verify this point directly. Otherwise, x_1 fixes the first coordinate of W_1, giving at most two curve points since the curve equation is quadratic in w. Subtracting Q_1 and checking all input constraints computes $\mathcal{R}(a)$ in expected polynomial bit time. On these finite inputs, $k = N$ and $\mathcal{B}_i = \{0, \ldots, X-1\}$.

For the average-cost analysis, fix y^* and sample the Q_i independently and uniformly from G, rejecting tuples that violate the public query conditions above. Infinity reports are retained. Before rejection, the W_i are independent and uniform on $y^* + G$, a set of M points. For $M \geq 16N^2$, public acceptance has constant probability. The approximations may depend jointly on all queries, subject to the stated error bounds and any supplied intervals. All input probabilities below concern the accepted tuples and a fixed permitted approximation rule.

Theorem 1.2 (EC-HNP recovery, informal)**.** *Fix $0 < \varepsilon < 1/5$ and $c \geq 0$. For a suitable $N = O(\varepsilon^{-1})$ and sufficiently large p, the conditions $M \geq \max\{16N^2, p/(\log p)^c\}$, and $X \leq p^{1/5-\varepsilon}$ give expected bit cost $\text{poly}_{\varepsilon,c}(\log p)$ and output $\{y^*\}$ with probability $1 - o(1)$. On every valid input, the algorithm returns exactly all consistent points with probability one. These guarantees hold uniformly for every fixed y^* and every permitted approximation rule; the expectation includes query sampling and any reconstruction randomness.*

Quadratic relations from the addition law. Building on [DDW+26], we use quadratic relations to keep the dual remainders on the scale X^2/u. For formal variables $z_1, \ldots, z_N$ and $i \neq j$, put $F_{ij} = z_i z_j - a_{ij}(z_i + z_j)$, using integer representatives of the a_{ij}. After substituting $z_h = \pi_1(y + Q_h)$, these become rational functions of the curve point y modulo p.

A pole of order r means that the expansion in a local coordinate t begins with ct^{-r} for $c \neq 0$. At $y = -Q_i$, z_i has a double pole while $z_j - a_{ij}$ vanishes, so $F_{ij} = z_i(z_j - a_{ij}) - a_{ij} z_j$ has at most a simple pole there. The same holds at $-Q_j$, and there are no other poles. Using Q_N as a common

reference, the addition law shows that the following combination cancels all remaining poles and is therefore constant:

$$F_{ij} - \frac{b_{ij}}{b_{iN}} F_{iN} + \frac{b_{ij}}{b_{jN}} F_{jN} \equiv \kappa_{ij} \pmod{p} \qquad (i < j < N).$$

The identity holds whenever all $z_h = \pi_1(y + Q_h)$ are finite. The constant κ_{ij} is publicly computable by evaluating at $y = \mathcal{O}$.

Take representatives in $\{0, \dots, p-1\}$ of the coefficients of the left-hand side minus κ_{ij}, then substitute $z_h = v_h + x_h$ to obtain integer polynomials P_{ij}. Append px_hx_N for $h < N$ and $px_2, \dots, px_N, px_1$. The x_ix_j terms for $i < j < N$ distinguish the P_{ij}, and the appended monomials give the remaining independent coefficient directions. Thus their scaled coefficient vectors form a basis of Λ of rank $m = N(N+1)/2$, with $P_m = px_1$. The appended linear polynomials give $\Phi(px_i) \in \Lambda$ for every i. Multiplying each P_{ij} by p and subtracting multiples of the appended polynomials also shows that $\Phi(px_ix_j) \in \Lambda$ for every $i < j$. Section 5 proves the identity and verifies these lattice properties.

Dual constraints with auxiliary residues. Set $Z_{\text{target}} = 1$ on inputs settled by the infinity check. For a finite input and $1 \le u < X^{1/N}$, let $\mathbf{w}$ witness $Z_{\text{target}} > u$. Put $s_i = \langle \Phi(px_i), \mathbf{w} \rangle \in \mathbb{Z}$. The norm bound gives $|s_i| < X/u$, and $s_1 \neq 0$ by the target condition. Since $X < p$, s_1 is also nonzero modulo p. With $\mathbf{v} = (v_1, \dots, v_N)$, define

$$A_{ij} = p\langle \Phi(F_{ij}(\mathbf{v} + \mathbf{x})), \mathbf{w} \rangle \in \mathbb{Z}, \qquad \theta_N = 0, \quad \theta_i \equiv A_{iN}/b_{iN} \pmod{p} \quad (i < N).$$

Since $\Phi(px_i), \Phi(px_ix_j) \in \Lambda$, each A_{ij} is an integer, and the three-term identities give $A_{ij} \equiv b_{ij}(\theta_i - \theta_j) \pmod{p}$. Substituting $v_h \equiv z_h^* - x_h^* \pmod{p}$ in these pairings gives

$$(z_j^* - a_{ij})s_i + (z_i^* - a_{ij})s_j - b_{ij}(\theta_i - \theta_j) \equiv r_{ij} \pmod{p}, \qquad |r_{ij}| \le 3X^2/u,$$

where r_{ij} are integers. In the claimed range, $6X^2 < p$ for sufficiently large p. The residues θ_i are not small, so counting all their values directly would introduce factors of p. We eliminate them instead.

Eliminating the residues with two anchor points. Only differences of the θ_i occur, so subtract θ_1 from all of them and work with $\theta_1 = 0$. For fixed W_1, s_1, s_i, r_{1i}, the equation for $(1, i)$ determines θ_i once W_i is chosen. Fix a second point W_b and the tuple (s_1, s_b, r_{1b}). The equation for $(1, b)$ then determines θ_b. For each remaining i, fix (s_i, r_{1i}, r_{bi}) and eliminate θ_i between the equations for $(1, i)$ and (b, i). Multiply the first equation by b_{bi} and subtract b_{1i} times the second. The θ_i terms cancel, and the left-hand side minus the right-hand side defines a rational function $H_i(W_i)$ whose zeros include all permitted W_i. In this equation, (a_{hi}, b_{hi}) vary with W_i as the coordinates of $W_i - W_h = Q_i - Q_h$. An uncancelled pole guarantees nonconstancy. On the curve, total pole order bounds the number of solutions to a nonconstant rational-function equation, playing the role of degree in polynomial root counting.

The anchor choice is used only in the counting argument. The key is to choose W_b before fixing the integers s_i and r_{ij} in the dual constraints, so that H_i is nonconstant for every allowed choice. Write $W_h = (z_h^*, w_h^*)$. We choose $b \neq 1$ satisfying

$$w_1^* \varphi'(z_b^*) - w_b^* \varphi'(z_1^*) \not\equiv 0 \pmod{p}, \qquad \varphi'(z) = 3z^2 + a_{\mathcal{E}}.$$

For a variable curve point $W = (z, w)$, use the local parameter $t = -z/w$ at $\mathcal{O}$. The addition-law calculation gives the coefficients of t^{-2} and t^{-1} in $H_i(W)$, respectively, as $w_1^* s_b - w_b^* s_1$ and

$\varphi'(z_b^*)s_1 - \varphi'(z_1^*)s_b$. Under the displayed anchor condition, these coefficients cannot both vanish when $s_1 \not\equiv 0 \pmod p$. Thus the same anchor guarantees nonconstancy for every choice of the bounded integer parameters.

As a function of W_b, the left-hand side of the anchor condition is nonzero, has total pole order at most four, and vanishes at $W_b = W_1$. Thus at most three other sampled points fail the nonvanishing condition on W_b, and $N \geq 5$ guarantees a suitable anchor. For this anchor, H_i has total pole order at most eight, so each integer choice permits at most eight values of W_i. The equation for $(1, i)$ then fixes θ_i uniquely. Section 5 gives the pole calculations.

Counting the sampled points. For fixed W_1, W_b, the tuple (s_1, s_b, r_{1b}) has $O_N(X^4/u^3)$ choices: two factors X/u and one X^2/u. Each choice determines θ_b. For every other i, the tuple (s_i, r_{1i}, r_{bi}) has $O_N(X^5/u^3)$ choices: one factor X/u and two X^2/u. Each tuple permits at most eight values of W_i, each of which determines θ_i uniquely. Thus the auxiliary residues introduce no additional factor of p.

For each fixed index b, condition only on W_1, W_b and use independence of the other points before rejection. Each is uniform on M points, giving a factor $O_N(X^5/(Mu^3))$ per remaining point. Summing over b and the anchor parameters, then conditioning on public acceptance, gives

$$\Pr[Z_{\text{target}} > u] \ll_N \frac{X^4}{u^3}\left(\frac{X^5}{Mu^3}\right)^{N-2} = \frac{X^{5N-6}}{M^{N-2}u^{3N-3}}.$$

The count allows every bounded integer choice and depends only on the W_i, so it covers every permitted approximation rule.

Expected cost and uniqueness. Here $k = N$, $|\mathcal{B}_1| = X$, and $3N - 3 > N$. The common tail integral therefore gives

$$\mathbb{E}\Big[\min\{X, Z_{\text{target}}^N\}\Big] \ll_N 1 + \frac{X^{5N-6}}{M^{N-2}}.$$

For $M \geq p/(\log p)^c$, this is polynomial in $\log p$ when $X \leq p^{(N-2)/(5N-6)}$. The exponent approaches $1/5$, so a sufficiently large $N = O(\varepsilon^{-1})$ covers $X \leq p^{1/5-\varepsilon}$. The factor X^5/M per remaining point explains this range.

Finally, for a fixed $\widehat{y} \neq y^*$, the function $\pi_1(\widehat{y} + Q) - \pi_1(y^* + Q)$ has total pole order four. Consistency permits $O(X)$ difference values, each with at most four preimages. Independence before rejection and a union bound over the $O(p)$ possible secrets $\widehat{y}$ give $O_N(p(X/M)^N) = o(1)$ after conditioning on public acceptance. Together with the common recovery guarantee, this proves Theorem 1.2.

1.2.5 HNPHM

For HNPHM, we first explain how to verify a candidate secret without enumerating the hidden multipliers. We then use the product relations to bound the expected search cost.

Setup. Let $N \geq 1$ and $1 \leq X < p$ be integers. The public input consists of approximations $v_1, \ldots, v_k \in \mathbb{Z}_p$, where $k = 2N + 1$, to a secret $y^* \neq 0$, hidden nonzero multipliers t_i^*, and their products modulo p. More precisely, $z_1^* = y^*$, $z_{i+1}^* = t_i^*$, and $z_{N+i+1}^* \equiv y^* t_i^* \pmod p$ are canonical residues with $z_j^* = v_j + x_j^*$ over the integers and $|x_j^*| \leq X$. We use $\mathcal{B} = [-X, X]^k \cap \mathbb{Z}^k$. For

leading-bit inputs, each v_j is the left endpoint of the original interval specified by the revealed bits; reconstruction also checks these supplied intervals.

For the average-cost analysis, fix $y^* \in \mathbb{Z}_p \setminus \{0\}$ and sample the t_i^* independently and uniformly from $\mathbb{Z}_p \setminus \{0\}$. The approximations may depend jointly on all sampled multipliers, provided they satisfy the error bounds and any supplied intervals. All probabilities below use this sampling model for a fixed permitted approximation rule.

Each multiplier–product pair gives the standard quadratic relation [HGNS03]

$$P_i(\mathbf{x}) = (v_1 + x_1)(v_{i+1} + x_{i+1}) - (v_{N+i+1} + x_{N+i+1}) \equiv 0 \pmod{p} \qquad (1 \le i \le N).$$

All these relations share the same error coordinate x_1. Append $px_2, \ldots, px_{N+1}, px_1$. The distinct quadratic monomials x_1x_{i+1} and the appended linear monomials give independent scaled coefficient vectors, so these vectors generate a lattice Λ of rank $m = 2N + 1$.

Verifying a secret without recovering the multipliers. For $a \in \mathcal{B}_1$, the candidate secret is $\widehat{y} = v_1 + a$. We require $1 \le \widehat{y} < p$ and check any supplied leading-bit interval for the secret. For each i, form the multiplier and product intervals I_i, J_i, centered at v_{i+1} and v_{N+i+1}, respectively, with radius X. Clip them to $[1, p-1]$ and $[0, p-1]$, respectively, and intersect them with any supplied leading-bit intervals. The required test is

$$\#\{t \in I_i : \widehat{y}t \bmod p \in J_i\} > 0 \qquad (1 \le i \le N).$$

Once $\widehat{y}$ is fixed, each multiplier can be chosen separately, so these tests are jointly sufficient. Each count is computable by Euclidean division through floor sums in deterministic polynomial bit time [yos21]. Thus $\mathcal{R}(a)$ returns $\{\widehat{y}\}$ exactly when all tests pass, and returns the empty set otherwise. Section 6 gives the counting formulas.

Dual constraints with a shared nonzero coefficient. For $1 \le u \le X$, let $\mathbf{w}$ witness $Z_{\text{target}} > u$. Put $s = \langle \Phi(px_1), \mathbf{w} \rangle$ and $s_i = \langle \Phi(px_{i+1}), \mathbf{w} \rangle$.

These are integers because the corresponding vectors belong to the lattice, and $s \neq 0$ by the target condition. Define $R_i = \langle \Phi(p(x_{N+i+1} - x_1x_{i+1})), \mathbf{w} \rangle + sx_{i+1}^* + x_1^*s_i$. Expanding the pairings with $\Phi(P_i)$ and substituting $t_i^* = v_{i+1} + x_{i+1}^*$ and $y^* = v_1 + x_1^*$ gives

$$st_i^* + y^*s_i \equiv R_i \pmod{p}, \qquad 0 < |s| < X/u, \quad |s_i| < X/u, \quad R_i \in \mathbb{Z}, \quad |R_i| \le 4X^2/u.$$

The combined quadratic and product-error pairing is an integer because $\Phi(p(x_{N+i+1} - x_1x_{i+1})) \in \Lambda$, as follows from P_i and the appended linear vectors. The quadratic and product-error pairings need not be integral separately; only their difference is used as an integer parameter. The common monomial bounds give magnitudes at most X^2/u and X/u for the pairings with $\Phi(px_1x_{i+1})$ and $\Phi(px_{N+i+1})$, and at most X^2/u for each of sx_{i+1}^* and $x_1^*s_i$. Since $0 < |s| < X < p$, s is invertible modulo p. Consequently, fixed s, s_i, R_i permit at most one nonzero multiplier t_i^*.

Counting the sampled multipliers. We count the $O(X/u)$ choices of s once. For each fixed s and each index i, allow every pair (s_i, R_i) within the displayed bounds. There are $O(X^3/u^2)$ such pairs, and each permits at most one multiplier among $p-1$ equally likely values. The probability that some pair satisfies the congruence is therefore $O(X^3/((p-1)u^2))$.

We do not require the chosen pairs across indices to arise from a common dual vector; relaxing this requirement only enlarges the event being counted. For fixed s and y^*, the event that some

integer pair (s_i, R_i) within the displayed bounds satisfies the congruence depends only on t_i^*, since the approximations v_j no longer occur. We can therefore multiply the probabilities across the independent multipliers, even when the approximations depend jointly on all samples. Summing over s gives

$$\Pr[Z_{\text{target}} > u] \ll_N \frac{X}{u}\left(\frac{X^3}{(p-1)u^2}\right)^N = \frac{X^{3N+1}}{(p-1)^N u^{2N+1}} \qquad (1 \le u \le X).$$

For $u \ge X$, the nonzero integer s cannot satisfy $|s| < X/u$, so the tail probability is zero.

Expected cost and uniqueness. Here $k = 2N+1$ matches the tail exponent, and the direct-scan cap is $|\mathcal{B}_1| = 2X+1$. The common tail integral therefore gives

$$\mathbb{E}\Big[\min\{2X+1, Z_{\text{target}}^{2N+1}\}\Big] \ll_N 1 + \frac{X^{3N+1}}{(p-1)^N}(1+\log X).$$

In particular, $X \le p^{N/(3N+1)}$ gives polynomial expected bit cost in $\log p$. Since $N/(3N+1)$ approaches $1/3$, choosing a sufficiently large $N = O(\varepsilon^{-1})$ covers $X \le p^{1/3-\varepsilon}$ for each fixed $0 < \varepsilon < 1/3$. The factor $X^3/(p-1)$ per multiplier explains this range. The expectation is over the sampled multipliers, and the bound holds uniformly for every fixed y^* and every permitted approximation rule.

Finally, any additional satisfying secret $\widehat{y}$ lies within $2X$ of y^*, leaving $O(X)$ possibilities. For each fixed $\widehat{y} \ne y^*$, let $\widehat{t}_i$ be a consistent multiplier. Then $\widehat{t}_i$ and $\widehat{y}\widehat{t}_i \bmod p$ differ from t_i^* and z_{N+i+1}^* by at most $2X$ each, giving $O(X^2)$ difference pairs. For each pair, subtracting the product congruences gives a nonzero coefficient $\widehat{y} - y^*$ on the sampled multiplier t_i^*, so at most one value is possible. Independence and a union bound give an additional-secret probability $O_N(X^{2N+1}/(p-1)^N) = o(1)$ in this range. The common search is deterministic here and returns the exact set of consistent secrets on every valid input; its output is therefore $\{y^*\}$ with probability $1 - o(1)$.

1.3 Related Work

Earlier provable polynomial-time algorithms for MIHNP, EC-HNP, and HNPHM use exact or approximate shortest or closest vector computations and bound the failure probability of recovering y^* [LSSW12, Sha17, HGNS03]. For the three nonlinear HNPs, our analysis proves polynomial input-averaged bit cost for the complete integer-quotient search. While [AH21] circumvents the unique shortest vector barrier by enumerating candidate lattice vectors with a verification predicate, our algorithm searches the parameter space directly via geometric subdivision and integer quotients.

Lower-degree polynomial relations are crucial for tightening our tail bounds. For MIHNP and HNPHM, standard quadratic formulations suffice and have been widely used in both provable and heuristic settings [LSSW12, XSH+19, Sar17]. In contrast, EC-HNP natively yields cubic relations from the elliptic curve group law [Sha17, XSWH26]. We therefore build on the observation of [DDW+26] that cubic EC-HNP relations can be reduced to quadratic ones modulo p. We use quadratic identities valid for admissible query tuples. A separate two-anchor counting argument bounds the number of possibilities for each remaining point once the bounded integer parameters are fixed. This count yields the tail bounds used to control the expected search cost.

[BKS23] also use the dual lattice, but for a different purpose. They show that polynomial equations obtained from lattice reduction permit efficient recovery with high probability under a specified distribution of partial approximate common divisor instances. For the three nonlinear HNPs, our target-sensitive dual estimates bound the expected cost of the complete integer-quotient search, while allowing short dual vectors orthogonal to the recovery target.

Organization. Section 2 introduces the input model and the lattice tools. Section 3 gives the recovery algorithm, its correctness proof, and its cost bound. Sections 4 to 6 construct the polynomial relations and prove the counting bounds for the three problems. Section 7 presents the experiments, and Section 8 discusses the results and further directions.

