

NTRU with Hints: Secret Key Recovery under Partial Leakage on NTRU-based Signatures

Honglin Shao*, Yuejun Liu*[✉], Mingyao Shao^{†‡}, Wei Cheng*[§], Yongbin Zhou*^{†✉}

* School of Cyber Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China

[†] Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100085, China

[‡] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

[§] LTCl, Télécom Paris, Institut Polytechnique de Paris, Palaiseau 91120, France

{shaohonglin, liuyuejun, wei.cheng, zhouyongbin}@njjust.edu.cn; shaomingyao@iie.ac.cn

Abstract—Post-quantum cryptography has become pivotal for ensuring communication security in the quantum era. NTRU-based signature schemes have gained attention due to their computational efficiency and compact public keys and signatures, which reduce communication overheads. However, the algebraic structure of NTRU lattices introduces vulnerabilities in physical attack scenarios, where partial secret key leakage can severely undermine system security. In this paper, we propose the Dimension-and-Sample Reduced NTRU Attack (DSRNA). The core principle of DSRNA is to apply a dimension-reduction strategy that transforms the NTRU instance into a lower-dimensional NTRU instance with fewer samples, which is efficiently solvable via lattice basis reduction. Furthermore, we design a unified hint-embedding technique that jointly exploits side information from both secret keys f and g , thereby improving both attack efficiency and success rates. We evaluate the residual security of Falcon and its variants Mitaka and Hawk, under perfect, modular and approximate leakage models. Experimental results demonstrate security degradation. Compared to the method of May et al. at Asiacrypt 2023, DSRNA achieves speedups of $5.8\times$ for Falcon-512 with 400 leaked coefficients and over $29.4\times$ for Falcon-1024 with 905 coefficients. This study reveals the potential vulnerability of NTRU-based signature schemes to partial secret key leakage.

Index Terms—Post-quantum cryptography, Lattice-based Cryptanalysis, NTRU-based Signature, Lattice reduction.

I. INTRODUCTION

With the rapid progress of quantum computing technology, traditional public-key cryptographic algorithms are facing unprecedented security challenges. The powerful computational capabilities of quantum computers enable them to break classical schemes based on number-theoretic problems, such as integer factorization and discrete logarithms [1], within a feasible timeframe. As a result, cryptographic systems such as RSA and elliptic-curve cryptography would become insecure. To address this challenge, the U.S. National Institute of Standards and Technology (NIST) launched the Post-Quantum Cryptography (PQC) standardization project in 2016, which aims to identify and standardize new cryptographic algorithms that can resist quantum attacks.

To date, NIST has standardized three lattice-based cryptographic schemes the key encapsulation mechanism Kyber [2], and the digital signature algorithms Dilithium [3] and Falcon [4]. Both Kyber and Dilithium are designed based on the hardness of the Module Learning With Errors (MLWE) problem [2], [3], whereas Falcon is constructed using an NTRU

(N-th degree Truncated Polynomial Ring Unit) lattice [4], [5] and relies on the hardness of the NTRU problem. As the standardization process led by NIST has entered the migration phase [6], major industry stakeholders are actively deploying post-quantum cryptographic solutions. For example, Google has by default enabled the latest Kyber specification for all desktop platforms starting from Chrome 124¹, applying it to TLS 1.3 and QUIC to provide post-quantum secure communication channels. NTRU-based cryptographic algorithms have also been actively adopted. For instance, the Algorand blockchain employs Falcon as the core signature algorithm for its state proof mechanism, enabling secure cross-chain communication with post-quantum resilience [7]. In addition, Google Cloud has deployed a hybrid key exchange mechanism based on NTRU-HRSS and X25519 in its internal Application Layer Transport Security (ALTS) protocol [8]. This approach enhances the system's resistance to future quantum attacks while maintaining compatibility with existing infrastructure. Notably, the NTRU-HRSS [9] is one of the third round finalists in NIST's post-quantum cryptography standardization project.

NTRU-based signature schemes, including Falcon and its variants Mitaka [10] and Hawk [11], [12], have received increasing attention due to their compact structure, efficient verification, and suitability for resource-constrained environments. Notably, Hawk has advanced to the second round of NIST's post-quantum signature standardization [11]. Falcon, Mitaka, and Hawk are concrete instantiations of the GPV hash-and-sign framework [13], operating on NTRU lattices generated from small-coefficient secret polynomials (f, g) . These coefficients are typically sampled from discrete Gaussian or centered binomial distributions.

However, the inherent algebraic structure of NTRU lattices also introduces unique cryptanalytic challenges. Classical attack vectors include hybrid attacks [14], subfield attacks [15], dense sublattice attacks [16], and recent lattice-isomorphism-based attacks [17]. From a practical implementation perspective, side-channel studies [18]–[21] have demonstrated that the core operations of Falcon, including ring multiplication, ring division, and discrete Gaussian sampling, are susceptible to

¹Google Chrome 124 updates: <https://chromeenterprise.google/policies/#PostQuantumKeyAgreementEnabled>

leakage regarding the secret polynomials f and g . This leakage typically manifests as perfect, modular, or approximate hints. Furthermore, the secret key may be compromised by structured information revealed through fault-injection attacks [22], [23], cold-boot attacks [24], and decryption-failure attacks [25].

To systematically analyze the security under such leakage models, Dachman-Soled, Ducas, Gong, and Rossi (DDGR) introduced the leaky-LWE-Estimator framework at Crypto 2020 [26], which embeds perfect, modular, approximate, and short-vector hints into a Distorted Bounded Distance Decoding (DBDD) formulation. However, the DDGR framework integrates hints sequentially, which leads to limited scalability for high-dimensional LWE and NTRU instances. At Asiacrypt 2023, May and Nowakowski (MN) proposed the approach LLL breaks LWE, leveraging a parallel embedding technique [27] that can incorporate substantially more hints in a single instance compared to [26]. When attacking NTRU, their method models NTRU as a Ring-LWE instance with $\mathbf{b} = \mathbf{0}^m$. Nevertheless, for perfect hints and modular hints, the LLL-based approach is significantly slower than for mod- q hints and requires additional scaling factors. Moreover, its capability to handle approximate hints is inferior to that of the DDGR framework. In addition, the MN method is restricted to secret-only hint types; for NTRU instances, it cannot combine joint leakage information from both f and g . Subsequently, Kirshanova, May, and Nowakowski [28] introduced an almost-parallel hint embedding strategy. More recently, the CJC method [29] further decomposes LWE with hints into Z-SIS and LWE components, enabling efficient lattice constructions in the regime of excessive hints. Although these works have laid important foundations, several core issues remain unresolved. Therefore, in the next section, we focus on the research motivations of this work.

A. Motivation

Among lattice-based post-quantum cryptographic schemes, NTRU-based digital signatures such as Falcon and its variants Mitaka and Hawk have attracted considerable attention due to their compact structure and high implementation efficiency. However, the strong algebraic structure underlying these schemes may also introduce potential vulnerabilities in physical-attack settings. In particular, side-channel and fault-injection attacks may cause partial leakage of the secret polynomials f and g , thereby providing exploitable side information that can significantly weaken the overall security of the system. Consequently, evaluating how such partial leakage affects the residual security of NTRU-type schemes has become a critical research problem.

A series of recent works, such as DDGR [26], MN [27], and CJC [29], has investigated the incorporation of hints into lattice reduction attacks. However, these methodologies are primarily tailored for standard LWE problems, leaving their applicability to structured NTRU lattices still limited. The DDGR framework analyzes residual security for NTRU-based key-encapsulation mechanisms (NTRU-KEM), such as NTRU-HPS and NTRU-HRSS, under partial-leakage scenarios where

the secret keys are ternary polynomials with coefficients in $\{-1, 0, 1\}$. However, DDGR does not address NTRU-based signature schemes (NTRU-Sign). For example, the NIST-standardized NTRU-Sign scheme falcon uses secrets sampled from a discrete Gaussian distribution with significantly larger coefficient ranges, in Falcon-512, the coefficients are integers within the range of $[-31, 31]$.

Moreover, although the MN method reports successful Falcon-512 key recovery using 233 perfect hints, its experimental setup relies on specially constructed hints that artificially reduce the lattice volume; such hints do not correspond to realistic physical leakage (e.g., known secret coefficients). Under realistic leakage models, 233 perfect hints are insufficient to recover the key, an issue that we examine in detail in this work. In addition, existing approaches such as MN [27] and CJC [29] cannot jointly exploit partial leakage from both f and g , leaving unresolved the practical challenge of efficiently integrating joint leakage, including perfect and modular hints from both secret polynomials to improve key recovery performance.

Another dimension that remains largely unexplored is the secure utilization of noisy secret leakage, i.e., approximate hints. To date, no prior work has systematically assessed the impact of jointly leaked approximate hints of f and g on the security of Falcon. The MN method [27] handles approximate hints poorly, while the CJC method [29] requires additional perfect hints to process approximate leakage, which is an unrealistic assumption in real world attacks, where adversaries typically obtain only noisy side information. Driven by this gap, the present work adopts the DDGR framework to quantitatively evaluate the residual security of Falcon under joint approximate-hint leakage on both f and g .

TABLE I: Comparison of supported hint types across different methods.

Method	Scheme	Integrating hint types			
		Perfect	Modular	Approximate	Joint leakage
DDGR [26]	NTRU-KEM	✓*	✓*	✓*	✓*
MN [27]	NTRU-Sign	✓	✓	×	×
CJC [29]	NTRU-Sign	✓	✓	✓**	×
This work	NTRU-Sign	✓	✓	✓	✓

✓ Supported for NTRU-Sign; × Not supported for NTRU-Sign.

✓* Supported only for NTRU-KEM.

✓** Requires additional perfect hints (not independent).

A consolidated comparison of supported hint types across different methods is summarized in Table I. Motivated by the identified gaps, this work aims to systematically quantify the impact of different types of secret-key leakage on the residual security of NTRU-based signature schemes. To assess security degradation under various leakage models, we adopt standard lattice-based metrics, using the BKZ block size parameter β to estimate the hardness of key recovery. The “core-SVP methodology” may further convert β into a bit-level security estimate [30]. Assuming access to an optimal Shortest Vector

Problem (SVP) oracle, the complexity of lattice sieving is approximately $\lfloor 0.292\beta \rfloor$ in the classical setting and $\lfloor 0.265\beta \rfloor$ in the quantum setting. These metrics allow for a precise and quantitative evaluation of security loss under different types of leakage.

B. Our Contributions

We investigate how partial perfect hints, modular hints, and approximate hints on the secret key affect the residual security of NTRU-based signature schemes such as Falcon, Mitaka, and Hawk, and how such information can be efficiently exploited to improve full secret key-recovery performance. Our main contributions are summarised as follows:

- We propose a dimension-reduction-based key-recovery method for Falcon under perfect-hint leakage. For leakage models in which partial secret coefficients are known, we design the DSRNA technique, which eliminates known coefficients and redundant samples to transform the original problem into a lower-dimensional NTRU instance with fewer samples, enabling more efficient lattice reduction and avoiding costly sublattice construction. Compared with the method of May–Nowakowski presented at Asiacrypt 2023, our attack achieves approximately a $5.8\times$ speedup for Falcon-512 when 400 coefficients are leaked, and more than a $29.4\times$ speedup for Falcon-1024 when 905 coefficients are leaked.
- We systematically evaluate the impact of modular hints on the residual security of NTRU-based schemes. In particular, we investigate mod- q hints and LSBs hints and analyze their influence on the remaining security of Falcon and Hawk, quantifying the resulting reduction in the required BKZ blocksize β . In addition, we examine the effect of mod- q hints on the security of Mitaka.
- We further analyze the effect of approximate hints on residual security, as leaked side information is often noisy in practical attack scenarios. Considering different quantities and noise levels of approximate hints on the secret polynomial f , we conduct comprehensive evaluations on Falcon-512 and Falcon-1024 and determine the BKZ blocksize required for full key recovery. Our results reveal a clear quantitative relationship among noise magnitude, the number of approximate hints, and the remaining security level.
- We present a unified joint-leakage framework capable of simultaneously integrating multiple types of leakage from both f and g . For perfect hints, mod- q hints, and LSBs hints, we propose methods that incorporate leakage from both secret polynomials in a unified manner, enabling the concurrent exploitation of side information from f and g and significantly improving practical key-recovery efficiency. We further evaluate the impact of such joint leakage on the residual security of NTRU-based signature schemes, including Falcon, Mitaka, and Hawk.

II. PRELIMINARIES

A. Notations

Let $\phi = X^n + 1$ for $n = 2^\kappa$, a power of two, and $q \in \mathbb{N}^*$. We denote by $\mathbb{R} = \mathbb{Z}[x]/(X^n + 1)$ and $\mathbb{R}_q = \mathbb{Z}_q[x]/(X^n + 1)$. Scalars are written in regular font. Bold lower-case letters denote vectors with coefficients in R or R_q and are assumed to be row vectors by default. Bold upper-case letters denote matrices. For a vector $\mathbf{b}$, we denote its transpose by $\mathbf{b}^T$. For a matrix $\mathbf{A}$ with row vectors $\mathbf{a}_1, \dots, \mathbf{a}_n$, we write $\mathbf{A} = [\mathbf{a}_1^T; \dots; \mathbf{a}_n^T]$, and denote the transpose by $\mathbf{A}^T$. The i -th standard basis vector of $\mathbb{R}^n$ is denoted by $\mathbf{e}_i$ (e.g., $\mathbf{e}_1 = (1, 0, \dots, 0)$). The n -dimensional identity matrix is denoted by $\mathbf{I}_n$, the all-zero $(n \times m)$ matrix by $\mathbf{0}_{n \times m}$, and the n -dimensional all-zero vector by $\mathbf{0}^n$. When dimensions are clear from context, we omit the subscripts. A polynomial $f(x) = f_0 + f_1x + \dots + f_{n-1}x^{n-1}$ in R_q is identified with its coefficient vector $\mathbf{f} = (f_0, f_1, \dots, f_{n-1})$, treated as a row vector. The Euclidean norm and inner product are denoted by $\|\cdot\|$ and $\langle \cdot, \cdot \rangle$, respectively.

