Improving BDD Cryptosystems in General Lattices

Michael Rose, Thomas Plantard, and Willy Susilo

Center for Computer and Information Security Research
School of Computer Science and Software Engineering
University of Wollongong
{mr79, thomaspl, wsusilo}@uow.edu.au

Abstract. A prime goal of Lattice-based cryptosystems is to provide an enhanced security assurance by remaining secure with respect to quantum computational complexity, while remaining practical on conventional computer systems. In this paper, we define and analyze a superclass of GGH-style nearly-orthogonal bases for use in private keys, together with a subclass of Hermite Normal Forms for use in Micciancio-style public keys and discuss their benefits when used in Bounded Distance Decoding cryptosystems in general lattices. We propose efficient methods for the creation of such nearly-orthogonal private bases and "Optimal" Hermite Normal Forms and discuss timing results for these methods. Finally, we propose a class of cryptosystems based on the use of these constructions and provide a fair comparison between this class of cryptosystems and related cryptosystems.

Key Words: lattice-based cryptography, BDD, HNF, nearly orthogonal basis

1 Introduction

1.1 Post-Quantum Cryptography

The intractability of both integer-factorization and the discrete logarithm problem have long been used to construct asymmetric-key cryptosystems and prove such cryptosystems secure. In 1994, Shor's paper "Algorithms for quantum computation: discrete logarithms and factoring" [1] rendered these two problems tractable on a quantum computer. As a result of the continual advancements in quantum computer architecture, new one-way trapdoor functions must be developed that will remain secure after quantum computers become practical. One promising avenue of research in this direction is lattice-based cryptography.

1.2 Lattice-based Cryptography Overview

Since Ajtai's seminal paper *"Generating hard instances of Lattice problems"* [2], there has been an ongoing effort by the cryptographic community to produce a practical yet secure lattice-based (see Def. 1) cryptosystem based on lattice problems [3]. It is believed that such lattice problems remain hard in quantum complexity [4], providing strong indications that lattice-based cryptosystems remain secure assuming the existence of quantum computers. This is immensely beneficial for parties who need to retain long term data privacy, beyond the foreseeable advent of practical, large-scale quantum computers.

Definition 1. *A lattice $\mathcal{L}$ is a discrete sub-group of $\mathbb{R}^n$, or equivalently the set of all the integral combinations of $d \leq n$ linearly independent vectors over $\mathbb{R}$.*

$$\mathcal{L} = \mathbb{Z}\, b_1 + \cdots + \mathbb{Z}\, b_d, \quad b_i \in \mathbb{R}^n.$$

$B = (b_1, ..., b_d)$ is called a basis of $\mathcal{L}$ and d, the dimension of $\mathcal{L}$. If $d = n$, the lattice is called full-rank.

We will refer $\mathcal{L}_B$ as a lattice of basis B. In this paper, full-rank lattices will be used (simplifying some definitions accordingly) and be represented by a matrix of the basis vectors, in row-vector form.

Many lattice problems are based on distance minimization. Distance is measured in a lattice system by a norm, denoted $\|.\|$. In this paper, the euclidean norm as defined in Def. 2 will be used for some calculations. In addition, the l_∞-norm, also known as the *infinity norm* or the *max norm*, will also be used as defined in Def. 3. The induced matrix norm as defined in Def. 4 is also required.

Definition 2. *Let $\mathcal{H}$ be a Hilbert space of some finite dimension d. Given a vector $v \in \mathcal{H}$, $\|v\|_2 = \sqrt{v_1^2 + v_2^2 + v_3^2 + ... + v_d^2}$.*

Definition 3. *Let $\mathcal{H}$ be a Hilbert space of some finite dimension d. Given a vector $v \in \mathcal{H}$, $\|v\|_\infty = \max\left\{|v_i|, 1 \leq i \leq d\right\}$.*

Definition 4. *Let $A \in \mathbb{R}^{n,n}$ be a matrix. Then a matrix norm denoted as $\|A\|$ is said to be* consistent *to a vector norm $\|.\|$, if $\|A\| = sup\left\{\|xA\|, \quad x \in \mathbb{R}^n, \quad \|x\| = 1\right\}$.*

Both matrix norm $\|.\|_\infty$ and $\|.\|_2$ can be easily computed [5].

In this paper, we also use $\lfloor x \rceil$ as the closest integer of x if $x \in \mathbb{R}$ or the closest vector with integer coefficient if $x \in \mathbb{R}^n$.

1.3 Existing Lattice Cryptosystems

Cryptosytems in q-ary lattices In 1997, Ajtai and Dwork [6] proposed a cryptosystem such that its security is based on a variant of the Shortest Vector Problem (SVP) called the *unique* Shortest Vector Problem. This cryptosystem received wide attention due to a surprising security proof based on worst-case assumptions. Nonetheless, due to extreme ciphertext expansion, this cryptosystem is merely a theoretical proposition and it cannot be used in practice. Furthermore, Nguyen and Stern presented a heuristic attack against this cryptosystem [7]. Since then, this initial proposition has been improved [8–10] and inspired many other cryptosystems based on SVP [11–13]. For a more detailed examination of SVP-based trapdoors, we refer the readers to [14].

