Security Evaluation on KpqC Round 1 Lattice-based Algorithms Using Lattice Estimator

Suhri Kim[1], Eunmin Lee[1], Joohee Lee[1]*, Minju Lee[1], and Hyuna Noh[1]

Sungshin Women's University, Seoul, Republic of Korea
{suhrikim, 20211089, jooheelee, 20211082, 20211056}@sungshin.ac.kr

Abstract. Post-quantum cryptography is expected to become one of the fundamental technologies in the field of security that requires public-key cryptosystems, potentially replacing standards such as RSA and ECC, as it is designed to withstand attacks using quantum computers. In South Korea, there is an ongoing standardization effort called the KpqC (Korean Post-Quantum Cryptography) competition for developing post-quantum cryptography as a national standard. The competition is in its first round, and it has introduced a total of 16 candidate algorithms for evaluation.

In this paper, we analyze the security of five algorithms among the eight lattice-based schemes in the first round of the KpqC competition. We assess their security using M. Albrecht's Lattice Estimator, focusing on problems related to LWE (Learning with Errors) and LWR (Learning with Rounding). Additionally, we compare the security analysis results with the claims in the proposal documents for each algorithm. When an algorithm fails to achieve the level of security in its proposal, we suggest potential types of attacks that need to be considered for further analysis and improvement.

Keywords: Post-Quantum Cryptography, KpqC Competition, LWE, LWR

1 Introduction

In 1994, Peter Shor proposed polynomial-time quantum algorithms for solving discrete logarithm and factoring problems, posing a significant threat to the security of standard public-key cryptosystems such as RSA and ECC [32]. Against this backdrop, there have been active international standardization efforts for Post-Quantum Cryptography (PQC), which aims to provide new standards that are resistant to attacks using quantum computers. Since the end of 2016, the National Institute of Standards and Technology (NIST) in the United States has been conducting a standardization project for PQC in the areas of Key Encapsulation Mechanism (KEM) and digital signature. Over three rounds of evaluations,

* corresponding author

NIST selected one KEM and three signature schemes as standards in 2022 [28]. Currently, there is an ongoing process for additional selections and evaluations in the fourth round and an on-ramp for the digital signature category [29]. Similarly, in South Korea, a national standardization competition for post-quantum cryptography, known as the KpqC competition, began in 2022 [10].

Lattice-based cryptography is a field of post-quantum cryptography that relies on the hard problems related to lattices, including NTRU [19], Learning with Errors (LWE) [31, 9], Learning with Rounding (LWR) [30], and Short Integer Solution (SIS) [1]. It has gained significant attention and recognition due to its fast computational speed and balanced performance in terms of communication overhead compared to other post-quantum cryptosystems: in the U.S. NIST standardization competition, the one selected KEM standard and two digital signature schemes out of a total of selected three post-quantum signatures are lattice-based schemes.

In the context of the KpqC competition, the lattice-based submissions in the first round include three and five schemes in the Key Encapsulation Mechanism (KEM) and digital signature categories, respectively. Each of these schemes is built upon specific underlying problems, which are summarized in Table 1.

Table 1: KpqC Competition - Round 1 Lattice-based Submissions

Category	Algorithm	Base Problem
KEM	NTRU+	NTRU, **RLWE**
	SMAUG	**MLWE**, **MLWR**
	TiGER	**RLWR**, **RLWE**
Signature	GCKSign	GCK
	HAETAE	**MLWE**, MSIS
	NCC-Sign	**RLWE**, RSIS
	Peregrine	NTRU, RSIS
	SOLMAE	NTRU, RSIS

In this paper, we analyze the security of Learning with Errors (LWE) and Learning with Rounding (LWR) based algorithms, a total of 5 schemes (NTRU+, SMAUG, TiGER, HAETAE, NCC-Sign), among the lattice-based algorithms in the 1st round of the KpqC competition. We analyze the security of the LWE/LWR problem instances used in each algorithm. For security analysis of the LWE/LWR problems, we utilize M. Albrecht's Lattice Estimator [3]. The Lattice Estimator is an open-source tool written in Sage that quantifies specific attack complexities for various types of LWE attacks, including those described in [3]. It takes LWE (LWR) parameters as inputs and computes the attack com-

plexities along with additional parameters required for the respective attack methods.

Using the Lattice Estimator for security analysis, we derive classical security estimation results for the 5 algorithms, as shown in Table 2. For the time complexity calculation of the BKZ algorithm, we employ the Core-SVP model [4], which is consistent with the methods used in the proposal documents for 4 of the 5 algorithms, excluding NCC-Sign. In Table 2, the column 'Claimed' is the claimed security shown in the proposal documents for each algorithm, and 'Estimated' is the security that we estimated by using the Lattice Estimator. For NTRU+, we observed that the description in the specification document is different from the reference implementation which is reflected in our security estimations with respective cases. More precisely, the LWE secret, which is sampled in the encapsulation phase and denoted as r in their scheme description, is sampled from $\{0,1\}^n$ according to the specification document (See Algorithm 6 and 9 of the NTRU+ document in [10]), while it is sampled from the centered binomial distribution in their implementation. We estimate both cases and denote the security estimation for the NTRU+ version of the specification document in parentheses. Also, for NCC-Sign, we additionally estimated the security without the Core-SVP model shown in the parentheses, since the proposal document of NCC-Sign presented the security result without using the Core-SVP model. The results are summarized as follows.

- We have observed a discrepancy between the claimed attack complexities for NTRU+ and the estimated attack complexities derived using the Lattice Estimator. For the NTRU+576, NTRU+768, and NTRU+864 parameters, we achieve 115.9 bits, 164.7 bits, and 189.2 bits, respectively, for the version of the reference implementation. These values exhibit a difference of 0.1 to 3.7 bits compared to the classical security levels claimed in the specification document. Also, larger gaps were observed between the claimed security and the estimation for the version of the specification document that utilizes the LWE with uniform binary secrets.
- For the SMAUG1280 parameters (Security level V) and TiGER256 (Security level V), classical security levels of 260.3 bits and 263 bits were claimed, but when measured using the Lattice Estimator, the attack complexities were found to be 259.2 bits and 262.0 bits, respectively.
- For HAETAE, the claimed parameters from the proposal document and the security analysis results of the Lattice Estimator are found to be similar, with an error range of less than 1 bit.
- For NCC-Sign, the proposal document presents security analysis results without using the Core-SVP model, and it is confirmed that the measured results using the Lattice Estimator were consistent. However, when measured using the Core-SVP model, it is determined that for parameters I, III, and V, the classical security levels were 123.2 bits, 190.1 bits, and 273.3 bits, respectively. This shows a difference of 18 to 24.5 bits compared to the results without the Core-SVP model in the NCC-Sign proposal document.