2 Preliminaries

Notation. We write $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{R}$ for the integers, rational numbers, and real numbers. For a prime p, let $\mathbb{Z}_p = \{0, \dots, p-1\}$ denote the set of standard residues modulo p. For an integer a, let $[a]_p \in \mathbb{Z}_p$ denote the least nonnegative residue of a modulo p. For an integer $r \geq 1$, write $[r] = \{1, \dots, r\}$. Vectors are denoted in boldface, and their components are denoted in standard font with subscripts (e.g., $\mathbf{u} = (u_1, \dots, u_d)$). For vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$, write $\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{i=1}^{d} u_i v_i$, $\|\mathbf{u}\|_1 = \sum_{i=1}^{d} |u_i|$, $\|\mathbf{u}\|_2 = \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle}$, and $\|\mathbf{u}\|_\infty = \max_{i \in [d]} |u_i|$.

2.1 Nonlinear Hidden Number Problems

The classical hidden number problem (HNP) asks to recover a secret y^* from approximations to $[t_i y^*]_p$ for known multipliers t_i [BV96]. Below, we formalize nonlinear variants of the HNP by replacing these linear multiples with polynomial relations and representing the partial information by integer error intervals.

Definition 2.1 (Nonlinear Hidden Number Problem)**.** The public parameters are a prime p and integers $X, n, k \geq 1$. The remaining public input consists of integer polynomials $F_1, \dots, F_n \in \mathbb{Z}[y, z_1, \dots, z_k]$, an approximation vector $\mathbf{v} \in \mathbb{Z}_p^k$, and an error box $\mathcal{B} = \prod_{j=1}^{k} \mathcal{B}_j \subseteq [-X, X]^k \cap \mathbb{Z}^k$, where each $\mathcal{B}_j$ is an integer interval. The system defined by $F_1, \dots, F_n$ is nonlinear in the joint unknowns $y, z_1, \dots, z_k$.

The input is promised to have a hidden secret $y^* \in \mathbb{Z}_p$ and an error vector $\mathbf{x}^* \in \mathcal{B}$ such that, with $z_j^* = v_j + x_j^*$ for $j \in [k]$,

$$z_j^* \in \mathbb{Z}_p \quad (j \in [k]), \qquad F_i(y^*, z_1^*, \dots, z_k^*) \equiv 0 \pmod{p} \quad (i \in [n]).$$

The input may admit several secrets. Our algorithms return exactly the secrets satisfying its constraints, and the application sections prove that this set equals $\{y^*\}$ with high probability under their sampling models.

The formulation above uses a scalar secret and requires $z_j^* = v_j + x_j^*$ as an equality in $\mathbb{Z}$. Leading-bit information corresponds to nonnegative error intervals. For EC-HNP, the secret is a point, and we allow $z_j^* \equiv v_j + x_j^* \pmod{p}$, as specified in Section 5.

For MIHNP and HNPHM, and for EC-HNP inputs with only finite observations, we substitute $z_j = v_j + x_j$ in the public relations and eliminate the secret variables to obtain public polynomials. For MIHNP and HNPHM, these transformations follow the relations in [BHHG01, HGNS03]. Our construction for EC-HNP is given in Section 5.

By taking polynomial multiples and integer linear combinations of these relation polynomials and adjoining polynomials divisible by p, such as px_j, we obtain a family $P_1, \dots, P_m \in \mathbb{Z}[x_1, \dots, x_k]$ satisfying

$$P_i(\mathbf{x}) \equiv 0 \pmod{p} \qquad (i \in [m])$$

for every $\mathbf{x} \in \mathcal{B}$ that, together with some secret, satisfies the original input constraints. Each application specifies the choice of $P_1, \dots, P_m$. The lattice construction below uses this polynomial family.

2.2 Lattices

Definition 2.2 (Lattice)**.** Let $\mathbf{b}_1, \ldots, \mathbf{b}_m \in \mathbb{R}^d$ be linearly independent over $\mathbb{R}$, where $1 \le m \le d$. Let $B \in \mathbb{R}^{m \times d}$ be the matrix with these vectors as rows. The *lattice* generated by B is

$$\Lambda = \left\{ \sum_{i=1}^{m} \alpha_i \mathbf{b}_i : \alpha_1, \ldots, \alpha_m \in \mathbb{Z} \right\} \subseteq \mathbb{R}^d.$$

Its *rank* is m, its *ambient dimension* is d, and its real span $V = \operatorname{span}_{\mathbb{R}} \Lambda$ has dimension m.

Constructing lattices from polynomials. We represent integer linear combinations of polynomials by their nonconstant coefficients, scaled according to the bound $|x_j| \le X$.

Definition 2.3 (Nonconstant coefficient map with scaling)**.** Fix an integer $X \ge 1$ and integer polynomials $P_1, \ldots, P_m \in \mathbb{Z}[x_1, \ldots, x_k]$. Let $M_1, \ldots, M_d$ be the distinct nonconstant monomials appearing in these polynomials, in a fixed order. For any polynomial of the form

$$f(\mathbf{x}) = c_0 + \sum_{a=1}^{d} c_a M_a(\mathbf{x}), \qquad c_0, \ldots, c_d \in \mathbb{Z},$$

define

$$\Phi(f) = \left(c_1 X^{\deg M_1}, \ldots, c_d X^{\deg M_d} \right) \in \mathbb{Z}^d,$$

where deg denotes total degree.

The map Φ is $\mathbb{Z}$-linear and omits the constant term. Since every scaling factor is nonzero, the vectors $\Phi(P_1), \ldots, \Phi(P_m)$ are linearly independent if and only if the unscaled nonconstant coefficient vectors are linearly independent.

Suppose these vectors are linearly independent over $\mathbb{R}$. Set $\mathbf{b}_i^{(0)} = \Phi(P_i)$, let $B_0 \in \mathbb{Z}^{m \times d}$ have these vectors as its rows, and let

$$\Lambda = \left\{ \sum_{i=1}^{m} a_i \mathbf{b}_i^{(0)} : a_1, \ldots, a_m \in \mathbb{Z} \right\}.$$

Then Φ restricts to a bijection

$$\Phi : \operatorname{span}_{\mathbb{Z}}\{P_1, \ldots, P_m\} \longrightarrow \Lambda.$$

Surjectivity follows from linearity. If $\Phi(\sum_i a_i P_i) = \mathbf{0}$, linear independence of the $\Phi(P_i)$ gives $a_i = 0$ for every i, proving injectivity. We write Φ^{-1} for the inverse of this restricted map. In particular, Φ^{-1} recovers the full polynomial in this span, including its constant term.

For every polynomial f in the domain of Φ and every $\mathbf{x} \in \mathcal{B}$, the scaling gives

$$|f(\mathbf{x}) - f(\mathbf{0})| \le \sum_{a=1}^{d} |c_a| X^{\deg M_a} = \|\Phi(f)\|_1 \le \sqrt{d}\, \|\Phi(f)\|_2.$$

The known constant term shifts the resulting value interval without changing its width (the difference between its endpoints).

2.2.1 Lattice Reduction

Lattice reduction seeks a basis of shorter vectors that generates the same lattice. We use the Lenstra–Lenstra–Lovász (LLL) algorithm [LLL82, NS09].

Definition 2.4 (LLL-reduced basis)**.** Let $\delta \in (1/4, 1)$ and let $\mathbf{b}_1, \dots, \mathbf{b}_m \in \mathbb{R}^d$ be a lattice basis. Define its Gram–Schmidt vectors recursively by $\widetilde{\mathbf{b}}_i = \mathbf{b}_i - \sum_{j<i} \mu_{ij} \widetilde{\mathbf{b}}_j$, where $\mu_{ij} = \langle \mathbf{b}_i, \widetilde{\mathbf{b}}_j \rangle / \|\widetilde{\mathbf{b}}_j\|_2^2$ for $j < i$. The basis is *LLL-reduced with parameter* δ if it satisfies the size-reduction conditions $|\mu_{ij}| \le 1/2$ for $1 \le j < i \le m$ and the Lovász conditions

$$\|\widetilde{\mathbf{b}}_i\|_2^2 \ge (\delta - \mu_{i,i-1}^2) \|\widetilde{\mathbf{b}}_{i-1}\|_2^2 \qquad (2 \le i \le m).$$

We use $\delta = 3/4$ throughout. The two reduction conditions then give $\|\widetilde{\mathbf{b}}_i\|_2^2 \ge \frac{1}{2} \|\widetilde{\mathbf{b}}_{i-1}\|_2^2$ for $2 \le i \le m$.

For this parameter, the LLL algorithm applied to B_0 computes a reduced basis matrix $B = UB_0$, where $U \in \mathbb{Z}^{m \times m}$ is unimodular, meaning that $\det U = \pm 1$. Thus the basis change consists of invertible integer row operations. Its bit running time is polynomial in m, d, and the bit-length of the entries of B_0.

Let $\mathbf{b}_j$ denote the jth row of B, and define $h_j = \Phi^{-1}(\mathbf{b}_j)$. Since $\mathbf{b}_j = \sum_i U_{ji} \Phi(P_i)$, linearity and injectivity of Φ on the integer span of the P_i give $h_j = \sum_i U_{ji} P_i$, including the constant terms. Because U^{-1} also has integer entries, each P_i is an integer linear combination of the h_j:

$$P_i = \sum_{j=1}^{m} (U^{-1})_{ij} h_j \qquad (i \in [m]).$$

2.2.2 Dual Lattice

Since $\Phi(h_j) = \mathbf{b}_j$, the preceding evaluation bound turns bounds on $\|\mathbf{b}_j\|_2$ into bounds on the variation of h_j over $\mathcal{B}$. We obtain these length bounds using the dual lattice.

Definition 2.5 (Dual lattice and dual basis)**.** The *dual lattice* of Λ within V is

$$\Lambda^* = \{\mathbf{w} \in V : \langle \mathbf{w}, \mathbf{v} \rangle \in \mathbb{Z} \text{ for every } \mathbf{v} \in \Lambda\}.$$

For a basis $\mathbf{b}_1, \dots, \mathbf{b}_m$ of Λ, its *dual basis* $\mathbf{b}_1^\vee, \dots, \mathbf{b}_m^\vee \in V$ is uniquely determined by $\langle \mathbf{b}_i, \mathbf{b}_j^\vee \rangle = \delta_{ij}$, where $\delta_{ij} = 1$ if $i = j$ and 0 otherwise. These vectors form a basis of Λ^*.

The following standard bound relates each reduced basis vector to its paired dual basis vector.

Lemma 2.6 (Paired length bound via dual lattice [LLL82, Bab86])**.** *Let $\mathbf{b}_1, \dots, \mathbf{b}_m$ be an LLL-reduced basis of Λ with parameter $\delta = 3/4$, and let $\mathbf{b}_1^\vee, \dots, \mathbf{b}_m^\vee$ be its dual basis. For every $j \in [m]$,*

$$\|\mathbf{b}_j\|_2 \|\mathbf{b}_j^\vee\|_2 \le 4^m.$$

Proof. Fix $j \in [m]$ and put $d_i = \|\widetilde{\mathbf{b}}_i\|_2$. The LLL conditions give $d_i^2 \ge d_{i-1}^2/2$ for $i \ge 2$, and hence $\|\mathbf{b}_j\|_2^2 = d_j^2 + \sum_{i<j} \mu_{ji}^2 d_i^2 \le 2^{j-1} d_j^2$. Write $\mathbf{b}_j^\vee = \sum_{i=j}^m c_i \widetilde{\mathbf{b}}_i / d_i^2$. The duality equations give $c_j = 1$ and $c_i = -\sum_{\ell=j}^{i-1} \mu_{i\ell} c_\ell$ for $i > j$. Induction using $|\mu_{i\ell}| \le 1/2$ yields $|c_i| \le \frac{1}{2}(3/2)^{i-j-1}$ for $i > j$. Since $d_i^{-2} \le 2^{i-j} d_j^{-2}$ for $i \ge j$, orthogonality gives

$$\|\mathbf{b}_j^\vee\|_2^2 \le \frac{1}{d_j^2} \left(1 + \frac{1}{2} \sum_{r=0}^{m-j-1} (9/2)^r \right) \le \frac{(9/2)^{m-j}}{d_j^2}.$$

Combining the bounds gives $\|\mathbf{b}_j\|_2 \|\mathbf{b}_j^\vee\|_2 \le 2^{(j-1)/2} (3/\sqrt{2})^{m-j} \le (3/\sqrt{2})^{m-1} \le 4^m$. □

3 Secret Recovery from Integer Quotients

This section gives the recovery algorithm used in Sections 4 to 6 and bounds its cost on each fixed public input. Each application supplies polynomial relations in the error variables and a procedure that reconstructs and verifies secret candidates from a proposed value of x_1. We specify the properties these data must satisfy, then prove correctness and a cost bound using only those properties. The application sections give the concrete polynomials and reconstruction procedures and analyze the cost under their sampling rules.

3.1 A Common Recovery Framework

Data supplied by each application. Let $\mathcal{B} = \prod_{i=1}^{k} \mathcal{B}_i \subseteq [-X, X]^k \cap \mathbb{Z}^k$ be the input's error box, where $X \geq 1$ and each $\mathcal{B}_i$ is a nonempty integer interval. A secret y and an error vector $\mathbf{x} \in \mathcal{B}$ are consistent with the input if together they satisfy the constraints in the application's problem definition. We call $\mathbf{x}$ consistent with the input if such a secret y exists. Each application specifies $\mathbf{P} = (P_1, \ldots, P_m)^T \in \mathbb{Z}[\mathbf{x}]^m$ with the following properties:

(i) $P_m = px_1$, and every P_i has total degree at most two.

(ii) For every secret y and every corresponding $\mathbf{x} \in \mathcal{B}$ consistent with the input, $P_i(\mathbf{x}) \equiv 0 \pmod{p}$ for all $i \in [m]$.

(iii) The vectors $\Phi(P_1), \ldots, \Phi(P_m)$ are linearly independent over $\mathbb{R}$, where Φ is the scaled coefficient map of Definition 2.3 for this family and X.

(iv) The public input has binary encoding length $\mathrm{poly}(\log p)$, and the full polynomials P_i can be constructed deterministically in $\mathrm{poly}(\log p)$ bit time.

These are the input requirements of the common algorithm. The constructions in Sections 4 to 6 verify them for the three problems. Condition (ii) is only a necessary condition for a valid secret. The congruences $P_i(\mathbf{x}) \equiv 0 \pmod{p}$ need not imply that $\mathbf{x}$ is consistent with the input. Reconstruction therefore checks the original input constraints.

Reconstruction from $a \in \mathcal{B}_1$. A proposed value a for x_1 gives at most one $\widehat{y}$ in MIHNP or HNPHM and at most two in finite EC-HNP inputs. Each $\widehat{y}$ is checked against the original input constraints. EC-HNP inputs reporting infinity are handled directly in Section 5, before applying this algorithm.

Definition 3.1 (Reconstructor)**.** A reconstructor is a procedure $\mathcal{R}$ that, given the public input and $a \in \mathcal{B}_1$, returns a set $\mathcal{R}(a)$ of at most r secret candidates, where $r \geq 1$ is a fixed constant. It satisfies:

(i) Every element of $\mathcal{R}(a)$ satisfies the original input constraints with probability one.

(ii) For every secret y and corresponding $\mathbf{x} \in \mathcal{B}$ consistent with the input, $\Pr[y \in \mathcal{R}(x_1)] = 1$.

(iii) For every public input and every $a \in \mathcal{B}_1$, computing $\mathcal{R}(a)$ has uniform expected bit cost $\mathrm{poly}(\log p)$. Its elements have canonical binary encodings of length $\mathrm{poly}(\log p)$.

Probabilities and expectations here are over reconstruction randomness.

Testing a means computing $\mathcal{R}(a)$, including reconstruction and verification. Condition (ii) ensures that testing the correct first error of any valid secret returns that secret. We keep k, m, d, r fixed in the complexity bounds, where d is the number of monomials recorded by Φ; polynomial bounds may depend on these fixed parameters.

The polynomial lattice and the quotient identity. Apply the polynomial-lattice construction of Section 2.2 to $\mathbf{P}$, retaining the unimodular matrix U with $B = UB_0$. Then

$$\mathbf{H} = (h_1, \ldots, h_m)^T = U\mathbf{P}, \qquad \mathbf{b}_j = \Phi(h_j).$$

Let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)$ be the last row of U^{-1}. The last coordinate of $\mathbf{P} = U^{-1}\mathbf{H}$ gives the polynomial identity

$$px_1 = \sum_{j=1}^{m} \alpha_j h_j(\mathbf{x}).$$

At every $\mathbf{x}$ consistent with the input, each $h_j(\mathbf{x})$ is divisible by p, since h_j is an integer linear combination of the P_i. In particular,

$$x_1^* = \sum_{j:\alpha_j \neq 0} \alpha_j q_j^*, \qquad q_j^* = \frac{h_j(\mathbf{x}^*)}{p} \in \mathbb{Z}. \tag{1}$$

The same identity holds for every error vector consistent with the input. We therefore need quotient bounds only for j with $\alpha_j \neq 0$.

Partitioning and searching $\mathcal{B}$. All the h_j depend on the same $\mathbf{x}$, so we bound their quotients on common subsets of $\mathcal{B}$. The following finite partition specifies how far the search may need to subdivide. It is determined by the computed basis lengths.

Definition 3.2 (Cell family)**.** Set

$$W = \max_{j:\alpha_j \neq 0} \|\mathbf{b}_j\|_1, \qquad b = \max\{1, \lceil 12W/p \rceil\}.$$

Partition each $\mathcal{B}_i$ into $\min\{b, |\mathcal{B}_i|\}$ consecutive integer intervals whose cardinalities differ by at most one. Call these the final intervals. Their Cartesian products form the family $\mathcal{C}$ of final cells, with $|\mathcal{C}| = \prod_{i=1}^{k} \min\{b, |\mathcal{B}_i|\}$.

The search starts with $C = \mathcal{B}$. At every stage, $C = \prod_i C_i$, where each C_i is a union of consecutive final intervals in $\mathcal{B}_i$. For every j with $\alpha_j \neq 0$, compute integer bounds $L_j(C) \leq h_j(\mathbf{x}) \leq R_j(C)$ for all $\mathbf{x} \in C$. We obtain $L_j(C)$ by summing the minima of the monomial terms of h_j on C, and $R_j(C)$ by summing their maxima. The constant term is included in both sums. For linear and mixed quadratic terms, coordinate endpoints suffice; for a square term, also check zero when its interval contains zero. For every $\mathbf{x} \in C$ consistent with the input, the integer quotients $h_j(\mathbf{x})/p$ lie in

$$I_j(C) = \left[\left\lceil \frac{L_j(C)}{p} \right\rceil, \left\lfloor \frac{R_j(C)}{p} \right\rfloor \right] \cap \mathbb{Z}.$$

If any $I_j(C)$ is empty, discard C. Otherwise let

$$J(C) = C_1 \cap \left[\sum_{j:\alpha_j \neq 0} \min_{q \in I_j(C)} \alpha_j q, \sum_{j:\alpha_j \neq 0} \max_{q \in I_j(C)} \alpha_j q \right] \cap \mathbb{Z}.$$

The minima and maxima also handle negative α_j. Thus $x_1 \in J(C)$ for every $\mathbf{x} \in C$ consistent with the input.

