On the Bounded Distance Decoding Problem for Lattices Constructed from Polynomials and Their Cryptographic Applications

Zhe Li, San Ling, Chaoping Xing and Sze Ling Yeo *†‡§¶

Abstract

In this paper, we propose new classes of trapdoor functions to solve the bounded distance decoding problem in lattices. Specifically, we construct lattices based on properties of polynomials for which the bounded distance decoding problem is hard to solve unless some trapdoor information is revealed. We thoroughly analyze the security of our proposed functions using state-of-the-art attacks and results on lattice reductions. Finally, we describe how our functions can be used to design quantum-safe encryption schemes with reasonable public key sizes. Our encryption schemes are efficient with respect to key generation, encryption and decryption.

1 Introduction

In today's digital world, protecting the confidentiality and integrity of digital information is of vital importance. At the core of providing data privacy, integrity and authenticity are a class of algorithms called public-key cryptosystems, first introduced by Diffie and Hellman in 1976 [13]. Essentially, these public-key cryptosystems are constructed from trapdoor functions. Recall that a trapdoor function f is a function satisfying:

- $f(x)$ is easy to evaluate for all inputs x;
- Given an output y of the function f, it is computationally infeasible to determine x such that $y = f(x)$ unless some trapdoor information is known.

*The research of San Ling was partially supported by Singapore Ministry of Education under Research Grant MOE2016-T2-2-014(S). The research of Chaoping Xing was supported by RG21/18 of MoE Tier 1 and NTU internal funding M4081575.

†Zhe Li is with School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 637371 (email: lzonline01@gmail.com).

‡San Ling is with School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 637371 (email:lingsan@ntu.edu.sg).

§Chaoping Xing is with School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China, and also with School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 637371 (email:xingcp@ntu.edu.sg).

¶Sze Ling Yeo is with Institute for Infocomm Research (I2R), Singapore 138632 (email:slyeo@i2r.a-star.edu.sg).

To date, most commonly deployed trapdoor functions rely on some computational number theory problems where no efficient classical algorithm is known, including the integer factorization problem and discrete logarithm problem in various finite groups. However, Shor showed in 1994 that there exists a quantum algorithm that can solve these problems in polynomial time [43].

As such, there is an urgent need to design new trapdoor functions based on different mathematical problems that are resistant to quantum algorithms. At present, a number of potential classes of mathematical problems are being considered and studied, namely, from coding theory, lattices, multi-variate polynomials, hash functions and isogenies of supersingular elliptic curves [7]. Among them, lattices seem to be among the most promising, spawning many new constructions with different properties and capabilities, most notably fully homomorphic encryption [17].

1.1 Previous work

The first code-based cryptosystem was proposed by McEliece in 1978 [35] with its security based on the hardness of decoding a random code. It provides efficient encoding and decoding while suffers from the disadvantage of having large public key sizes for standard security levels. The main ingredient of code-based schemes is a class of codes with efficient decoding which are indistinguishable from random codes. Many families of codes have been proposed including Goppa codes [35], algebraic geometric codes [25], Reed-Muller codes [44], Reed Solomon codes [39] and MDPC codes [37]. To date, Goppa codes and MDPC codes are among the few families of codes that possess the desired properties for code-based schemes.

On the other hand, early lattice-based encryption schemes include the Ajtai-Dwork Encryption [1], Goldreich-Goldwasser-Halevi (GGH) encryption [20] and NTRU encryption [23]. A breakthrough of modern lattice-based cryptography is the invention of the learning with error (LWE) and Ring-LWE problems [40, 34]. Consequently several LWE-based encryption schemes have been proposed [8, 9, 18, 21].

The GGH encryption scheme [20] is an analog of the famous code-based encryption scheme– McEliece encryption. However, it was shown in [38] that the structure of the error provided an inherent weakness and together with the embedding technique, this weakness can be exploited to attack the GGH instances. Even though suggestions were put forth in [38] to mend the scheme, the corresponding parameters will make the scheme impractical for use.

Nonetheless, the ideas underpinning the GGH scheme remained interesting and motivated other lattice-based trapdoor functions including those based on RLWE/SIS problems [19]. Besides, it is interesting that the code-based analog McEliece cryptosystem is still secure against quantum computers. As such, one might hope to improve the lattice-based GGH scheme such that it provides the same level of security but is more efficient than the McEliece scheme.