B. Lattice Hard Problems

Definition 1 (Lattice). Let $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n \in \mathbb{R}^m$ be linearly independent vectors. We form the basis matrix $\mathbf{B} \in \mathbb{R}^{n \times m}$ whose rows are the basis vectors, i.e., $\mathbf{B} = (\mathbf{b}_1; \mathbf{b}_2; \dots; \mathbf{b}_n)$. The lattice generated by $\mathbf{B}$ is $\mathcal{L}(\mathbf{B}) = \{\mathbf{x}\mathbf{B} : \mathbf{x} \in \mathbb{Z}^n\}$. The volume of Λ is defined as $\text{Vol}(\Lambda) = \sqrt{\det(\mathbf{B}\mathbf{B}^T)}$.

Definition 2 (Learning with Errors, LWE [31]). Let $n, m, q \in \mathbb{Z}$, and let χ be a distribution over $\mathbb{Z}$ with zero mean and standard deviation σ . Given $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{b} = \mathbf{s}\mathbf{A} + \mathbf{e} \bmod q$, where $\mathbf{s} \leftarrow \chi^n$ and $\mathbf{e} \leftarrow \chi^m$, the LWE problem is to recover the secret vector $\mathbf{s}$ and the noise vector $\mathbf{e}$. The tuple $(\mathbf{A}, \mathbf{b}, q)$ is called an LWE instance.

Definition 3 (NTRU problem [5]). Given the public key $h \equiv g/f \pmod{q}$ where f, g are sampled from a distribution χ , the NTRU problem is to recover (f, g) up to rotation. For Falcon, this is equivalent to solving a Ring-LWE instance $\mathbf{f}\mathbf{H}_h - \mathbf{g} \equiv \mathbf{0} \pmod{q}$, where $\mathbf{H}_h \in \mathbb{Z}_q^{n \times n}$ is the negacyclic matrix of h . The coefficients of the secret polynomials f and g are sampled i.i.d. from a discrete Gaussian distribution with variance $\sigma_{f,g} = 1.17\sqrt{q/(2n)}$ [4], leading to an expected secret norm $\mathbb{E}[\|(f, g)\|] \approx 1.17\sqrt{q}$.

Definition 4 (Hint Matrix [27]). Let $\mathbf{v}_i \in \mathbb{Z}^{1 \times n}$, $\bar{\mathbf{v}}_i = (\mathbf{v}_i, \ell_i) \in \mathbb{Z}^n \times \mathbb{Z}$ for $i = 1, \dots, k$. We define the corresponding hint matrix $\mathbf{H} \in \mathbb{Z}_q^{(n+1) \times k}$ as:

$$\mathbf{H} = \text{Hint}(\bar{\mathbf{v}}_1, \dots, \bar{\mathbf{v}}_k) := \begin{pmatrix} \mathbf{V} \\ \boldsymbol{\ell} \end{pmatrix} = \begin{pmatrix} \mathbf{v}_1^T & \dots & \mathbf{v}_k^T \\ \ell_1 & \dots & \ell_k \end{pmatrix}$$

C. NTRU with Hints

In post-quantum cryptographic schemes based on NTRU lattices, the secret polynomials f and g may satisfy certain structural constraints, or partial information about them may be leaked through attack methods such as side-channel analysis. These constraints or leaked information are collectively referred to as hints. According to [26], [28], for the NTRU

problem, an attacker may obtain several types of hint information related to the secret $\mathbf{f}$ and $\mathbf{g}$ or the lattice Λ . Let us assume the attacker knows the parameters $\mathbf{v}, l, k, \sigma, \epsilon_\sigma, q$,

- Perfect hints: $\langle \mathbf{f}, \mathbf{v} \rangle = \ell$ define a hyperplane intersecting the lattice.
- Modular hints: $\langle \mathbf{f}, \mathbf{v} \rangle \equiv \ell \pmod k$ sparsify the lattice.
- Approximate hints: $\langle \mathbf{f}, \mathbf{v} \rangle = \ell + \epsilon_\sigma$ reduce the covariance of the secret.
- Short vector hints: $\mathbf{v} \in \Lambda$ can be projected away from the lattice.

D. Lattice Reduction

Lattice reduction is fundamental to cryptanalysis. It aims to find a basis with short, nearly orthogonal vectors to solve hard problems like SVP and Closest Vector Problem (CVP). The BKZ algorithm [32], [33] enhances the LLL algorithm by solving SVP on sublattices of block size β . This parameter β determines both the algorithm's effectiveness and its computational cost.

The success of BKZ is typically evaluated using the following heuristic. For a lattice Λ and a target vector $\mathbf{t}$, the BKZ algorithm with block size β can solve a unique SVP (uSVP) instance if

$$\sqrt{\beta / \dim(\Lambda)} \cdot \|\mathbf{t}\| \leq \delta_\beta^{2\beta - \dim(\Lambda) - 1} \cdot \text{vol}(\Lambda)^{1/\dim(\Lambda)}, \quad (1)$$

where δ_β is the root-Hermite factor estimated under the Gaussian Heuristic [34] as

$$\delta_\beta = \left((\pi\beta)^{1/\beta} \cdot \frac{\beta}{2\pi e} \right)^{1/(2\beta-2)}. \quad (2)$$

In our analysis, the minimal block size β required to solve the lattice instance serves as the primary metric for residual security. Accurate estimation of this cost is therefore essential for evaluating the strength of lattice-based schemes.

E. The Primal Attack [27]

For an LWE instance $(\mathbf{A}, \mathbf{b})$ satisfying $\mathbf{s}\mathbf{A} + \mathbf{e} \equiv \mathbf{b} \pmod q$, the standard embedding yields the LWE lattice with basis

$$\mathbf{B}^{\text{LWE}} = \begin{pmatrix} q\mathbf{I}_m & 0 & 0 \\ \mathbf{A} & \mathbf{I}_n & 0 \\ \mathbf{b} & 0 & 1 \end{pmatrix}, \quad (3)$$

which contains the short vector $\mathbf{t}_{\text{LWE}} = (-\mathbf{e}, \mathbf{s}, -1)$. A primal lattice reduction attack applies BKZ to $\mathbf{B}^{\text{LWE}}$ to recover this vector.

NTRU as a Structured R-LWE Instance. For NTRU, let $\mathbf{f} \in \mathbb{Z}_q^n, \mathbf{g} \in \mathbb{Z}_q^m$ denote the coefficient vectors of the secret polynomials, and let $\mathbf{H}_h \in \mathbb{Z}_q^{n \times m}$ be the negacyclic matrix induced by the public key $h = g/f \pmod q$. Embedding this system yields the NTRU lattice with basis

$$\mathbf{B}^{\text{NTRU}} = \begin{pmatrix} q\mathbf{I}_m & 0 & 0 \\ \mathbf{H}_h & \mathbf{I}_n & 0 \\ \mathbf{0} & 0 & 1 \end{pmatrix}, \quad (4)$$

which contains the short vector $\mathbf{t}_{\text{NTRU}} = (\mathbf{g}, \mathbf{f}, -1)$.

III. INTEGRATING PSCL OF SECRET AS PERFECT HINTS IN NTRU INSTANCE

This section studies the security impact of Partial Secret Coefficient Leakage (PSCL) on the NTRU secret polynomial f and how to recover the remaining coefficients of f . Since NTRU-based signature schemes typically sample the secret polynomials f and g from the same distribution, we investigate the impact on security strength not only under the partial coefficient leakage of f alone but also under the simultaneous leakage of coefficients from both f and g . The leakage restricted to f provides perfect hints of the form $\langle \mathbf{f}, \mathbf{v} \rangle = \ell$, where the adversary knows the vector $\mathbf{v} \in \mathbb{Z}^{1 \times n}$ and the scalar $\ell \in \mathbb{Z}$. We examine how different choices of $\mathbf{v}$ affect the lattice volume, quantify the resulting security degradation, and present an efficient attack that reduces the overall computational cost.

A. Effect of Hint Construction on Lattice Volume and Security

In the experimental framework established by [27], hint vectors are conventionally assumed to be uniformly distributed in $\mathbb{Z}_q^n$. However, this assumption is invalidated when side-channel leakage reveals specific partial coefficients of the secret key. Let the hint matrix be defined as $\mathbf{H} = \text{Hint}(\bar{\mathbf{v}}_1, \dots, \bar{\mathbf{v}}_k) \in \mathbb{Z}_q^{(n+1) \times k}$, where $\mathbf{V} = [\mathbf{v}_1^T, \dots, \mathbf{v}_k^T] \in \mathbb{Z}^{n \times k}$ comprises k linear constraint vectors, and $\ell = (\ell_1, \dots, \ell_k) \in \mathbb{Z}^{1 \times k}$ represents the corresponding leaked values such that $\ell_i = \mathbf{f}\mathbf{v}_i^T$. We denote $\mathcal{L}(\mathbf{H})$ as the lattice spanned by $\mathbf{H}$, Λ^{NTRU} as the original NTRU lattice, and $\Lambda_{\mathbf{H},k}^{\text{NTRU}}$ as the sub-lattice derived after integrating k perfect hints. Since each hint imposes an additional linear constraint, the determinant of the resulting lattice evolves according to the relation $\det \Lambda_{\mathbf{H},k}^{\text{NTRU}} = \det \Lambda^{\text{NTRU}} \cdot \det \mathcal{L}(\mathbf{H})$. The impact on the lattice geometry depends critically on the construction of the hint vectors $\mathbf{v}_i$, which we categorize into two primary types:

Type I (Random Hint Vectors). Each $\mathbf{v}_i$ is sampled uniformly from $\mathbb{Z}_q^n$, consistent with the model in [27]. The resulting leakage values ℓ_i typically possess large magnitudes, causing $\det \mathcal{L}(\mathbf{H})$ to grow approximately as $(\sqrt{k}/(12e) \cdot q)^k$. A significant increase in the lattice volume facilitates the identification of short vectors during the primal lattice reduction attack, thereby furthering the recovery of the secret key $\mathbf{f}$.

Type II (Standard Basis Hint Vectors). These hints directly reveal individual coefficients of f via $\mathbf{v}_i = \mathbf{e}_{j_i}$. If the indices $\{j_1, \dots, j_k\}$ are distinct, the matrix $\mathbf{V}$ forms a sub-matrix of the identity $\mathbf{I}_n$, yielding a lattice determinant of $\det \mathcal{L}(\mathbf{V}) = 1$. Because ℓ is strictly determined by $\mathbf{V}$ and $\mathbf{f}$, these appended scalars introduce no additional degrees of freedom, resulting in $\det \mathcal{L}(\mathbf{H}) = 1$. Consequently, Type II hints induce a reduction in the dimension of the primal lattice while keeping the determinant invariant.

In summary, the specific construction of $\mathbf{v}_i$ dictates the hardness of the primal attack. Type I hints facilitate the identification of short vectors during primal lattice reduction more effectively than Type II hints by significantly increasing the lattice volume, thereby furthering the recovery of the

secret key $\mathbf{f}$. However, prior work mainly focuses on Type I hints. Here, we focus on Type II hints, with a complementary analysis of Type I hints.

B. Evaluating the Impact of PSCL on Residual Security

Perfect hints of the form $\langle \mathbf{f}, \mathbf{v} \rangle = \ell$ provide critical auxiliary information for secret-key recovery. In NTRU-HPS, the fixed-weight ternary structure of the secret vector naturally induces perfect hints. For Falcon, side-channel analyses [18], [19], [21] have shown that partial coefficients of the secret polynomial f can leak, necessitating a quantitative assessment of security degradation relative to the leakage scale.

To model coefficient leakage, DDGR [26] proposed integrating perfect hints into a DBDD instance by recursively intersecting the LWE lattice with orthogonal constraints (e.g., $\mathbf{e}_{m+i}^\perp$). This iterative process constructs a sequence of sublattices $\Lambda_i = \Lambda_{i-1} \cap \mathbf{e}_{m+i}^\perp$ until all k constraints are embedded. Although theoretically polynomial-time, this sequential methodology is computationally inefficient for large-scale cryptanalysis. To resolve this bottleneck, the MN framework [27] enables simultaneous, single-shot hint integration, facilitating scalable and parallelizable attacks against high-dimensional instances.

We evaluate the concrete risks for Falcon at $n = 512$ and $n = 1024$ by characterizing the relationship between the number of leaked coefficients (f or g) and the residual security strength. Using the leaky-LWE-Estimator [26], we embed perfect-hint constraints into the Primal attack pipeline to quantify the reduction in residual lattice dimension d' and the required BKZ blocksize β . This analysis identifies the critical thresholds where partial leakage renders attacks practically feasible.

TABLE II: Residual d' and β for Falcon-512 with perfect hints of f .

Perfect Hints	400	405	410	415	420	425	430	435	440
Remaining d'	295	284	273	261	247	232	216	198	177
Remaining β	45.67	39.02	31.69	23.69	18.41	12.76	8.72	4.82	2

TABLE III: Residual d' and β for Falcon-1024 with perfect hints of f .