Table 2: Claimed vs. Estimated Security for the Round 1 Lattice-based Submissions. For NTRU+, the estimated results for the specification document version are reported in parentheses. For NCC-Sign, the estimated results without the Core-SVP model are reported in parentheses.

	Security Level	Claimed	Estimated
NTRU+	I ($n = 576$)	116	115.9 **(108.9)**
	I ($n = 768$)	161	**164.7 (156.5)**
	III	188	**189.2 (175.4)**
	V	264	263.4 **(243.5)**
SMAUG	I	120.0	120.0
	III	180.2	180.2
	V	260.3	**259.2**
TiGER	I	130	130.5
	III	200	**206.1**
	V	263	**262.0**
HAETAE	I	125	125.5
	III	236	236.1
	V	288	287.1
NCC-Sign	I	147.7	123.2 (147.7)
	III	211.5	190.1 (211.5)
	V	291.3	273.3 (291.3)

Based on the results, the additional attacks that each scheme needs to further consider through the Lattice Estimator are as follows:

- For SMAUG, it is confirmed that the SMAUG512 and SMAUG768 parameters achieve the claimed security levels. However, in the case of SMAUG1280, the claimed values and the estimated values differ for all attacks (`usvp`, `bdd`, `bdd_hybrid`, `dual`, `dual_hybrid`), which are displayed in Table 3. We remark that the displayed measurements are for the LWR instances.
- For TiGER, it is confirmed that the TiGER128 and TiGER192 parameters achieved the claimed security levels. However, for the TiGER256 parameters, the security level against `dual_hybrid` attacks differs between the claimed and the estimated values, which are displayed in Table 4. The displayed measurements are for the LWR instances when the `dual_hybrid` attack is applied.

Paper Organization This paper is structured as follows: In Chapter II, we introduce lattice-based hard problems, LWE and LWR, and provide definitions for

Table 3: Claimed vs. Estimated Classical Security for the SMAUG1280 parameter set

	claimed	estimated
`usvp`	317.1	316.2
`bdd`	319.5	318.4
`bdd_hybrid`	290.0	288.5
`dual`	329.1	328.2
`dual_hybrid`	260.3	259.2

Table 4: Claimed vs. Estimated Classical Security for the TiGER256 parameter set

	claimed	estimated
dual_hybrid	≥ 263	262.0

KEM and digital signatures. In Chapter III, we briefly describe the key features of the KpqC 1st round candidates, including three KEM schemes and two digital signature schemes. In Chapter IV, we present the time complexity computation methods for the BKZ algorithm used in the security analysis of LWE/LWR-based algorithms and discuss the attacks covered in the Lattice Estimator. In Chapter V, we present the security analysis results obtained from the Lattice Estimator for each scheme's proposed parameters and compare them with the claimed security. Finally, in Chapter VI, we summarize the main results and conclude the paper.

2 Preliminaries

2.1 The LWE and LWR Problem

In this section, we introduce lattice-based hard problems, Learning with Errors (LWE)[31] and Learning with Rounding (LWR)[30].

2.1.1 LWE

Let m, n, q be positive integers, $s \in \mathbb{Z}_q^n$ be a secret vector and χ be an error distribution on $\mathbb{Z}$. The LWE distribution $A_{m,n,q,\chi}^{LWE}(s)$ consisting of m samples is obtained as follows: For each $i \in \{1, 2, \ldots, m\}$, compute

$$b_i = \langle \vec{a}_i, \vec{s} \rangle + e_i \mod q$$

by choosing a vector $\vec{a}_i \in \mathbb{Z}_q^n$ uniformly and a small error $e_i \in \mathbb{Z}$ from the distribution χ, and then output $\{(\vec{a}_i, b_i)\}_{i=1}^m$ as the result.

The decision LWE problem is to distinguish either given samples $\{(\vec{a}_i, b_i)\}_{i=1}^m$ is from the distribution $A_{m,n,q,\chi}^{LWE}$ or from the uniform distribution. The search LWE problem is to find $s \in \mathbb{Z}_q^n$, given independent samples $\{(\vec{a}_i, b_i)\}_{i=1}^m$ from $A_{m,n,q,\chi}^{LWE}(s)$.

Variants of LWE. Let n and q be positive integers and $f(x) \in \mathbb{Z}[x]$ an irreducible polynomial of degree n. We define a polynomial ring $\mathcal{R} = \mathbb{Z}[x]/f(x)$ and its quotient ring $\mathcal{R}_q = \mathbb{Z}_q[x]/(f(x))$ modulo q. The Module LWE (MLWE) problem [8] is a variant of the LWE problem defined over a module $\mathcal{R}_q^k$ for positive integers k. The distribution $A_{m,n,q,k,\chi}^{MLWE}(\vec{s})$ for the secret value $\vec{s} \in \mathcal{R}_q^k$ is defined as follows: For $i \in \{1, 2, \ldots, m\}$, sample uniform random $\vec{a}_i \in \mathcal{R}_q^k$ and $e_i \in \mathcal{R} \leftarrow \chi^n$, calculate $b_i = \langle \vec{a}_i, \vec{s} \rangle + e_i \mod q \in \mathcal{R}_q$ and return the set of pairs $\{(\vec{a}_i, b_i)\}_{i=1}^m$ as results. It is also classified into the decision MLWE and search MLWE problems as in the LWE problem. For the specific case of MLWE when the dimension of module k is 1, we call it as Ring-LWE (RLWE) problem [25].