If $J(C)$ is empty, discard C. If $J(C) = \{a\}$, compute $\mathcal{R}(a)$ and stop searching C. By Definition 3.1(ii), $\mathcal{R}(a)$ returns every secret consistent with an error vector in C with probability one. The remaining error coordinates therefore need not be determined. If $|J(C)| > 1$, choose an index i for which C_i contains at least two final intervals. Split C_i into two adjacent integer intervals, each a union of final intervals, such that the numbers of final intervals in the two parts differ by at most one. The proof below shows that a final cell never needs a split. The final intervals and their endpoints are represented implicitly; the algorithm does not list all cells of $\mathcal{C}$.

Let $\mathcal{Q}$ store unprocessed C, $\mathcal{T}$ store tested values a, and $\mathcal{Y}$ store returned secrets. Testing an untested a adds $\mathcal{R}(a)$ to $\mathcal{Y}$ and a to $\mathcal{T}$. All sets are stored without duplicates. A visit consists of removing one box C from $\mathcal{Q}$ and processing it as described above. After $|\mathcal{B}_1|$ visits, any unfinished search is replaced by a scan of the untested values in $\mathcal{B}_1$. For concreteness, process boxes in depth-first order, split the smallest-index coordinate containing at least two final intervals, and process the left child first.

Algorithm 1 Secret candidate recovery from integer quotients

Input: Public input, $\mathbf{P}$, and a reconstructor $\mathcal{R}$
Output: $\mathcal{Y}$: all secrets satisfying the original input constraints

Form B_0; run LLL to obtain $B = UB_0$.
$\mathbf{H} \leftarrow U\mathbf{P}$, $\alpha_j \leftarrow (U^{-1})_{m,j}$ $(j \in [m])$; compute b.
$\mathcal{Y} \leftarrow \varnothing$, $\mathcal{T} \leftarrow \varnothing$, $\mathcal{Q} \leftarrow \{\mathcal{B}\}$, $c \leftarrow 0$.
while $\mathcal{Q} \neq \varnothing$ **and** $c < |\mathcal{B}_1|$ **do**
    Remove C from $\mathcal{Q}$; $c \leftarrow c + 1$.
    Compute $I_j(C)$ for j with $\alpha_j \neq 0$.
    if all these $I_j(C)$ are nonempty **then**
        Compute $J(C)$.
        if $J(C) = \{a\}$ **then**
            Test a if $a \notin \mathcal{T}$.
        else if $|J(C)| > 1$ **then**
            Split C; insert its children into $\mathcal{Q}$.
if $\mathcal{Q} \neq \varnothing$ **then**
    Test all $a \in \mathcal{B}_1 \setminus \mathcal{T}$.
return $\mathcal{Y}$.

The intervals are necessary tests for a solution in C. Only $\mathcal{R}(a)$ determines which secrets are returned.

3.2 Correctness

Proposition 3.3 (Complete candidate recovery)**.** *Under the polynomial and reconstruction requirements above, Algorithm 1 terminates with probability one and returns exactly the secrets satisfying the original input constraints. Its output contains y^* and equals $\{y^*\}$ whenever those constraints determine the secret uniquely.*

Proof. Every returned secret belongs to some $\mathcal{R}(a)$ and hence satisfies the original constraints with probability one.

First, a final cell needs no further subdivision. If $b \geq |\mathcal{B}_i|$, its ith interval is a singleton. Otherwise that interval has width at most

$$\left\lceil \frac{|\mathcal{B}_i|}{b} \right\rceil - 1 < \frac{|\mathcal{B}_i|}{b} \leq \frac{3X}{b},$$

because $|\mathcal{B}_i| \leq 2X + 1 \leq 3X$. For $u, v, u', v' \in [-X, X]$, $|uv - u'v'| \leq X|u - u'| + X|v - v'|$. Thus a linear monomial has range width at most $3X/b$ and a quadratic monomial has range width at most $6X^2/b$ on a final cell. Adding the widths of the monomial terms, and using $\mathbf{b}_j = \Phi(h_j)$, gives

$$R_j(C) - L_j(C) \leq \frac{6\|\mathbf{b}_j\|_1}{b} \leq \frac{6W}{b} \leq \frac{p}{2} < p \qquad (\alpha_j \neq 0).$$

Each $I_j(C)$ therefore contains at most one integer. If one is empty, C is discarded; otherwise their weighted sum gives at most one integer in $J(C)$. In particular, $|J(C)| > 1$ implies that C can be split as specified.

Now fix any secret y and corresponding $\mathbf{x} \in \mathcal{B}$ consistent with the input. For every visited C containing $\mathbf{x}$, each $I_j(C)$ contains $h_j(\mathbf{x})/p$, and the quotient identity gives $x_1 \in J(C)$. Such a C cannot be discarded. If it is split, one child still contains $\mathbf{x}$. Consequently, either a descendant has $J(C) = \{x_1\}$, or the visit cap triggers the scan over $\mathcal{B}_1$. In both cases x_1 is tested, possibly during an earlier test of the same value. Definition 3.1 then gives $y \in \mathcal{R}(x_1)$ with probability one. Since the secret encodings and the possible first errors are finite, all valid secrets are returned with probability one.

The partition, the search tree, and the scan range are finite. Each reconstruction call has finite expected running time and therefore terminates with probability one. The whole algorithm terminates with probability one and returns exactly the required set. □

3.3 Complexity

Counting visits and calls to $\mathcal{R}$. Fully expanding the specified binary subdivision produces a tree with $|\mathcal{C}|$ leaves and $2|\mathcal{C}| - 1$ nodes. Discarding C, or stopping its subdivision when $J(C)$ is a singleton, can only reduce the number of visits. With the visit cap, the number of visits is at most $\min\{|\mathcal{B}_1|, 2|\mathcal{C}| - 1\}$. Each visit tests at most one new a. If the final scan is used, an unprocessed C remains after $|\mathcal{B}_1|$ visits, so $|\mathcal{B}_1| < 2|\mathcal{C}| - 1$. The scan tests at most $|\mathcal{B}_1|$ values. Visits, scan iterations, and distinct tested values together therefore number

$$O(1 + \min\{|\mathcal{B}_1|, |\mathcal{C}|\}).$$

It remains to bound $|\mathcal{C}| \leq b^k$ in terms of the input.

Bounding b using the dual lattice. Let $\mathbf{b}_1^\vee, \ldots, \mathbf{b}_m^\vee$ be the dual basis inside $\operatorname{span}_{\mathbb{R}} \Lambda$, as in Definition 2.5. Set $\mathbf{v}_{\text{tar}} = \Phi(px_1) \in \Lambda$. Applying Φ to the polynomial identity for px_1 gives

$$\mathbf{v}_{\text{tar}} = \sum_{i=1}^{m} \alpha_i \mathbf{b}_i, \qquad \langle \mathbf{v}_{\text{tar}}, \mathbf{b}_j^\vee \rangle = \sum_{i=1}^{m} \alpha_i \delta_{ij} = \alpha_j.$$

Thus, when $\alpha_j \neq 0$, the dual vector $\mathbf{b}_j^\vee$ has a nonzero inner product with $\mathbf{v}_{\text{tar}}$. Define

$$\lambda_{\text{target}} = \min\{\|\mathbf{w}\|_2 : \mathbf{w} \in \Lambda^*, \quad \langle \mathbf{v}_{\text{tar}}, \mathbf{w} \rangle \neq 0\}, \qquad Z_{\text{target}} = \max\{1, (p\lambda_{\text{target}})^{-1}\}.$$

The set in this minimum is nonempty: some $\alpha_j \neq 0$ because $\mathbf{v}_{\text{tar}} \neq \mathbf{0}$. The quantity Z_{target} will bound the number of final intervals per coordinate. It is used only for analysis; the algorithm computes its partition from W.

The only nonzero coordinate of $\Phi(px_1)$ is pX, at the monomial x_1, so $\|\mathbf{v}_{\mathrm{tar}}\|_2 = pX$. For any $\mathbf{w}$ in the minimum above, its inner product with $\mathbf{v}_{\mathrm{tar}}$ is a nonzero integer. Hence

$$1 \le |\langle \mathbf{v}_{\mathrm{tar}}, \mathbf{w}\rangle| \le pX\|\mathbf{w}\|_2, \qquad \lambda_{\mathrm{target}} \ge \frac{1}{pX}, \qquad 1 \le Z_{\mathrm{target}} \le X.$$

For every j with $\alpha_j \neq 0$, $\|\mathbf{b}_j^{\vee}\|_2 \ge \lambda_{\mathrm{target}}$. Lemma 2.6 and $\|\mathbf{b}_j\|_1 \le \sqrt{d}\|\mathbf{b}_j\|_2$ therefore give

$$W \le \frac{\sqrt{d}\,4^m}{\lambda_{\mathrm{target}}}, \qquad b \le 1 + \frac{12\sqrt{d}\,4^m}{p\lambda_{\mathrm{target}}} \le KZ_{\mathrm{target}}, \tag{2}$$

where $K = 1 + 12\sqrt{d}\,4^m$ depends only on m, d. Consequently $|\mathcal{C}| \le (KZ_{\mathrm{target}})^k$.

Theorem 3.4 (Secret recovery cost)**.** *Under the polynomial and reconstruction requirements above, Algorithm 1 returns the set in Proposition 3.3 with probability one in expected bit time*

$$\mathrm{poly}(\log p)\big[1 + \min\{|\mathcal{B}_1|, (KZ_{\mathrm{target}})^k\}\big].$$

For a fixed public input, the expectation is over reconstruction randomness. The bound is deterministic when $\mathcal{R}$ is deterministic.

Proof. By the input requirements, B_0 has polynomially bounded bit length in $\log p$. LLL, basis tracking, and exact linear algebra compute $B, U, \mathbf{H}, \boldsymbol{\alpha}$, and b in $\mathrm{poly}(\log p)$ bit time, with polynomially bounded intermediate bit lengths [LLL82, NS09]. The final intervals are represented by their counts and endpoints, computed as needed by integer division.

Each visit computes $L_j(C), R_j(C), I_j(C)$ and $J(C)$ using a fixed number of exact arithmetic operations on integers of polynomial bit length. Its bit cost is therefore $\mathrm{poly}(\log p)$. Each tested a incurs the uniform expected cost of computing $\mathcal{R}(a)$ in Definition 3.1. Store tested values and canonical secret encodings in balanced search trees. Each operation has polynomial bit cost: at most $r|\mathcal{B}_1|$ secret candidates are inserted, and $\log|\mathcal{B}_1| \le \mathrm{poly}(\log p)$ by the input encoding bound. Combining the count of visits and tests with (2) gives

$$\mathrm{poly}(\log p)\big[1 + \min\{|\mathcal{B}_1|, |\mathcal{C}|\}\big] \le \mathrm{poly}(\log p)\big[1 + \min\{|\mathcal{B}_1|, (KZ_{\mathrm{target}})^k\}\big].$$

Correctness is Proposition 3.3. □

Averaging over sampled inputs. Fix k, m, d, r and, for each public choice of p, X, a secret y^* in the application's secret domain. The sampling rules in Sections 4 to 6 specify which shifts, query points, or multipliers are random and how the approximations are formed. Suppose every sampled input either satisfies the common requirements or is handled separately in polynomial bit time before lattice construction, as for EC-HNP infinity inputs. Use Z_{target} as defined above on the former inputs; for the input average, set $Z_{\mathrm{target}} = 1$ on the latter. Because the fallback scan caps the number of visits and tests at $O(|\mathcal{B}_1|)$, taking the input average in Theorem 3.4 gives polynomial expected bit cost whenever

$$\mathbb{E}\big[\min\{2X + 1, Z_{\mathrm{target}}^k\}\big] = \mathrm{poly}(\log p). \tag{3}$$

Indeed, $|\mathcal{B}_1| \le 2X + 1$ and K^k is constant, so $\min\{|\mathcal{B}_1|, (KZ_{\mathrm{target}})^k\} \le K^k \min\{2X + 1, Z_{\mathrm{target}}^k\}$. This expectation is over the sampled inputs; the theorem has already included the expected cost of reconstruction on each fixed input.

For any random variable $Z \geq 1$ and real $T \geq 1$, the tail-integral identity and the substitution $t = u^k$ give

$$\mathbb{E}[\min\{T, Z^k\}] = 1 + \int_1^T \Pr[Z > t^{1/k}]\, dt = 1 + k \int_1^{T^{1/k}} u^{k-1} \Pr[Z > u]\, du.$$

Apply this identity with $T = 2X + 1$ and $Z = Z_{\text{target}}$. It therefore suffices to control $\Pr[Z_{\text{target}} > u]$ for $1 \leq u \leq (2X+1)^{1/k}$.

The tail estimate must compensate for the weight u^{k-1}. For example, a bound $\Pr[Z_{\text{target}} > u] \leq \text{poly}(\log p)\, u^{-k}$ leaves only a logarithmic integral. A fixed decay exponent greater than k gives a bounded integral. The application-specific estimates control both this decay and the remaining factors involving p and X. For an input passed to the common algorithm and $u \geq 1$, the definition gives

$$Z_{\text{target}} > u \quad \Longleftrightarrow \quad \exists\, \mathbf{w} \in \Lambda^* : \|\mathbf{w}\|_2 < \frac{1}{pu}, \quad \langle \mathbf{v}_{\text{tar}}, \mathbf{w} \rangle \neq 0.$$

On inputs handled separately, $Z_{\text{target}} > u$ is false for $u \geq 1$. No independence assumption on the sampled inputs is needed for this reduction.

Counting inputs with short dual vectors. We bound this event by expressing its constraints through bounded integer parameters. Let $\mathbf{w}$ satisfy the two short-vector conditions above. For every nonconstant monomial M recorded by Φ, we have $\|\Phi(M)\|_2 = X^{\deg M}$. Using $\Phi(pM) = p\Phi(M)$, Cauchy–Schwarz gives

$$\big|\langle \Phi(pM), \mathbf{w} \rangle\big| = p\big|\langle \Phi(M), \mathbf{w} \rangle\big| < \frac{X^{\deg M}}{u}.$$

This quantity is an integer whenever $\Phi(pM) \in \Lambda$. For $M = x_1$, it is nonzero because $\Phi(px_1) = \mathbf{v}_{\text{tar}}$. Linear and quadratic monomials give the scales X/u and X^2/u, respectively.

Each application uses its polynomial identities and error bounds to obtain necessary congruences on the sampled values, indexed by bounded integer parameters. For each fixed parameter tuple, we bound the probability that these congruences hold, then sum over the tuples. Every input admitting such a vector satisfies the congruences for at least one tuple, so this union bound covers all possible short dual vectors. Sections 4 to 6 derive the parameter ranges and count the satisfying inputs under their respective sampling models. The applications also bound the probability of a second consistent secret, so the exact-output guarantee yields unique recovery with high probability in the stated ranges.

4 MIHNP

In MIHNP, the public values v_i approximate $z_i^* = [(y^* + t_i)^{-1}]_p$, where y^* is hidden and the shifts t_i are public. Knowing x_1^* fixes $z_1^* = v_1 + x_1^*$ and hence $y^* = [(z_1^*)^{-1} - t_1]_p$. We apply Section 3 by constructing the polynomials P_i and the reconstructor $\mathcal{R}$ defined below, then bounding $\mathbb{E}[Z_{\text{target}}^N]$ under the sampling rule below.

Definition 4.1 (MIHNP)**.** Let p be prime and let N, X be integers with $3 \leq N < p$ and $1 \leq X < p$. The public input consists of distinct shifts $t_1, \ldots, t_N \in \mathbb{Z}_p$ and approximations $\mathbf{v} \in \mathbb{Z}_p^N$. Put $\mathcal{B}_i = \{0, \ldots, X-1\}$ for $i \in [N]$ and $\mathcal{B} = \prod_{i=1}^N \mathcal{B}_i$. The input is promised to have a hidden secret $y^* \in \mathbb{Z}_p$ and $\mathbf{x}^* \in \mathcal{B}$ such that

$$(y^* + t_i)(v_i + x_i^*) \equiv 1 \pmod{p}, \qquad v_i + x_i^* \in \mathbb{Z}_p \qquad (i \in [N]). \tag{4}$$

The goal is to recover y^*. For leading-bit observations, X is a power of two and $v_i \in X\mathbb{Z}$ for every i.

For the average-case analysis, fix $y^* \in \mathbb{Z}_p$ and sample $t_1, \dots, t_N$ uniformly without replacement from $\mathbb{Z}_p \setminus \{[-y^*]_p\}$. Set

$$z_i^* = [(y^* + t_i)^{-1}]_p, \qquad v_i = X\lfloor z_i^*/X \rfloor, \qquad x_i^* = z_i^* - v_i \qquad (i \in [N]).$$

All inverses in this section are taken modulo p. The map $t \mapsto [(y^* + t)^{-1}]_p$ is a bijection from the sampling domain onto $\mathbb{Z}_p \setminus \{0\}$, so $z_1^*, \dots, z_N^*$ are uniform distinct nonzero residues. All probabilities and expectations below use this sampling rule.

Theorem 4.2 (MIHNP recovery)**.** *Fix $N \geq 3$ and $A \geq 0$. Let k_N be the nonnegative integer satisfying $k_N(k_N+1) \leq N < (k_N+1)(k_N+2)$, and put*

$$\xi_N = \frac{N + k_N^2 - 1}{2(N + k_N(k_N+1))}. \tag{5}$$

Algorithm 1, with the polynomials and reconstructor below, deterministically returns exactly the secrets satisfying (4) *with $\mathbf{x} \in \mathcal{B}$ on every promised input. Its output contains y^* and has at most X elements. It uses $k = N$ error variables and a lattice of rank $m = N(N+1)/2$ in ambient dimension $d = m$.*

For sufficiently large p and $X \leq p^{\xi_N}(\log p)^A$, its expected bit running time is $\mathrm{poly}_{N,A}(\log p)$, and its output equals $\{y^*\}$ with probability at least $1 - O_N(X^N/p^{N-1}) = 1 - o(1)$. The expectation and probability are over the shifts sampled above, uniformly for every fixed y^*.