Perfect Hints	905	910	915	920	925	930	935	940	945	950
Remaining d'	300	290	279	266	253	239	223	206	186	175
Remaining β	40.72	34.16	26.34	21.25	15.4	11.28	8.1	4.68	2	2

For Falcon-512 and Falcon-1024, we view the NTRU problem as an RLWE instance and apply the leaky-LWE-Estimator for security evaluation [26]. Because [26] is based on a Primal lattice-reduction attack, the lattice dimension without any integrated hints is $m + n + 1$. In the Falcon-512 case, the no-hint dimension is 1025, and the estimated BKZ blocksize is approximately $\beta = 480$ [19]. For Falcon-1024, the no-hint dimension is 2049, with an estimated security level of about $\beta = 974$. With both the perfect hints from partially known coefficients and the lattice's inherent short-vector hints

integrated, the residual security strength, reduced dimension d , and the corresponding β -level changes across all perfect-hint leakage levels under the Primal lattice-reduction attack are summarized in Tables II and III. Based on our evaluation, for Falcon-512, leaking 400 coefficients of the secret polynomial f reduces the residual security to $\beta = 45.67$. This implies that a BKZ blocksize of only $\beta = 45.67$ is sufficient to recover the remaining 112 coefficients of f , representing a reduction of $\Delta\beta = 434.33$ relative to the no-leakage (no-hint) baseline. Similarly, for Falcon-1024, leaking 905 coefficients yields $\beta = 40.72$, meaning a blocksize of $\beta = 40.72$ is required to recover the remaining 119 coefficients of f , which corresponds to a reduction of $\Delta\beta = 933.28$ compared with the no-leakage baseline.

C. Integrating Partial Coefficients f into an NTRU Instance

In this section, we propose Algorithm 1, called the Dimension-and-Sample Reduced NTRU Attack (DSRNA). The goal of DSRNA is to transform a high-dimensional NTRU instance with a large number of samples into a lower-dimensional instance with fewer samples, by jointly exploiting partial coefficient leakage of the secret key and sample elimination techniques. The reduced instance is then subjected to BKZ lattice basis reduction in order to recover the complete secret key.

When invoking the BKZ algorithm, employing an excessive number of samples is typically redundant, since BKZ can inherently disregard unnecessary samples during the reduction process [26]. Current security estimators indicate that an overabundance of samples may cause the estimators to overestimate the required block size, while the actual block size β needed for successful secret recovery remains largely unaffected [27]. This observation suggests that merely increasing the number of samples does not intrinsically enhance the success probability of the attack. Therefore, in order to improve attack efficiency and reduce computational overhead, our DSRNA method reduces the number of processed samples via random sampling, which probabilistically filters out redundant samples.

In the DSRNA framework, the efficiency and feasibility of the primal attack are jointly determined by the interplay among the lattice dimension, the lattice volume, and the norm of the target vector. As shown in Table IV, we first eliminate the known coefficients of the secret key, thereby obtaining the number of remaining unknown coefficients, denoted by $n' = n - k$, where n' is known. We then employ a heuristic leaky-LWE-Estimator to determine, for the minimal BKZ block size β , the optimal total lattice dimension d' required by the primal attack. Based on this estimate, the optimal number of samples is defined as $m' = d' - n' - 1$, representing the minimal sample count required to maintain target vector distinguishability while minimizing computational effort; the formal proof of the optimality of m' is provided in Appendix B. Here, m' denotes an arbitrary selection of m' samples, which preserves the inclusion of the secret key within the constructed lattice while simultaneously reducing the computational cost

Algorithm 1 Dimension-and-Sample Reduced NTRU Attack (DSRNA)

Input: NTRU instance $(\mathbf{H}_h, \mathbf{b} = \mathbf{0}^m, q)$; known secret coefficients $\mathbf{f}^{[1:k]}$; optimized sample size m' .

Output: Recovered secret coefficients $\mathbf{f}' \in \mathbb{Z}^{n-k}$ or $\perp$.

- 1: **Preprocessing:** Set $n' = n - k$ and partition the secret vector as $\mathbf{f} = (\mathbf{f}^{[1:k]}, \mathbf{f}^{[k+1:n]})$.
- 2: **Residual Computation (row-consistent):** Eliminate the contribution of the known secret coefficients:

$$\mathbf{b}' = \mathbf{b} - \mathbf{f}^{[1:k]} \cdot \mathbf{H}_h^{[1:k,:]} \pmod{q}.$$

- 3: **Row-Restricted NTRU System:** Define the reduced NTRU matrix by retaining only the rows associated with the unknown secret coefficients:

$$\mathbf{H}'_h = \mathbf{H}_h^{[k+1:n,:]} \in \mathbb{Z}_q^{n' \times m}.$$

- 4: **Sample Reduction:** Select a uniformly random subset of m' samples indexed by $\mathcal{I} = \{i_1, i_2, \dots, i_{m'}\} \subseteq \{1, \dots, m\}$, where $i_1 < i_2 < \dots < i_{m'}$, and restrict the system to these samples:

$$\mathbf{H}''_h = \mathbf{H}'_h[:, \mathcal{I}] \in \mathbb{Z}_q^{n' \times m'}, \quad \mathbf{b}'' = \mathbf{b}'[\mathcal{I}].$$

- 5: **Reduced Primal Lattice Construction:** Construct the reduced NTRU primal lattice basis:

$$\mathbf{B}_{\text{red}}^{\text{NTRU}} = \begin{pmatrix} q\mathbf{I}_{m'} & 0 & 0 \\ \mathbf{H}''_h & \mathbf{I}_{n'} & 0 \\ \mathbf{b}'' & 0 & 1 \end{pmatrix}.$$

- 6: **Lattice Basis Reduction:** Apply BKZ with block size β to $\mathbf{B}_{\text{red}}^{\text{NTRU}}$ and search for a short vector $\mathbf{t} = (\mathbf{g}', \mathbf{f}', -1)$.
 - 7: **Verification:**
 - 8: **if** $\|(\mathbf{g}', \mathbf{f}')\|$ is sufficiently small and $\mathbf{f}'\mathbf{H}''_h - \mathbf{g}' \equiv \mathbf{b}'' \pmod{q}$ **then**
 - 9: **return** $\mathbf{f}'$.
 - 10: **else**
 - 11: **return** $\perp$.
 - 12: **end if**
-

of the attack. The validity and correctness of Algorithm 1 are established in Appendix A.

For Falcon-512, the optimal values of m' and the minimal block size β determined by the leaky-LWE-Estimator are reported in Table IV. Although the optimal m' can, in principle, be automatically identified during the BKZ reduction process [27], DSRNA explicitly determines m' prior to launching the attack. By discarding redundant samples in advance, this framework substantially reduces the running time and avoids the LLL-based sublattice construction step required in previous approaches [27]. More detailed performance comparisons are presented in Figs. 3 and 4.

An NTRU instance can be represented as $(\mathbf{H}_h, \mathbf{b} = \mathbf{0}^m, q)$, where $\mathbf{H}_h$ denotes the negacyclic matrix induced by the public-key polynomial h . The core idea of DSRNA is to eliminate NTRU samples corresponding to known secret co-

TABLE IV: Dim-and-Sample Reduced NTRU Attack for Falcon-512.

Perfect Hints	400	405	410	415	420	425	430	435	440
Remaining d'	295	284	273	261	247	232	216	198	177
Remaining n'	112	107	102	97	92	87	82	77	72
Samples m'	182	176	170	163	154	145	133	120	104
BKZ β	45.64	38.99	31.66	23.67	18.38	12.75	8.71	4.82	2

efficients, thereby reducing the dimension of the primal lattice attack and further decreasing the required BKZ block size. The primary objective of the attack is to reconstruct the complete private key by recovering the unknown coefficients of the secret polynomial f . Once f is successfully recovered, the corresponding polynomial g can be efficiently derived from the public key h , enabling full reconstruction of the secret key sk . The DSRNA method proposed in this paper is applicable to attacking known secret-key coefficient leakages in Falcon, Mitaka, and Hawk. As a case study, in Section VI-B, we perform key recovery attacks on Falcon-512 and Falcon-1024 under various scenarios of partial-coefficient leakage using the DSRNA method. In Appendix C-A, the DSRNA approach is utilized to conduct key recovery attacks on Mitaka-512.

D. Integrating PSCL of f and g as Perfect and Mod- q Hints in NTRU Instances

In practical cryptanalysis of NTRU-like schemes, the secret key typically comprises two polynomials, f and g . Under realistic physical attack scenarios, such as side-channel analysis, an adversary may observe partial secret coefficient leakage (PSCL) for both polynomials concurrently. A fundamental challenge remains: how to exploit this leakage from both f and g to enhance the success probability of full secret-key recovery. Existing approaches still suffer from limited flexibility in this regard. Although the framework proposed by Dachman-Soled et al. [26] allows the combination of leakage information from both f and g , it relies on a sequential integration of hints, which incurs substantial computational overhead and restricts the number of hints that can be feasibly incorporated within a practical time budget. Moreover, their analysis is primarily tailored to ternary secret distributions, rather than secret keys sampled from the discrete Gaussian distributions with larger coefficients. In contrast, the methods proposed by MN [26] and CJC [29] are limited to secret-only leakage and lack a mechanism for simultaneously integrating hints originating from both f and g .

To address these limitations, we propose a unified lattice framework that enables the simultaneous integration of partial coefficient leakage from both components of the secret key. Throughout this formalization, we strictly adhere to the row-vector convention: let $\mathbf{f} \in \mathbb{Z}^{1 \times n}$ and $\mathbf{g} \in \mathbb{Z}^{1 \times m}$ represent the secret polynomials, and let $\mathbf{H}_h \in \mathbb{Z}_q^{n \times m}$ be the public matrix satisfying the NTRU relation $\mathbf{f}\mathbf{H}_h \equiv \mathbf{g} \pmod{q}$.

Type I (Independent perfect hints). Although the hints on $\mathbf{f}$ and $\mathbf{g}$ are independent at the leakage level, all constraints are

eventually embedded as linear relations on $\mathbf{f}$ through the public NTRU relation. The independent perfect hints relations are defined as $\mathbf{f}\mathbf{v}_i^T = \ell_{1i}$ and $\mathbf{g}\mathbf{w}_j^T = \ell_{2j}$, where $\mathbf{v}_i \in \mathbb{Z}^{1 \times n}$ for $i = 1, \dots, k_f$ and $\mathbf{w}_j \in \mathbb{Z}^{1 \times m}$ for $j = 1, \dots, k_g$, $\ell_{1i}, \ell_{2j} \in \mathbb{Z}$, $\mathbf{Q}_1 = q\mathbf{I}_{k_g}$. For independent perfect hints, the indices i and j must be mutually independent.

By substituting the NTRU relation, each hint on $\mathbf{g}$ induces a modular hint on $\mathbf{f}$: $(\mathbf{f}\mathbf{H}_h)\mathbf{w}_j^T \equiv \ell_{2j} \pmod{q} \implies \mathbf{f}(\mathbf{H}_h\mathbf{w}_j^T) \equiv \ell_{2j} \pmod{q}$. We define $\mathbf{V}_{k_f} = [\mathbf{v}_1^T, \dots, \mathbf{v}_{k_f}^T] \in \mathbb{Z}^{n \times k_f}$, $\mathbf{W}_h = [\mathbf{H}_h\mathbf{w}_1^T, \dots, \mathbf{H}_h\mathbf{w}_{k_g}^T] \in \mathbb{Z}_q^{n \times k_g}$. Let $\ell_1 = (\ell_{11}, \dots, \ell_{1k_f}) \in \mathbb{Z}^{1 \times k_f}$ and $\ell_2 = (\ell_{21}, \dots, \ell_{2k_g}) \in \mathbb{Z}^{1 \times k_g}$. The unified primal NTRU basis $\mathbf{B}_{\mathbf{H}_1}^{\text{NTRU}}$ is constructed as

$$\mathbf{B}_{\mathbf{H}_1}^{\text{NTRU}} = \begin{pmatrix} q\mathbf{I}_m & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{H}_h & \mathbf{V}_{k_f} & \mathbf{W}_h & \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \ell_1 & \ell_2 & \mathbf{0} & 1 \end{pmatrix}. \quad (5)$$

Theorem 1. *Under the above construction, the lattice $\Lambda(\mathbf{B}_{\mathbf{H}_1}^{\text{NTRU}})$ contains the short vector $\mathbf{t}_{\mathbf{H}_1} = (\mathbf{g}, \mathbf{0}^{k_f}, \mathbf{0}^{k_g}, \mathbf{f}, -1)$.*

Proof. To prove that $\mathbf{t}_{\mathbf{H}_1} = (\mathbf{g}, \mathbf{0}^{k_f}, \mathbf{0}^{k_g}, \mathbf{f}, -1) \in \Lambda(\mathbf{B}_{\mathbf{H}_1}^{\text{NTRU}})$, we define the integer coefficient vector $\mathbf{x}_1 = (\mathbf{r}, \mathbf{c}, \mathbf{f}, -1)$, where $\mathbf{r} \in \mathbb{Z}^m$ and $\mathbf{c} \in \mathbb{Z}^{k_g}$ are quotient vectors. Specifically, $\mathbf{r}q + \mathbf{f}\mathbf{H}_h = \mathbf{g}$ follows from the NTRU relation, while $\mathbf{f}\mathbf{V}_{k_f} - \ell_1 = \mathbf{0}$ and $\mathbf{c}\mathbf{Q}_1 + \mathbf{f}\mathbf{W}_h - \ell_2 = \mathbf{0}$ follow from the perfect and modular hints. The product $\mathbf{x}_1\mathbf{B}_{\mathbf{H}_1}^{\text{NTRU}}$ is computed by block multiplication as $\mathbf{x}_1\mathbf{B}_{\mathbf{H}_1}^{\text{NTRU}} = (\mathbf{r}q\mathbf{I}_m + \mathbf{f}\mathbf{H}_h, \mathbf{f}\mathbf{V}_{k_f} - \ell_1, \mathbf{c}\mathbf{Q}_1 + \mathbf{f}\mathbf{W}_h - \ell_2, \mathbf{f}\mathbf{I}_n, -1) = \mathbf{t}_{\mathbf{H}_1}$. Since $\mathbf{x}_1$ consists of integer entries, $\mathbf{t}_{\mathbf{H}_1}$ is an integer linear combination of the rows of $\mathbf{B}_{\mathbf{H}_1}^{\text{NTRU}}$, completing the proof. $\square$