2.1.2 LWR

The LWR problem introduced by Banerjee et al. [6] obfuscates the secret by applying a deterministic rounding procedure ($\lfloor \cdot \rceil$) to linear equations instead of adding errors sampled from discrete Gaussian distributions. Given positive integers m, n, q, p, let $\vec{s} \in \mathbb{Z}_q^n$ be an n-dimensional secret vector. The LWR distribution $A_{m,n,q,p}^{LWR}(\vec{s})$ over $\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_p^m$ consisting of m samples is obtained as follows : For $i \in \{1, 2, \ldots, m\}$, compute $b_i = \lfloor (p/q) \cdot (\langle \vec{a}_i, \vec{s} \rangle \mod q) \rceil$ where $\vec{a}_i \in \mathbb{Z}_q^n$ is uniformly sampled, and return the set of pairs $\{(\vec{a}_i, b_i)\}_{i=1}^m$. The decision LWR problem is to distinguish either given samples $\{(\vec{a}_i, b_i)\}_{i=1}^m$ is from the distribution $A_{m,n,q,\chi}^{LWR}$ or from the uniform distribution. The search LWR problem is to find $\vec{s} \in \mathbb{Z}_q^n$, given independent samples $\{(\vec{a}_i, b_i)\}_{i=1}^m$ from $A_{m,n,q,p}^{LWR}(\vec{s})$. This definition can be extended to Ring-LWR (RLWR) and Module-LWR (MLWR) by using vectors of polynomials as in the LWE problem.

2.2 The Round 1 LWE/LWR-based Candidates

2.2.1 KEM

A Key Encapsulation Mechanism (KEM) is a triple of algorithms, Π=(KeyGen, Encaps, Decaps), where

- $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$: The key generation algorithm takes security parameter $\lambda > 0$ as an input and then outputs the pair of public key and private key (pk, sk).
- $(c, K) \leftarrow \mathsf{Encaps}(pk)$: The encapsulation algorithm takes the public key pk as an input and then outputs a pair of secret key K and ciphertext c.

– $(K \text{ or } \perp) \leftarrow \mathsf{Decaps}(sk, c)$: The decapsulation algorithm takes the private key sk and the ciphertext c as input, and then outputs the shared key K or $\perp$.

For correctness, it is required that, for all $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and for all $(c, K) \leftarrow \mathsf{Encaps}(pk)$, $\mathsf{Decaps}(sk, c) = K$ holds. In this section, we review the distinguished features of the KpqC Round 1 lattice-based KEMs NTRU+, SMAUG, and TiGER.

NTRU+. NTRU+ is an algorithm that improves the efficiency of the existing NTRU scheme [19]. It follows the strategy to construct NTT (Number Theory Transform)-friendly settings for NTRU which has been introduced in NTTRU [26] and NTRU-B [17]. The security of NTRU+ is based on the NTRU and RLWE problems. The main features are as follows:

– NTRU+ utilizes the NTT-friendly polynomial rings $\mathcal{R}_q = \mathbb{Z}_q[x]/(f(x))$, where $f(x) = x^n - x^{n/2} + 1$ is a cyclotomic trinomial of degree $n = 2^i 3^j$, and adapt NTT in all computations.
– In the encapsulation and decapsulation, new methods for secret key encoding (SOTP) and decoding (Inv) were proposed. The SOTP and Inv operations for $m \in \{0,1\}^n$, $u = (u_1, u_2) \in \{0,1\}^{2n}$, and $y \in \{-1,0,1\}^n$ are designed as follows.

$$\mathsf{SOTP}(m, u) = (m \oplus u_1) - u_2 \in \{-1,0,1\}^n \tag{1}$$

$$\mathsf{Inv}(y, u) = (y + u_2) \oplus u_1 \tag{2}$$

One can easily check $\mathsf{Inv}(\mathsf{SOTP}(m, u), u) = m$.
– To satisfy IND-CCA (Indistinguishability against adaptive Chosen-Ciphertext Attacks) security, NTRU+ applies a modified transform of the conventional Fujisaki-Okamoto (FO) transform [18]. The difference is that the decapsulation procedures require re-encryption when applying the FO transform, while NTRU+ removes the re-encryption in the decapsulation by recovering the random polynomial (denoted by r in their scheme) used in the encapsulation twice and then comparing between them.

SMAUG. SMAUG is designed based on the hardness of MLWE and MLWR problems, both of which utilize the sparse ternary secrets following the approaches in Lizard [14] and RLizard [21]. The main features are as follows.

– SMAUG KEM is obtained by first constructing an IND-CPA (Indistinguishability against Chosen-Plaintext Attacks) secure public-key encryption (PKE) scheme and then applying the FO transform [18] on it to achieve the IND-CCA security.
– The secret keys for MLWE and MLWR are sampled as sparse ternary vectors with fixed Hamming weights, respectively.
– The moduli q and p are set to powers of 2 in order to replace the rounding operations in the encapsulation with bit-wise shift operations.

TiGER. TiGER is designed based on the RLWE and RLWR problems with sparse secrets. The main features are as follows.

- TiGER consists of an IND-CPA PKE scheme, and an IND-CCA KEM obtained by applying the FO transform to it.
- All integer modulus in the scheme are set to be power of 2 for the same reason as in SMAUG, in order to replace the rounding operations with bitwise shifts.
- TiGER pre-defines the Hamming weight of the secrets of RLWE and RLWR and generates sparse vectors. Additionally, the errors for RLWE are also sampled as sparse vectors.
- The sizes of ciphertexts and public keys are relatively small because of using a small modulus of 1 byte ($q = 256$) for all suggested parameters.
- When encoding the secret key in TiGER KEM, they employ an Error Correcting Code (ECC) to reduce decryption failure rates. Therefore, it is possible to adjust the decryption failure rate to be negligible in the security parameter, despite using the small modulus q. They utilize XEf [5], D2 [4] for the ECC methods.