4.1 Recovery data and lemmas

Quadratic relations and the lattice. Let $y \in \mathbb{Z}_p$ and $\mathbf{x} \in \mathcal{B}$ satisfy (4), with $y, \mathbf{x}$ in place of $y^*, \mathbf{x}^*$, and write $z_i = v_i + x_i$. Subtracting $(y + t_i) \equiv z_i^{-1} \pmod p$ and $(y + t_j) \equiv z_j^{-1} \pmod p$ gives

$$(t_i - t_j) z_i z_j + z_i - z_j \equiv 0 \pmod p \qquad (1 \leq i < j \leq N).$$

For each $i < j$, let $a_{ij} \in (-p/2, p/2] \cap \mathbb{Z}$ represent $(t_i - t_j)^{-1}$ modulo p. Multiplying by a_{ij} yields the pair polynomials of [XSH+19, Section 2.3]:

$$P_{ij}(\mathbf{x}) = (v_i + x_i)(v_j + x_j) + a_{ij}(v_i + x_i - v_j - x_j) \qquad (i < j).$$

Take the P_{ij}, followed by $px_2, \dots, px_N, px_1$, as the entries of $\mathbf{P}$. Thus $P_m = px_1$, and every entry vanishes modulo p at each $\mathbf{x}$ satisfying the input constraints. All entries have degree at most two. Each P_{ij} has coefficient one on $x_i x_j$, and no other entry has that quadratic monomial. In any zero linear combination of the nonconstant coefficient vectors, the quadratic coordinates therefore force the coefficients multiplying the P_{ij} to be zero. The remaining px_i have distinct linear coordinates. Thus the nonconstant coefficient vectors $\Phi(P_i)$ from Definition 2.3 are linearly independent, with

$$k = N, \qquad m = d = \binom{N}{2} + N = \frac{N(N+1)}{2}.$$

Using the scaled coefficient map Φ from Definition 2.3, set

$$\Lambda = \mathrm{span}_{\mathbb{Z}}\{\Phi(P_h) : h \in [m]\}.$$

Its quadratic and linear coefficients already carry weights X^2 and X, respectively. Let $J \in \mathbb{Z}^{\binom{N}{2} \times N}$ be the matrix of linear coefficients of the P_{ij}:

$$J_{(i,j),i} = v_j + a_{ij}, \qquad J_{(i,j),j} = v_i - a_{ij}, \qquad J_{(i,j),h} = 0 \quad (h \notin \{i, j\}).$$

In the monomial order $(x_ix_j)_{i<j}, x_1, \dots, x_N$, a basis matrix of Λ is

$$\begin{pmatrix} X^2 I_{\binom{N}{2}} & XJ \\ 0 & pXI_N \end{pmatrix}. \tag{6}$$

The px_i rows are reordered here only to display the identity block; the algorithm uses px_1 last. The vector $\mathbf{v}_{\mathrm{tar}}$ representing px_1 has entry pX in the x_1 coordinate and zero elsewhere. Constructing these polynomials and their scaled coefficients takes $\mathrm{poly}_N(\log p)$ bit time.

Reconstructing y from a proposal for x_1. Given $a \in \mathcal{B}_1$, first check $v_1 + a \in \mathbb{Z}_p \setminus \{0\}$. If this fails, set $\mathcal{R}(a) = \varnothing$ and stop. Otherwise compute $\widehat{y} = [(v_1 + a)^{-1} - t_1]_p$. If $\widehat{y} + t_i \equiv 0 \pmod p$ for any $i \in [N]$, set $\mathcal{R}(a) = \varnothing$ and stop. Otherwise, for every $i \in [N]$, compute $\widehat{z}_i = [(\widehat{y} + t_i)^{-1}]_p$ and $\widehat{x}_i = \widehat{z}_i - v_i$. Set $\mathcal{R}(a) = \{\widehat{y}\}$ if $\widehat{\mathbf{x}} \in \mathcal{B}$, and $\mathcal{R}(a) = \varnothing$ otherwise. These inverse and interval checks take $\mathrm{poly}_N(\log p)$ bit time.

Lemma 4.3 (Reconstruction and uniqueness)**.** *On every promised input, $\mathcal{R}$ is a deterministic reconstructor with $r = 1$ in the sense of Definition 3.1. In particular, $\mathcal{R}(x_1^*) = \{y^*\}$. Under the sampling rule above, the probability that some $y \neq y^*$ satisfies* (4) *with an error vector in $\mathcal{B}$ is at most*

$$\frac{4^N (X-1)^N}{(p-1)^{N-1}}.$$

From Z_{target} to the expected cost. Use Z_{target} from Section 3.3 for the lattice (6) and $\mathbf{v}_{\mathrm{tar}}$ above. Since $k = N$, Theorem 3.4 bounds the cost by a polynomial in $\log p$ times Z_{target}^N. We therefore need a bound on $\mathbb{E}[Z_{\mathrm{target}}^N]$. The following two lemmas bound its tail and control that bound throughout $1 \le u \le X$.

Lemma 4.4 (Tail bound for short dual vectors)**.** *Fix $N \ge 3$ and integers $X = X(p) \ge 1$ with $X^2 = o(p)$. Under the sampling rule above, for sufficiently large p and every $1 \le u \le X$, $\Pr[Z_{\mathrm{target}} > u] \le \Psi_X(u)$, where $E_j(u) = (X^2/(pu))^j + p^{-1/2}$ for $j \in [N]$ and*

$$\Psi_X(u) = C_N \sum_{\kappa=1}^{N} (X/u)^\kappa \prod_{j=1}^{\kappa-1} E_j(u)\, E_\kappa(u)^{N-\kappa}. \tag{7}$$

C_N depends only on N.

In the proof of Lemma 4.4, κ counts the distinct nonzero values among the integer pairings $\langle \Phi(px_i), \mathbf{w} \rangle$, $i \in [N]$, for a short dual vector $\mathbf{w}$. There are $O_N((X/u)^\kappa)$ choices for these pairing values, while the $E_j(u)$ factors bound successive conditional probabilities for the sampled inverse values.

Lemma 4.5 (Endpoint bounds)**.** *For fixed $N \ge 3$, $A \ge 0$, sufficiently large p, and $1 \le X \le p^{\xi_N} (\log p)^A$, there is $B = \mathrm{poly}_{N,A}(\log p)$ such that*

$$\Psi_X(1) \le B, \qquad X^N \Psi_X(X) \le B.$$

Moreover, $\Psi_X(u)$ is a finite sum of powers of u^{-1} with positive coefficients.

The positive-coefficient form allows the bounds at $u = 1, X$ to control $u^N \Psi_X(u)$ between them. We first use this fact to prove Theorem 4.2.

Proof of Theorem 4.2. The construction of $\mathbf{P}$ and Lemma 4.3 verify the hypotheses of Theorem 3.4. Thus the algorithm returns exactly the secrets satisfying (4) with $\mathbf{x} \in \mathcal{B}$. Each $a \in \mathcal{B}_1$ determines at most one secret, so the output has at most X elements. Since m, d depend only on N, the per-input cost is at most

$$\mathrm{poly}_N(\log p)\big[1 + \min\{X, (KZ_{\mathrm{target}})^N\}\big] \le \mathrm{poly}_N(\log p) Z_{\mathrm{target}}^N,$$

where K is the constant in (2). It remains to average over the shifts.

The stated bound on X gives $X^2 = o(p)$ because $\xi_N < 1/2$, so Lemma 4.4 applies. By Lemma 4.5, write $\Psi_X(u) = \sum_{\ell=1}^{L} c_\ell u^{-r_\ell}$ for a finite L, with $c_\ell > 0$ and $r_\ell \ge 0$. For $X > 1$ and $1 < u < X$, put $t = \log u / \log X$. Hölder's inequality gives

$$\begin{aligned} u^N \Psi_X(u) &= \sum_{\ell=1}^{L} c_\ell^{1-t} \big(c_\ell X^{N-r_\ell}\big)^t \\ &\le \left(\sum_{\ell=1}^{L} c_\ell\right)^{1-t} \left(\sum_{\ell=1}^{L} c_\ell X^{N-r_\ell}\right)^t \\ &= \Psi_X(1)^{1-t} \big(X^N \Psi_X(X)\big)^t \le B. \end{aligned}$$

The endpoint bounds give the same inequality at $u = 1, X$. Using $1 \le Z_{\mathrm{target}} \le X$ and the tail-integral identity from Section 3.3, we obtain

$$\begin{aligned} \mathbb{E}[Z_{\mathrm{target}}^N] - 1 &= N \int_1^X u^{N-1} \Pr[Z_{\mathrm{target}} > u]\, du \\ &\le N \int_1^X u^{N-1} \Psi_X(u)\, du \le NB \log X = \mathrm{poly}_{N,A}(\log p). \end{aligned}$$

For $X = 1$, $Z_{\mathrm{target}} = 1$. This proves the expected running-time bound.

The output contains y^*. By Lemma 4.3, the probability of any additional secret is $O_N(X^N/p^{N-1})$. This tends to zero in the stated range because $N\xi_N < N - 1$ for $N \ge 3$. □

4.2 Proofs of the lemmas

Proof of Lemma 4.3. Every $\widehat{y} \in \mathcal{R}(a)$ has canonical inverse values $\widehat{z}_i$ with $\widehat{z}_i - v_i \in \mathcal{B}_i$, so it satisfies the input constraints. Conversely, fix $y \in \mathbb{Z}_p$ and $\mathbf{x} \in \mathcal{B}$ satisfying those constraints. At $a = x_1$, the procedure computes $\widehat{y} = [(v_1 + x_1)^{-1} - t_1]_p = y$. For each i, it then obtains $\widehat{z}_i = v_i + x_i$ and $\widehat{x}_i = x_i$, so $\mathcal{R}(x_1) = \{y\}$. Together with the polynomial bit cost and canonical encoding, this verifies Definition 3.1 with $r = 1$.

For uniqueness, fix $y' \in \mathbb{Z}_p \setminus \{y^*\}$. If y' satisfies the input constraint at a shift t, the canonical inverses $z' = [(y' + t)^{-1}]_p$ and $z^* = [(y^* + t)^{-1}]_p$ lie in the same interval of X consecutive integers. Their difference $\delta = z' - z^*$ therefore satisfies $0 < |\delta| \le X - 1$ and

$$\delta(t + y')(t + y^*) \equiv y^* - y' \pmod{p}.$$

For each such δ, this is a quadratic congruence in t with nonzero leading coefficient modulo p, since $0 < |\delta| < p$. It therefore has at most two solutions. There are $2(X - 1)$ possible integers δ, so at most $4(X - 1)$ shifts can satisfy this necessary condition for the fixed y'.

The N shifts are sampled uniformly without replacement from $\mathbb{Z}_p \setminus \{[-y^*]_p\}$, a set of size $p - 1$. The probability that all sampled shifts belong to a fixed subset is no larger than under sampling

with replacement. Thus the probability that this y' satisfies every input constraint is at most

$$\left(\frac{4(X-1)}{p-1}\right)^N.$$

A union bound over the $p-1$ choices of $y' \neq y^*$ gives

$$\Pr[\text{some } y' \neq y^* \text{ satisfies all input constraints}] \leq \frac{4^N(X-1)^N}{(p-1)^{N-1}},$$

as claimed. □

Counting simultaneous rational constraints. The proof of Lemma 4.4 will reduce the event $Z_{\text{target}} > u$ to inequalities $|f_h(z)|_p \leq R$ for several rational functions of one sampled inverse z. Here $|a|_p = \min\{[a]_p, p - [a]_p\}$. The next lemma bounds the probability of these inequalities using Cochrane–Pinner [CP06, Theorem 1.1].

A rational function is a quotient of polynomials, evaluated where its denominator is nonzero. A finite pole of order h at β means that, after cancellation, the function has the form $(z-\beta)^{-h}g(z)$ with g defined and nonzero at β. Pole orders are considered over an algebraic closure. At infinity, use $t = 1/z$ and take the pole order of $f(1/t)$ at $t = 0$. A pole of order one is called *simple*.

Lemma 4.6 (Bounded rational values)**.** *Fix integers $j, D \geq 1$. Let $f_1, \ldots, f_j$ be rational functions modulo p whose reduced numerators and denominators have degree at most D. Suppose every linear combination with coefficients not all zero modulo p is nonconstant and has only simple poles, including at infinity. Let $\Omega \subseteq \mathbb{Z}_p$ consist of points where all f_h are defined, with $|\mathbb{Z}_p \setminus \Omega| \leq D$. For sufficiently large p, an integer $1 \leq R \leq (p-1)/4$, and z uniform on Ω,*

$$\Pr[|f_h(z)|_p \leq R \text{ for every } h \in [j]] \ll_{j,D} (R/p)^j + p^{-1/2}.$$

Proof of Lemma 4.6. Let $\mathcal{U} \subseteq \mathbb{Z}_p$ be the set where all f_h are defined. The denominators exclude at most jD residues, and $\Omega \subseteq \mathcal{U}$ with $|\Omega| \geq p - D \geq p/2$ for sufficiently large p.

Put $I = \{[-R]_p, [-R+1]_p, \ldots, [R]_p\}$ and $H = 2R+1$. We bound its indicator 1_I by a nonnegative function β whose Fourier coefficients have bounded absolute sum. The product $\prod_{h=1}^{j} \beta(f_h(z))$ is then an upper bound for the indicator that all j constraints hold simultaneously. For a set $A \subseteq \mathbb{Z}_p$, $\mathbf{1}_A$ is one on A and zero elsewhere. Write $e_p(x) = \exp(2\pi \mathrm{i}[x]_p/p)$. For functions on $\mathbb{Z}_p$, use the convolution and Fourier conventions

$$(f * g)(x) = \sum_{a \in \mathbb{Z}_p} f(a)g([x-a]_p), \qquad \widehat{\beta}(t) = \frac{1}{p}\sum_{x \in \mathbb{Z}_p} \beta(x)e_p(-tx).$$

When $H \leq p/4$, take $A = \{0, \ldots, 2H-1\}$ and set

$$\beta(x) = \frac{1}{H}(\mathbf{1}_A * \mathbf{1}_{-A})([x+R]_p), \qquad -A = \{[-a]_p : a \in A\}.$$

For $x \in I$, the overlap counted by the convolution is at least H. Thus $\beta \geq \mathbf{1}_I$ and $\beta \geq 0$. The convolution formula and Parseval's identity give

$$\widehat{\beta}(0) = \frac{4H}{p}, \qquad |\widehat{\beta}(t)| = \frac{|\sum_{x \in A} e_p(-tx)|^2}{Hp}, \qquad \sum_{t \in \mathbb{Z}_p} |\widehat{\beta}(t)| = 2.$$

When $H > p/4$, take $\beta = 1$. Then $\beta \geq \mathbf{1}_I$, its mean is at most $4H/p$, and its Fourier coefficients have absolute sum one. In both cases,

$$\widehat{\beta}(0) = O(R/p), \qquad \sum_{t \in \mathbb{Z}_p} |\widehat{\beta}(t)| \leq 2.$$

For $\mathbf{c} = (c_1, \ldots, c_j) \in \mathbb{Z}_p^j \setminus \{\mathbf{0}\}$, put $f_{\mathbf{c}} = \sum_{h=1}^{j} c_h f_h$ modulo p. By hypothesis, $f_{\mathbf{c}}$ is nonconstant and has only simple poles. The pure additive case of Cochrane–Pinner [CP06, Theorem 1.1] therefore gives

$$\left| \sum_{z \in \mathcal{U}} e_p(f_{\mathbf{c}}(z)) \right| = O_{j,D}(\sqrt{p}).$$

The pole orders, including the order at infinity when present, are one and hence prime to p. Restricting the sum from the finite points where $f_{\mathbf{c}}$ is defined to $\mathcal{U}$ removes at most jD terms.

Fourier inversion expands $\prod_{h=1}^{j} \beta(f_h(z))$ as a sum over $\mathbf{c} \in \mathbb{Z}_p^j$. The term $\mathbf{c} = \mathbf{0}$ contributes $|\mathcal{U}|\widehat{\beta}(0)^j = O_j(p(R/p)^j)$. For all other terms, the preceding character-sum bound and the absolute sum of the Fourier coefficients give

$$\begin{aligned} \sum_{z \in \mathcal{U}} \prod_{h=1}^{j} \beta(f_h(z)) &\leq O_j(p(R/p)^j) + O_{j,D}(\sqrt{p}) \left(\sum_{t \in \mathbb{Z}_p} |\widehat{\beta}(t)| \right)^j \\ &\ll_{j,D} p(R/p)^j + \sqrt{p}. \end{aligned}$$

Since $\beta \geq \mathbf{1}_I$ and the summands are nonnegative, the desired probability is at most this sum divided by $|\Omega|$. Using $|\Omega| \geq p/2$ proves the lemma. □

Proof of Lemma 4.4. Fix $1 \leq u \leq X$. By Section 3.3, $Z_{\mathrm{target}} > u$ implies that some $\mathbf{w} \in \Lambda^*$ satisfies

$$\|\mathbf{w}\|_2 < \frac{1}{pu}, \qquad \langle \mathbf{v}_{\mathrm{tar}}, \mathbf{w} \rangle \neq 0.$$

Write $\mathbf{w} = (\mathbf{w}_{\mathrm{quad}}, \mathbf{w}_{\mathrm{lin}})$ in the monomial order of (6). The inner product with every row of that matrix is an integer. Its lower block gives $pX\mathbf{w}_{\mathrm{lin}} = \mathbf{s} \in \mathbb{Z}^N$. Its upper block then gives $X^2\mathbf{w}_{\mathrm{quad}} + J\mathbf{s}/p = \mathbf{h} \in \mathbb{Z}^{\binom{N}{2}}$. Consequently,

$$\mathbf{w} = \left(\frac{p\mathbf{h} - J\mathbf{s}}{pX^2}, \frac{\mathbf{s}}{pX} \right), \qquad \langle \mathbf{v}_{\mathrm{tar}}, \mathbf{w} \rangle = s_1 \neq 0.$$

Each coordinate has magnitude less than $1/(pu)$. Thus $|s_i| < X/u$ and, using row (i,j) of J,

$$|(v_j + a_{ij})s_i + (v_i - a_{ij})s_j|_p < \frac{X^2}{u} \qquad (i < j).$$

Substituting $v_i = z_i^* - x_i^*$ separates the expression into

$$(z_j^* + a_{ij})s_i + (z_i^* - a_{ij})s_j - (x_j^* s_i + x_i^* s_j).$$

The last term has magnitude at most $2X^2/u$. Moreover, $a_{ij} \equiv z_i^* z_j^*/(z_j^* - z_i^*) \pmod{p}$ by the inverse relation. The triangle inequality for $|\cdot|_p$ therefore gives

$$\left| \frac{s_i(z_j^*)^2 - s_j(z_i^*)^2}{z_j^* - z_i^*} \right|_p \leq \frac{3X^2}{u} \qquad (i < j). \tag{8}$$

The denominators are nonzero because the z_i^* are distinct. Fractions in this formula are evaluated modulo p.

We bound the probability that some integer vector $\mathbf{s}$ with $s_1 \neq 0$ and $|s_i| < X/u$ satisfies (8). For each such vector, let κ be the number of its distinct nonzero values. There are $O_N(1)$ ways to specify which coordinates are zero and which have equal nonzero values. For each such specification, choosing the κ distinct nonzero integers gives $O_N((X/u)^\kappa)$ vectors $\mathbf{s}$. These integers are also distinct modulo p for sufficiently large p, since their differences have magnitude less than $2X = o(p)$.