Type II (Joint perfect hints). Suppose we obtain k joint perfect hints of the form $\mathbf{f}\mathbf{v}_i^T + \mathbf{g}\mathbf{w}_i^T = \ell_i$ for $i = 1, \dots, k$. Substituting $\mathbf{g} \equiv \mathbf{f}\mathbf{H}_h \pmod{q}$ yields $\mathbf{f}\mathbf{v}_i^T + (\mathbf{f}\mathbf{H}_h)\mathbf{w}_i^T = \mathbf{f}(\mathbf{v}_i^T + \mathbf{H}_h\mathbf{w}_i^T) \equiv \ell_i \pmod{q}$. We define $\ell = (\ell_1, \dots, \ell_k) \in \mathbb{Z}^{1 \times k}$, the integrated hint column vectors $\mathbf{u}_i = \mathbf{v}_i^T + \mathbf{H}_h\mathbf{w}_i^T \in \mathbb{Z}_q^{n \times 1}$ and the corresponding matrix $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_k] \in \mathbb{Z}_q^{n \times k}$, $\mathbf{Q} = q\mathbf{I}_k$. The primal NTRU basis for joint leakage is

$$\mathbf{B}_{\mathbf{H}_2}^{\text{NTRU}} = \begin{pmatrix} q\mathbf{I}_m & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} & \mathbf{0} & \mathbf{0} \\ \mathbf{H}_h & \mathbf{U} & \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \ell & \mathbf{0} & 1 \end{pmatrix}. \quad (6)$$

Theorem 2. *Under the above construction, the lattice $\Lambda(\mathbf{B}_{\mathbf{H}_2}^{\text{NTRU}})$ contains the short vector $\mathbf{t}_{\mathbf{H}_2} = (\mathbf{g}, \mathbf{0}^k, \mathbf{f}, -1)$.*

Proof. To prove that $\mathbf{t}_{\mathbf{H}_2} = (\mathbf{g}, \mathbf{0}^k, \mathbf{f}, -1) \in \Lambda(\mathbf{B}_{\mathbf{H}_2}^{\text{NTRU}})$, we define the integer coefficient vector $\mathbf{x}_2 = (\mathbf{r}, \mathbf{c}, \mathbf{f}, -1)$, where $\mathbf{r} \in \mathbb{Z}^m$ and $\mathbf{c} \in \mathbb{Z}^k$ are quotient vectors. Specifically, $\mathbf{r}q + \mathbf{f}\mathbf{H}_h = \mathbf{g}$ follows from the NTRU relation, while $\mathbf{c}\mathbf{Q} + \mathbf{f}\mathbf{U} - \ell = \mathbf{0}$ follows from the joint hints $\mathbf{f}\mathbf{u}_i \equiv \ell_i \pmod{q}$. The product $\mathbf{x}_2\mathbf{B}_{\mathbf{H}_2}^{\text{NTRU}}$ is computed by block multiplication as: $\mathbf{x}_2\mathbf{B}_{\mathbf{H}_2}^{\text{NTRU}} = (\mathbf{r}q\mathbf{I}_m + \mathbf{f}\mathbf{H}_h, \mathbf{c}\mathbf{Q} + \mathbf{f}\mathbf{U} - \ell, \mathbf{f}\mathbf{I}_n, -1) = (\mathbf{g}, \mathbf{0}^k, \mathbf{f}, -1) = \mathbf{t}_{\mathbf{H}_2}$. Since $\mathbf{x}_2$ consists of integer entries, $\mathbf{t}_{\mathbf{H}_2}$ is an integer linear combination of the rows of $\mathbf{B}_{\mathbf{H}_2}^{\text{NTRU}}$, completing the proof. $\square$

This transformation enables the direct application of the combined modular and perfect hint integration method of [27], allowing joint leakage on $\mathbf{f}$ and $\mathbf{g}$ to be efficiently embedded into the NTRU lattice. As a result, a primal lattice reduction attack can be applied to recover $\mathbf{f}$ and ultimately reconstruct the secret key sk .

IV. INTEGRATING MODULAR HINTS

In this section, we focus on two common types of Modular Hints, namely the mod- q hint and the mod-2 hint that reflects low-bit leakage of secret key coefficients. Let $(\mathbf{H}_h, \mathbf{b}, q) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m \times \mathbb{N}$ denote an NTRU instance with secret polynomials f and g , and let $\bar{\mathbf{v}}_i = (\mathbf{v}_i, \ell_i, m_i) \in \mathbb{Z}^n \times \mathbb{Z} \times \mathbb{N}$ be a modular hint such that $\langle \mathbf{f}, \mathbf{v}_i \rangle \equiv \ell_i \pmod{m_i}$, for $i = 1, \dots, k$, where $\mathbf{f}$ denotes the secret. For modular hints, we interpret them as noiseless RLWE samples reduced modulo m_i .

A. Integrating Mod- q Hints into NTRU Instances

In NTRU-based cryptographic schemes, the leakage of mod- q hint information is a common and significant security concern. For instance, during the key generation phase of the Falcon signature scheme, Number Theoretic Transform (NTT) operations are applied to the secret polynomial f . If such operations are subjected to side-channel analysis, they may result in the leakage of mod- q hint information on f .

Regarding the resolution of LWE instances under mod- q leakage, the framework established in [27] primarily addresses the integration of *secret-only* mod- q hints. In contrast, this work focuses on the NTRU problem and investigates *joint* mod- q information leakage concerning the secret polynomials $\mathbf{f}$ and $\mathbf{g}$. In NTRU, the secret is the pair $(\mathbf{f}, \mathbf{g})$ satisfying $\mathbf{g} \equiv \mathbf{f}\mathbf{H}_h \pmod{q}$. We consider two representative forms of joint leakage on $(\mathbf{f}, \mathbf{g})$, which we classify as Independent Mod- q Leakage and Joint Mod- q Leakage.

Type I (Independent Mod- q hints). The hints on $\mathbf{f}$ and $\mathbf{g}$ are given independently as $\mathbf{f}\mathbf{v}_i^T \equiv \ell_{1i} \pmod{q}$ for $i = 1, \dots, k_f$ and $\mathbf{g}\mathbf{w}_j^T \equiv \ell_{2j} \pmod{q}$ for $j = 1, \dots, k_g$. Utilizing the public NTRU relation $\mathbf{g} \equiv \mathbf{f}\mathbf{H}_h \pmod{q}$, the hints on $\mathbf{g}$ are transformed into equivalent modular constraints on $\mathbf{f}$: $(\mathbf{f}\mathbf{H}_h)\mathbf{w}_j^T \equiv \ell_{2j} \pmod{q} \implies \mathbf{f}(\mathbf{H}_h\mathbf{w}_j^T) \equiv \ell_{2j} \pmod{q}$. We define $\mathbf{V}_{k_f} = [\mathbf{v}_1^T, \dots, \mathbf{v}_{k_f}^T] \in \mathbb{Z}^{n \times k_f}$, $\mathbf{W}_h = [\mathbf{H}_h\mathbf{w}_1^T, \dots, \mathbf{H}_h\mathbf{w}_{k_g}^T] \in \mathbb{Z}_q^{n \times k_g}$, $\mathbf{Q}_2 = q\mathbf{I}_{k_f}$. Let $\ell_1 = (\ell_{11}, \dots, \ell_{1k_f}) \in \mathbb{Z}^{1 \times k_f}$ and $\ell_2 = (\ell_{21}, \dots, \ell_{2k_g}) \in \mathbb{Z}^{1 \times k_g}$. The unified basis for embedding these hints into the NTRU primal lattice is

$$\mathbf{B}_{\mathbf{H}_3}^{\text{NTRU}} = \begin{pmatrix} q\mathbf{I}_m & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{H}_h & \mathbf{V}_{k_f} & \mathbf{W}_h & \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \ell_1 & \ell_2 & \mathbf{0} & 1 \end{pmatrix}. \quad (7)$$

Theorem 3. *Under the above construction, the lattice $\Lambda(\mathbf{B}_{\mathbf{H}_3}^{\text{NTRU}})$ contains the short vector $\mathbf{t}_{\mathbf{H}_3} = (\mathbf{g}, \mathbf{0}^{k_f}, \mathbf{0}^{k_g}, \mathbf{f}, -1)$.*

Proof. To prove that $\mathbf{t}_{\mathbf{H}_3} = (\mathbf{g}, \mathbf{0}^{k_f}, \mathbf{0}^{k_g}, \mathbf{f}, -1) \in \Lambda(\mathbf{B}_{\mathbf{H}_3}^{\text{NTRU}})$, we define the integer coefficient vector $\mathbf{x}_3 =$

$(\mathbf{r}, \mathbf{c}_1, \mathbf{c}_2, \mathbf{f}, -1)$, where $\mathbf{r} \in \mathbb{Z}^m$, $\mathbf{c}_1 \in \mathbb{Z}^{k_f}$, and $\mathbf{c}_2 \in \mathbb{Z}^{k_g}$ are quotient vectors. Specifically, $\mathbf{r}\mathbf{q} + \mathbf{f}\mathbf{H}_h = \mathbf{g}$ follows from the NTRU relation, while $\mathbf{c}_1\mathbf{Q}_2 + \mathbf{f}\mathbf{V}_{k_f} - \ell_1 = \mathbf{0}$ and $\mathbf{c}_2\mathbf{Q}_1 + \mathbf{f}\mathbf{W}_h - \ell_2 = \mathbf{0}$ follow from the modular hints $\mathbf{f}\mathbf{V}_{k_f} \equiv \ell_1 \pmod{q}$ and $\mathbf{f}\mathbf{W}_h \equiv \ell_2 \pmod{q}$. The product $\mathbf{x}_3\mathbf{B}_{\mathbf{H}_3}^{\text{NTRU}}$ is computed by block multiplication as: $\mathbf{x}_3\mathbf{B}_{\mathbf{H}_3}^{\text{NTRU}} = (\mathbf{r}\mathbf{q}\mathbf{I}_m + \mathbf{f}\mathbf{H}_h, \mathbf{c}_1\mathbf{Q}_2 + \mathbf{f}\mathbf{V}_{k_f} - \ell_1, \mathbf{c}_2\mathbf{Q}_1 + \mathbf{f}\mathbf{W}_h - \ell_2, \mathbf{f}\mathbf{I}_n, -1) = (\mathbf{g}, \mathbf{0}^{k_f}, \mathbf{0}^{k_g}, \mathbf{f}, -1) = \mathbf{t}_{\mathbf{H}_3}$. Since $\mathbf{x}_3$ consists of integer entries, $\mathbf{t}_{\mathbf{H}_3}$ is an integer linear combination of the rows of $\mathbf{B}_{\mathbf{H}_3}^{\text{NTRU}}$, completing the proof. $\square$

Type II (Joint Mod- q hints). Joint leakage is observed in the form $\mathbf{f}\mathbf{v}_i^T + \mathbf{g}\mathbf{w}_i^T \equiv \ell_i \pmod{q}$ for $i = 1, \dots, k$. Substituting the relation $\mathbf{g} \equiv \mathbf{f}\mathbf{H}_h \pmod{q}$ yields: $\mathbf{f}\mathbf{v}_i^T + (\mathbf{f}\mathbf{H}_h)\mathbf{w}_i^T = \mathbf{f}(\mathbf{v}_i^T + \mathbf{H}_h\mathbf{w}_i^T) \equiv \ell_i \pmod{q}$. We define $\mathbf{u}_i = \mathbf{v}_i^T + \mathbf{H}_h\mathbf{w}_i^T \in \mathbb{Z}_q^{n \times 1}$ and the corresponding hint matrix $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_k] \in \mathbb{Z}_q^{n \times k}$. Let $\ell = (\ell_1, \dots, \ell_k) \in \mathbb{Z}^{1 \times k}$ and $\mathbf{Q} = q\mathbf{I}_k$. The unified basis for embedding these hints into NTRU primal lattice is:

$$\mathbf{B}_{\mathbf{H}_4}^{\text{NTRU}} = \begin{pmatrix} q\mathbf{I}_m & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} & \mathbf{0} & \mathbf{0} \\ \mathbf{H}_h & \mathbf{U} & \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \ell & \mathbf{0} & 1 \end{pmatrix}. \quad (8)$$

Theorem 4. Under the above construction, the lattice $\Lambda(\mathbf{B}_{\mathbf{H}_4}^{\text{NTRU}})$ contains the short vector $\mathbf{t}_{\mathbf{H}_4} = (\mathbf{g}, \mathbf{0}^k, \mathbf{f}, -1)$.

Proof. To prove that $\mathbf{t}_{\mathbf{H}_4} = (\mathbf{g}, \mathbf{0}^k, \mathbf{f}, -1) \in \Lambda(\mathbf{B}_{\mathbf{H}_4}^{\text{NTRU}})$, we define the integer coefficient vector $\mathbf{x}_4 = (\mathbf{r}, \mathbf{c}, \mathbf{f}, -1)$, where $\mathbf{r} \in \mathbb{Z}^m$ and $\mathbf{c} \in \mathbb{Z}^k$ are quotient vectors. Specifically, $\mathbf{r}\mathbf{q} + \mathbf{f}\mathbf{H}_h = \mathbf{g}$ follows from the NTRU relation, while $\mathbf{c}\mathbf{Q} + \mathbf{f}\mathbf{U} - \ell = \mathbf{0}$ follows from the joint modular hints $\mathbf{f}\mathbf{U} \equiv \ell \pmod{q}$. The product $\mathbf{x}_4\mathbf{B}_{\mathbf{H}_4}^{\text{NTRU}}$ is computed via block multiplication: $\mathbf{x}_4\mathbf{B}_{\mathbf{H}_4}^{\text{NTRU}} = (\mathbf{r}\mathbf{q}\mathbf{I}_m + \mathbf{f}\mathbf{H}_h, \mathbf{c}\mathbf{Q} + \mathbf{f}\mathbf{U} - \ell, \mathbf{f}\mathbf{I}_n, -1) = (\mathbf{g}, \mathbf{0}^k, \mathbf{f}, -1) = \mathbf{t}_{\mathbf{H}_4}$. Since $\mathbf{x}_4$ consists of integer entries, $\mathbf{t}_{\mathbf{H}_4}$ is an integer linear combination of the rows of $\mathbf{B}_{\mathbf{H}_4}^{\text{NTRU}}$, completing the proof. $\square$

B. Integrating LSBs Hints into the NTRU Instances

As shown in [35], an adversary may obtain the least significant bits (LSBs) of secret key coefficients during the signing phase of Hawk [11]. In NTRU-based signature schemes, such leakage naturally arises in practical implementations. This type of leakage can be rigorously modeled as modular hints; in particular, LSBs leakage corresponds exactly to mod-2 modular hints on the secret key.