However, each of these cryptosytems, like most lattice based cryptographic protocols [15, 4, 16–20] are based on q-ary lattices [4]. In 1998, a non broken q-ary lattice cryptosystem was discovered: NTRU, for N^{th} degree truncated polynomial ring units. NTRU was proposed in 1998 by Hoffstein, Pipher and Silverman [21]. This cryptosystem was not modelled initially as a Goldreich-GoldwasserHalevi (GGH) type cryptosystem, however, it can been represented as one, which has been useful in evaluating the security of the cryptosystem [22].

We note that while some problems in conditional lattices (such as q-ary or ideal lattices) are reducible to problems in general lattices, the opposite is not true. As such, we feel that providing a practical alternative based only on general lattices is prudent. It is clear that general lattices are at least as secure as any conditional form of lattice and should there be any conditional lattice assumption broken in the future, a practical alternative would be needed. Therefore, in this paper, we will work with the class of cryptosystems defined by GGH and Micciancio, in general lattices constructed by some non-random, nearly-orthogonal private basis, to allow the public basis to have no specific constraints.

GGH In 1996, Goldreich, Goldwasser and Halevi [23] proposed an efficient way to build a cryptosystem that uses lattice theory, inspired by McEliece cryptosystem [24] and based on Bounded Distance Decoding (see Def. 5). Their practical proposition of a cryptosystem was strongly attacked and broken by Nguyen in 1999 [25]. However, the general idea is still viable, as can be seen by the many variants of the basic GGH cryptosystem that have been proposed since (see [26–28]).

Definition 5. *Let $\mathcal{L}$ be a lattice, $\lambda_1(\mathcal{L})$ be the norm of the shortest non-zero vector of the lattice $\mathcal{L}$. Given a vector w such that $\exists u \in \mathcal{L}, \|w - u\| \leq \alpha\lambda_1(\mathcal{L})$, the α-Bounded Distance Decoding Problem (BDD_α) is to find such a vector u.*

The three general cryptographic algorithms for the GGH class of cryptosystems are as follows:

- **Setup:** Compute a "good basis" A and a "bad basis" B of a lattice $\mathcal{L}$. i.e. $\mathcal{L}(A) = \mathcal{L}(B)$. Provide B as public and keep A private.
- **Encrypt:** To encrypt a plaintext vector-message p: Use the bad basis to create a random vector $r \in \mathcal{L}$. Publish the encrypted message which is the addition of the vector message with the random vector: $c = p + r$.
- **Decrypt:** Use the good basis to find the closest vector in the lattice to the encrypted ciphertext message c. The closest vector of the encrypted message c is the random vector r[1]. Subtract the random vector from the encrypted message to obtain the vector message $p = c - r$.

The security of the GGH class of cryptosystems relies on the following three assumptions.

[1] under the supposition that the norm of p is sufficiently small

i) It is easy to compute a “bad basis” from a “good basis”, but it is difficult to compute a “good basis” from a “bad basis”.
ii) It is easy to create a random vector of a lattice even with a “bad basis”.
iii) It is easy to find the closest vector with a “good basis” but difficult to do so with a “bad basis”.

Goldreich et al. suggested the use of Babai’s Round-Off [29] method to address assumption (iii).

After Nguyen’s first attack, the utilization of the initial GGH proposition requires lattices with higher dimension (> 500) to ensure security. As a result, the computation of the closest vector, even with a “good basis”, becomes very expensive. Another significant drawback to this class of cryptosystems is the enormous key sizes in practical implementations, particularly in such higher dimensions. Due to these two drawbacks, we feel that this cryptosystem has limited practicality.

Micciancio In 2001, Micciancio [27] proposed some major improvements of the speed and the security of GGH. In this scheme, the public key is of a Hermite Normal Form (HNF) (see Def. 6). Such a HNF basis is not only compact in storage requirements as will be shown in later sections of this paper, it is also provably the hardest basis to transform to a “good basis” compared to other bases [27]. Furthermore, the HNF of a lattice (and hence the public key of the Micciancio cryptosystem) is unique [30] and can be computed from any lattice basis in polynomial time [31].

Definition 6. *Let $\mathcal{L}$ be an integer full-rank lattice of dimension d and $H \in \mathbb{Z}^{d,d}$ be a basis of $\mathcal{L}$. H is a Hermite Normal Form basis of $\mathcal{L}$ if and only if*

$$\forall 1 \leq i,j \leq d \quad 0 \leq H_{i,j} \begin{cases} = 0 & \textit{if } i < j \\ \geq 1 & \textit{if } i = j \\ < H_{j,j} & \textit{if } i > j \end{cases}$$

The use of such HNF public keys gives space complexity advantages as the resulting public key is much smaller in a data storage sense than those proposed by GGH, as will be shown later in this paper. Rather than Babai’s Round-Off method [29] as suggested by Goldreich et al., Micciancio suggested the use of Babai’s Nearest-Plane method [29] to address the BDD problem. However, implementations of this algorithm are extremely slow, again limiting practicality. It is possible however to adapt this cryptosystem to use Babai’s Round-Off method and in doing so, much faster decryption speeds are seen, provided that the matrix inverse is precomputed. Unfortunately, the storage requirement for this matrix inverse is extremely large and practicality is again limited. We will show later in this paper how we can reduce this storage requirement through a special sub-class of Hermite Normal Forms, without reducing security.