Fix one vector $\mathbf{s}$ before sampling the inverse values. Choose indices $i_1, \dots, i_\kappa$, one for each distinct nonzero value, and put $a_h = s_{i_h}$. First condition successively on $\beta_h = z_{i_h}^*$ for $h = 1, \dots, \kappa$, then on the other inverse values in increasing index order. This order depends only on $\mathbf{s}$. Suppose j of the β_h are already fixed and the next inverse value is $z = z_i^*$, with $b = s_i$. Its constraints with the fixed β_h are $|f_h(z)|_p \leq 3X^2/u$ for $h \in [j]$, where

$$f_h(z) = \frac{a_h z^2 - b\beta_h^2}{z - \beta_h} = a_h(z + \beta_h) + \frac{(a_h - b)\beta_h^2}{z - \beta_h}.$$

For $a_h \neq b$, the function has a simple pole at β_h with nonzero coefficient. These points are distinct, so such poles cannot cancel in a linear combination of the f_h. There is at most one index with $a_h = b$; its function is $a_h(z+\beta_h)$ with $a_h \neq 0$. Thus every nonzero linear combination is nonconstant. All finite poles are simple, and the polynomial part has degree at most one, so any pole at infinity is also simple.

Given all previously fixed inverse values, z is uniform on the nonzero residues excluding those values. This set omits at most N residues and contains no pole of the f_h. For $j \geq 1$, apply Lemma 4.6 with $D = N + 1$ and $R = \lceil 3X^2/u \rceil$. Since $1 \leq X^2/u$ and $X^2 = o(p)$, $R \leq 4X^2/u = o(p)$ uniformly for $1 \leq u \leq X$. For every choice of the preceding inverse values,

$$\Pr\big[|f_h(z)|_p \leq 3X^2/u \text{ for all } h \in [j] \mid \text{preceding inverse values}\big] \leq C_N E_j(u).$$

For $z_{i_1}^*, \dots, z_{i_\kappa}^*$, the numbers of these constraints are $0, 1, \dots, \kappa - 1$. Each of the remaining $N - \kappa$ inverse values has κ constraints with $\beta_1, \dots, \beta_\kappa$. Successive conditioning therefore bounds the probability for this fixed $\mathbf{s}$ by

$$C_N \prod_{j=1}^{\kappa-1} E_j(u)\, E_\kappa(u)^{N-\kappa}.$$

We used only constraints with these selected indices; satisfying all of (8) implies satisfying them. A union bound over the integer vectors $\mathbf{s}$, grouped by κ, gives (7). □

Proof of Lemma 4.5. First consider $1 \leq X \leq p^{\xi_N}$. Write $X = p^\xi$ with $0 \leq \xi \leq \xi_N < 1/2$, and put $g_j = \min\{j(1 - 2\xi), 1/2\}$. At $u = 1$,

$$E_j(1) = p^{-j(1-2\xi)} + p^{-1/2} \leq 2p^{-g_j}.$$

Hence the κth summand of $\Psi_X(1)$ is at most $O_N(p^{e_\kappa})$, where

$$e_\kappa = \kappa\xi - \sum_{j=1}^{\kappa-1} g_j - (N - \kappa)g_\kappa.$$

The increments of g_j are nonnegative and nonincreasing. It follows that

$$e_{\kappa+1} - e_\kappa = \xi - (N - \kappa - 1)(g_{\kappa+1} - g_\kappa)$$

is nondecreasing in κ. The maximum of $e_1, \dots, e_N$ therefore occurs at $\kappa = 1$ or $\kappa = N$. Since $g_j \le 1/2$ and $g_1 = \min\{1 - 2\xi, 1/2\} \ge 1/2 - \xi$, we have $g_j - g_1 \le \xi$ and hence

$$e_N - e_1 = (N-1)\xi - \sum_{j=1}^{N-1} (g_j - g_1) \ge 0.$$

Thus all $e_\kappa \le 0$ exactly when

$$N\xi \le \sum_{j=1}^{N-1} \min\{j(1-2\xi), 1/2\}. \tag{9}$$

The left side comes from the X^N choices when all entries of $\mathbf{s}$ are distinct and nonzero. The right side sums the probability gains from the successive rational constraints; the cap $1/2$ is the exponent of their $p^{-1/2}$ error term.

The right side of (9) minus the left side decreases strictly with ξ. At $\xi = \xi_N$, the defining inequalities for k_N give

$$k_N(1 - 2\xi_N) \le \frac{1}{2} \le (k_N + 1)(1 - 2\xi_N).$$

Consequently,

$$\begin{aligned} \sum_{j=1}^{N-1} \min\{j(1-2\xi_N), 1/2\} &= \frac{k_N(k_N+1)}{2}(1-2\xi_N) + \frac{N-1-k_N}{2} \\ &= N\xi_N. \end{aligned}$$

This proves $\Psi_X(1) = O_N(1)$ for $X \le p^{\xi_N}$. After expanding the factors in (7), each term is a positive power of X of degree at most N^2, multiplied by a factor depending on p, N. Allowing $X \le p^{\xi_N}(\log p)^A$ therefore increases this bound by at most $(\log p)^{AN^2}$.

For the other endpoint, $E_j(X) = (X/p)^j + p^{-1/2} \le 2p^{-1/2}$ for sufficiently large p in the stated range. Each summand of $\Psi_X(X)$ contains $N - 1$ such factors, so

$$X^N \Psi_X(X) = O_N\big(X^N p^{-(N-1)/2}\big).$$

Equation (9) at ξ_N gives $N\xi_N \le (N-1)/2$. Thus this endpoint is also bounded by $\mathrm{poly}_{N,A}(\log p)$. Finally, expanding (7) gives a finite sum of powers of u^{-1} with positive coefficients. □

4.3 Recovery from leading bits

If b low bits of each z_i^* are unrevealed, take $X = 2^b$. The bound $X \le p^{\xi_N}$ therefore gives a sufficient leading-bit fraction $1 - \xi_N$, up to rounding in the fixed $\lceil \log_2 p \rceil$-bit representation. The following choice of N makes this fraction at most $1/2 + \varepsilon$.

Corollary 4.7 (MIHNP recovery from leading bits)**.** *For every fixed $0 < \varepsilon < 1/2$ and sufficiently large p, Algorithm 1 has expected bit cost $\mathrm{poly}_\varepsilon(\log p)$ and returns $\{y^*\}$ with probability $1 - o(1)$ whenever $X \le p^{1/2-\varepsilon}$ under the sampling rule above. These guarantees hold uniformly for every fixed secret, using $N = O(\varepsilon^{-2})$ samples and lattice rank $O(\varepsilon^{-4})$. Thus a $1/2 + \varepsilon$ fraction of leading bits suffices, up to rounding, to recover y^*.*

Proof. Choose

$$q = \max\left\{2, \left\lfloor \frac{1}{4\varepsilon} \right\rfloor + 1\right\}, \qquad N = q(q+1).$$

Then $k_N = q$ and

$$\xi_N = \frac{1}{2} - \frac{1}{4q} > \frac{1}{2} - \varepsilon.$$

Apply Theorem 4.2 with $A = 0$. Since $q = O(\varepsilon^{-1})$, this uses $N = O(\varepsilon^{-2})$ samples and a lattice of rank $N(N+1)/2 = O(\varepsilon^{-4})$. □

5 EC-HNP

For finite EC-HNP inputs, the public values v_i approximate $z_i^* = \pi_1(y^* + Q_i)$ modulo p, where y^* is hidden and the Q_i are public curve points. Knowing z_1^* gives at most two possibilities for $y^* + Q_1$, and subtracting Q_1 gives the corresponding candidates for y^*. We construct quadratic polynomials and two reconstruction branches for Section 3. To bound their expected recovery cost, we derive constraints on $y^* + Q_i$ from a short vector in the dual lattice. Two suitably chosen points then restrict each remaining point to at most eight choices for each fixed set of bounded integer parameters.

Let $p > 3$ be prime and let $\mathcal{E}/\mathbb{F}_p$ be the nonsingular curve

$$w^2 = \phi(z) = z^3 + a_{\mathcal{E}} z + b_{\mathcal{E}}.$$

Here $a_{\mathcal{E}}, b_{\mathcal{E}} \in \mathbb{Z}_p$. We represent finite coordinates by elements of $\mathbb{Z}_p$. Thus $\mathcal{E}(\mathbb{F}_p)$ consists of the point $\mathcal{O}$ and the pairs $(z, w) \in \mathbb{Z}_p^2$ satisfying the curve equation modulo p. Write $+$ for its group operation, with identity $\mathcal{O}$, and put $\pi_1(z, w) = z$, $\pi_2(z, w) = w$ for finite points. Arithmetic in coordinates is modulo p unless integer representatives or bounds are specified.

Definition 5.1 (EC-HNP)**.** The public parameters are $\mathcal{E}/\mathbb{F}_p$, a subgroup $G = \langle R_0 \rangle$ of known order M, and integers $N \geq 5$ and $1 \leq X < p$. The input consists of queries $Q_1, \ldots, Q_N \in G$ and, for each index i, either a finite observation with an integer v_i or an infinity observation. It is promised to have a hidden point $y^* \in \mathcal{E}(\mathbb{F}_p)$ satisfying these observations. For a finite observation, $y^* + Q_i \neq \mathcal{O}$ and

$$z_i^* = \pi_1(y^* + Q_i) \equiv v_i + x_i^* \pmod{p}, \qquad x_i^* \in \{0, \ldots, X-1\}. \tag{10}$$

An infinity observation is supplied exactly when $y^* + Q_i = \mathcal{O}$. The goal is to recover y^*.

For leading-bit observations, the input also specifies the integer interval determined by the revealed bits. It has left endpoint v_i, contains at most X canonical values, and contains z_i^*. Here X is chosen as a power of two. Then $z_i^* = v_i + x_i^*$ holds in $\mathbb{Z}$. Any promised subgroup or coset restriction on y^* is supplied with a polynomial-time verifier and is part of the input constraints. For fixed N, the total public encoding length, including the approximations and restriction description, is bounded by a fixed polynomial in $\log p$.

The queries must satisfy

$$Q_i \neq \mathcal{O}, \qquad Q_j - Q_i \notin \mathcal{E}[2] \quad (i \neq j), \qquad \mathcal{E}[2] = \{T \in \mathcal{E}(\overline{\mathbb{F}_p}) : 2T = \mathcal{O}\}. \tag{11}$$

Here $\overline{\mathbb{F}}_p$ is an algebraic closure. The finite points in $\mathcal{E}[2]$ may have coordinates in that closure. Negation sends (z, w) to $(z, -w)$, so $\mathcal{E}[2]$ consists of $\mathcal{O}$ and the points $(z, 0)$ with $\phi(z) = 0$. There are at most four such points in total. Condition (11) therefore makes the queries distinct and gives a nonzero second coordinate for every difference $Q_j - Q_i$. We call an input *finite* when all its observations are finite.

For the average-case analysis, fix y^* satisfying the promised restriction. Sample the queries independently and uniformly from G by scalar multiplication of R_0, rejecting tuples that fail (11).

The approximations may be functions of all the queries, provided they satisfy (10) for this y^* and, for leading bits, the specified integer intervals. Before rejection, a query equals $\mathcal{O}$ or $-y^*$ with probability at most $2/M$, and a pair difference lies in $\mathcal{E}[2]$ with probability at most $4/M$. Consequently, the probability of public rejection or an infinity observation is at most

$$\frac{2N}{M} + \frac{4\binom{N}{2}}{M} = \frac{2N^2}{M}.$$

For $M \geq 16N^2$, public acceptance has probability at least $7/8$. Infinity observations are retained and processed directly. All input probabilities below refer to the accepted tuples and the chosen approximation rule.

For finite $y^* + Q_i$, we use the same type of additive query access as [Sha17], whose algorithm uses a central query and symmetric pairs. Our Q_i are independent uniform samples from G before the public rejection in (11). Definition 5.1 explicitly specifies infinity reports, which we handle by checking $-Q_i$ against the original input constraints.

Theorem 5.2 (EC-HNP recovery)**.** *For fixed $N \geq 5$, Algorithm 1, with the reconstructor below and a preliminary check for infinity observations, is Las Vegas and returns exactly the points satisfying the input constraints on every admissible promised input. Its output contains y^* and has at most $2X$ elements. On finite inputs, it uses $k = N$ error variables and a lattice of rank $m = N(N+1)/2$ in ambient dimension $d = m$. If $M \geq 16N^2$ and $6X^2 < p$, its input-averaged expected bit cost is*

$$\mathrm{poly}_N(\log p)\left(1 + \min\{X, \Delta_N, \Delta_N^{N/(3N-3)}\}\right), \qquad \Delta_N = \frac{X^{5N-6}}{M^{N-2}}. \tag{12}$$

In particular, fix $A, c \geq 0$ and put $\rho_N = (N-2)/(5N-6)$. If $M \geq \max\{16N^2, p/(\log p)^c\}$ and $X \leq p^{\rho_N}(\log p)^A$, the expected cost is $\mathrm{poly}_{N,A,c}(\log p)$ and the output equals $\{y^*\}$ with probability $1 - o(1)$ as $p \to \infty$. These guarantees hold uniformly for every fixed y^* and every approximation rule in the sampling model above.

5.1 Recovery data and lemmas

Quadratic relations and the lattice. For finite inputs, put $\mathcal{B}_i = \{0, \ldots, X-1\}$ and $\mathcal{B} = \prod_{i=1}^N \mathcal{B}_i$, so $k = N$. Define the public difference coordinates

$$a_{ij} = \pi_1(Q_j - Q_i), \qquad b_{ij} = \pi_2(Q_j - Q_i) \neq 0 \qquad (i \neq j).$$

They satisfy $a_{ji} = a_{ij}$ and $b_{ji} = -b_{ij}$ modulo p. For variables $z_1, \ldots, z_N$, put

$$F_{ij} = z_i z_j - a_{ij}(z_i + z_j).$$

After substituting $z_h = \pi_1(y + Q_h)$, the subtraction in F_{ij} cancels the double-pole terms at $-Q_i$ and $-Q_j$. The remaining poles are simple, and the following combination cancels them to obtain a constant.

Lemma 5.3 (Quadratic source polynomials)**.** *For every $i < j < N$, there is a publicly computable $\kappa_{ij} \in \mathbb{Z}_p$ such that*

$$F_{ij} - \frac{b_{ij}}{b_{iN}} F_{iN} + \frac{b_{ij}}{b_{jN}} F_{jN} = \kappa_{ij} \tag{13}$$

is an identity of rational functions modulo p after substituting $z_h = \pi_1(y + Q_h)$. The constant κ_{ij} is computed by evaluating the identity at $y = \mathcal{O}$, since every Q_h is finite. In particular, it holds

at every y for which these translates are finite. Subtract κ_{ij}, lift the coefficients to their canonical integer representatives, and substitute $z_h = v_h + x_h$ to obtain P_{ij}. Together with px_hx_N for $h < N$ and px_h for $h \in [N]$, these give $m = N(N+1)/2$ polynomials of degree at most two. They vanish modulo p at every $\mathbf{x} \in \mathcal{B}$ associated with a point satisfying the input constraints. Their nonconstant coefficient vectors are linearly independent over $\mathbb{R}$.

Place px_1 last in $\mathbf{P}$, as required in Section 3. Using the scaled coefficient map Φ from Definition 2.3, set

$$\Lambda = \operatorname{span}_{\mathbb{Z}}\{\Phi(P_h) : h \in [m]\}.$$

Its quadratic and linear coefficients already carry weights X^2 and X, respectively. Its nonconstant monomials are x_ix_j for $i < j$ and x_i for $i \in [N]$, so

$$d = \binom{N}{2} + N = m.$$

The vector $\mathbf{v}_{\mathrm{tar}}$ representing px_1 has entry pX in the x_1 coordinate and zero elsewhere. The lifted coefficients and full polynomials, including their constant terms, can be constructed in $\mathrm{poly}_N(\log p)$ bit time.

Reconstructing y from a proposal for x_1. Before constructing Λ, check for infinity observations. An infinity observation at index i forces $y = -Q_i$. Check this point against every observation and the promised restriction, returning $\{-Q_i\}$ if all checks pass and $\varnothing$ otherwise.

For a finite input and a proposal $a \in \mathcal{B}_1$, put $\widehat{z}_1 = [v_1 + a]_p$. If $\phi(\widehat{z}_1)$ has no square root modulo p, set $\mathcal{R}(a) = \varnothing$ and stop. Otherwise compute any square root $\widehat{w} \in \mathbb{Z}_p$ and set

$$\widehat{y}_1 = (\widehat{z}_1, \widehat{w}) - Q_1, \qquad \widehat{y}_2 = (\widehat{z}_1, [-\widehat{w}]_p) - Q_1.$$

For each $j \in \{1, 2\}$, test $\widehat{y}_j$ as follows. Every $\widehat{y}_j + Q_i$ must be finite. Writing $\widehat{z}_i = \pi_1(\widehat{y}_j + Q_i)$, require

$$[\widehat{z}_i - v_i]_p \in \mathcal{B}_i \qquad (i \in [N]).$$

For leading bits, also require $\widehat{z}_i$ to lie in its original integer interval. Finally, apply the verifier for any promised restriction.

Let $\mathcal{R}(a)$ be the set of distinct points among $\widehat{y}_1, \widehat{y}_2$ that pass all these tests. Then $|\mathcal{R}(a)| \le 2$. Choosing the other square root only permutes the two proposed points and therefore does not affect $\mathcal{R}(a)$. Quadratic-residuosity testing and square-root extraction admit Las Vegas algorithms with expected bit cost polynomial in $\log p$ [Sho09, Sections 12.4–12.5]. All remaining tests have polynomial cost by the input encoding and verifier assumptions. Encode finite points by their canonical coordinates and use a separate encoding for $\mathcal{O}$.

Lemma 5.4 (Reconstruction and uniqueness)**.** *On every finite admissible promised input, $\mathcal{R}$ is a Las Vegas reconstructor with $r = 2$ in the sense of Definition 3.1. An infinity observation determines the secret directly. Under the sampling model above, if $M \ge 16N^2$, then*

$$\Pr[\textit{some } y \neq y^* \textit{ satisfies all input constraints}] = O_N\Big(p(X/M)^N\Big).$$

From Z_{target} to the expected cost. Use Z_{target} from Section 3.3 for Λ and $\mathbf{v}_{\text{tar}}$ above. Define

$$Y = \min\{X, Z_{\text{target}}^N\} \tag{14}$$

on finite inputs, and put $Y = 1$ on infinity inputs. The cap is $X = |\mathcal{B}_1|$: if unprocessed boxes remain after X visits, Algorithm 1 tests the untested $a \in \mathcal{B}_1$ directly. For fixed N, Theorem 3.4 therefore bounds the per-input expected cost by $\text{poly}_N(\log p)(1 + Y)$. It remains to bound $\mathbb{E}Y$ over the sampled queries. For $1 \le t < X$, the event $Y > t$ on finite inputs is equivalent to $Z_{\text{target}} > t^{1/N}$. The next lemma therefore states the short-dual estimate directly as a tail bound for Y.