Since each LSBs reveals only one bit of information, such low-bit modular leakage is typically insufficient to directly recover the full secret key in practice. Nevertheless, existing approaches, including the MN method [27] and the CJC method [29], investigate modular hint leakage affecting only a single secret component, and do not consider scenarios in which modular hints on both $\mathbf{f}$ and $\mathbf{g}$ are simultaneously available. The primary objective of this section is to quantitatively evaluate the impact of low-bit modular leakage on the security of the Falcon and Hawk signature schemes, and to study how mod-2 hints on both $\mathbf{f}$ and $\mathbf{g}$ can be jointly exploited within a lattice-based attack framework. To this end, we identify and

analyze two representative forms of joint modular leakage. Throughout this section, we assume $m = n$.

It is worth emphasizing that, unlike the mod- q setting, joint LSBs constraints are embedded directly as LSBs hints on both $\mathbf{f}$ and $\mathbf{g}$, rather than converting the LSBs leakage on $\mathbf{g}$ into mod- q hints on $\mathbf{f}$. This distinction arises because when q is odd, mod- q hints and LSBs hints cannot be directly transformed into each other. We next describe two representative types of joint LSBs leakage on $\mathbf{f}$ and $\mathbf{g}$.

Type I (Independent LSBs hints). Assume two independent sets of LSBs hints such that $\mathbf{f}\mathbf{v}_i^T \equiv \ell_{1i} \pmod{2}$ and $\mathbf{g}\mathbf{w}_j^T \equiv \ell_{2j} \pmod{2}$. To distinguish from the converted hints in previous sections, we define $\mathbf{V}_{k_f} = [\mathbf{v}_1^T, \dots, \mathbf{v}_{k_f}^T] \in \mathbb{Z}^{n \times k_f}$, $\mathbf{W}_{k_g} = [\mathbf{w}_1^T, \dots, \mathbf{w}_{k_g}^T] \in \mathbb{Z}^{m \times k_g}$. Let $\ell_1 = (\ell_{11}, \dots, \ell_{1k_f}) \in \mathbb{Z}^{1 \times k_f}$ and $\ell_2 = (\ell_{21}, \dots, \ell_{2k_g}) \in \mathbb{Z}^{1 \times k_g}$. Defining $\mathbf{M}_1 = 2\mathbf{I}_{k_f}$ and $\mathbf{M}_2 = 2\mathbf{I}_{k_g}$, the unified primal basis is

$$\mathbf{B}_{\mathbf{H}_5}^{\text{NTRU}} = \begin{pmatrix} q\mathbf{I}_m & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{M}_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{H}_h & \mathbf{V}_{k_f} & \mathbf{W}_{k_g} & \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \ell_1 & \ell_2 & \mathbf{0} & 1 \end{pmatrix}. \quad (9)$$

Theorem 5. Under the above construction, the lattice $\Lambda(\mathbf{B}_{\mathbf{H}_5}^{\text{NTRU}})$ contains the short vector $\mathbf{t}_{\mathbf{H}_5} = (\mathbf{g}, \mathbf{0}^{k_f}, \mathbf{0}^{k_g}, \mathbf{f}, -1)$.

Proof. To prove that $\mathbf{t}_{\mathbf{H}_5} = (\mathbf{g}, \mathbf{0}^{k_f}, \mathbf{0}^{k_g}, \mathbf{f}, -1) \in \Lambda(\mathbf{B}_{\mathbf{H}_5}^{\text{NTRU}})$, we define the integer coefficient vector $\mathbf{x}_5 = (\mathbf{r}, \mathbf{c}_1, \mathbf{c}_2, \mathbf{f}, -1)$, where $\mathbf{r} \in \mathbb{Z}^m$, $\mathbf{c}_1 \in \mathbb{Z}^{k_f}$, and $\mathbf{c}_2 \in \mathbb{Z}^{k_g}$ are quotient vectors. Specifically, $\mathbf{r}\mathbf{q} + \mathbf{f}\mathbf{H}_h = \mathbf{g}$ follows from the NTRU relation, while $\mathbf{c}_1(2) + \mathbf{f}\mathbf{V}_{k_f} - \ell_1 = \mathbf{0}$ and $\mathbf{c}_2(2) + \mathbf{g}\mathbf{W}_{k_g} - \ell_2 = \mathbf{0}$ follow from the LSB hints. The product $\mathbf{x}_5\mathbf{B}_{\mathbf{H}_5}^{\text{NTRU}}$ is computed by block multiplication as: $\mathbf{x}_5\mathbf{B}_{\mathbf{H}_5}^{\text{NTRU}} = (\mathbf{r}\mathbf{q}\mathbf{I}_m + \mathbf{f}\mathbf{H}_h, \mathbf{c}_1\mathbf{M}_1 + \mathbf{f}\mathbf{V}_{k_f} - \ell_1, \mathbf{c}_2\mathbf{M}_2 + \mathbf{r}\mathbf{q}\mathbf{W}_{k_g} + \mathbf{f}\mathbf{H}_h\mathbf{W}_{k_g} - \ell_2, \mathbf{f}\mathbf{I}_n, -1) = (\mathbf{g}, \mathbf{0}^{k_f}, \mathbf{c}_2(2\mathbf{I}_{k_g}) + (\mathbf{r}\mathbf{q}\mathbf{I}_m + \mathbf{f}\mathbf{H}_h)\mathbf{W}_{k_g} - \ell_2, \mathbf{f}, -1) = (\mathbf{g}, \mathbf{0}^{k_f}, \mathbf{c}_2(2\mathbf{I}_{k_g}) + \mathbf{g}\mathbf{W}_{k_g} - \ell_2, \mathbf{f}, -1) = (\mathbf{g}, \mathbf{0}^{k_f}, \mathbf{0}^{k_g}, \mathbf{f}, -1) = \mathbf{t}_{\mathbf{H}_5}$. Since $\mathbf{x}_5$ consists of integer entries, $\mathbf{t}_{\mathbf{H}_5}$ is an integer linear combination of the rows of $\mathbf{B}_{\mathbf{H}_5}^{\text{NTRU}}$, completing the proof. $\square$

Type II (Joint LSBs hints). For joint leakage of the form $\mathbf{f}\mathbf{v}_i^T + \mathbf{g}\mathbf{w}_i^T \equiv \ell_i \pmod{2}$, we define the integrated matrices $\mathbf{V} = [\mathbf{v}_1^T, \dots, \mathbf{v}_k^T] \in \mathbb{Z}^{n \times k}$ and $\mathbf{W} = [\mathbf{w}_1^T, \dots, \mathbf{w}_k^T] \in \mathbb{Z}^{m \times k}$. Let $\ell = (\ell_1, \dots, \ell_k) \in \mathbb{Z}^{1 \times k}$ and $\mathbf{M} = 2\mathbf{I}_k$. To incorporate these constraints while maintaining the target vector's norm, we lift the modular relations to the integers: $\mathbf{f}\mathbf{v}_i^T + \mathbf{g}\mathbf{w}_i^T = \ell_i + 2r_i$ for some unknown quotients $r_i \in \mathbb{Z}$. Following the modular embedding approach, the unified primal basis is constructed as

$$\mathbf{B}_{\mathbf{H}_6}^{\text{NTRU}} = \begin{pmatrix} q\mathbf{I}_m & \mathbf{W} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} & \mathbf{0} & \mathbf{0} \\ \mathbf{H}_h & \mathbf{V} & \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \ell & \mathbf{0} & 1 \end{pmatrix}. \quad (10)$$

Theorem 6. Under the above construction, the lattice $\Lambda(\mathbf{B}_{\mathbf{H}_6}^{\text{NTRU}})$ contains the short target vector $\mathbf{t}_{\mathbf{H}_6} = (\mathbf{g}, \mathbf{0}^k, \mathbf{f}, -1)$.

Proof. To prove that $\mathbf{t}_{\mathbf{H}_6} = (\mathbf{g}, \mathbf{0}^k, \mathbf{f}, -1) \in \Lambda(\mathbf{B}_{\mathbf{H}_6}^{\text{NTRU}})$, we define the integer coefficient vector $\mathbf{x}_6 = (\mathbf{r}, \mathbf{c}, \mathbf{f}, -1)$, where $\mathbf{r} \in \mathbb{Z}^m$ and $\mathbf{c} \in \mathbb{Z}^k$ are quotient vectors. Specifically, $\mathbf{r}\mathbf{q} + \mathbf{f}\mathbf{H}_h = \mathbf{g}$ follows from the NTRU relation, and $\mathbf{c}\mathbf{M} + \mathbf{f}\mathbf{V} + \mathbf{g}\mathbf{W} - \ell = \mathbf{0}$ (with $\mathbf{M} = 2\mathbf{I}_k$) follows from the lifted joint LSB constraints $\mathbf{f}\mathbf{v}_i^T + \mathbf{g}\mathbf{w}_i^T = \ell_i + 2r_i$. The product $\mathbf{x}_6\mathbf{B}_{\mathbf{H}_6}^{\text{NTRU}}$ is computed via block multiplication as: $\mathbf{x}_6\mathbf{B}_{\mathbf{H}_6}^{\text{NTRU}} = (\mathbf{r}\mathbf{q}\mathbf{I}_m + \mathbf{f}\mathbf{H}_h, \mathbf{r}\mathbf{W} + \mathbf{c}\mathbf{M} + \mathbf{f}\mathbf{V} - \ell, \mathbf{f}\mathbf{I}_n, -1) = (\mathbf{g}, \mathbf{0}^k, \mathbf{f}, -1) = \mathbf{t}_{\mathbf{H}_6}$. The second block vanishes because substituting $\mathbf{g} = \mathbf{r}\mathbf{q} + \mathbf{f}\mathbf{H}_h$ into the joint LSB relation $\mathbf{g}\mathbf{W} + \mathbf{f}\mathbf{V} - \ell \in 2\mathbb{Z}^k$ ensures the term is neutralized by the quotient offset $\mathbf{c}\mathbf{M}$. Since $\mathbf{x}_6$ has integer entries, $\mathbf{t}_{\mathbf{H}_6} \in \Lambda(\mathbf{B}_{\mathbf{H}_6}^{\text{NTRU}})$. $\square$

C. LSBs Leakage Impact on Falcon Security

In this subsection we evaluate the leakage of the least significant bits (LSBs) of the secret-key coefficients in Falcon-512. We model LSB leakage on f as a mod-2 hint, i.e., a congruence constraint on each coefficient. The residual security under such hint leakage is assessed with the leaky-LWE-Estimator tool [26].

Before integrating any auxiliary hints, the original primal embedding has dimension $n = 1025$; in this setting, a successful key recovery requires BKZ with blocksize about $\beta = 480$. The exact value of β depends on the public key and on the coefficients of the secret polynomials f and g sampled from a discrete Gaussian.

TABLE V: Residual BKZ blocksize β for Falcon-512 with LSBs hint.

LSBs hints	450	460	470	480	490	500	510	512
Remaining β	436.75	435.82	434.89	433.96	433.04	432.12	431.20	431.02
$\Delta\beta$	43.25	44.18	45.11	46.04	46.96	47.88	48.80	48.98

TABLE VI: Residual BKZ blocksize β for Falcon-1024 with LSBs hint.

LSBs hints	960	970	980	990	1000	1010	1020	1024
Remaining β	894.34	893.56	892.79	892.02	891.25	890.48	889.71	889.40
$\Delta\beta$	79.66	80.44	81.21	81.98	82.75	83.52	84.29	84.60

When incorporating only LSB-based hints, the lattice dimension within the primal embedding framework remains constant, specifically 1025 for Falcon-512 and 2049 for Falcon-1024. However, the BKZ block size β necessary for a successful attack exhibits a consistent decrease. Tables V and VI summarize the residual block size β alongside its corresponding reduction under varying quantities of LSB hints for Falcon-512 and Falcon-1024. These empirical results suggest that while LSB leakage of secret coefficients in isolation may not facilitate a complete key recovery, it provides a rigorous quantitative measure of how such side-channel information degrades the residual security of the scheme. Consequently, our findings underscore the necessity for Falcon implementations to incorporate robust countermeasures against the leakage of secret key LSBs.

D. Impact of LSB Leakage on Hawk Security

The security of the Hawk signature scheme is predicated on the hardness of finding short vectors in specific lattices. However, the signing process involves a preimage computation during discrete Gaussian sampling (Algorithm 15, Step 8 in [12]), which is potentially vulnerable to side-channel attacks. Specifically, advanced side-channel techniques, such as Correlation Power Analysis (CPA) or template attacks targeting integer multiplications, can recover the LSBs of the coefficients of the secret polynomials $\mathbf{f}, \mathbf{F}, \mathbf{g}$, and $\mathbf{G}$ [35]. We model the recovery of these LSBs as the integration of independent modular hints into a primal lattice reduction attack. For a Hawk instance with parameter n , recovering the LSBs for all four secret polynomials corresponds to obtaining $k = 4n$ independent modular hints.