2.2.2 Digital Signatures

Digital signatures is a triple of algorithms Π=(KeyGen, Sign, Verify). The key generation (KeyGen) algorithm generates a pair of a public key and a private key. The signing (Sign) algorithm takes the private key and a message as inputs to generate a signature. The verification (Verify) algorithm takes the public key, message, and signature value as inputs to verify the validity of the signatures. These can be summarized as follows:

- $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$: The key generation algorithm takes security parameter λ as an input and then outputs a pair of public key and private key (pk, sk).
- $\sigma \leftarrow \mathsf{Sign}(sk, m)$: The signature algorithm takes the private key sk and a message m as inputs and then outputs a signature σ.
- 1 or $0 \leftarrow \mathsf{Verify}(pk, m, \sigma)$: The verification algorithm takes the public key pk, a message m, and a signature σ as inputs. It outputs 1 if the signature is valid, and 0 otherwise.

In this section, we summarize the distinguished features of the KpqC Round 1 lattice-based signature schemes HAETAE and NCC-Sign.

HAETAE. HAETAE utilizes the Fiat-Shamir with Aborts paradigm [23, 24] as in the CRYSTALS-Dilithium [16], one of the standards selected in the NIST PQC standardization project. HAETAE uses a bimodal distribution proposed in the rejection sampling of BLISS signatures [15]. The main features are as follows:

- In lattice-based digital signature algorithms, the distribution used for rejection sampling has a significant impact on the signature size. HAETAE uses

a hyperball uniform distribution to reduce the signature size, albeit at the cost of speed compared to Dilithium.
– HAETAE leverages a module structure and uses a predefined polynomial ring $\mathcal{R}_q = \mathbb{Z}_q[x]/(x^{256}+1)$ for all parameter sets, making it easy to adjust parameters according to the required security level.

NCC-Sign. NCC-Sign is a digital signature algorithm that combines the design rationale of CRYSTALS-Dilithium and NTRU prime [7], which were also round 3, 4 candidates for NIST PQC standardization project KEM algorithms. NCC-Sign also adopts Fiat-Shamir with Aborts paradigm as in HAETAE and Dilithium, but instead of using a cyclotomic polynomial ring of $\mathcal{R}_q = \mathbb{Z}_q[x]/(x^n+1)$, it uses the non-cyclotomic polynomial ring of the form $\mathcal{R}_q = \mathbb{Z}_q[x]/(x^p+x+1)$, where p is a prime. The main features are as follows:

– Due to the use of a non-cyclotomic ring, NTT cannot be applied to polynomial multiplications. In NCC-Sign, polynomial multiplication is computed using the Toom-Cook method, one of the divide-and-conquer techniques. For a prime p such that $p \leq 4n, n \in \mathbb{Z}$, the algorithm computes polynomial multiplication of degree $4n$ and exploits Toom-Cook-4-way and Karatsuba multiplication.

3 Security Analysis Methods

3.1 Time complexity Estimation of the BKZ algorithm

The BKZ algorithm [12] is a state-of-the-art lattice basis reduction algorithm used to find short bases within a given lattice, and it exhibits exponential time complexity. To analyze the security of the LWE/LWR-based algorithms, the instances of LWE/LWR used in the algorithms are induced to the problems to find short vectors in lattices which are given by the choices of attack strategies such as Dual and Primal attacks. Hence, it can be solved by using the BKZ algorithm.

The core idea behind the BKZ algorithm is to iteratively apply a Shortest Vector Problem (SVP) solver to sub-lattices of dimension smaller than the original lattice. When the dimension of the sub-lattice to which the SVP solver is applied is $\beta > 0$, it is referred to as β-BKZ, and this sub-lattice is called a 'block'.

The Core-SVP model [4] is a measurement model used to estimate the time complexity of the BKZ algorithm from a conservative perspective. When calculating the time complexity of the BKZ algorithm using the Core-SVP model, the time complexity of β-BKZ is estimated to be $2^{c\cdot\beta}$, which is a lower bound of the time complexity of a single application of the SVP solver $\left(2^{c\cdot\beta+o(\beta)}\right)$, where $c \in [0,1]$ is constant. This conservative model is designed to ensure that the security predictions of the BKZ algorithm remain unaffected by improvements in the efficiency of either the number of iterations of applying the SVP solver or

the efficiency of the SVP solver itself, thus preserving the algorithm's security guarantees.

In the Core-SVP model, the constant $c \in [0, 1]$ used for calculating the BKZ time complexity is determined based on the efficiency of the SVP solver. In [4], it was employed as shown in Table 5. For quantum SVP solvers, continuous improvements in efficiency have led to the existence of algorithms with $c_Q = 0.257$ [11]. In this paper, we calculate the BKZ time complexity using $c = 0.292$ for classical security. When using the Core-SVP model, quantum security (in bits) can be simply estimated by multiplying classical security (in bits) with $c_Q/0.292$.

Table 5: The BKZ time complexity (T) for classical security and quantum security in the Core-SVP model

	classical	quantum
c	0.292	0.265
T	$2^{0.292\beta}$	$2^{0.265\beta}$

3.2 Dual Attack

The dual attack identifies a short vector v that is orthogonal to matrix A. Given $(A, \vec{b}) \in \mathbb{Z}_q^{k\times l} \times \mathbb{Z}_q^k$ either from the LWE distribution or the uniform distribution, a lattice $\Lambda_m^{\mathsf{dual}}$ can be defined as follow. Let $A_{[m]}$ be the upmost $m \times l$ sub-matrix of A for $m \leq k$.

$$\Lambda_m^{\mathsf{dual}} := \left\{ (\vec{u}, \vec{v}) \in \mathbb{Z}^m \times \mathbb{Z}^l : A_{[m]}^\top \vec{u} + \vec{v} = 0 \bmod q \right\}$$

If it is the case $\vec{b} = A\vec{s} + \vec{e}$, with a short non-zero element $(\vec{u}, \vec{v})$, an attacker can compute $\langle \vec{u}, \vec{b}_{[m]} \rangle = -\langle \vec{v}, \vec{s} \rangle + \langle \vec{u}, \vec{e}_{[m]} \rangle$, where $\vec{b}_{[m]}$ and $\vec{e}_{[m]}$ are the upmost m-dimensional sub-vector of $\vec{b}$. Hence, the attacker can determine it is an LWE instance if $\langle \vec{u}, \vec{b}_{[m]} \rangle$ is short enough. Therefore, finding a sufficiently short non-zero vector in the lattice $\Lambda_m^{\mathsf{dual}}$ implies solving the decision-LWE problem. To find a short lattice element of $\Lambda_m^{\mathsf{dual}}$, the attack employs the β-BKZ lattice basis reduction algorithm.