Other cryptosystems in general lattices The use of general lattices seems to preclude any notion of provable security based on hard lattice assumptions, as we are obliged to create a good basis of a particular stucture in order to allow decryption. Other ideas have been proposed however, based on random bases in conditional lattices. In 2000, Fischlin and Seifert [26] proposed an original lattice construction with a good basis with which BDD is easy to perform. In this cryptosystem, the tensor product of the lattice is used to obtain a divide and conquer way to solve the BDD problem.

In 2003, Paeng, Jung and Ha [28] proposed to use a lattice built on polynomial rings. However, in 2007, Han, Kim, and Yeom [32] used a lattice reduction to cryptanalyse this scheme. Their attack recovered the secret key, even in huge dimensions (> 1000) and hence make the PJH scheme unusable.

Imperfect Decryption We note that all existing lattice-based cryptosystems proposed in the literature have some small non-zero probability of decryption error. This small but evaluable probability of decryption error has been shown to lead to a cryptanalysis of several schemes, a successful example of such being Proos' attack on NTRU [33]. Since NTRU is the most practical cryptosystem that is related to lattice theory, NTRU has been studied exhaustively in the literature. For a more elaborate discussion on the decryption error problem in NTRU, we refer the readers to [34].

Although there are some methods to protect cryptosystems against decryption errors [35], these adversely affect the efficiency of the cryptosystems. Based on past attacks on existing lattice cryptosystems it seems to be beneficial in terms of security confidence to construct a lattice-based cryptosystem that is *free* from decryption errors by design. Construction of such a cryptosystem with a zero error-probability a design goal will remove the algorithmic and computational complexity involved in using "tack-on" systems such as that described above.

2 Rotated Nearly-Orthogonal Bases

2.1 Motivation

While the LLL-reduced [36] private bases proposed by Micciancio can be seen to exhibit good orthogonality, the LLL-reduction step is computationally expensive. While work is constantly being done to improve the computational speed of this reduction (see [37–39]), we propose a faster method to produce nearly-orthogonal bases, bypassing the LLL step altogether. The private basis construction proposed by GGH has exceptionally good orthogonality as well as being easy and quick to construct. However, the construction restricts the bases to be oriented along the axes. This is unnecessary as the only requirement for efficient BDD is orthogonality and, as such, we feel that this poses a limitation on the private keyspace which could possibly be exploited in the future.

Moreover, the particular structure of the GGH diagonally-dominant basis allows some attacks. For example, instead of looking for some short vector in $\mathcal{L}(kId+M)$, it is easier to look for some close vector of $(k, 0, \ldots, 0)$ in $\mathcal{L}(kId+M)$. Since the distance is short ($\|M[1]\|_2$), this limits the security of the private basis. To counteract this, we propose a rotation step to be added to the GGH private basis construction phase, which will rotate the lattice probabilistically through a number of planes, greatly increasing the private key-space and alleviating the aforementioned security issue. Since these bases are in general random lattices, they will still need to be structured to allow decryption. While we have designed these bases to be resistant to all known attacks on GGH-style bases in general lattices, several properties still distinguish these rotated bases from general bases, such as the length of the first minima compared to the Gaussian heuristic. Further research into this area needs to be conducted to provide a greater security confidence that these properties are not able to be exploited.

2.2 Construction approaches

An obvious method to construct a randomized orthogonal basis non-aligned with the axes is to use the Q matrix from a QR decomposition of a matrix A having uniformly distributed, random coefficients. Since, however, the R matrix is not needed, yet carries significant information from the original matrix, it is not only wasteful from an information theoretic sense to generate this extra, unnecessary entropy, it also results in a higher computational complexity [40]. Instead, a method for generating a uniformly distributed randomized Q matrix directly is sufficient. Several methods exist to create such Q matrices directly [40].

The butterfly orthogonal matrices used in Fast Fourier Transforms show promise for two reasons. Firstly, butterfly matrices are computationally inexpensive to generate in fixed precision as they are a product of $\frac{n}{2}$ Givens rotations. Secondly, the product of a set of butterfly orthogonal matrices is uniformly distributed if the dimension is a power of 2 [40]. Unfortunately, when the dimension is not a power of 2 however, the product of a set of butterfly orthogonal matrices exhibit significant bias with some zero coefficients. These biases can be mitigated through the generation of multiple sets of butterfly orthogonal matrices combined with intermediary permutation matrices which act to distribute the bias, however, this approach requires n permutated butterfly orthogonal matrices to distribute the bias uniformly.

We can generalize the creation of Q to be a product of some number of independent Givens rotations. To address the shortcoming discussed above, a randomized approach can be taken to the creation of Q and uniformity can be achieved significantly quicker in practice if the dimension is not a power of 2.