Lemma 5.5 (Capped tail bound)**.** *Fix $N \ge 5$, $M \ge 16N^2$, and $6X^2 < p$. Under the sampling model above, for every $1 \le t < X$,*

$$\Pr[Y > t] \le \min\{1, C_N \Delta_N t^{-\beta}\}, \qquad \beta = \frac{3N-3}{N}, \tag{15}$$

uniformly over the fixed secret and the approximation rule. Here C_N depends only on N.

Proof of Theorem 5.2. On finite inputs, Lemmas 5.3 and 5.4 verify the polynomial and reconstruction assumptions of Theorem 3.4. Thus the algorithm returns exactly the points satisfying the input constraints. The direct check gives the same guarantee on infinity inputs. Each $a \in \mathcal{B}_1$ gives at most two points, so the output contains at most $2X$ points and includes y^*.

For finite inputs, Theorem 3.4 gives

$$\text{poly}_N(\log p)\big[1 + \min\{X, (KZ_{\text{target}})^N\}\big] \le \text{poly}_N(\log p)(1 + Y),$$

where K depends only on N. Infinity inputs have polynomial cost and $Y = 1$. For $X > 1$, Lemma 5.5 and the tail-integral identity give, with $d_0 = C_N \Delta_N$,

$$\mathbb{E}Y - 1 \le \int_1^X \min\{1, d_0 t^{-\beta}\}\, dt \ll_N \min\{X, d_0, d_0^{1/\beta}\}.$$

For $d_0 \le 1$, integrating $d_0 t^{-\beta}$ gives $O_N(d_0)$. For $d_0 > 1$, splitting at $d_0^{1/\beta}$ gives $O_N(d_0^{1/\beta})$. The integral is also at most X. For $X = 1$, $Y = 1$. These bounds prove (12). The expectation includes both the sampled input and the internal randomness in reconstruction.

In the stated range,

$$\Delta_N \le (\log p)^{A(5N-6)+c(N-2)}.$$

Since $\rho_N < 1/5$, the condition $6X^2 < p$ holds for sufficiently large p. By Lemma 5.4, the probability of an additional output is at most $p^{1-N(1-\rho_N)}$ times a fixed power of $\log p$. This tends to zero for $N \ge 5$. □

5.2 Proofs of the lemmas

Rational functions on the curve. We use the following standard facts about rational functions on $\mathcal{E}$. A local parameter t is a formal coordinate vanishing to order one at a point. A pole of order h means that the formal Laurent expansion starts with ct^{-h} for $c \ne 0$. The functions $z = \pi_1$ and $w = \pi_2$ have their only poles at $\mathcal{O}$, of orders two and three. For a nonzero rational function, the total orders of its zeros and poles are equal, counted with multiplicity over the algebraic closure by [Sut25, Section 23.1, Corollary 23.10]. Thus a nonconstant rational function with total pole order at most D takes a prescribed value at at most D points. A rational function with no poles is constant.

For the local calculations, let $T = (z, w)$ and let $D = (a, b)$ be a fixed finite point. The addition law from [Sil09, Sections III.2 and IV.1] gives

$$\lambda = \frac{w-b}{z-a}, \qquad \pi_1(T+D) = \lambda^2 - z - a, \qquad \pi_2(T+D) = -w + \lambda\big(z - \pi_1(T+D)\big).$$

These are rational identities modulo p. At $\mathcal{O}$, use $t = -z/w$ and write $O(t^h)$ for terms of formal degree at least h. The curve equation gives $t^{-2} = z + a_{\mathcal{E}}/z + b_{\mathcal{E}}/z^2$, hence

$$z = t^{-2} - a_{\mathcal{E}}t^2 + O(t^4), \qquad w = -t^{-3} + a_{\mathcal{E}}t + O(t^3).$$

It follows that

$$\lambda = -t^{-1} - at - bt^2 - a^2t^3 + O(t^4).$$

Substituting this expansion into the addition law yields

$$\begin{aligned} \pi_1(T+D) &= a + 2bt + \phi'(a)t^2 + O(t^3), \\ \pi_2(T+D) &= b + \phi'(a)t + O(t^2), \end{aligned} \tag{16}$$

where $\phi'(a) = 3a^2 + a_{\mathcal{E}}$. We use these expansions first to prove the quadratic identity and then to bound the number of points in the tail calculation.

Proof of Lemma 5.3. Regard F_{ij} as a rational function of y after substituting $z_h = \pi_1(y + Q_h)$. Its only possible poles are $-Q_i$ and $-Q_j$. Near $y = -Q_i$, put $T = y + Q_i$. Then $y + Q_j = T + (Q_j - Q_i)$, so Equation (16) gives

$$z_i = t^{-2} + O(t^2), \qquad z_j = a_{ij} + 2b_{ij}t + O(t^2).$$

Consequently,

$$F_{ij} = z_i(z_j - a_{ij}) - a_{ij}z_j = 2b_{ij}t^{-1} + O(1).$$

At $-Q_j$, the simple-pole coefficient is $-2b_{ij}$. The three terms $F_{ij} - (b_{ij}/b_{iN})F_{iN} + (b_{ij}/b_{jN})F_{jN}$ therefore cancel the pole at each of $-Q_i, -Q_j, -Q_N$. There are no other possible poles, so this function is a constant κ_{ij}. Every Q_h is finite, so evaluating at $y = \mathcal{O}$ computes κ_{ij} from public data.

After lifting and substituting $z_h = v_h + x_h$, the coefficient of x_ix_j in P_{ij} is one. The only other quadratic monomials in that polynomial are x_ix_N and x_jx_N. Consider any linear combination of the nonconstant coefficient vectors that equals zero. For each $i < j < N$, only the vector corresponding to P_{ij} has a nonzero x_ix_j coordinate. The coefficient of this vector in the linear combination must therefore be zero. The remaining polynomials px_hx_N and px_h have distinct monomials. Their vectors are independent as well, giving rank

$$\binom{N-1}{2} + (N-1) + N = \frac{N(N+1)}{2}.$$

Identity (13) holds for every point y whose translates are finite. Substitution of the errors associated with any point satisfying the input constraints therefore makes each P_{ij} zero modulo p. The appended polynomials are multiples of p. □

Proof of Lemma 5.4. Every point in $\mathcal{R}(a)$ passes all the input tests. Conversely, fix a point y and $\mathbf{x} \in \mathcal{B}$ satisfying the constraints on a finite input. At $a = x_1$, the procedure obtains $\widehat{z}_1 = [v_1 + x_1]_p = \pi_1(y + Q_1)$. Hence y is among the proposed points $\widehat{y}_1, \widehat{y}_2$ and passes all the tests, so $y \in \mathcal{R}(x_1)$. The expected cost and canonical encoding properties established above verify the remaining requirements

of Definition 3.1. For an infinity observation at index i, every satisfying point must be $-Q_i$, so the preliminary check returns the entire solution set.

For uniqueness, fix $y' \neq y^*$ and consider

$$f(Q) = \pi_1(y' + Q) - \pi_1(y^* + Q).$$

This rational function has double poles at the distinct points $-y'$ and $-y^*$. Its total pole order is four, so each prescribed value has at most four preimages. If y' and y^* both satisfy a finite observation, their first-coordinate difference is congruent to the difference of two errors in $\{0, \dots, X-1\}$. Thus $f(Q)$ is one of the residues represented by $-(X-1), \dots, X-1$. At most $4(2X-1)$ choices of $Q \in G$ meet this condition.

Before rejecting query tuples, the Q_i are independent. For this fixed y', the probability that every Q_i meets the condition is therefore $O_N((X/M)^N)$. The condition depends only on Q_i, y', y^*, X, so it also covers approximations chosen jointly from all queries. There are at most $2p+1$ points in $\mathcal{E}(\mathbb{F}_p)$: each of the p first coordinates gives at most two finite points, in addition to $\mathcal{O}$. Take a union bound over $y' \neq y^*$ and divide by the public acceptance probability, which is at least $7/8$. This proves the stated bound. An infinity observation already determines y^* uniquely. □

Constraints imposed by a short dual vector. We first derive the constraints used to prove Lemma 5.5. Fix a finite input and $1 \leq u \leq X$ with $Z_{\text{target}} > u$. By Section 3.3, some $\mathbf{w} \in \Lambda^*$ satisfies

$$\|\mathbf{w}\|_2 < \frac{1}{pu}, \qquad \langle \mathbf{v}_{\text{tar}}, \mathbf{w} \rangle \neq 0.$$

The lattice contains the scaled coefficient vector of each px_i and each px_ix_j with $i < j$. For $i < j < N$, the latter follows by multiplying P_{ij} by p and subtracting integer multiples of the available px_hx_N and px_h. The remaining nonconstant term is px_ix_j; the map Φ omits the constant term. The polynomials px_ix_N are already included.

The inner products with these vectors are integers. Hence the coordinates of $\mathbf{w}$ indexed by x_i and x_ix_j, respectively, have the form

$$\frac{s_i}{pX}, \qquad \frac{\ell_{ij}}{pX^2}, \qquad s_i, \ell_{ij} \in \mathbb{Z}, \qquad |s_i| < X/u, \quad |\ell_{ij}| < X^2/u.$$

Put $\ell_{ji} = \ell_{ij}$. The target vector has its only nonzero entry pX in the x_1 coordinate, so

$$\langle \mathbf{v}_{\text{tar}}, \mathbf{w} \rangle = s_1 \neq 0.$$

Let $\mathbf{s} = (s_1, \dots, s_N)^\mathsf{T}$. For $i \neq j$, let J_{ij} be the row of linear coefficients of $F_{ij}(\mathbf{v} + \mathbf{x})$, using integer representatives of a_{ij}. Then

$$J_{ij}\mathbf{s} = (v_j - a_{ij})s_i + (v_i - a_{ij})s_j.$$

Pairing the scaled nonconstant coefficients of $F_{ij}(\mathbf{v}+\mathbf{x})$ with $\mathbf{w}$ gives $(\ell_{ij} + J_{ij}\mathbf{s})/p$. Since $\Phi(P_{ij}) \in \Lambda$, the pairing $\langle \Phi(P_{ij}), \mathbf{w} \rangle$ is an integer. Represent the coefficient ratios in the three-term identity by integers. After substitution, the resulting combination differs from P_{ij} only by a constant and integer multiples of px_h and px_hx_ℓ. The constant is omitted by Φ, while the scaled vectors of the other terms belong to Λ by the inclusions above. Thus these changes preserve integrality of the pairing. Multiplying by p, reducing modulo p, and dividing by b_{ij} in (13) therefore gives

$$\frac{\ell_{ij} + J_{ij}\mathbf{s}}{b_{ij}} \equiv \frac{\ell_{iN} + J_{iN}\mathbf{s}}{b_{iN}} - \frac{\ell_{jN} + J_{jN}\mathbf{s}}{b_{jN}} \pmod{p} \qquad (i < j < N).$$

Define residues $\theta_i \in \mathbb{Z}_p$ by

$$\theta_N = 0, \qquad \theta_i \equiv \frac{\ell_{iN} + J_{iN}\mathbf{s}}{b_{iN}} \pmod{p} \quad (i < N).$$

It follows that $\ell_{ij} + J_{ij}\mathbf{s} \equiv b_{ij}(\theta_i - \theta_j) \pmod{p}$ for every pair $i \neq j$.

Write $W_i = y^* + Q_i = (z_i^*, w_i^*)$. Substituting $v_i \equiv z_i^* - x_i^* \pmod{p}$ yields

$$(z_j^* - a_{ij})s_i + (z_i^* - a_{ij})s_j - b_{ij}(\theta_i - \theta_j) \equiv r_{ij} \pmod{p}, \qquad |r_{ij}| \leq 3X^2/u, \tag{17}$$

where the integer $r_{ij} = -\ell_{ij} + x_j^* s_i + x_i^* s_j$ has the stated bound. Set $r_{ji} = r_{ij}$. Thus $Z_{\text{target}} > u$ requires bounded integers s_i, r_{ij} and residues θ_i satisfying this system on the points W_i.

Restricting a point with two fixed points. To count solutions of (17), we will fix two points W_1, W_b, called *anchors*, and use the equations for $(1, i)$ and (b, i) to restrict each other W_i. Their differences satisfy $W_j - W_i = Q_j - Q_i$. For a variable point W, write $d_h(W) = \pi_1(W - W_h)$ and $e_h(W) = \pi_2(W - W_h)$. At $W = W_i$, these equal a_{hi} and b_{hi}. The next lemma eliminates the unknown θ_i from these two equations. Its variables W, s, θ, r_1, r_b correspond to $W_i, s_i, \theta_i, r_{1i}, r_{bi}$ in (17).

Lemma 5.6 (Two-anchor point bound)**.** *Let $N \geq 5$ and let $W_i = (z_i, w_i)$, $i \in [N]$, be distinct finite points with $W_j - W_i \notin \mathcal{E}[2]$ for $i \neq j$. Fix a nonzero residue s_1 modulo p. There is an index $b \neq 1$ with*

$$w_1\phi'(z_b) - w_b\phi'(z_1) \neq 0. \tag{18}$$

For any such index b, fix residues $s_b, \theta_1, \theta_b, s, r_1, r_b$ modulo p. There are at most eight finite points $W = (z, w)$ with $W - W_1, W - W_b \notin \mathcal{E}[2]$ for which some residue θ satisfies

$$(z - d_h(W))s_h + (z_h - d_h(W))s - e_h(W)(\theta_h - \theta) = r_h, \qquad h \in \{1, b\}. \tag{19}$$

Here the equations and condition (18) *are modulo p. Each such W determines θ uniquely.*

Proof. Consider the rational function $g(z, w) = w_1\phi'(z) - \phi'(z_1)w$. The functions $\phi'(z) = 3z^2 + a_{\mathcal{E}}$ and w have pole orders four and three at $\mathcal{O}$, so their nonzero terms cannot cancel. Both coefficients could vanish only if $w_1 = \phi'(z_1) = 0$, when both partial derivatives of $w^2 - \phi(z)$ would vanish at W_1, contradicting nonsingularity. Thus g is nonzero with total pole order at most four. It vanishes at W_1 but cannot vanish at all $N \geq 5$ distinct points. This gives an index b satisfying (18).

Fix any such index b, and put

$$A_h(W) = (z - d_h(W))s_h + (z_h - d_h(W))s - r_h \qquad (h \in \{1, b\}).$$

Equation (19) says $A_h(W) = e_h(W)(\theta_h - \theta)$. Eliminating θ therefore makes W a zero of

$$H(W) = e_b(W)A_1(W) - e_1(W)A_b(W) - (\theta_1 - \theta_b)e_1(W)e_b(W).$$

The only poles of d_h and e_h are at W_h, of orders two and three. At W_1, neither e_b nor A_b has a pole, while A_1 has pole order at most two. Thus H has pole order at most three. The same holds at W_b. At $\mathcal{O}$, neither d_h nor e_h has a pole, while the z term gives A_h pole order at most two. There are no other possible poles. The total pole order of H is therefore at most $3 + 3 + 2 = 8$.

To prove H is nonconstant, apply (16) with $T = W$ and $D = -W_h = (z_h, -w_h)$. At $\mathcal{O}$, this gives

$$d_h(W) = z_h - 2w_h t + \phi'(z_h)t^2 + O(t^3), \qquad e_h(W) = -w_h + \phi'(z_h)t + O(t^2).$$

Since $A_h(W) = s_h t^{-2} + O(1)$, the coefficients of t^{-2} and t^{-1} in H are, respectively,

$$w_1 s_b - w_b s_1, \qquad \phi'(z_b) s_1 - \phi'(z_1) s_b.$$

If both were zero, then $s_1(w_1\phi'(z_b) - w_b\phi'(z_1)) = 0$, contrary to $s_1 \neq 0$ and (18). Thus H is nonconstant and has at most eight zeros. Finally, $e_1(W) \neq 0$ by the condition on $W - W_1$, and the first equation gives $\theta = \theta_1 - A_1(W)/e_1(W)$ uniquely. □

Proof of Lemma 5.5. Fix $1 \leq t < X$ and put $u = t^{1/N}$. If $Y > t$, the input is finite and $Z_{\text{target}} > u$, so (17) holds for some s_i, r_{ij}, θ_i as derived above. Put $S = X/u$ and $R = 3X^2/u$. Then $S, R \geq 1$, and $6X^2 < p$ gives $S, R < p/2$. In particular, the nonzero integer s_1 is also nonzero modulo p. We bound the probability that all W_i are finite, $W_j - W_i \notin \mathcal{E}[2]$ for $i \neq j$, and some integers $|s_i| \leq S$, $|r_{ij}| \leq R$ and residues θ_i satisfy (17) with $s_1 \neq 0$.

Before rejecting query tuples, the points $W_i = y^* + Q_i$ are independent and uniform on $y^* + G = \{y^* + Q : Q \in G\}$, which has M points. For each $b \neq 1$, condition only on W_1, W_b and count tuples for which they are finite, $W_b - W_1 \notin \mathcal{E}[2]$, and (18) holds. Lemma 5.6 ensures that every finite admissible tuple has at least one such b. A union bound over b costs at most a factor $N - 1$.

Fix such b, W_1, W_b. Subtracting θ_1 from each θ_i preserves all differences, so we may set $\theta_1 = 0$. There are $O(S^2R)$ choices for s_1, s_b, r_{1b}. For each choice, the $(1, b)$ equation in (17) fixes θ_b because $b_{1b} \neq 0$.

For every remaining index i, the integers s_i, r_{1i}, r_{bi} have $O(SR^2)$ choices. For each choice, Lemma 5.6 permits at most eight points W_i and fixes θ_i. The union of these permitted points therefore has size $O(SR^2)$. With $W_1, W_b, s_1, s_b, r_{1b}$ fixed, this set depends only on the constraints for index i. The remaining W_i are independent and uniform on $y^* + G$, so the probability that each belongs to its permitted set is at most $(C_N SR^2/M)^{N-2}$. Summing over s_1, s_b, r_{1b} and b gives

$$O_N\left(S^2 R \left(\frac{SR^2}{M}\right)^{N-2}\right) = O_N\left(\frac{S^N R^{2N-3}}{M^{N-2}}\right).$$

The other pair constraints can only reduce the set being counted. Each θ_i is fixed once its point and the chosen integers are fixed, so no factor p is introduced by these residues.

Substituting $S = X/u$ and $R = 3X^2/u$ gives $C_N \Delta_N u^{-(3N-3)}$. Dividing by the public acceptance probability changes only C_N. Since $u = t^{1/N}$, this proves (15). The event we counted permits every integer choice within the stated bounds and depends only on the W_i. It therefore covers every approximation rule allowed by the sampling model. □

5.3 Recovery from leading bits

On finite inputs, if at most b low bits of each z_i^* are unrevealed, the corresponding integer interval contains at most 2^b values. Taking $X = 2^b$, the bound $X \leq p^{\rho_N}$ gives a sufficient leading-bit fraction $1 - \rho_N$, up to rounding in the fixed $\lceil \log_2 p \rceil$-bit representation. The following choice of N makes this fraction at most $4/5 + \varepsilon$.