3.3 Primal Attack

The primal attack on LWE addresses the bounded distance decoding (BDD) problem directly. In other words, when provided with LWE samples (A, b), it

seeks a vector $w = As$ such that $\|b - w\|$ is unusually small. There are two main strategies to solve BDD: the first strategy is to utilize Babai's nearest algorithm with lattice basis reduction [22], and the second is to reduce BDD problem into unique-SVP (uSVP) problem and solve it using the lattice basis reduction algorithms [2, 4]. Here, we will elaborate on the second method, which is more widely considered.

Given an LWE instance $(A, b = As + e) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$, a lattice Λ_m can be defined as follow. $B = \left(A_{[m]} \,|I_m|\, b_{[m]}\right) \in \mathbb{Z}_q^{m \times (n+m+1)}$.

$$\Lambda_m = \left\{v \in \mathbb{Z}_q^{n+m+1} : Bv \bmod q\right\}$$

Therefore, a short non-zero vector in the lattice Λ_m can be transformed into the non-trivial solutions for the LWE equation. This attack utilizes the β-BKZ algorithm to find the sufficiently short vector in the lattice Λ_m.

3.4 Hybrid Attack

An attack that combines techniques, such as meet-in-the-middle, with either Primal or Dual attacks is known as a hybrid attack. Hybrid attacks are generally not as efficient as Primal or Dual attacks, but they can be effective in cases where the secret key in LWE follows a specialized distribution. In [20], by incorporating lattice reduction techniques and implementing a meet-in-the-middle (MITM) strategy, it is possible to diminish the complexity of the attack on the NTRUEncrypt private key from $2^{84.2}$ to $2^{60.3}$ for the parameter set for 80-bit security. Also, Jung Hee Cheon et al. [13] introduced a hybrid attack strategy that integrates dual lattice attacks with the MITM approach. This approach involves increasing the error size while simultaneously reducing the dimension and Hamming weights of the secret vector. As the MITM attack cost is strongly correlated with the dimension of the secret vector but less affected by error size, this trade-off significantly reduces the overall cost of the MITM attack when applying it to the LWE with sparse secrets.

4 KpqC Round 1 LWE/LWR-based algorithms Security analysis

4.1 Parameters

In this section, we summarize the proposed parameters used in the underlying LWE/LWR instances in the respective schemes. For simplicity, we use the same notations as in the original specification documents.

NTRU+ Parameters. NTRU+ is based on NTRU and RLWE, and the proposed parameters used to analyze attack complexities of RLWE are as shown in Table 6. They use the quotient ring $\mathcal{R}_q = \mathbb{Z}[X]/(X^n - X^{n/2} + 1)$ for dimension $n = 2^i 3^j$ and fixed modulus $q = 3457$ for all parameters. For the RLWE

secret distribution and error distribution, they utilize the uniform distribution on $(0, 1)$ and the centered binomial distribution in their specification document and reference implementation, respectively.

Table 6: NTRU+ Proposed parameter sets

	576	768	864	1152
n	576	768	864	1152
q	3457	3457	3457	3457
security level	I	I	III	V

SMAUG Parameters. SMAUG is based on MLWE/MLWR, and the parameters used for the attack on MLWE/MLWR are as shown in Table 7. They use the quotient ring $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^n + 1)$ for power of 2 integer n and positive integer q. The secret keys for each LWE and LWR instance, denoted as s and r are sampled as sparse vectors with fixed Hamming weights, where the Hamming weights are denoted as h_s, h_r, respectively. σ is the standard deviation of the discrete Gaussian distribution to sample the errors in LWE.

Table 7: SMAUG Proposed parameter sets

	`SMAUG128`	`SMAUG192`	`SMAUG256`
n	512	768	1280
m	512	768	1280
q	1024	1024	1024
p	256	256	256
h_r	132	147	140
h_s	140	150	145
σ	1.0625	1.0625	1.0625
security level	I	III	V

TiGER Parameters. TiGER is based on RLWR/RLWE and the parameters used for the attack are as shown in Table 8. They use the quotient ring $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^n + 1)$ for a power of 2 integer n and a positive integer q. k_1 and k_2 are power of 2's and represents the modulus used for ciphertext compression.

h_s and h_r are the Hamming weights of the secret key and the ephemeral secret used for encapsulation. h_e is the Hamming weight of the LWE error.

Table 8: TiGER Proposed parameter sets

	TiGER128	TiGER192	TiGER256
n	512	1024	1024
m	512	1024	1024
q	256	256	256
p	128	64	128
h_r	128	84	198
h_s	160	84	198
h_e	32	84	32
k_1	64	64	128
k_2	64	4	4
security level	I	III	V

HAETAE Parameters. HAETAE is based on MLWE/MSIS and the parameters used for the attack are as shown in Table 9. They use the quotient ring $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^n + 1)$ for positive integers n and q which are set to 256 and 64513, respectively, for all parameter sets. (k, ℓ) denotes the matrix size of the module structure. They select the private key from the uniform distribution over $[-\eta, \eta]$, and τ refers to the Hamming weight of the binary challenge.

Table 9: HAETAE Proposed parameter sets

	HAETAE120	HAETAE180	HAETAE260
n	256	256	256
q	64513	64513	64513
(k, ℓ)	$(2, 4)$	$(3, 6)$	$(4, 7)$
η	1	1	1
τ	39	49	60
security level	I	III	V

NCC-Sign Parameters. NCC-Sign is based on RLWE/RSIS and the parameters used for the attack are shown in Table 10. They use the ring $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^p - X - 1)$ for prime numbers p and q. Also, they select the private key from the distribution over $[-\eta, \eta]$, and τ refers to the number of nonzero coefficients in $\{-1, 0, 1\}$.