According to Lemma 13 in [26], integrating k modular hints modulo m increases the lattice volume (determinant) by a factor of m^k . For k hints modulo 2, the volume increases by a factor of 2^k . Furthermore, as established in [27], integrating LSB hints increases the lattice volume without altering the lattice dimension or the norm of the target short vector. Consequently, the following discussion focuses on the impact of LSB hints on the residual security strength of Hawk. In the standard specification of Hawk [12], the baseline lattice volume is analyzed by scaling the lattice by a factor of $\sqrt{2n}$ to balance the target vector norm. Given that the basis $\mathbf{B}$ is unimodular ($\det(\mathbf{Q}) = 1$), the scaled volume is defined as $\text{Vol}(\Lambda) = \sqrt{2n^{2n}} \det(\mathbf{Q}) = 2n^n$. Thus, the initial logarithmic volume is $\ln(\text{Vol}(\Lambda)) = n \ln(2n)$. Integrating k modular hints adds a term of $\ln(2^k) = k \ln(2)$ to this logarithmic baseline.

To rigorously evaluate the residual security strength of the Hawk signature scheme under the leakage of secret key LSBs, we employ the official Hawk complexity estimation script, as detailed in Appendix D. While previous studies, such as [35], focused on partial leakage during Gaussian sampling, our analysis demonstrates that the LSBs leakage of secret key coefficients results in a significantly more severe security degradation. The integration of $4n$ constraints from the leaked LSBs increases the lattice volume, thereby significantly reducing the required residual security strength β .

TABLE VII: Residual β with f_{LSBs} and $(f, F, g, G)_{\text{LSBs}}$ for Hawk.

Instance	No Leakage	f_{LSBs} hints	$(f, F, g, G)_{\text{LSBs}}$ hints
Hawk-256	211	159.568	81.249
Hawk-512	452	365.657	214.354
Hawk-1024	940	790.960	511.089

We evaluate this impact across three standard Hawk instances where $n \in \{256, 512, 1024\}$, corresponding to lattice dimensions $\dim(\Lambda) = 2n \in \{512, 1024, 2048\}$. Table VII presents the residual security strength for various Hawk dimensions after integrating the LSB hints of the secret polynomial $\mathbf{f}$, as well as the residual strength following the simultaneous integration of LSBs hints from $\mathbf{f}, \mathbf{g}, \mathbf{F}$, and $\mathbf{G}$. The results

demonstrate that the concurrent leakage of LSB hints for $\mathbf{f}, \mathbf{g}, \mathbf{F}$, and $\mathbf{G}$ leads to a severe degradation in security. Specifically, for Hawk-512, the required blocksize β decreases significantly from a secure level of 452 down to 214.354, representing a reduction in security strength of 237.646. Similarly, for Hawk-1024, the required β is reduced by from 940 to 511.089, a reduction of 428.911.

V. INTEGRATING APPROXIMATE HINTS

In the DDGR framework [26], evaluating approximate hints was limited to small-scale LWE instances ($n = m \approx 80$). As dimensions scale to $n = m > 200$, the computational overhead of hint integration and primal lattice reduction becomes prohibitive. Furthermore, DDGR’s analysis of NTRU-based schemes was restricted to ternary secrets, neglecting cases where secret coefficients follow a discrete Gaussian distribution over a broader range, such as $[-31, 31]$ for Falcon-512 and $[-15, 15]$ for Falcon-1024.

Existing alternatives also present limitations: while MN [27] supports approximate hints, the feature remains undocumented due to its perceived lower efficiency compared to DDGR. Meanwhile, CJC [29] relies on the availability of perfect hints to integrate approximate ones, a condition often unmet in realistic attack scenarios. In practical side-channel or fault-injection attacks, adversaries typically obtain noisy leakage modeled as approximate hints. For instance, Simple Power Analysis (SPA) on Falcon-512 can approximate secret polynomials $\mathbf{f}$ and $\mathbf{g}$ during the signing phase [19]. Thus, quantifying the impact of approximate hints with varying noise levels on the residual security of NTRU-based signatures is of critical importance.

In this section, we utilize the leaky-LWE-Estimator from the DDGR framework to systematically evaluate how the intensity (σ) and quantity of approximate hints affect the required BKZ blocksize (β) for key-recovery attacks on Falcon-512 and Falcon-1024. We model these hints as: $\langle \mathbf{f}, \mathbf{v}_i \rangle = \ell_i + \varepsilon_{\sigma_i}$, where $\mathbf{v}_i \in \mathbb{Z}_q^{1 \times n}$, $\ell_i \in \mathbb{Z}$, and ε_{σ_i} denotes Gaussian noise with mean 0 and standard deviation σ_i .

In our experiments, we focus on the scenario where approximate hints of the secret key coefficients are leaked, and we study how different levels of noise in these leaked coefficients affect the security of the scheme. Our experiments are based on the DDGR framework using the DBDD_predict [26] class for performance evaluation, with detailed experimental settings described in Section VI-A. When only approximate hints on $\mathbf{f}$ and $\mathbf{g}$ are available, the required BKZ blocksize β decreases monotonically as the number of hints increases; for a fixed number of hints, β decreases as the standard deviation σ of the injected noise becomes smaller. This trend indicates that larger noise weakens the effectiveness of the approximate hints, thus requiring stronger lattice reduction (i.e., larger β) for successful key recovery. As shown in Fig. 1, for Falcon-512, the residual security under different noise levels is summarized as follows. For hints = 160, when $\sigma = 1, 5, 10$, the corresponding BKZ blocksize is $\beta = 392.94, 424.29, 438.72$, respectively. For hints = 200, when $\sigma = 1, 5, 10$, the corresponding BKZ

blocksize is $\beta = 374.79, 411.80, 429.16$, respectively. Overall, as the number of hints increases from 160 to 200, the BKZ blocksize β consistently decreases, demonstrating that additional leakage information facilitates more efficient key recovery.

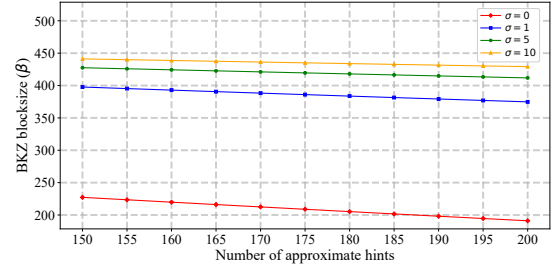


Fig. 1: BKZ blocksize β for Falcon-512 with approximate hints of $\mathbf{f}$.

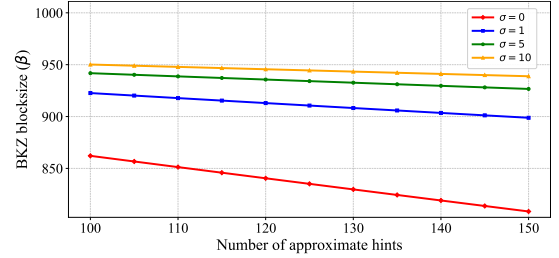


Fig. 2: BKZ blocksize β for Falcon-1024 with approximate hints of $\mathbf{f}$.

Due to the high dimensionality of Falcon-1024, using the leaky-LWE-Estimator [26] to integrate approximate hints incurs substantial computational overhead. In our experiments, integrating 150 noisy approximate hints required approximately 50 hours of computation. To maintain practical feasibility, we limited our evaluation to cases involving 100–150 approximate hints. The noise term ε follows a zero-mean Gaussian distribution with standard deviation $\sigma \in \{1, 5, 10\}$, and results were compared with the noise-free case. As shown in Fig. 2, smaller noise levels lead to greater reductions in residual security after hint embedding. Specifically, when $\sigma = 1$, embedding 150 approximate hints reduces the required BKZ blocksize for Falcon-1024 from 974 to 898.81, a reduction of 75.19; in contrast, when $\sigma = 10$, the blocksize β decreases only slightly from 974 to 938.85, a reduction of 35.15.

VI. EXPERIMENTAL ANALYSIS

A. Experimental setup

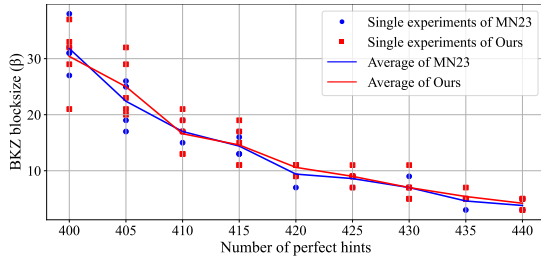
All experiments were conducted on a server running Ubuntu 20.04.6 LTS with a 5.15.0-130-generic Linux kernel (x86_64 architecture). The server is equipped with two Intel(R) Xeon(R) Gold 5220R CPUs@2.20GHz, providing a total of 48 physical cores and 96 logical threads. The system is configured with 503 GiB of main memory and a 122 GiB swap space. For GPU acceleration, the server hosts three NVIDIA RTX A4000 graphics cards, each equipped with 16 GiB of dedicated memory. The computational experiments

were primarily executed using SageMath version 10.4, which was compiled from source to ensure compatibility with the server environment.

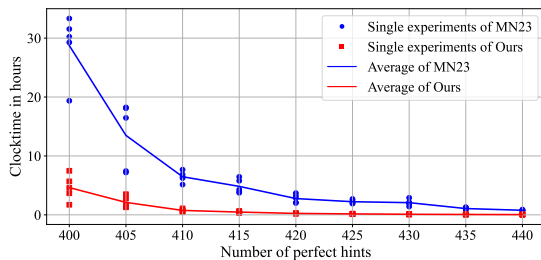
In our experiments, we first consider the case of perfect hints, where certain coefficients of the secret polynomial f are assumed to be leaked. Such hints can be represented as inner products of the form $\langle \mathbf{v}_i, \mathbf{f} \rangle = \ell_i$, where $\mathbf{v}_i$ may be viewed as a standard basis vector. Leakage of coefficients of the secret polynomial g is modeled analogously. For example, the i -th standard basis vector is given by $\mathbf{e}_{j_i} = (0, \dots, 1, \dots, 0)$. We further evaluate the impact of joint leakage where partial coefficients of both f and g are revealed simultaneously, and analyze its effect on the residual security level and the corresponding time complexity. For modular hints, we again focus on the leakage of partial coefficients. In particular, we consider the Mod- q hint commonly observed in practical implementations, which reveals a larger amount of information and can be expressed as $\langle \mathbf{v}_i, \mathbf{f} \rangle = \ell_i \bmod q$. For each type of hint, we perform experiments on five different NTRU instances for Falcon-512 and Falcon-1024. The residual security of the primal lattice reduction attack is evaluated by the minimal BKZ blocksize β required for successful key recovery, while the time complexity is measured by the actual clock time needed to complete the attack.

B. Experimental Results for secret-only f perfect hint

In the scenario of partial coefficient leakage of the secret key f (i.e., perfect hints), we evaluate five NTRU instances generated from Falcon-512 and Falcon-1024. The discrete points in each figure represent the outcome of individual attacks, while the solid lines depict the averaged performance across the five instances.



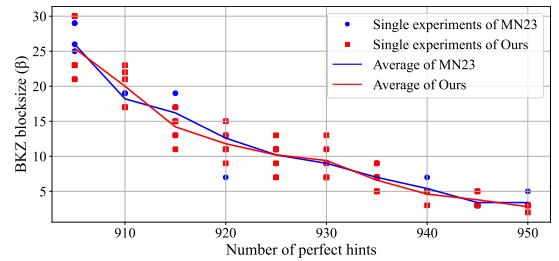
(a) Effect of PSCL of f on BKZ blocksize in Falcon-512.



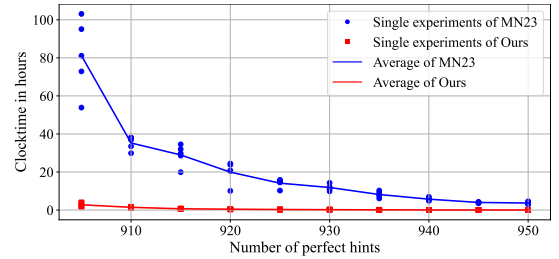
(b) Effect of PSCL of f on attack clocktime in Falcon-512.

Fig. 3: BKZ blocksize β and clocktime for Falcon-512 with PSCL of f .

For Falcon-512, the two subfigures in Fig. 3 illustrate the BKZ blocksize and clocktime required to recover the full secret key when the leaked coefficients of f range from indices 400 to 440. As shown in Fig. 3a, the required BKZ blocksize β decreases as more coefficients become known. The security degradation trend and numerical values of our method closely match those of the MN approach, confirming the correctness of our DSRNA attack framework. Fig. 3b compares the attack clocktime of the two methods. When 400 and 405 perfect hints are available, our method requires only 6 and 2 hours on average, respectively, whereas the MN method takes approximately 35 and 15 hours. This corresponds to speedups of roughly $5.8\times$ and $7.5\times$. Overall, our approach achieves comparable success probability while significantly reducing computation cost, demonstrating superior efficiency and practicality.



(a) Effect of PSCL of f on BKZ blocksize in Falcon-1024.



(b) Effect of PSCL of f on attack clocktime in Falcon-1024.

Fig. 4: BKZ blocksize β and clocktime for Falcon-1024 with PSCL of f .

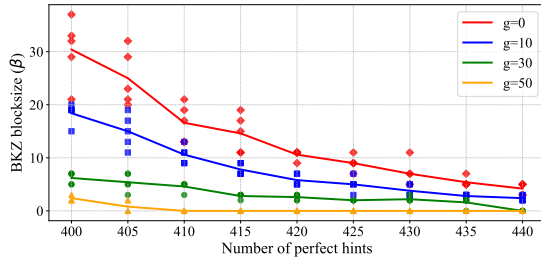
For Falcon-1024, the two subfigures in Fig. 4 illustrate the BKZ blocksize and clocktime required to recover the full secret key when the leaked coefficients of f range from indices 905 to 950. As shown in Fig. 4a, the remaining security levels produced by our method closely match those of the MN approach, confirming the correctness of our DSRNA attack framework. Fig. 4b compares the attack clocktime of the two methods. When 905 perfect hints are available, our method requires only 2.8 hours on average, whereas the MN method takes approximately 81.1 hours. This corresponds to a speedup of roughly $29.4\times$. When 910 perfect hints are leaked, our method needs about 1.5 hours, whereas MN requires around 35.2 hours, yielding a speedup of approximately $24.3\times$. When the number of leaked coefficients reaches 950, our method reduces the average attack time to below 0.03 hours, while MN still takes about 3.68 hours, resulting in a speedup exceeding $122\times$. Overall, these results demonstrate that our approach achieves comparable success probability while significantly

reducing computation cost, highlighting its superior efficiency and scalability for high-dimensional NTRU parameter sets.