2.3 Givens rotations

Definition 7. *Let $i, j \in \mathbb{Z}, i \neq j$ represent two distinct axes of a Hilbert space of finite dimension d. Let $\theta \in \mathbb{R}$ be an angle, $-\pi \leq \theta \leq \pi$. A matrix multiplicative* *transform $G(i, j, \theta)$ is defined as the rotation by θ through the plane defined by the union of the axes i, j, where*

$$\forall p, q \in \mathbb{Z}, 1 \leq p, q \leq d \quad G_{p,q} \begin{cases} = \sin\theta & \textit{if } p = j \textit{ and } q = i \\ = -\sin\theta & \textit{if } p = i \textit{ and } q = j \\ = \cos\theta & \textit{if } p = q = i \textit{ or } j \\ = 1 & \textit{if } p = q \neq i \textit{ or } j \\ = 0 & \textit{otherwise} \end{cases}$$

This definition can be simplifed by describing the rotation transform as an identity matrix with the following exceptions: $G_{i,i} = G_{j,j} = \cos\theta$ and $G_{j,i} = -G_{i,j} = \sin\theta$.

A Givens rotation transform G has two properties that allows the construction of a uniformly distributed orthogonal basis. Firstly, since the product of a basis A with a Givens rotation transform $G(i, j, \theta)$ is geometrically represented as a rotation through an axial plane defined by the axes (i, j), it follows that the multiplication of any basis with a Givens rotation transform will preserve orthogonality. Secondly, since the identity matrix is trivially orthogonal, the product of an identity matrix with a Givens rotation transform will also be orthogonal, therefore the Givens rotation matrix itself is orthogonal.

By creating a transform T being the product of a number of independent Givens matrices, choosing each rotation plane and angle randomly, uniformity can be achieved at little computational cost. To ensure uniformity, enough Givens matrices are multiplied to ensure that every coefficient of the resultant basis T is strictly greater than zero.

To perturb this orthogonal basis and ensure it is integral, the basis is multiplied by some small parameter k and the coefficients are then randomly rounded off to one of the two nearest integers. This can be expressed as $R \in \mathbb{Z}^{n,n}$ such that $R = kT + M$ where $M \in \mathbb{R}^{n,n}$. To simplify, this can be represented as $R = \lfloor kG \rfloor + M'$ with $M' \in [0, 1]^{n,n}$. This is a direct, rotated analog of the original GGH specification, alleviating the security issue described above while still maintaining high orthogonality.

2.4 Decryption Error

Due to attacks on lattice cryptosystems with decryption errors (such as Proos' attack on NTRU [33], for a more elaborate discussion see [34]), our scheme has been designed to avoid decryption error through a choice of parameter k and the plaintext domain as described in Theorem 1.

Theorem 1. *Let $k \in \mathbb{R}^+$, $T \in \mathbb{R}^{n,n}$ a rotation matrix, $M \in \mathbb{R}^{n,n}$ with $|M_{i,j}| \leq 1$ and $R = kT + M$. Then for any vectors $c, p \in \mathbb{R}^n$ with $\|p\|_2 < \frac{k-n}{2}$ and $q \in \mathbb{Z}^n$ if $c = p + qR$ then $q = \lfloor cR^{-1} \rceil$.* [2]

[2] Therefore, p can be computed correctly from c and B using Babai's Round-Off algorithm.

Proof (Theorem 1).
Let's study $c - \lfloor cR^{-1} \rceil R$.

$$\begin{aligned} c - \lfloor cR^{-1} \rceil R &= p + qR - \lfloor (p+qR)R^{-1} \rceil R \\ &= p + qR - \lfloor pR^{-1} + q \rceil R \\ &= p - \lfloor pR^{-1} \rceil R \end{aligned}$$

Let's study $\|pR^{-1}\|_\infty$.

$$\begin{aligned} \|pR^{-1}\|_\infty &\leq \|pR^{-1}\|_2 \\ &\leq \|p\|_2 \|R^{-1}\|_2 \\ &\leq \|p\|_2 \|(kT+M)^{-1}\|_2 \\ &\leq \|p\|_2 \|(kT)^{-1}(Id + MT^{-1}k^{-1})^{-1}\|_2 \\ &\leq \|p\|_2 k^{-1} \|T^{-1}\|_2 \|(Id + MT^{-1}k^{-1})^{-1}\|_2 \end{aligned}$$

Let's study $\|(MT^{-1}k^{-1})\|_2$.

$$\|(MT^{-1}k^{-1})\|_2 \leq \|M\|_2 \|T^{-1}\|_2 k^{-1}$$

T^{-1} is a rotation matrix (unitary matrix) then $\|T^{-1}\|_2 = 1$ ([5], Chapter 2.5.6) and $\|M\|_2 \leq n \max |M_{i,j}| \leq n$ ([5], Chapter 2.3.2, Equation 2.3.8). As by $\frac{k-n}{2} > 0$, we obtain $n^{-1} > k^{-1}$ and

$$\|(MT^{-1}k^{-1})\|_2 < n \times 1 \times n^{-1} = 1.$$

Therefore, as $\|(MT^{-1}k^{-1})\|_2 < 1$, we have $\|(Id + MT^{-1}k^{-1})\|_2 \leq \frac{1}{1 - \|MT^{-1}k^{-1}\|_2}$ ([5], Chapter 2.3.4, Lemma 2.3.3). Therefore, we obtain

$$\|pR^{-1}\|_\infty \leq \frac{\|p\|_2 k^{-1} \|T^{-1}\|_2}{1 - \|MT^{-1}k^{-1}\|_2} \leq \frac{\|p\|_2 \|T^{-1}\|_2}{k - \|M\|_2 \|T^{-1}\|_2}$$