Table 10: NCC-Sign Proposed parameter sets

	I	III	V
p	1021	1429	1913
q	8339581	8376649	8343469
η	2	2	2
τ	25	29	32
security level	I	III	V

4.2 Analysis using the Lattice Estimator

In this section, we report our estimated results for the lattice attacks in [3] outlined in Section 3. The results of security analysis using the Lattice Estimator for NTRU+, SMAUG, TiGER, HAETAE, and NCC-Sign schemes are shown in Table 11, Table 12, Table 13, Table 14, Table 15, Table 16, and Table 17.

The column names in each table, "sec" and "β" represent classical security in bits and BKZ block size respectively. For the BKZ time complexity estimation, we use the Core-SVP model except Table 17. Among the row names in each table, "`usvp`" refers to the attack complexity for the Primal attack described in Section 3.3, and `bdd, bdd_hybrid, bdd_mitm_hybrid` attacks are variations of the Primal attack. Also, "`dual`" means the attack complexity for the Dual attack explained in Section 3.2, and `dual_hybrid, dual_mitm_hybrid` are variations of the Dual attack. For more details about the attacks, we recommend to see [3]. We remark that when analyzing the security of SMAUG and TiGER, we measured attack complexities for both LWE and LWR instances, and reported the minimum value. In the case of NTRU+, since it does not use a sparse secret key in the LWE instance, during the security analysis, we did not measure the attack complexities for `bdd_mitm_hybrid` and `dual_mitm_hybrid`, which are expected to be less efficient compared to other attacks.

In the case of NTRU+, Table 11 shows `dual_hybrid` has the smallest attack complexity. In Table 12, overall attack complexities have increased, and `usvp` has the smallest complexity. In the case of SMAUG, according to Table 13, the most effective attack differs for each parameter set: the most effective attack for SMAUG128 is `usvp`, `dual_hybrid` for SMAUG192, and `dual_hybrid` for SMAUG256. In the case of TiGER, as listed in Table 14, TiGER128 exhibits

the smallest complexity for Primal attack `usvp`. For TiGER192 and TiGER256, `dual_hybrid` is the most effective method.

In the case of HAETAE, in Table 15, for the claimed security of 120 bits and 260 bits, the most effective attack method is `dual_hybrid` followed by `usvp`. For the security of 180 bits, `usvp` has the smallest attack complexity. In the case of NCC-Sign, Table 16 and Table 17 show similar results. In Table 16, `usvp` is confirmed to have the smallest attack complexity, while in Table 17, `bdd` exhibits the smallest attack complexity.

Table 11: NTRU+ Security Estimation

	576		768		864		1152	
	sec	β	sec	β	sec	β	sec	β
`usvp`	109.8	376	**156.5**	536	180.2	617	252.9	866
`bdd`	110.8	375	157.4	535	181.0	617	253.7	865
`bdd_hybrid`	111.0	375	157.4	535	181.2	617	316.1	864
`dual`	114.8	393	162.4	556	186.9	640	261.3	895
`dual_hybrid`	**108.9**	372	153.0	523	**175.4**	599	**243.5**	833

Table 12: NTRU+ Security Estimation _ rev

	576		768		864		1152	
	sec	β	sec	β	sec	β	sec	β
`usvp`	**115.9**	397	**164.7**	564	189.8	650	266.0	911
`bdd`	116.9	397	165.7	563	190.7	649	266.9	911
`bdd_hybrid`	193.2	397	264.1	563	300.0	649	408.9	911
`dual`	120.9	414	171.1	586	196.5	673	274.8	941
`dual_hybrid`	117.2	400	164.9	564	**189.2**	647	**263.4**	901

Table 13: SMAUG Security Estimation

	128		192		256	
	sec	β	sec	β	sec	β
usvp	**120.0**	411	187.2	641	316.2	1083
bdd	120.9	411	188.5	642	318.4	1090
bdd_hybrid	121.3	411	189.0	642	288.5	674
bdd_mitm_hybrid	166.5	410	221.0	496	277.8	680
dual	125.9	431	195.3	669	328.2	1124
dual_hybrid	122.7	399	**180.2**	575	**259.2**	749

Table 14: TiGER Security Estimation

	128		192		256	
	sec	β	sec	β	sec	β
usvp	**130.5**	447	277.4	950	279.7	958
bdd	131.4	445	281.5	964	280.7	958
bdd_hybrid	131.4	445	220.2	472	280.7	958
bdd_mitm_hybrid	173.8	419	212.7	503	316.5	730
dual	137.5	471	290.5	995	291.7	999
dual_hybrid	131.9	428	**206.1**	535	**262.0**	835

Table 15: HAETAE Security Estimation

	120		180		260	
	sec	β	sec	β	sec	β
usvp	**125.6**	430	238.0	815	290.2	994
bdd	126.6	429	238.8	815	291.1	993
bdd_hybrid	126.6	429	238.8	815	291.1	993
bdd_mitm_hybrid	219.3	429	390.9	815	472.7	993
dual	130.5	447	245.6	841	298.4	1022
dual_hybrid	126.4	432	**236.1**	808	**287.1**	982

Table 16: NCC-Sign Security Estimation with the Core-SVP model

	1		3		5	
	sec	β	sec	β	sec	β
usvp	**123.2**	422	**190.1**	651	**273.3**	936
bdd	124.6	421	191.0	651	274.3	935
bdd_hybrid	124.6	421	191.0	651	274.3	935
bdd_mitm_hybrid	270.0	421	406.1	651	588.6	935
dual	126.4	433	194.2	665	278.6	954
dual_hybrid	124.8	427	191.1	654	273.6	937

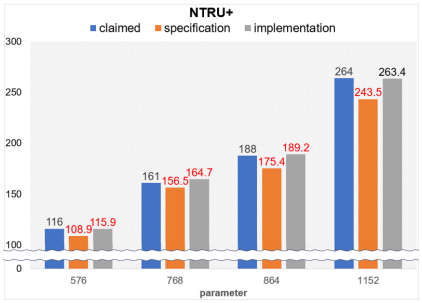
Table 17: NCC-Sign Security Estimation without the Core-SVP model as they evaluated in the Round 1 Proposal (less conservative)