Corollary 5.7 (EC-HNP recovery from leading bits)**.** *Fix $0 < \varepsilon < 1/5$ and $c \geq 0$, and take $N = \max\{5, 1 + \lceil 1/(5\varepsilon) \rceil\}$. For $M \geq \max\{16N^2, p/(\log p)^c\}$ and $X \leq p^{1/5-\varepsilon}$, the algorithm has input-averaged expected bit cost $\text{poly}_{\varepsilon,c}(\log p)$ and its output equals $\{y^*\}$ with probability $1 - o(1)$. These guarantees hold uniformly for every fixed y^* and every leading-bit approximation rule in the sampling model above. Thus a $4/5 + \varepsilon$ fraction of leading bits suffices, up to rounding.*

Proof. We have $\rho_N = 1/5 - 4/(25N - 30)$. The chosen N gives $25N - 30 \geq 5/\varepsilon - 5 > 4/\varepsilon$, so $1/5 - \varepsilon < \rho_N$. Apply Theorem 5.2. □

6 HNPHM

In HNPHM, the secret, the multipliers, and their modular products are known only approximately [HGNS03]. Write v_1 for the approximation to y^* and $x_1^* = y^* - v_1$ for its integer error. Knowing x_1^* then gives $y^* = v_1 + x_1^*$. We construct quadratic polynomials for the search in Section 3 and a procedure that checks each proposed secret by counting its possible multipliers. To bound the expected search cost, we show that a short vector in the dual lattice imposes a linear congruence on each sampled multiplier. For each fixed choice of bounded integer parameters, this congruence permits at most one multiplier.

Definition 6.1 (HNPHM)**.** Let p be prime, let $N \geq 1$ and $1 \leq X < p$ be integers, and put $k = 2N + 1$. The public input consists of approximations $\mathbf{v} = (v_1, \ldots, v_k) \in \mathbb{Z}_p^k$. It is promised to have a hidden secret $y^* \in \mathbb{Z}_p \setminus \{0\}$ and multipliers $t_1^*, \ldots, t_N^* \in \mathbb{Z}_p \setminus \{0\}$ satisfying

$$\begin{aligned} |y^* - v_1| &\leq X, \\ |t_i^* - v_{i+1}| \leq X, \qquad & |[y^* t_i^*]_p - v_{N+i+1}| \leq X \qquad (i \in [N]). \end{aligned} \tag{20}$$

The goal is to recover y^*. Here N counts multiplier–product pairs, while k counts all approximated integers, including the secret. The bound X is the radius of each approximation interval, and the differences in (20) are in $\mathbb{Z}$.

For leading-bit observations, the input also supplies the original integer intervals specified by the revealed bits. These intervals lie in $\mathbb{Z}_p$ and contain the corresponding values y^*, t_i^*, and $[y^* t_i^*]_p$. They are checked in addition to (20).

A secret $y \in \mathbb{Z}_p \setminus \{0\}$ is consistent with the input if there exist multipliers $t_1, \ldots, t_N \in \mathbb{Z}_p \setminus \{0\}$ such that the displayed approximation bounds hold with y, t_i in place of y^*, t_i^*, and all supplied leading-bit constraints also hold.

For the average-case analysis, fix $y^* \in \mathbb{Z}_p \setminus \{0\}$ and sample $t_1^*, \ldots, t_N^*$ independently and uniformly from $\mathbb{Z}_p \setminus \{0\}$. The approximations may be functions of all the samples, provided they satisfy (20) for this y^* and these multipliers. Any supplied leading-bit intervals must also contain the corresponding values. All probabilities and input averages below use this sampling model.

Theorem 6.2 (HNPHM recovery)**.** *Fix $N \geq 1$ and put*

$$\Delta_N = \frac{X^{3N+1}}{(p-1)^N}, \qquad \sigma_N = \frac{N}{3N+1}.$$

With the polynomials and reconstruction procedure below, Algorithm 1 deterministically returns exactly the secrets satisfying the input constraints on every promised input. Its output contains y^* and has at most $2X + 1$ elements. It uses $k = 2N + 1$ error variables and a lattice of rank $m = 2N + 1$ in ambient dimension $d = 3N + 1$. Its input-averaged bit cost is at most

$$\mathrm{poly}_N(\log p)(1 + \Delta_N(1 + \log X)). \tag{21}$$

For every fixed $B \geq 0$, this cost is $\mathrm{poly}_{N,B}(\log p)$ throughout $X \leq p^{\sigma_N}(\log p)^B$. In this window, the output equals $\{y^*\}$ with probability $1 - o(1)$ as $p \to \infty$. These guarantees hold uniformly for every fixed y^* and every approximation rule in the sampling model above.

6.1 Recovery data and lemmas

Quadratic relations and the lattice. For a secret y and multipliers $t_1, \dots, t_N$ satisfying the input constraints, define

$$\begin{aligned} x_1 &= y - v_1, \\ x_{i+1} &= t_i - v_{i+1}, \qquad x_{N+i+1} = [yt_i]_p - v_{N+i+1} \qquad (i \in [N]). \end{aligned}$$

Then $\mathbf{x} = (x_1, \dots, x_k)$ belongs to $\mathcal{B} = \prod_{j=1}^{k} \mathcal{B}_j$, where $\mathcal{B}_j = [-X, X] \cap \mathbb{Z}$. Write $\mathbf{x}^*$ for the vector obtained from $y^*, t_1^*, \dots, t_N^*$. The modular product equation gives the polynomials

$$P_i(\mathbf{x}) = (v_1 + x_1)(v_{i+1} + x_{i+1}) - (v_{N+i+1} + x_{N+i+1}) \qquad (i \in [N]). \tag{22}$$

Set $m = 2N + 1$ and append

$$P_{N+i} = px_{i+1} \quad (i \in [N]), \qquad P_m = px_1.$$

This places px_1 last, as required in Section 3. The appended polynomials will give integer parameters for $x_1, \dots, x_{N+1}$ in the dual calculation.

Use the scaled coefficient map Φ from Definition 2.3. Its coordinates correspond to the N monomials x_1x_{i+1} and the $2N + 1$ linear monomials x_j; their coefficients already carry weights X^2 and X, respectively. Let

$$\Lambda = \operatorname{span}_{\mathbb{Z}}\{\Phi(P_h) : h \in [m]\}.$$

The target vector $\mathbf{v}_{\mathrm{tar}} = \Phi(px_1)$ has entry pX in the x_1 coordinate and zero elsewhere.

Lemma 6.3 (Quadratic source polynomials)**.** *The polynomials $P_1, \dots, P_m$ above have degree at most two and satisfy $P_h(\mathbf{x}) \equiv 0 \pmod{p}$ for every $h \in [m]$ and every $\mathbf{x}$ associated with a secret and multipliers satisfying the input constraints. The vectors $\Phi(P_1), \dots, \Phi(P_m)$ are linearly independent over $\mathbb{R}$. Consequently, Λ has rank $m = 2N + 1$ in ambient dimension $d = 3N + 1$. The full polynomials, including their constant terms, and their scaled coefficient vectors can be constructed deterministically in $\mathrm{poly}_N(\log p)$ bit time.*

The target restriction matters even for simple secrets. For leading-bit inputs with $y^* = 1$ and $X \geq 2$, we have $v_1 = 0$. Let $\mathbf{w}$ have coordinate $1/(pX)$ at x_2 and zero elsewhere. It is a real multiple of $\Phi(px_2)$, and its inner products with the lattice generators are 1 for $\Phi(px_2)$ and 0 for all others. Thus $\mathbf{w} \in \Lambda^*$, with $\|\mathbf{w}\|_2 = 1/(pX)$ and $\langle \mathbf{v}_{\mathrm{tar}}, \mathbf{w} \rangle = 0$. Using the unrestricted dual minimum in our cost bound would therefore leave only the direct-scan bound for every input with this fixed secret.

Reconstructing y from a proposal for x_1. For a proposal $a \in \mathcal{B}_1$, set $\widehat{y} = v_1 + a$. If $1 \leq \widehat{y} < p$ does not hold, set $\mathcal{R}(a) = \varnothing$ and stop. For each $i \in [N]$, form the integer intervals

$$\begin{aligned} [L_i, U_i] &= [v_{i+1} - X, v_{i+1} + X] \cap [1, p-1], \\ [S_i, V_i] &= [v_{N+i+1} - X, v_{N+i+1} + X] \cap [0, p-1]. \end{aligned}$$

If leading-bit intervals are supplied, intersect these intervals with the original multiplier and product intervals, respectively, and also require $\widehat{y}$ to lie in the original secret interval. If this check fails or any intersection is empty, set $\mathcal{R}(a) = \varnothing$ and stop.

For each $i \in [N]$, count the integers $t \in [L_i, U_i]$ with $[\widehat{y}t]_p \in [S_i, V_i]$:

$$C_i(\widehat{y}) = \sum_{t=L_i}^{U_i} \left(\left\lfloor \frac{\widehat{y}t - S_i}{p} \right\rfloor - \left\lfloor \frac{\widehat{y}t - V_i - 1}{p} \right\rfloor \right). \tag{23}$$

The summand equals one exactly when $[\widehat{y}t]_p \in [S_i, V_i]$, and zero otherwise. Thus $\widehat{y}$ satisfies the input constraints exactly when $C_i(\widehat{y}) > 0$ for every $i \in [N]$: the multipliers for different samples can be chosen separately once $\widehat{y}$ is fixed. Set $\mathcal{R}(a) = \{\widehat{y}\}$ if all counts are positive, and $\mathcal{R}(a) = \varnothing$ otherwise. The Euclidean floor-sum algorithm [yos21] computes these counts in polynomial time without enumerating the integers t; its recurrence is given below. The next lemma verifies Definition 3.1 for $\mathcal{R}$ with $r = 1$.

Lemma 6.4 (Reconstruction and uniqueness)**.** *On every promised input, $\mathcal{R}$ is a deterministic reconstructor with $r = 1$ in the sense of Definition 3.1. In particular, $\mathcal{R}(x_1^*) = \{y^*\}$. Under the sampling model above,*

$$\Pr[\textit{some } y \neq y^* \textit{ satisfies the input constraints}] \leq C_N \frac{X^{2N+1}}{(p-1)^N},$$

where C_N depends only on N.

From Z_{target} to the expected cost. Use Z_{target} from Section 3.3 for Λ and $\mathbf{v}_{\text{tar}}$ above. It satisfies $1 \leq Z_{\text{target}} \leq X$. Since $k = 2N + 1$, Theorem 3.4 bounds the recovery cost by a polynomial in $\log p$ times $1 + Z_{\text{target}}^k$. We therefore bound $\mathbb{E}[Z_{\text{target}}^k]$ using the following tail estimate.

Lemma 6.5 (Target-dual tail bound)**.** *For fixed $N \geq 1$ and every $1 \leq X < p$,*

$$\Pr[Z_{\text{target}} > u] \leq C_N \Delta_N u^{-k} \qquad (1 \leq u \leq X), \qquad k = 2N + 1, \tag{24}$$

under the sampling model above, uniformly over the fixed secret and the approximation rule. Here C_N depends only on N.

Proof of Theorem 6.2. Lemmas 6.3 and 6.4 verify the polynomial and reconstruction hypotheses of Theorem 3.4. The algorithm therefore returns exactly the secrets satisfying the input constraints. For any such secret y, its first error is $x_1 = y - v_1$, and $\mathcal{R}(x_1) = \{y\}$. There are $2X + 1$ possible integers in $\mathcal{B}_1$, so the output has at most $2X + 1$ elements and contains y^*.

All steps are deterministic, and Theorem 3.4 gives the per-input bit-cost bound

$$\text{poly}_N(\log p)[1 + \min\{2X + 1, (KZ_{\text{target}})^k\}] \leq \text{poly}_N(\log p)(1 + Z_{\text{target}}^k),$$

where K is the constant in (2). For fixed N, it is absorbed into the polynomial factor. For $X > 1$, the tail-integral identity and Lemma 6.5 give

$$\begin{aligned}\mathbb{E}[Z_{\text{target}}^k] - 1 &= k \int_1^X u^{k-1} \Pr[Z_{\text{target}} > u]\, du \\ &\leq kC_N\Delta_N \int_1^X \frac{du}{u} = kC_N\Delta_N \log X.\end{aligned}$$

For $X = 1$, we have $Z_{\text{target}} = 1$. This proves (21). The expectation is over $t_1^*, \ldots, t_N^*$ with the approximation rule fixed.

If $X \leq p^{\sigma_N}(\log p)^B$, then

$$\Delta_N \leq \left(\frac{p}{p-1}\right)^N (\log p)^{B(3N+1)},$$

so (21) is polynomial in $\log p$. The probability that the output contains a secret other than y^* is, by Lemma 6.4, at most a constant times

$$\left(\frac{p}{p-1}\right)^N p^{-N^2/(3N+1)} (\log p)^{B(2N+1)}.$$

This tends to zero, proving the uniqueness claim. □

6.2 Proofs of the lemmas

Proof of Lemma 6.3. For any secret y and multipliers t_i satisfying the input constraints, substitution of their associated $\mathbf{x}$ into (22) gives $P_i(\mathbf{x}) = yt_i - [yt_i]_p$. This is divisible by p. Each appended polynomial is also divisible by p.

Suppose $\sum_{h=1}^{m} \alpha_h \Phi(P_h) = \mathbf{0}$ with $\alpha_1, \ldots, \alpha_m \in \mathbb{R}$. For each $i \in [N]$, the monomial $x_1 x_{i+1}$ occurs only in P_i, with coefficient one. Its coordinate therefore forces $\alpha_i = 0$. The remaining polynomials are $px_2, \ldots, px_{N+1}, px_1$; their distinct linear coordinates force all remaining coefficients to be zero. Scaling coordinates by the positive numbers X and X^2 preserves this independence. There are N quadratic monomials and $2N+1$ linear monomials, giving $d = 3N+1$ and $m = 2N+1$. For fixed N, the inputs have $O(\log p)$-bit entries, and forming the full polynomials and scaling their coefficient vectors takes $\mathrm{poly}_N(\log p)$ bit time. □

Computing $C_i(\widehat{y})$ by Euclidean division. To evaluate (23), define, for this calculation,

$$F(n, M, A, B) = \sum_{j=0}^{n-1} \left\lfloor \frac{Aj + B}{M} \right\rfloor, \qquad n \geq 0,\ M \geq 1,\ A, B \in \mathbb{Z}.$$

Here n and M are integers. With $n_i = U_i - L_i + 1$, substituting $t = L_i + j$ gives

$$C_i(\widehat{y}) = F(n_i, p, \widehat{y}, \widehat{y}L_i - S_i) - F(n_i, p, \widehat{y}, \widehat{y}L_i - V_i - 1).$$

It therefore suffices to evaluate two sums of the form F.

Write $A = q_A M + A_0$ and $B = q_B M + B_0$ by Euclidean division, with $0 \leq A_0, B_0 < M$. Then

$$F(n, M, A, B) = q_A \frac{n(n-1)}{2} + q_B n + F(n, M, A_0, B_0).$$

This identity also covers negative A or B. The last term is zero if $n = 0$ or $A_0 = 0$. Otherwise set $q = \lfloor (A_0 n + B_0)/M \rfloor$ and $r = [A_0 n + B_0]_M$. The Euclidean floor-sum recurrence [yos21] is

$$F(n, M, A_0, B_0) = F(q, A_0, M, r). \tag{25}$$

To see this, the term at column j counts the positive integers ℓ with $M\ell \leq A_0 j + B_0$. Counting the same pairs (j, ℓ) by ℓ gives

$$\begin{aligned} F(n, M, A_0, B_0) &= \sum_{\ell=1}^{q} \left(n - \left\lceil \frac{M\ell - B_0}{A_0} \right\rceil \right) \\ &= \sum_{h=0}^{q-1} \left\lfloor \frac{Mh + r}{A_0} \right\rfloor, \qquad h = q - \ell. \end{aligned}$$

If level q is reached only at the excluded column $j = n$, its contribution is zero. The second equality uses $A_0 n + B_0 = Mq + r$.

The recurrence replaces modulus M by $A_0 < M$. At the next normalization, the new slope remainder is $M \bmod A_0$, so successive moduli follow the Euclidean algorithm. For the two sums defining $C_i(\widehat{y})$, there are $O(\log p)$ stages. Initially $n_i < p$, $1 \leq \widehat{y} < p$, and the offsets $\widehat{y}L_i - S_i$ and $\widehat{y}L_i - V_i - 1$ have $O(\log p)$ bits. In the normalized recurrence, $q \leq n$ and the modulus decreases, so the new lengths and moduli remain at most p. The quotient contributions and accumulated sums are bounded by a fixed power of p and hence also have $O(\log p)$ bits. Exact integer arithmetic therefore computes every $C_i(\widehat{y})$ in polynomial bit time.

Proof of Lemma 6.4. We first verify that the test using $C_i(\widehat{y})$ is exact and polynomial-time. For a trial multiplier t, write $\widehat{y}t = qp + z$, where $z = [\widehat{y}t]_p$. Since $0 \le S_i \le V_i < p$, the summand in (23) equals

$$\left\lfloor \frac{z - S_i}{p} \right\rfloor - \left\lfloor \frac{z - V_i - 1}{p} \right\rfloor = \begin{cases} 1, & S_i \le z \le V_i, \\ 0, & \text{otherwise.} \end{cases}$$

Thus $C_i(\widehat{y})$ counts exactly the multipliers $t \in [L_i, U_i] \cap \mathbb{Z}$ with $[\widehat{y}t]_p \in [S_i, V_i]$. When every count is positive, choose one such t_i for each i. Together with the checks on $\widehat{y}$, these choices satisfy all input constraints. Conversely, any allowed multipliers make all the counts positive. The Euclidean calculation above evaluates the counts in $\mathrm{poly}_N(\log p)$ bit time.

Now fix any $y, t_1, \ldots, t_N$ satisfying the input constraints, with associated errors $\mathbf{x} \in \mathcal{B}$. At $a = x_1$, we have $\widehat{y} = v_1 + x_1 = y$. Each t_i belongs to its allowed interval and has its product in the required interval, so $C_i(y) > 0$. All checks on y also pass, including any supplied leading-bit interval. It follows that $\mathcal{R}(x_1) = \{y\}$. Every returned secret is a canonical integer with $O(\log p)$ bits. This verifies Definition 3.1 for $\mathcal{R}$ with $r = 1$.