Then, $\|pR^{-1}\|_\infty \leq \frac{\|p\|_2}{k-n} < \frac{1}{2}$.Therefore $\lfloor pR^{-1} \rceil = 0$. □

In all scenarios tested, $k = 2n$ where n is the basis dimension. Given this choice of k, the correct plaintext domain used was $[-\lceil \frac{\sqrt{n}}{2} \rceil + 1, \lceil \frac{\sqrt{n}}{2} \rceil - 1]^n$.

3 Optimal Hermite Normal Forms

3.1 Definition

We define an "Optimal" Hermite Normal Form as a subclass of general Hermite Normal Forms, where only one column (given a row-vector representation) is non-trivial. See Def. 8.

Definition 8. *Let $\mathcal{L}$ be an integer full-rank lattice of dimension d and $H \in \mathbb{Z}^{d,d}$ a basis of $\mathcal{L}$. H is an Optimal Hermite Normal Form basis of $\mathcal{L}$ if and only if H is a HNF basis and $\forall 1 < i \leq d, H_{i,i} = 1$.*

3.2 Motivation

An analysis of the Micciancio cryptosystem reveals that due to the ciphertext being the modulo reduction of the plaintext by the public basis, which itself is of Hermite Normal Form, if this public basis is of a form whereby only one column is non-trivial (i.e. only one diagonal is strictly greater than one), the ciphertext is reduced to a form whereby only one coefficient is non-zero. This allows the ciphertext to be represented as a scalar rather than a vector with no loss of information. This construction was previously defined and used in GGH-based signature scheme by Plantard et. al. [41]. While this construction has benefits in and of itself, such as simplified ciphertext transmission, if the decryption method is changed from Nearest-Plane to Round-Off, the primary benefits come from the associated storage reduction of the private key.

While it is not strictly required to store the inverse of the private key to perform Babai's Round-Off, doing so dramatically reduces the computation costs of the decryption phase and it is assumed that this inversion occurs in the key generation phase. However, it can be seen that the coefficients of the inverse will be significantly larger than those of the private basis itself, if stored as rational, due to the fact that they are not bounded by some arbitrary small number used in the construction. As such, a significant drawback of existing lattice based cryptosystems utilizing such a precomputation step is extremely large storage costs associated with the private key.

If Babai's Round-Off algorithm [29] is used rather than the Nearest-Plane method proposed by Micciancio, the first step of ciphertext decryption is to multiply the ciphertext by the inverse of the private basis before rounding off. Since the ciphertext vector only has one non-zero coefficient, only the corresponding row in the private basis inverse needs to be stored, again without any loss of information. Since the inverse is many times larger than the basis, this reduction in storage size gives a storage requirement decrease approximately linear in the dimension. In addition, since the speed of vector-matrix multiplication is approximately linear in the size of the basis inverse, a dramatic reduction in decryption times is also seen.

3.3 Naive testing

A simple way to find such "optimal" Hermite Normal Form bases is to start with a good basis, reduce to Hermite Normal Form and check for optimality.

We define the following:

- $Pr[optimal]$ – the probability of the Hermite Normal Form of a random *good* basis being of the aforementioned optimal form
- T_{create} – the time to create a good basis
- T_{HNF} – the time to HNF a basis and check it is optimal

As such, it can be seen that the mean time taken to generate an optimal hermite normal form via naive testing is:

$$T_{naive} = \frac{(T_{create} + T_{HNF})}{Pr[optimal]}$$

3.4 Coprimality testing

It can be seen that a square matrix with a prime determinant will produce an optimal HNF if no vectors lie on an axis, as since the HNF matrix is triangular, the determinant is the product of the diagonal. Since the product is prime, only one diagonal can be populated and if no vector lies on an axis, all resultant vectors in the HNF produced must be non-trivial. Furthermore, empirical testing reveals that suboptimal matrices predominantly have low valued diagonals in the non-optimal column, which would not exist had the determinant been coprime with these smaller values. With this in mind, we propose an improved method over the naive method discussed above. An orthogonal private basis is first created and the determinant calculated. If the determinant is coprime with some set $\mathcal{P}$ of n smallest primes, the resultant HNF of this matrix has a much higher probability of being optimal. We specifically define in our implementation that $n = 9$, as the product of the 9 smallest primes is the largest such product that is strictly less than 2^{32} and hence can be represented as an integer on common 32-bit consumer platforms.