	1		3		5	
	sec	β	sec	β	sec	β
usvp	149.7	422	213.9	651	294,0	936
bdd	**147.7**	413	**211.5**	641	**291.3**	924
bdd_hybrid	**147.7**	413	**211.5**	641	**291.3**	924
bdd_mitm_hybrid	261.8	421	394.9	651	574.2	935
dual	153.8	433	219.6	668	302.4	962
dual_hybrid	150.5	421	214.9	651	295.5	937

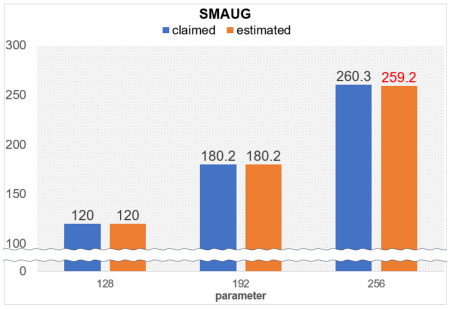
4.3 Comparisons with the Claimed Security

We present the comparison of the claimed vs. estimated (classical) security in bits for each scheme in Fig. 1a, Fig. 1b, Fig. 1c, Fig. 1d, and Fig. 1e.

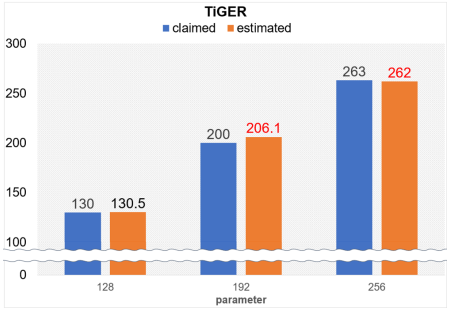
For NTRU+, we measured the security based on both the specification document and the implementation. The result from the implementation was similar to the claimed security in the proposal document. However, the result based on the specification document indicated lower security than the implementation result. The reason for these different results occurred from the process of sampling the secret 'r' value in the LWE instances using the H function in the Encaps algorithm in NTRU+ (See Algorithm 6 and 9 in the NTRU+ specification document). The specification samples the secret 'r' with uniform binary values, however, the implementation samples it with ternary values following the centered binomial distribution.



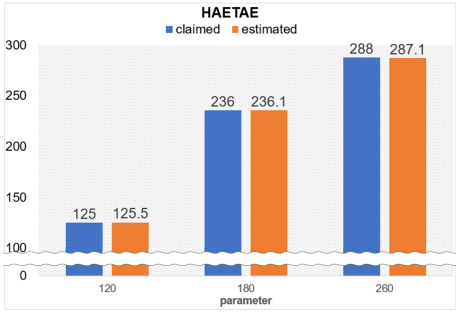
(a) Comparison of the claimed security and estimated results in which estimated results are measured for the versions of specification and implementation for NTRU+, respectively



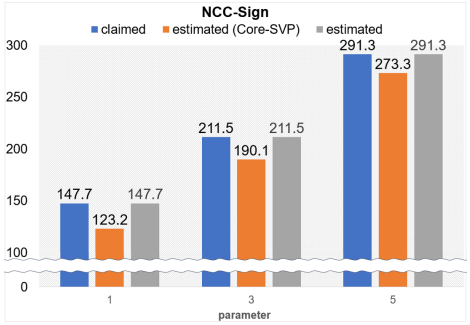
(b) Comparson of the claimed security and estimated results for SMAUG parameters



(c) Comparison of the claimed security and estimated results for TiGER parameters



(d) Comparison of the claimed security and estimated results for HAETAE parameters



(e) Comparison of the claimed security and estimated results with Core-SVP and without Core-SVP for NCC-Sign parameters

The differences between the analysis by using the Lattice Estimator and the analysis presented in the proposal document can be summarized as follows.

- For the SMAUG1280 parameters, the claimed security in the proposal document is of 260.3 bits, but the estimated result using Lattice Estimator resulted in an attack amount of 259.2 bits.
- In the case of TiGER256(Security level V), the classical security of 263 bits was claimed, but the estimated result was 262.0 bits.
- The estimated results of NTRU+ were found different from the claimed attack complexities for all parameters. For the NTRU+576, NTRU+768, NTRU+864, and NTRU+1152 parameters, they each satisfy classical security levels of 115.9 bits, 164.7 bits, 189.2 bits, and 263.4 bits, respectively, for the implementation version. These values differ by 0.1 to 3.7 bits from the classical security levels claimed in the proposal document, which were 116 bits, 161 bits, 188 bits, and 264 bits. For the document version of NTRU+ using LWE with uniform binary secrets, the gaps between the claimed and estimated security get larger.
- For HAETAE, the result claimed in the proposal document and security analysis results were similar about all parameters, with an error range of less than 1 bit.
- In the case of NCC-Sign, the proposal document provided results of security analysis without using the Core-SVP model and the estimations using the Lattice Estimator were found to match these results. When we measured using the Core-SVP model, it was observed that parameters I, III, and V achieve classical security levels of 123.2 bits, 190.1 bits, and 273.3 bits, respectively. This represents differences of 18 to 24.5 bits compared to the results presented in the NCC-Sign proposal document.

5 Conclusion

In this paper, we discussed the results of a security analysis using the Lattice Estimator for five Round 1 lattice-based candidates proposed in the KpqC Competition. It was found that NTRU+ had differences of approximately 0.1 to 3.7 bit compared to the claimed results of security analysis for all parameters when using the centered binomial distribution as a secret distribution in LWE. For SMAUG and TiGER, the classical security of parameters in the security level V was observed to differ by approximately 1 bit from the estimated results. In the case of HAETAE and NCC-Sign, we confirmed that the claimed parameters are closely similar to the security analysis results. We also remark that the Lattice Estimator does not exhaustively cover all recent attacks for LWE including [27]. We will analyze the KpqC Round 1 lattice-based schemes further by applying various recent LWE attacks for future works.

Acknowledgement. This work is the result of commissioned research project supported by the affiliated institute of ETRI [2023-080].