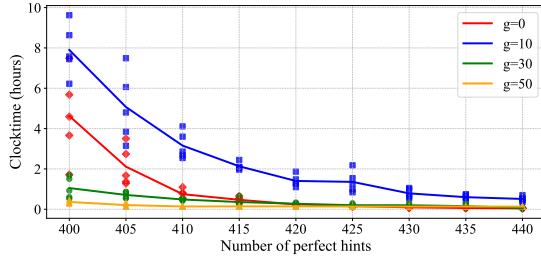
C. Experimental Results for secret f and g perfect hint

Fig. 5 and Fig. 6 present, respectively for Falcon-512 and Falcon-1024, the impact of PSCL of both f and g under varying amounts of leaked coefficients of f . For each leakage configuration, we additionally incorporate different numbers of independently leaked coefficients of $g \in \{0, 10, 30, 50\}$, run five independent trials, and report the averaged results.

For Falcon-512, Fig. 5a and Fig. 5b illustrate how joint leakage of f and g affects both the required BKZ blocksize and the attack clocktime. The horizontal axis denotes the number of leaked coefficients of f , while the four curves correspond to different leakage patterns on g : the case $g = 0$ leaks only coefficients of f , whereas $g = 10$, $g = 30$, and $g = 50$ additionally leak 10, 30, and 50 independent coefficients of g , respectively.



(a) Effect of PSCL of f and g on BKZ blocksize in Falcon-512.

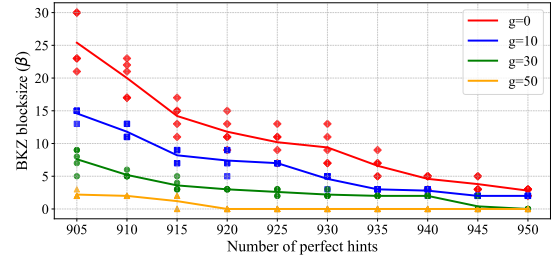


(b) Effect of PSCL of f and g on clocktime in Falcon-512.

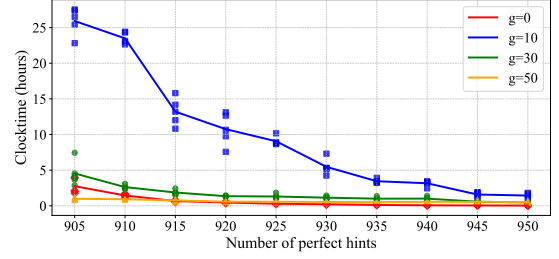
Fig. 5: BKZ blocksize β and clocktime for Falcon-512 with PSCL of f and g .

When 400 coefficients of f are known, leaking only f in the $g = 0$ case requires an average blocksize of approximately $\beta = 30.4$ and an average attack time of about 4.6 hours. Under the same conditions, leaking 50 additional coefficients of g in the $g = 50$ case reduces the average blocksize to $\beta = 2.4$ and the average clocktime to roughly 0.36 hours.

Even when the leakage on f increases to 440 coefficients, additional leakage from g continues to provide clear benefits: the average blocksize drops from $\beta = 4.2$ for the $g = 0$ scenario to $\beta = 0.8$ for the $g = 50$ scenario. Overall, Fig. 5 quantitatively demonstrates that exploiting joint leakage of f and g can substantially reduce the BKZ hardness and the practical running time for Falcon-512, thereby significantly accelerating full secret key recovery.



(a) Effect of PSCL of f and g on BKZ blocksize in Falcon-1024.



(b) Effect of PSCL of f and g on clocktime in Falcon-1024.

Fig. 6: BKZ blocksize β and clocktime for Falcon-1024 with PSCL of f , g .

For Falcon-1024, Fig. 6a and Fig. 6b illustrate the PSCL evaluation under higher-dimensional parameters. The horizontal axis denotes the number of leaked coefficients of f , with curves representing 0, 10, 30, or 50 additional leaked coefficients of g . When 905 coefficients of f are leaked, the baseline $g = 0$ case requires approximately $\beta = 25.4$ and 2.76 hours on average. In contrast, with 50 leaked coefficients of g in the $g = 50$ case, the required blocksize collapses to $\beta = 2.2$ and the attack time drops to approximately 0.98 hours. As the leakage on f reaches 950 coefficients, all curves converge to $\beta \leq 3$ with execution times well under one hour. The joint-leakage curves for $g = 30$ and $g = 50$ consistently stay below the baseline, underscoring the significant efficiency gains from simultaneous leakage of f and g in Falcon-1024.

VII. CONCLUSION

Our research focuses on the impact of partial secret key information leakage on the security of NTRU lattice-based signature schemes. First, we analyze the effects of different types of leakage (including perfect, modular, and approximate hints) on theoretical security. Subsequently, we propose an efficient attack method, namely the DSRNA approach, which enables full secret key recovery by exploiting partial secret key coefficient leakage. We investigate the joint leakage of various hint types. Our experiments primarily target Falcon and its variants Mitaka and Hawk, providing a systematic assessment of vulnerability and recovery efficiency under various leakage models. Inspired by the unified framework of Hermelink et al. [36] for LWE, our future work will adapt their distributional-hint methodology to NTRU-based schemes, utilizing greedy algorithms or belief propagation to enhance key recovery efficiency.

ACKNOWLEDGEMENTS

This work is supported in part by the National Natural Science Foundation of China (No. U2336205, No. 62302226, No. 62302224, No. 62502207) and Yunnan Provincial New R&D Institution Cultivation Project (No. 202404BQ040148).

REFERENCES

- [1] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 1994, pp. 124–134.
- [2] J. W. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, "CRYSTALS - kyber: A cca-secure module-lattice-based KEM," in *IEEE European Symposium on Security and Privacy, EuroS&P*. IEEE, 2018, pp. 353–367.
- [3] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé, "Crystals-dilithium: A lattice-based digital signature scheme," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2018, no. 1, pp. 238–268, 2018.
- [4] P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, R. Ricosset, G. Seiler, W. Whyte, Z. Zhang *et al.*, "Falcon: Fast-fourier lattice-based compact signatures over ntru," *Submission to the NIST's post-quantum cryptography standardization process*, vol. 36, no. 5, pp. 1–75, 2018, <https://falcon-sign.info/falcon.pdf>.
- [5] J. Hoffstein, J. Pipher, and J. H. Silverman, "NTRU: A ring-based public key cryptosystem," in *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings*, ser. Lecture Notes in Computer Science, vol. 1423. Springer, 1998, pp. 267–288.
- [6] National Institute of Standards and Technology (NIST). (2024) Migration to post-quantum cryptography. National Cybersecurity Center of Excellence. <https://www.nccoe.nist.gov/crypto-agility-considerations-migrating-post-quantum-cryptographic-algorithms>.
- [7] Algorand, "The algorand blockchain is on a path to post-quantum readiness," 2023, <https://algorand.com/technology/post-quantum>.
- [8] Google Cloud, "Why google now uses post-quantum cryptography for internal comms," 2022, <https://cloud.google.com/blog/products/identity-security>.
- [9] A. Hülsing, J. Rijneveld, J. M. Schanck, and P. Schwabe, "High-speed key encapsulation from NTRU," in *Cryptographic Hardware and Embedded Systems - CHES*, ser. Lecture Notes in Computer Science, vol. 10529. Springer, 2017, pp. 232–252.
- [10] T. Espitau, P. Fouque, F. Gérard, M. Rossi, A. Takahashi, M. Tibouchi, A. Wallet, and Y. Yu, "Mitaka: A simpler, parallelizable, maskable variant of falcon," in *EUROCRYPT 2022*, ser. Lecture Notes in Computer Science, vol. 13277. Springer, 2022, pp. 222–253.
- [11] L. Ducas, E. W. Postlethwaite, L. N. Pulles, and W. P. J. van Woerden, "Hawk: Module LIP makes lattice signatures fast, compact and simple," in *ASIACRYPT*, ser. Lecture Notes in Computer Science, vol. 13794. Springer, 2022, pp. 65–94.
- [12] J. W. Bos, O. Bronchain, L. Ducas, S. Fehr, Y.-H. Huang, T. Pornin, E. W. Postlethwaite, T. Prest, L. N. Pulles, and W. van Woerden, "HAWK," National Institute of Standards and Technology (NIST), Technical report, 2023, <https://hawk-sign.info/>.
- [13] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*. ACM, 2008, pp. 197–206.
- [14] N. Howgrave-Graham, "A hybrid lattice-reduction and meet-in-the-middle attack against NTRU," in *CRYPTO 2007*, ser. Lecture Notes in Computer Science, vol. 4622. Springer, 2007, pp. 150–169.
- [15] M. R. Albrecht, S. Bai, and L. Ducas, "A subfield lattice attack on overstretched NTRU assumptions - cryptanalysis of some FHE and graded encoding schemes," in *CRYPTO 2016*, ser. Lecture Notes in Computer Science, vol. 9814. Springer, 2016, pp. 153–178.
- [16] P. Kirchner and P. Fouque, "Revisiting lattice attacks on overstretched NTRU parameters," in *EUROCRYPT 2017*, ser. Lecture Notes in Computer Science, vol. 10210, 2017, pp. 3–26.
- [17] H. Bambury and P. Q. Nguyen, "Improved provable reduction of NTRU and hypercubic lattices," in *Post-Quantum Cryptography, PQCrypto 2024*, ser. Lecture Notes in Computer Science, vol. 14771. Springer, 2024, pp. 343–370.
- [18] E. Karabulut and A. Aysu, "FALCON down: Breaking FALCON post-quantum signature scheme through side-channel attacks," in *58th ACM/IEEE Design Automation Conference, DAC*. IEEE, 2021, pp. 691–696.
- [19] M. Guereau, A. Martinelli, T. Ricosset, and M. Rossi, "The hidden parallelepiped is back again: Power analysis attacks on falcon," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2022, no. 3, pp. 141–164, 2022.
- [20] S. Zhang, X. Lin, Y. Yu, and W. Wang, "Improved power analysis attacks on falcon," in *EUROCRYPT 2023*, ser. Lecture Notes in Computer Science, vol. 14007. Springer, 2023, pp. 565–595.
- [21] S. Chen, H. Shao, J. Ming, Y. Liu, Y. Gao, and Y. Zhou, "Serp-sca: A strengthened side-channel attack framework for falcon signature," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1–1, 2025.
- [22] C. H. Kim and J. Quisquater, "Faults, injection methods, and fault attacks," *IEEE Des. Test Comput.*, vol. 24, no. 6, pp. 544–545, 2007.
- [23] P. Ravi, A. Chattopadhyay, J. D'Anvers, and A. Baksı, "Side-channel and fault-injection attacks over lattice-based post-quantum schemes (kyber, dilithium): Survey and new results," *ACM Trans. Embed. Comput. Syst.*, vol. 23, no. 2, pp. 35:1–35:54, 2024.
- [24] M. R. Albrecht, A. Deo, and K. G. Paterson, "Cold boot attacks on ring and module LWE keys under the NTT," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2018, no. 3, pp. 173–213, 2018.
- [25] N. Howgrave-Graham, P. Q. Nguyen, D. Pointcheval, J. Proos, J. H. Silverman, A. Singer, and W. Whyte, "The impact of decryption failures on the security of NTRU encryption," in *CRYPTO 2003*, ser. Lecture Notes in Computer Science, vol. 2729. Springer, 2003, pp. 226–246.
- [26] D. Dachman-Soled, L. Ducas, H. Gong, and M. Rossi, "LWE with side information: Attacks and concrete security estimation," in *CRYPTO 2020*, ser. Lecture Notes in Computer Science, vol. 12171. Springer, 2020, pp. 329–358.
- [27] A. May and J. Nowakowski, "Too many hints - when LLL breaks LWE," in *ASIACRYPT 2023*, ser. Lecture Notes in Computer Science, vol. 14441. Springer, 2023, pp. 106–137.
- [28] E. Kirshanova, A. May, and J. Nowakowski, "New NTRU records with improved lattice bases," in *Post-Quantum Cryptography, PQCrypto2023*, ser. Lecture Notes in Computer Science, vol. 14154. Springer, 2023, pp. 167–195.
- [29] J. Cao, H. Jiang, and Q. Cheng, "Refined attack on LWE with hints: Constructing lattice via gaussian elimination," in *CRYPTO 2025*, ser. Lecture Notes in Computer Science, vol. 16000. Springer, 2025, pp. 385–416.
- [30] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, "Post-quantum key exchange - A new hope," in *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016*, T. Holz and S. Savage, Eds. USENIX Association, 2016, pp. 327–343, <https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim>.
- [31] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*. ACM, 2005, pp. 84–93.
- [32] C. Schnorr, "A hierarchy of polynomial time lattice basis reduction algorithms," *Theor. Comput. Sci.*, vol. 53, pp. 201–224, 1987.
- [33] Y. Aono, Y. Wang, T. Hayashi, and T. Takagi, "Improved progressive BKZ algorithms and their precise cost estimation by sharp simulator," in *EUROCRYPT 2016*, ser. Lecture Notes in Computer Science, vol. 9665. Springer, 2016, pp. 789–819.
- [34] Y. Chen and P. Q. Nguyen, "BKZ 2.0: Better lattice security estimates," in *ASIACRYPT*, ser. Lecture Notes in Computer Science, D. H. Lee and X. Wang, Eds., vol. 7073. Springer, 2011, pp. 1–20.
- [35] M. Guereau and M. Rossi, "A not so discrete sampler: Power analysis attacks on HAWK signature scheme," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2024, no. 4, pp. 156–178, 2024.
- [36] J. Hermelink, S. Streit, E. Mårtensson, and R. Petri, "A generic framework for side-channel attacks against lwe-based cryptosystems," in *EUROCRYPT 2025*, ser. Lecture Notes in Computer Science, vol. 15608. Springer, 2025, pp. 3–32.