We define the following:

- $\mathcal{P} = \{2, 3, 5, 7, 11, 13, 17, 19, 23\}$.
- $Pr[coprime]$ – probability of the determinant of a random good basis being coprime with all elements of the set $\mathcal{P}$.
- T_{check} – the time to calculate the determinant and check coprimality with the set $\mathcal{P}$.

It can be seen that the time to find an optimal basis using this method is the time to create and check coprimality, multiplied by the average number of times this will need to be done in order to achieve determinant coprimality with the set $\mathcal{P}$. The time to calculate the HNF of this basis and check optimality is then added on and the result is multiplied by the average number of times this will need to be done to get optimal, given the co-primality of the determinant:

$$T_{optimized} = \frac{\frac{T_{create}+T_{check}}{Pr[coprime]} + T_{HNF}}{Pr[optimal|coprime]}.$$

3.5 Comparison

We coded these test cases in C++ using Victor Shoup's NTL [42], compiled against GNU MultiPrecision Library (GMP) [43]. These timing values were obtained on a 2.1Ghz Intel Core 2 Duo platform with 4Gb RAM. The probability values were assessed as an average across all test cases as no statistically significant deviation in probabilities was seen between dimensions.

A significant correlation between coprimality and optimality was noted $Pr[optimal|coprime] \approx 0.9991$ The extreme time required to compute the HNF transform matrix compared to checking for coprimality, combined with the probability of a random HNF matrix being of optimal form ($Pr[optimal] \approx 0.435$) contributed to the naive method being more than twice as slow on average as the optimized method even if $Pr[optimal]$ is small, ≈ 0.0724. See Fig. 1

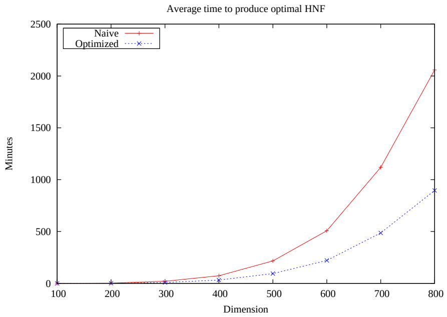

Fig. 1. Average time of optimal HNF basis generation methods

4 Our New Scheme

4.1 Construction

Key setup A key-pair is produced consisting of a rotated nearly-orthogonal private basis R and an optimal public basis B, using the coprimality method above. The basis R is stored and the populated column of B is stored as a vector of length n. The inverse of R is also precomputed and the row corresponding to the populated column of B is stored as a private vector. See Algorithm 1.

Algorithm 1: Key Setup

```
Input  : $n \in \mathbb{Z}$ the dimension
Output: $R, B \in \mathbb{Z}^{n,n}$ such that $\mathcal{L}_R \equiv \mathcal{L}_B$, $R_{INV} \in \mathbb{Q}^n$
begin
    $P[9] \leftarrow \{2, 3, 5, 7, 11, 13, 17, 19, 23\}$
    $T \leftarrow$RotationMatrix($n$), $k \leftarrow 2n$
    repeat
        repeat
            $M \leftarrow \{0, 1\}^{n,n}$
            $R \leftarrow \lfloor kT \rceil + M$
        until isCoprime(determinant(R), P)
        $B \leftarrow$ HNF($R$)
    until isOptimal(B)
    $R_{INV} \leftarrow R^{-1}$
    return $\{B.Col(1)\}$, $\{R, R_{INV}.Row(1)\}$
end
```

Encryption The plaintext vector is reduced modulo the public basis to obtain the ciphertext. As the public basis is of optimal HNF, the ciphertext can be represented as a scalar. See Algorithm 2.

```
Algorithm 2: Encryption
Input  : $p \in \mathbb{Z}^n$ the plaintext vector, $B \in \mathbb{Z}^n$ the public basis first column
Output: $c \in \mathbb{Z}$ the ciphertext scalar
begin
    $c \leftarrow p[1]$
    for $i \leftarrow 2$ to $n$ do  $c \leftarrow c - (p[i] \times B[i])$
    $c \leftarrow c \mod B[1]$
end
```

Decryption To decrypt, the ciphertext scalar is multiplied with the R inverse row vector and the values are rounded to the nearest integer. The result is then multiplied by the R basis and subtracted from the ciphertext vector. Due to the form of the ciphertext vector, this is only performed on the first value of the resultant vector. See Algorithm 3.

```
Algorithm 3: Decryption
Input  : $c \in \mathbb{Z}$ the ciphertext scalar, $R \in \mathbb{Z}^{n,n}$ the private basis, $R_{INV} \in \mathbb{Q}^n$
         the private basis inverse first row
Output: $p \in \mathbb{Z}^n$ the plaintext vector
begin
    $p \leftarrow -\lfloor c \times R_{INV} \rceil \times R$
    $p[1] = p[1] + c$
end
```

4.2 Results

All three cryptosystems were coded in C++ using Victor Shoup's NTL [42] compiled against GNU MultiPrecision Library (GMP) [43]. With runtime speed and numerical correctness design goals in both libraries, we feel that this choice of implementation forms a good basis for comparison. These values were obtained on a 2.1Ghz Intel Core 2 Duo platform with 4Gb RAM. For a practical assessment and approximate comparison of keysizes, all keys were compressed with the bzip2 algorithm. Decompression times were not included in timing results.