References

[1] Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing. p. 99–108. STOC '96, Association for Computing Machinery, New York, NY, USA (1996). https://doi.org/10.1145/237814.237838, https://doi.org/10.1145/237814.237838

[2] Albrecht, M.R., Göpfert, F., Virdia, F., Wunderer, T.: Revisiting the expected cost of solving usvp and applications to lwe. In: Takagi, T., Peyrin, T. (eds.) Advances in Cryptology – ASIACRYPT 2017. pp. 297–322. Springer International Publishing, Cham (2017)

[3] Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. Cryptology ePrint Archive, Paper 2015/046 (2015), https://eprint.iacr.org/2015/046, https://eprint.iacr.org/2015/046

[4] Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key Exchange—A new hope. In: 25th USENIX Security Symposium (USENIX Security 16). pp. 327–343. USENIX Association, Austin, TX (Aug 2016), https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim

[5] Baan, H., Bhattacharya, S., Fluhrer, S., Garcia-Morchon, O., Laarhoven, T., Rietman, R., Saarinen, M.J., Tolhuizen, L., Zhang, Z.: Round5: Compact and Fast Post-quantum Public-Key Encryption, pp. 83–102 (07 2019). https://doi.org/10.1007/978-3-030-25510-7_5

[6] Banerjee, A., Peikert, C., Rosen, A.: Pseudorandom functions and lattices. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 719–737. Springer (2012)

[7] Bernstein, D.J., Chuengsatiansup, C., Lange, T., van Vredendaal, C.: Ntru prime: reducing attack surface at low cost. Cryptology ePrint Archive, Paper 2016/461 (2016), https://eprint.iacr.org/2016/461, https://eprint.iacr.org/2016/461

[8] Boudgoust, K., Jeudy, C., Roux-Langlois, A., Wen, W.: On the hardness of module-lwe with binary secret. In: Cryptographers' Track at the RSA Conference. pp. 503–526. Springer (2021)

[9] Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors (2013)

[10] Center, K.R.: Kpqc competition round 1, available from: https://www.kpqc.or.kr/competition.html [last accessed June 2023]

[11] Chailloux, A., Loyer, J.: Lattice sieving via quantum random walks (2021)

[12] Chen, Y., Nguyen, P.Q.: Bkz 2.0: Better lattice security estimates. In: Lee, D.H., Wang, X. (eds.) Advances in Cryptology – ASIACRYPT 2011. pp. 1–20. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)

[13] Cheon, J.H., Hhan, M., Hong, S., Son, Y.: A hybrid of dual and meet-in-the-middle attack on sparse and ternary secret lwe. IEEE Access **7**, 89497–89506 (2019). https://doi.org/10.1109/ACCESS.2019.2925425

[14] Cheon, J.H., Kim, D., Lee, J., Song, Y.: Lizard: Cut off the tail! a practical post-quantum public-key encryption from lwe and lwr. In: International Conference on Security and Cryptography for Networks. pp. 160–177. Springer (2018)

[15] Ducas, L., Durmus, A., Lepoint, T., Lyubashevsky, V.: Lattice signatures and bimodal gaussians. In: Annual Cryptology Conference. pp. 40–56. Springer (2013)

[16] Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: Crystals-dilithium: A lattice-based digital signature scheme. IACR Transactions on Cryptographic Hardware and Embedded Systems pp. 238–268 (2018)
[17] Duman, J., Hövelmanns, K., Kiltz, E., Lyubashevsky, V., Seiler, G., Unruh, D.: A thorough treatment of highly-efficient ntru instantiations. In: IACR International Conference on Public-Key Cryptography. pp. 65–94. Springer (2023)
[18] Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Wiener, M. (ed.) Advances in Cryptology — CRYPTO' 99. pp. 537–554. Springer Berlin Heidelberg, Berlin, Heidelberg (1999)
[19] Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In: International algorithmic number theory symposium. pp. 267–288. Springer (1998)
[20] Howgrave-Graham, N.: A hybrid lattice-reduction and meet-in-the-middle attack against ntru. In: Menezes, A. (ed.) Advances in Cryptology - CRYPTO 2007. pp. 150–169. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
[21] Lee, J., Kim, D., Lee, H., Lee, Y., Cheon, J.H.: Rlizard: Post-quantum key encapsulation mechanism for iot devices. IEEE Access **7**, 2080–2091 (2018)
[22] Lindner, R., Peikert, C.: Better key sizes (and attacks) for lwe-based encryption. Cryptology ePrint Archive, Paper 2010/613 (2010), https://eprint.iacr.org/2010/613, https://eprint.iacr.org/2010/613
[23] Lyubashevsky, V.: Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 598–616. Springer (2009)
[24] Lyubashevsky, V.: Lattice signatures without trapdoors. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 738–755. Springer (2012)
[25] Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. Journal of the ACM (JACM) **60**(6), 1–35 (2013)
[26] Lyubashevsky, V., Seiler, G.: Nttru: truly fast ntru using ntt. Cryptology ePrint Archive (2019)
[27] May, A.: How to meet ternary lwe keys. Cryptology ePrint Archive, Paper 2021/216 (2021), https://eprint.iacr.org/2021/216, https://eprint.iacr.org/2021/216
[28] NIST: Post-quantum cryptography, available from: https://csrc.nist.gov/projects/post-quantum-cryptography [last accessed June 2023]
[29] NIST: Standardization of additional digital signature schemes, available from: https://csrc.nist.gov/projects/pqc-dig-sig/standardization [last accessed August 2023]
[30] Pointcheval, D., Johansson, T.: Advances in cryptology – eurocrypt 2012 : 31st annual international conference on the theory and applications of cryptographic techniques, cambridge, uk, april 15-19, 2012. proceedings. Lecture notes in computer science, 7237, EUROCRYPT (31st : 2012 : Cambridge, England), Springer, Berlin (2012)
[31] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM **56**(6) (sep 2009). https://doi.org/10.1145/1568318.1568324, https://doi.org/10.1145/1568318.1568324
[32] Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Journal on Computing **26**(5), 1484–1509 (oct 1997). https://doi.org/10.1137/s0097539795293172, https://doi.org/10.1137%2Fs0097539795293172