For uniqueness, suppose another secret $y' \ne y^*$ satisfies the input constraints. The secret approximation gives $|y' - y^*| \le 2X$. Hence all possible alternatives lie in the fixed set

$$\{y' \in \{1, \ldots, p-1\} : 0 < |y' - y^*| \le 2X\},$$

which has $O(X)$ elements and is independent of the sampled multipliers and the approximations. Fix one such y'. If it has allowed multipliers t_i', put

$$\delta_i = t_i' - t_i^*, \qquad e_i = [y' t_i']_p - [y^* t_i^*]_p.$$

The shared approximation intervals give $|\delta_i|, |e_i| \le 2X$, and subtraction of the product congruences yields

$$(y' - y^*) t_i^* + y' \delta_i \equiv e_i \pmod{p}.$$

For each i, consider the event that *some* integers δ_i, e_i with these bounds satisfy this congruence. There are $O(X^2)$ pairs to consider. Since distinct canonical secrets satisfy $y' - y^* \not\equiv 0 \pmod{p}$, each pair permits at most one $t_i^* \in \mathbb{Z}_p \setminus \{0\}$. This event has probability at most $O(X^2/(p-1))$. For the fixed y', the event at index i depends only on t_i^*, so independence gives the bound $O_N((X^2/(p-1))^N)$ for all indices. Taking a union over the $O(X)$ alternatives proves the stated bound. The events use only y^*, X and the sampled multipliers, so the bound is uniform over the approximation rule. □

Proof of Lemma 6.5. Fix $1 \le u \le X$. By Section 3.3, $Z_{\mathrm{target}} > u$ implies the existence of

$$\mathbf{w} \in \Lambda^* \subseteq \mathrm{span}_{\mathbb{R}} \Lambda, \qquad \|\mathbf{w}\|_2 < \frac{1}{pu}, \qquad \langle \mathbf{v}_{\mathrm{tar}}, \mathbf{w} \rangle \ne 0.$$

Here Λ^* is the dual within the real span of Λ. We index coordinates of $\mathbf{w}$ by their monomials: w_{x_j} for x_j and $w_{x_1 x_{i+1}}$ for $x_1 x_{i+1}$. Pairing $\mathbf{w}$ with the scaled rows of px_1 and px_{i+1} gives integers

$$s = \langle \mathbf{v}_{\mathrm{tar}}, \mathbf{w} \rangle = pX w_{x_1}, \qquad s_i = pX w_{x_{i+1}} \qquad (i \in [N]).$$

The nonzero pairing gives $s \ne 0$, and the norm bound gives $|s|, |s_i| < X/u$.

For each $i \in [N]$, let $\eta_i \in \mathbb{Z}$ be the pairing of $\mathbf{w}$ with the scaled coefficient vector of P_i. Expanding (22) and omitting its constant term gives

$$p\eta_i = s v_{i+1} + v_1 s_i + pX^2 w_{x_1 x_{i+1}} - pX w_{x_{N+i+1}}.$$

Therefore the integer $r_i = p\eta_i - sv_{i+1} - v_1 s_i$ satisfies

$$r_i = pX^2 w_{x_1 x_{i+1}} - pX w_{x_{N+i+1}}, \qquad |r_i| < \frac{X^2 + X}{u} \le \frac{2X^2}{u}.$$

The last bound follows by bounding each coordinate of $\mathbf{w}$ by $\|\mathbf{w}\|_2$.

Here r_i combines the quadratic and product-error contributions into one integer parameter. The two contributions need not be integral separately, so the count does not require additional rows px_{N+i+1}.

We next express these congruences using the sampled t_i^* and the fixed y^*. Substitute $t_i^* = v_{i+1} + x_{i+1}^*$ and $y^* = v_1 + x_1^*$, and reduce the definition of r_i modulo p. This gives

$$\begin{gathered} st_i^* + y^* s_i \equiv R_i \pmod p, \qquad R_i = -r_i + sx_{i+1}^* + x_1^* s_i \in \mathbb{Z}, \\ 0 < |s| \le X/u, \qquad |s_i| \le X/u, \qquad |R_i| \le 4X^2/u. \end{gathered} \tag{26}$$

Here the two correction terms have absolute values at most X^2/u each because $|x_1^*|, |x_{i+1}^*| \le X$.

It remains to count sampled tuples for which *some* integers s, s_i, R_i satisfy (26). This event contains $\{Z_{\text{target}} > u\}$ and no longer refers to the approximations. Set $S = X/u$ and $R = 4X^2/u$, both at least one. There are $O(S)$ possible nonzero integers s. Fix one such s. For a given index i, there are $O(SR)$ pairs of integers (s_i, R_i) with $|s_i| \le S$ and $|R_i| \le R$. Since $0 < |s| \le X < p$, the congruence $st_i^* + y^* s_i \equiv R_i \pmod p$ permits at most one nonzero canonical multiplier for each pair. Thus the probability that some (s_i, R_i) satisfies this congruence is at most $O(SR/(p-1))$.

For this fixed s, each index's event depends only on its own multiplier t_i^*. Independence gives the product of these probability bounds, and a union over the $O(S)$ choices of s yields

$$\begin{aligned} \Pr[Z_{\text{target}} > u] &\le O_N\left(S \left(\frac{SR}{p-1} \right)^N \right) \\ &= O_N\left(\frac{X^{3N+1}}{(p-1)^N} u^{-(2N+1)} \right). \end{aligned}$$

This is (24). The counted event depends only on the sampled multipliers, so the estimate holds for every approximation rule allowed by the sampling model. □

6.3 Recovery from leading bits

If at most b lower bits of each value are unknown, the revealed leading bits specify an integer interval containing at most 2^b canonical values. Choose each v_j as the left endpoint of its interval and take $X = 2^b$. Then $0 \le x_j^* < X$, which also satisfies the radius bound $|x_j^*| \le X$ used in this section. The reconstruction procedure checks the original intervals as well. Since $\sigma_N = N/(3N+1)$ approaches $1/3$, the sufficient leading-bit fractions approach $2/3$.

Corollary 6.6 (HNPHM recovery from leading bits)**.** *Fix $0 < \varepsilon < 1/3$ and take $N = \lceil 1/(9\varepsilon) \rceil$. For $X \le p^{1/3-\varepsilon}$, the algorithm has input-averaged bit cost $\mathrm{poly}_\varepsilon(\log p)$, and its output equals $\{y^*\}$ with probability $1 - o(1)$ for every fixed secret under the sampling model above. In the fixed $\lceil \log_2 p \rceil$-bit representation, it suffices to leave at most $\lfloor (1/3 - \varepsilon) \log_2 p \rfloor$ lower bits unknown in the secret, each multiplier, and each modular product. Thus a $2/3 + \varepsilon$ fraction of leading bits of each value suffices, up to rounding.*

Proof. We have $\sigma_N = 1/3 - 1/(9N+3)$. The chosen N satisfies $9N + 3 \ge 1/\varepsilon + 3$, so $1/(9N+3) < \varepsilon$ and $1/3 - \varepsilon < \sigma_N$. Apply Theorem 6.2. □

7 Experiments

Setup. The reported timings were obtained on a workstation with an Intel Core i9-14900KF processor (24 cores, 32 hardware threads) and 32 GiB of RAM. The main benchmark compares leading-bit recovery over the 384-bit P-384 prime, using the P-384 curve for EC-HNP. Each z_j^* is represented using 384 bits. We vary the percentage of leading bits revealed and report percentages rounded to one decimal place. Each main-benchmark setting uses ten inputs for each of three fixed secrets, giving 30 test inputs. Both methods receive the same public inputs. Code, data, and the detailed settings are available at `https://github.com/kangting/Improved-bounds-for-heuristic-free-nonlinear-HNPs`.

Implementation. Our lattices use the first 12 of the 30 supplied MIHNP values, the first eight of the 17 EC-HNP values, or all 16 HNPHM multiplier/product pairs together with the secret approximation. Using subsets for MIHNP and EC-HNP reduces their lattice ranks. Candidate verification uses all supplied values. The baselines are concrete LLL/Babai implementations of the constructions in [LSSW12, Sha17, HGNS03], using all 30 MIHNP values, 17 EC-HNP values, or 16 HNPHM pairs. Both methods use `flatter` [RH23] followed by LLL.

The plotted EC-HNP comparison uses a central query and eight symmetric query pairs to match the baseline's query structure. The first eight queries comprise the central query, three complete opposite-shift pairs, and one query from the fourth pair. We also test our method on inputs generated by sampling all 17 queries independently and uniformly from the public subgroup before rejecting tuples that violate the public query conditions. All supplied queries in both models satisfy these conditions.

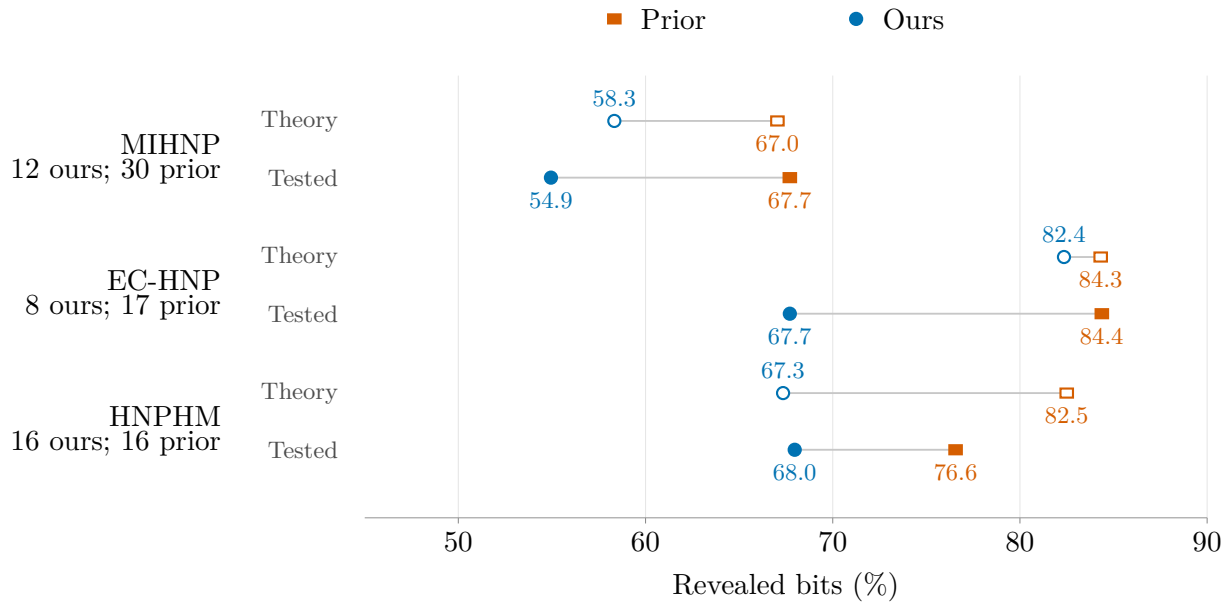


Figure 1: Theory (open markers): asymptotic sufficient leading-bit fractions at the indicated observation counts, with positive margins omitted. Tested (filled markers): lowest tested fractions at which all 30 runs output $\{y^*\}$. Prior refers to the cited provable constructions. Counts below each problem refer to the values used in the lattices for MIHNP and EC-HNP, and to multiplier/product pairs for HNPHM. Verification uses all supplied values. For EC-HNP, our theoretical bound uses independent queries before public rejection, while the plotted tests use symmetric query pairs.

For leading-bit HNPHM inputs, the implementation uses $\mathcal{B} = [0, X-1]^k \cap \mathbb{Z}^k$. It splits a coordinate containing the most remaining final intervals, breaking ties by the smallest index. These choices preserve the correctness and visit-count arguments of Section 3. The main-benchmark runner defaults to a 600-second time limit. Our search also has a 4,096-node limit, counting all visited tree nodes. Interrupted runs are recorded as unfinished.

Results. The prior theoretical fractions follow from the fixed-sample failure bounds in [LSSW12, Sha17, HGNS03], requiring success probability $1 - o(1)$. The corresponding error exponents are $(k-1)/(3k-2)$ for k MIHNP values, $(k-1)/(6k)$ for k EC-HNP queries, and $(N-2)/(5N)$ for N HNPHM multiplier/product pairs. For our theoretical points, the finite-sample bounds in Sections 4 to 6 give sufficient leading-bit fractions $1 - \xi_{12} = 7/12$ for MIHNP, $1 - \rho_8 = 14/17$ for EC-HNP, and $1 - \sigma_{16} = 33/49$ for HNPHM.

At the tested fractions displayed for our method in Figure 1, all 30 runs return $\{y^*\}$. On the same inputs, none of the corresponding baseline runs recovers y^* within the recorded limits. For both EC-HNP query models, all 30 runs return $\{y^*\}$ at 67.7% revealed bits. At 67.4% and 67.2%, all 30 runs per model and setting reach the 4,096-node limit.

Search behavior and elapsed time. All 240 completed runs of our method in the main benchmark terminated at the root box and made one call to $\mathcal{R}(a)$. All 150 unfinished runs reached the node limit: 120 EC-HNP runs across the two query models and 30 HNPHM runs at 67.7% revealed bits. These main-benchmark runs do not establish a performance gain attributable to nontrivial subdivision. Table 2 reports the recorded parent-process elapsed times at the successful settings. Separate stage timings were not retained.

Setting	Revealed bits (%)	Median (s)	Maximum (s)
MIHNP	54.9	6.36	19.98
EC-HNP, paired queries	67.7	1.71	2.79
EC-HNP, independent queries	67.7	1.71	3.07
HNPHM	68.0	1.12	9.19

Table 2: Recorded elapsed times for our method on the 384-bit P-384 modulus. Each row contains 30 runs returning $\{y^*\}$. The parent-process timer includes process startup, lattice preparation, search, and candidate verification.

Subdivision beyond the root box. A post-hoc examination of three previously recorded auxiliary P-384 HNPHM inputs with $N = 5$ and 119 unrevealed low bits (69.0% revealed bits) gives visited-node counts $1, 5, 1$. All three runs return $\{y^*\}$. On the saved reduced basis of the five-node input, the combined root target interval contains more than 10^{35} integers after imposing the supplied leading-bit intervals. The adaptive search discards two empty-target subtrees and completes with one call to $\mathcal{R}$, without exhaustive target scanning. These auxiliary runs used an earlier NTL-based implementation with a 10,000-node limit and no per-run wall-clock timeout; they are reported separately from the $N = 16$ main benchmark.

8 Conclusion

We improve the provable recovery ranges for MIHNP, EC-HNP, and HNPHM by recovering x_1^* from bounded integer quotients and verifying the resulting secret candidates. For every valid input, the algorithm returns exactly the secrets satisfying the original input constraints. For fixed observation counts, its input-averaged expected bit cost is polynomial in $\log p$ under the stated sampling models and parameter conditions. The experiments in Section 7 illustrate recovery with less revealed information on the tested instances.

A natural next question is which other polynomial congruence and small-root problems admit the same approach. The main requirements are efficient reconstruction of candidate secrets from a proposed x_1 and rigorous control of the average quotient-search cost. Extensions to higher-degree relations or reconstruction from several coordinates also merit further study.

Our analysis uses error boxes that are products of integer intervals, including those arising from leading-bit observations. Middle-bit or scattered-bit leakage can leave possible values in separated intervals. Extending the analysis to these leakage patterns calls for suitable representations, such as unions of intervals or additional error variables, together with corresponding bounds on reconstruction and average search costs.

AI methodology

We used ChatGPT and Gemini to help with ideation, forming the basic framework, improving phrasing, and writing experimental scripts. The tools materially assisted with the overall framework in Section 3, the proof strategies in Sections 4 to 6, and the implementation and evaluation in Section 7. All mathematical theorems, rigorous proofs, and analyses were developed, checked, and completed by the authors. The authors confirm the correctness and originality of all content, including all mathematical arguments and references, and take full responsibility for the entire work.

References

[AH21] Martin R Albrecht and Nadia Heninger. On bounded distance decoding with predicate: Breaking the "lattice barrier" for the hidden number problem. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 528–558. Springer, 2021.

[Bab86] László Babai. On Lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.

[BH19] Joachim Breitner and Nadia Heninger. Biased nonce sense: Lattice attacks against weak ECDSA signatures in cryptocurrencies. In *International Conference on Financial Cryptography and Data Security*, pages 3–20. Springer, 2019.

[BHHG01] Dan Boneh, Shai Halevi, and Nick Howgrave-Graham. The modular inversion hidden number problem. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 36–51. Springer, 2001.

[BKS23] Riley Baird, Bryce Kerr, and Igor Shparlinski. Bohr sets generated by polynomials and Coppersmith's method in many variables. *arXiv preprint arXiv:2310.20342*, 2023.

[BV96] Dan Boneh and Ramarathnam Venkatesan. Hardness of computing the most significant bits of secret keys in Diffie–Hellman and related schemes. In *Annual International Cryptology Conference*, pages 129–142. Springer, 1996.

[CP06] Todd Cochrane and Christopher Pinner. Using Stepanov's method for exponential sums involving rational functions. *Journal of Number Theory*, 116(2):270–292, 2006.

[DDW+26] Zhaopeng Ding, Zhaopeng Dai, Baofeng Wu, Yanshuo Zhang, and Kejun Zhang. Computing asymptotic bounds for the automated Coppersmith method via linear programming. *Cryptology ePrint Archive*, 2026. Paper 2026/1027. Online.

[HGNS03] Nick Howgrave-Graham, Phong Nguyen, and Igor Shparlinski. Hidden number problem with hidden multipliers, timed-release crypto, and noisy exponentiation. *Mathematics of Computation*, 72(243):1473–1485, 2003.

[LLL82] Arjen K. Lenstra, Hendrik Willem Lenstra, Jr., and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982. DOI.

[LSSW12] 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.

[NS09] Phong Q Nguyen and Damien Stehlé. An LLL algorithm with quadratic complexity. *SIAM Journal on Computing*, 39(3):874–903, 2009.

[RH23] Keegan Ryan and Nadia Heninger. Fast practical lattice reduction through iterated compression. In *Annual International Cryptology Conference*, pages 3–36. Springer, 2023.

[Sar17] Santanu Sarkar. Analysis of hidden number problem with hidden multiplier. *Advances in Mathematics of Communications*, 11(4):805–811, 2017.

[Sha17] Barak Shani. On the bit security of elliptic curve Diffie–Hellman. In *IACR International Workshop on Public Key Cryptography*, pages 361–387. Springer, 2017.

[Sho09] Victor Shoup. *A Computational Introduction to Number Theory and Algebra*. Cambridge University Press, second edition, 2009. Online.

[Sil09] Joseph H Silverman. *The arithmetic of elliptic curves*, volume 106 of *Graduate Texts in Mathematics*. Springer, second edition, 2009.

[Sut25] Andrew V. Sutherland. 18.783 elliptic curves, lecture 23: Divisors and the Weil pairing. MIT OpenCourseWare, 2025. Online.

[XSH+19] Jun Xu, Santanu Sarkar, Lei Hu, Huaxiong Wang, and Yanbin Pan. New results on modular inversion hidden number problem and inversive congruential generator. In *Annual International Cryptology Conference*, pages 297–321. Springer, 2019.

[XSWH26] Jun Xu, Santanu Sarkar, Huaxiong Wang, and Lei Hu. New results on elliptic curve hidden number problem for ECDH key exchange. *Journal of Cryptology*, 39(2):20, 2026.

[yos21] yosupo. C – Floor Sum: Official editorial. AtCoder Library Practice Contest, 2021. Accessed 2 November 2026. Online.