A brief overview of the space and time complexity of the cryptosystems discussed is displayed in Table 1. The public and private key sizes of the three cryptosystems discussed were compared for a low dimension 400 and a high dimension 800 (see Table 2). As we are comparing these cryptosystems on both a theoretic and practical basis, we define the private key as also containing redundant, pre-computed private basis inverse. As can be seen, due to the reduced storage requirements of the private basis inverse resulting from the use of Optimal Hermite Normal Forms, our cryptosystem has an extremely small private key in comparison in both cases. Similarly, a great improvement in both encryption and decryption speeds is seen over both the existing cryptosystems discussed (See Fig. 2).[3]

[3] To give an indication of speed, a fast implementation of RSA-2048 as displayed on Bernstein's eBACS page at *http://bench.cr.yp.to/*, is around 5643 kbps/69 kbps for

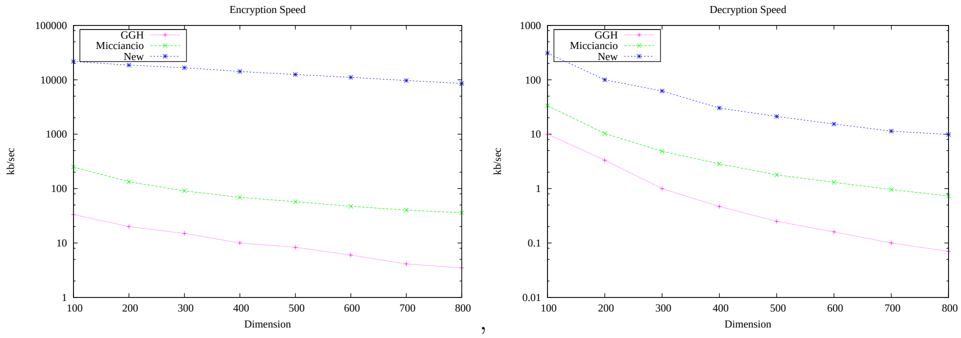

Fig. 2. Encryption and Decryption Results

Table 1. Complexity of Encryption Schemes.

	New	GGH	Mic
Private Key Size	$O(n^2 \log n)$	$O(n^2)$	$O(n^2 \log n)$
Public Key Size	$O(n^2 \log n)$	$O(n^3)$	$O(n^2 \log n^2)$
Message Size	$O(n \log n)$	$O(n)$	$O(n)$
Encrypted Size	$O(n \log n)$	$O(n \log(n))$	$O(n \log(n))$
Encryption Time	$O(n^2 \log n^2)$	$O(n^2 \log n)$	$O(n^2 \log n)$
Decryption Time	$O(n^2 \log n^2)$	$O(n^2 \log n)$	$O(n^5 \log(n)^2$
Decryption Error	No	Yes	Yes

5 Conclusion

We have described a new method for the creation of private bases in GGH-like BDD-based cryptosystems that provides a much larger private key-space without significantly decreasing key generation speed. This new method is based on planar rotations and retains the high orthogonality of the GGH private key at minimal computational expense. Furthermore, the use of these bases addresses a security issue inherent in the original GGH diagonally-dominant bases and is provably without decryption error.

We have also discussed the notion of "optimal" Hermite Normal Forms and the benefits arising from the use of such bases in lattice-based cryptosystems. Furthermore, we have described two methods to find lattices with optimal Hermite Normal Forms and compared these methods to assess suitability for implementing such a cryptosystem.

Finally, we have created a class of lattice based cryptosystems, using the two aforementioned improvements. We have provided a fair comparison between this cryptosystem and the two existing cryptosystems discussed.

encryption and decryption respectively on a similar processor to that used for testing in this paper

Table 2. Average compressed Key-sizes for low and high security.

Dimension	400		800	
	Public Key	Private Key	Public Key	Private Key
GGH	18.7 MB	43.0 MB	153.0 MB	374.4 MB
Micciancio (Round-Off)	241.4 kB	95.4 MB	1.1 MB	861.6 MB
New Scheme	200.8 kB	281.2 kB	885.8 kB	1.1 MB