APPENDIX A

PROOF OF CORRECTNESS FOR ALGORITHM 1

To establish the correctness of the DSRNA (1), we prove the existence of the target vector $\mathbf{t} = (\mathbf{g}', \mathbf{f}', -1)$ in the lattice

$\Lambda(\mathbf{B}_{\text{red}}^{\text{NTRU}})$ and verify its recovery criteria.

A. Existence of the Target Vector $\mathbf{t}$ in $\Lambda(\mathbf{B}_{\text{red}}^{\text{NTRU}})$

Proof. Let the original NTRU relation be defined by $\mathbf{f}\mathbf{H}_h \equiv \mathbf{g} \pmod{q}$. Following the partition $\mathbf{f} = (\mathbf{f}^{[1:k]}, \mathbf{f}^{[k+1:n]})$, we expand the relation as: $\mathbf{f}^{[1:k]}\mathbf{H}_h^{[1:k,:]} + \mathbf{f}^{[k+1:n]}\mathbf{H}_h^{[k+1:n,:]} \equiv \mathbf{g} \pmod{q}$. In Step 2, the residual $\mathbf{b}'$ is computed as $\mathbf{b}' \equiv -\mathbf{f}^{[1:k]}\mathbf{H}_h^{[1:k,:]} \pmod{q}$ (assuming initial $\mathbf{b} = \mathbf{0}$). Substituting this into the above relation, we obtain $\mathbf{f}^{[k+1:n]}\mathbf{H}_h \equiv \mathbf{g} + \mathbf{b}' \pmod{q}$. Restricting this to the sample subset $\mathcal{I}$ (Step 4), let $\mathbf{f}' = \mathbf{f}^{[k+1:n]}$ and $\mathbf{g}' = \mathbf{g}^{[\mathcal{I}]}$. The relation becomes $\mathbf{f}'\mathbf{H}_h'' - \mathbf{g}' \equiv \mathbf{b}'' \pmod{q}$.

This implies there exists an integer quotient vector $\mathbf{r}_1 \in \mathbb{Z}^{m'}$ such that $-\mathbf{g}' = \mathbf{b}'' + q\mathbf{r}_1 - \mathbf{f}'\mathbf{H}_h''$. The lattice membership of $\mathbf{t} = (\mathbf{g}', \mathbf{f}', -1)$ is verified by the integer linear combination $\mathbf{x} = (-\mathbf{r}_1, \mathbf{f}', -1)$:

$$\begin{aligned} \mathbf{x}\mathbf{B}_{\text{red}}^{\text{NTRU}} &= (-\mathbf{r}_1, \mathbf{f}', -1) \begin{pmatrix} q\mathbf{I}_{m'} & 0 & 0 \\ \mathbf{H}_h'' & \mathbf{I}_{n'} & 0 \\ \mathbf{b}'' & 0 & 1 \end{pmatrix} \\ &= (-q\mathbf{r}_1 + \mathbf{f}'\mathbf{H}_h'' - \mathbf{b}'', \mathbf{f}', -1) = \mathbf{t}. \end{aligned} \quad (11)$$

This confirms that the target vector resides in the constructed lattice $\Lambda(\mathbf{B}_{\text{red}}^{\text{NTRU}})$. $\square$

B. Success Conditions under the Gaussian Heuristic

Proof. The recovery of the secret key is predicated on the shortness of the NTRU secret vector relative to the lattice volume. According to Eq. (1), the BKZ algorithm with block size β is expected to identify the target vector $\mathbf{t}$ if the following condition holds for a total lattice dimension d' : $\sqrt{\frac{\beta}{d'}} \cdot \|\mathbf{t}\| \leq \delta_{\beta}^{2\beta-d'-1} \cdot \text{vol}(\Lambda)^{1/d'}$.

In this framework, the optimal number of samples m' is implicitly derived from the relation $m' = d' - n' - 1$, where n' is the fixed count of unknown secret coefficients. Since the target norm $\|\mathbf{t}\|$ is designed to be significantly smaller than the Gaussian Heuristic prediction for random vectors in a reduced lattice of volume $q^{m'}$, $\mathbf{t}$ forms a uSVP instance, ensuring its successful recovery through basis reduction. $\square$

APPENDIX B

OPTIMALITY OF SAMPLE COUNT m'

To justify the selection of m' , we transform the empirical dimension selection into an auditable optimization problem based on the interaction between lattice hardness and sample availability.

A. Lattice Properties and Target Vector Estimation

Definition 5 (Reduced Lattice Geometry). *The reduced lattice $\Lambda(\mathbf{B}_{\text{red}}^{\text{NTRU}})$ constructed in Algorithm 1 possesses a volume $\text{Vol}(\Lambda(\mathbf{B}_{\text{red}}^{\text{NTRU}})) = q^{m'}$. The total dimension of the embedding is denoted as d' , which serves as the primary parameter for the BKZ cost model.*

Lemma 1 (Target Vector Norm Estimation). *The target vector is $\mathbf{t} = (\mathbf{g}', \mathbf{f}', -1) \in \mathbb{Z}^{d'}$. Let $n' = n - k$ be the fixed number of remaining unknown secret coefficients. For a given total*

dimension d' , the number of samples utilized is determined by $m' = d' - n' - 1$. Assuming coefficients of f and g are i.i.d. variables with variance σ^2 , the expected norm of $\mathbf{t}$ is: $\|\mathbf{t}(d')\| \approx \sqrt{(d' - 1)\sigma^2 + 1}$.

B. Optimality Proof via Cost Minimization

Theorem 7 (Heuristic Optimality of m'). *The optimal sample count m' is implicitly determined by the optimal lattice dimension d' that minimizes the BKZ block size β required to satisfy the uSVP success condition.*

Proof. The recovery of the target vector $\mathbf{t}$ is predicated on the uSVP success condition. By substituting the reduced lattice volume $\text{Vol}(\Lambda(\mathbf{B}_{\text{red}}^{\text{NTRU}}))$ and the target norm $\|\mathbf{t}(d')\|$ into the success criteria, we define the success discriminant as a function of the dimension d' : $\sqrt{\frac{\beta}{d'}} \cdot \|\mathbf{t}(d')\| \leq \delta_{\beta}^{2\beta-d'-1} \cdot q^{\frac{d'-n'-1}{d'}}$, where $m' = d' - n' - 1$ represents the necessary number of samples.

The above analysis reveals that for a fixed number of unknown coefficients n' , the required block size β behaves as a convex-like function of the total dimension d' . In practice, the attacker identifies the optimal reduction dimension d' that yields the global minimum of the required block size $\beta_{\text{est}}(d')$, thereby minimizing the total computational complexity $2^{\mathcal{O}(\beta)}$.

Once this complexity-minimizing dimension d' is determined, the heuristic-optimal sample count m' is explicitly fixed by the structural requirement $m' = d' - n' - 1$. This demonstrates that m' is not an arbitrary parameter, but the critical number of constraints required to bridge the gap between the fixed secret entropy n' and the optimal reduction dimension d' . $\square$

APPENDIX C

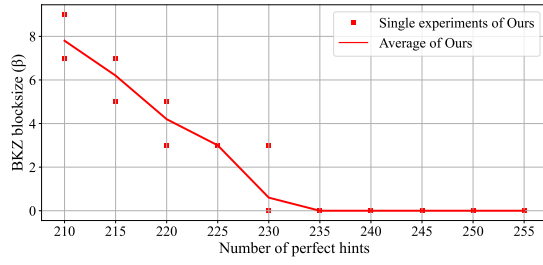
EXPERIMENTAL RESULTS FOR MITAKA WITH PERFECT AND MOD- q HINTS

In this section, we focus on investigating the impact of perfect and mod- q hints on the security strength of Mitaka, as well as the computational overhead required for the attacks.

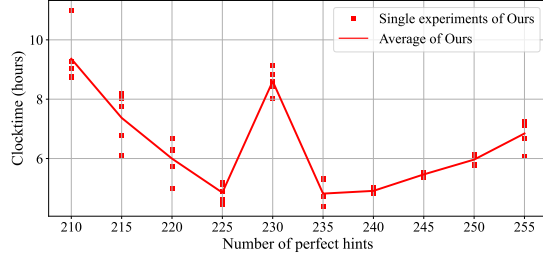
A. Experimental Results for Mitaka with Perfect Hints on f

For Mitaka-512, we evaluate the impact of partial leakage of polynomial f using the perfect-hint embedding technique. As illustrated in Fig. 7a, the required BKZ blocksize β decreases rapidly as the number of perfect hints increases from 210 to 255. Specifically, the average blocksize drops from $\beta = 7.8$ at 210 hints to $\beta = 3.0$ at 225 hints, and further vanishes to $\beta = 0$ once 235 or more hints are available.

The corresponding attack clocktime is reported in Fig. 7b. While the average runtime starts at approximately 9.36 hours with 210 hints and generally trends downward, it exhibits minor fluctuations (e.g., 4.85 hours at 225 hints versus 6.84 hours at 255 hints) due to the varying complexity of basis reduction across specific instances. Overall, the results in Fig. 7 demonstrate that the security of Mitaka-512 degrades significantly under the perfect-hint model.



(a) Effect of perfect hints of f on BKZ blocksize in Mitaka-512.



(b) Effect of perfect hints of f on attack clocktime in Mitaka-512.

Fig. 7: BKZ blocksize β and clocktime for Mitaka-512 with PSCL of f .

B. Experimental Results for Mitaka with Mod- q Hints on f

For Mitaka-512, Fig. 8 illustrates how the number of mod- q hints on the secret polynomial f affects the remaining security level. The horizontal axis shows the number of available mod- q hints, ranging from 420 to 455. Each data point represents five independent NTRU instances, and the plotted curves correspond to their averaged results.

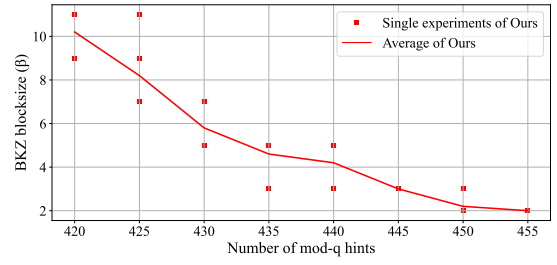
As shown in Fig. 8a, the required BKZ blocksize decreases monotonically as more mod- q hints become available. When 420 hints are leaked, the average blocksize is approximately $\beta = 10.2$; with 430 hints, it drops to about $\beta = 5.8$, and when the leakage reaches 455 hints, it further decreases to roughly $\beta = 2.0$. This trend clearly demonstrates that additional mod- q information on f significantly reduces the residual hardness of lattice reduction. Fig. 8b reports the corresponding attack clocktime. With 420 mod- q hints, the average running time is about 3.80 hours; at 430 hints, the runtime decreases to approximately 2.01 hours; and with 455 hints, it is further reduced to around 0.48 hours. Overall, Fig. 8 shows that under the mod- q leakage model of [27], the security of Mitaka-512 degrades significantly as the number of mod- q hints on f increases.

APPENDIX D

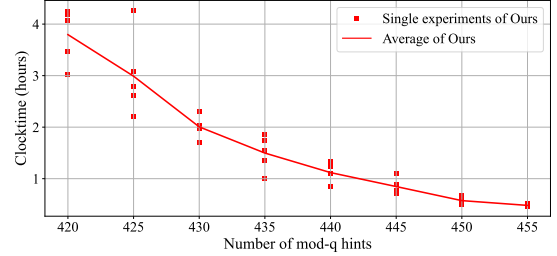
ESTIMATION OF THE RESIDUAL COMPLEXITY OF HAWK WITH LSBs HINTS

To rigorously evaluate the residual security strength of the Hawk signature scheme under the leakage of secret key LSBs, we employ the official Hawk complexity estimation script. To quantify the residual security strength, we utilize the official Hawk complexity estimation tool². Specifically, the estimation

²<https://github.com/hawk-sign/aux/blob/main/code>



(a) Effect of mod- q hints of f on BKZ blocksize in Mitaka-512.



(b) Effect of mod- q hints of f on attack clocktime in Mitaka-512.

Fig. 8: BKZ blocksize β and clocktime for Mitaka-512 with mod- q hints of f .

of the residual complexity with f_{LSBs} hints is presented in Listing 1, while the estimation of the residual complexity with $(f, F, g, G)_{\text{LSBs}}$ hints is detailed in Listing 2.

Listing 1: Estimation of the residual complexity with f_{LSBs} hints.

```
from sage.all import log
for n in {512, 1024, 2048}:
    current_logvol = n * log(n) / 2 + n * log(2)
    beta, _ = predict_beta_and_prev_sd(
        n,
        current_logvol,
        lift_union_bound=True,
        number_targets=n,
        tours=1
    )
    print(f"n={n}, beta={beta}")
```

Listing 2: Estimation of the residual complexity with $(f, F, g, G)_{\text{LSBs}}$ hints.

```
from sage.all import log
for n in {512, 1024, 2048}:
    current_logvol = n * log(n) / 2 + 4 * n * log(2)
    beta, _ = predict_beta_and_prev_sd(
        n,
        current_logvol,
        lift_union_bound=True,
        number_targets=n,
        tours=1
    )
    print(f"n={n}, beta={beta}")
```