References

1. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring, IEEE Press (1994) 124–134
2. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: STOC. (1996) 99–108
3. Boas, P.V.E.: Another NP-complete problem and the complexity of computing short vectors in lattices. Technical Report 81-04, U. of Amsterdam (1981)
4. In Bernstein, D.J., Buchmann, J., Dahmen, E., eds.: Post-quantum Cryprography. Springer (2008)
5. Golub, G.H., Loan, C.F.V.: Matrix Computations, Third Edition. The Johns Hopkins University Press (1996)
6. Ajtai, M., Dwork, C.: A public-key cryptosystem with worst-case/average-case equivalence. In: STOC. (1997) 284–293
7. Nguyen, P.Q., Stern, J.: Cryptanalysis of the Ajtai-Dwork cryptosystem. In: Advances in Cryptology - CRYPTO 1998. (1998) 223–242
8. Goldreich, O., Goldwasser, S., Halevi, S.: Eliminating decryption errors in the Ajtai-Dwork cryptosystem. In: CRYPTO. (1997) 105–111
9. Cai, J.Y., Cusick, T.W.: A lattice-based public-key cryptosystem. In: Selected Areas in Cryptography. (1998) 219–233
10. Kawachi, A., Tanaka, K., Xagawa, K.: Multi-bit cryptosystems based on lattice problems. In: Public Key Cryptography. (2007) 315–329
11. Regev, O.: Improved inapproximability of lattice and coding problems with preprocessing. In: IEEE Conference on Computational Complexity. (2003) 363–370
12. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC. (2005) 84–93
13. Ajtai, M.: Representing hard lattices with O(n log n) bits. In: STOC. (2005) 94–103
14. Regev, O.: Lattice cryptography. In: Advances in Cryptology - Crypto 2006, Lecture Notes in Computer Science 4117, Springer-Verlag. (2006) 131–141
15. Micciancio, D.: Generalized compact knapsacks, cyclic lattices and efficient one-way functions. Computational Complexity **16** (2007) 365–411
16. Lyubashevsky, V., Micciancio, D., Peikert, C., Rosen, A.: SWIFFT: A modest proposal for FFT hashing. In: FSE. (2008) 54–72
17. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC, ACM (2008) 197–206
18. Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. In: Advances in Cryptology - Crypto 2008. (2008)
19. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: STOC. (2008) 187–196

20. Lyubashevsky, V., Micciancio, D.: Asymptotically efficient lattice-based digital signatures. In: Theory of cryptography conference - TCC 2008. (2008) 37–54
21. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In: ANTS 1998, LNCS 1423. (1998) 267–288
22. Coppersmith, D., Shamir, A.: Lattice attacks on NTRU. In: EUROCRYPT. (1997) 52–61
23. Goldreich, O., Goldwasser, S., Halevi, S.: Public-key cryptosystems from lattice reduction problems. Electronic Colloquium on Computational Complexity **3** (1996)
24. McEliece, R.J.: A public-key cryptosystem based on algebraic coding theory. Deep Space Network Progress Report **44** (1978) 114–116
25. Nguyen, P.Q.: Cryptanalysis of the Goldreich-Goldwasser-Halevi cryptosystem from Crypto '97. In: Crypto 1999, LNCS 1666, Springer-Verlag. (1999) 288–304
26. Fischlin, R., Seifert, J.P.: Tensor-based trapdoors for CVP and their application to public key cryptography. In: IMA Int. Conf. (1999) 244–257
27. Micciancio, D.: Improving lattice based cryptosystems using the Hermite normal form. In: Cryptography and Lattices Conference (CaLC 2001). (2001) 126–145
28. Paeng, S.H., Jung, B.E., Ha, K.C.: A lattice based public key cryptosystem using polynomial representations. In: Public Key Cryptography. (2003) 292–308
29. Babai, L.: On Lovász' lattice reduction and the nearest lattice point problem. Combinatorica **6** (1986) 1–13
30. Cohen, H.: A course in computational algebraic number theory. Volume 138 of Graduate Texts in Mathematics. Springer-Verlag (1993)
31. Kannan, R., Bachem, A.: Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. SIAM J. of Comp. **8** (1979) 499–507
32. Han, D., Kim, M.H., Yeom, Y.: Cryptanalysis of the Paeng-Jung-Ha cryptosystem from PKC 2003. In: Public Key Cryptography. (2007) 107–117
33. Proos, J.: Imperfect decryption and an attack on the NTRU encryption scheme. IACR ePrint Archive (2003)
34. Howgrave-Graham, N., Nguyen, P.Q., Pointcheval, D., Proos, J., Silverman, J.H., Singer, A., Whyte, W.: The impact of decryption failures on the security of NTRU encryption. In: CRYPTO. (2003) 226–246
35. Dwork, C., Naor, M., Reingold, O.: Immunizing encryption schemes from decryption errors. In: EUROCRYPT. (2004) 342–360
36. Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring polynomials with rational coefficients. Mathematische Annalen, Springer-Verlag **261** (1982) 513–534
37. Nguyen, P.Q., Stehlé, D.: Floating-point LLL revisited. In: Proceedings of Eurocrypt 2005, LNCS 3494, Springer-Verlag. (2005) 215–233
38. Schnorr, C.P.: Fast LLL-type lattice reduction. Information and Computation **204** (2006) 1–25
39. Morel, I., Stehlé, D., Villard, G.: H-LLL: using householder inside LLL. In: ISSAC. (2009) 271–278
40. Genz, A.: Methods for generating random orthogonal matrices. In Niederreiter, H., Spanier, J., eds.: Monte Carlo and Quasi-Monte Carlo Methods. (1999) 199–213
41. Plantard, T., Susilo, W., Win, K.T.: A digital signature scheme based on CVP_∞. In Cramer, R., ed.: Proceedings of PKC08. Volume 4939 of LNCS. (2008) 288–307
42. Shoup, V.: (NTL (Number Theory Library) - http://www.shoup.net/ntl)
43. GMP: (GNU Multiple Precision Arithmetic library - http://gmplib.org/)