1.2 Our work

In this paper, we seek to design new trapdoor functions in which the function inversion involves solving the Bounded distance decoding problem (BDD), one of the well-known hard lattice problems. Our construction is primarily inspired by the GGH construction [20] and the McEliece code-based cryptosystem [35]. In particular, our scheme can be viewed as the lattice analog of the Goppa-code based McEliece cryptosystem. Like these schemes, our function involves constructing a point that is sufficiently close to a certain point in a lattice determined by the input. Hence, inverting this function will require one to solve BDD for a related lattice.

More precisely, we choose n distinct elements $\alpha_1, \alpha_2, \ldots, \alpha_n$ of $\mathbb{F}_q$ and t distinct monic irreducible polynomials $c_1(x), c_2(x), \ldots, c_t(x)$ of degree d_0 such that $\gcd(\prod_{i=1}^{n}(x-\alpha_i), \prod_{i=1}^{t} c_i(x)) = 1$. An integer point $(a_1, a_2, \ldots, a_n) \in \mathbb{Z}^n$ is a lattice point if and only if $\prod_{i=1}^{n}(x-\alpha_i)^{a_i} \equiv 1 \pmod{c(x)}$, where $c(x) = \prod_{i=1}^{t} c_i(x)$. Then a basis of this lattice can be computed efficiently. We show in this paper that, given q, n and certain range of $d = td_0$, and the basis of this lattice, the embedding technique does not work well to tackle the BDD of this lattice. On the other hand, with information on $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ and the polynomial set $\{c_1(x), c_2(x), \ldots, c_t(x)\}$, we are able to efficiently solve the BDD of this lattice. With this trapdoor, we can design an encryption algorithm that is similar to the one in the GGH encryption scheme. Furthermore, we conduct a thorough security analysis on our trapdoor function.

1.3 Comparison

The main contribution of this paper is to design a new lattice-based trapdoor function whose security relies on the hardness of the BDD problem on lattices. Our construction is analogous to the construction of Goppa codes that admit an efficient decoding based on polynomial operations. Unlike binary Goppa codes, our construction works over finite rings and its error decodability does not depend on the norm of the constructed lattices or the Hamming weight of the associated code.

Similar to the GGH construction which is based on the hardness of the closest vector problem (CVP), our function relies on the infeasibility of the related BDD problem. Nonetheless, we believe that our function provides the following advantages over the GGH construction:

- We do not restrict to a basis that admits a solution via CVP solving algorithms as a trapdoor. Instead, by using lattices constructed from polynomial functions, one can invert the function efficiently as long as one has access to the polynomials and points involved. As such, the vector norms of our basis can be better controlled, one of the main limitations of the GGH scheme.
- For our construction, we do not need to multiply the basis by unimodular matrices to hide the underlying structure.
- Our construction allows us to consider part of the generator matrix as the public key instead of using the entire square matrix, thereby reducing the public key size of our scheme.
- The security of our scheme against embedding attacks can be analyzed concretely.

The Chor-Rivest scheme proposed in [11] can be thought of as a special case of our scheme by considering an irreducible polynomial in our construction. For our scheme in Section 6, we propose to use a product of linear polynomials instead. This has the advantage of making key generation more efficient. Moreover, as discussed in Section 5, such a scheme is more secure.

1.4 Organization of the paper

This paper is organized as follows. In the next section, we briefly summarize some important background on lattices as well as the two encryption schemes (namely, GGH and McEliece schemes) that inspire our work. In Section 3, we describe a family of lattices constructed from polynomials. We then present our new trapdoor functions based on these lattices in Section 4. This is followed by a security analysis on these trapdoor functions in Section 5. In Section 6, we give details of a

semantically secure encryption scheme based on our trapdoor functions. In addition, we propose some possible parameters for our scheme.

2 Preliminaries

2.1 Background on Lattices

In this section, we briefly review some of the important definitions and notions on lattices. We refer the reader to [36, 7] for more background materials.

Let n be a positive integer. By usual convention, we will represent vectors in $\mathbb{R}^n$ in the row form. For $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$, denote by $\|\mathbf{x}\|$ the Euclidean norm of $\mathbf{x}$, that is, $\|\mathbf{x}\| = \sqrt{\sum_{i=1}^{n} x_i^2}$.

Lattice: A lattice L is a discrete additive subgroup of $\mathbb{R}^n$. Concretely, for $m \leq n$, let $\{\mathbf{b}_1, \ldots, \mathbf{b}_m\}$ be m linearly independent vectors in $\mathbb{R}^n$. Then a lattice L is a set $\{a_1\mathbf{b}_1 + a_2\mathbf{b}_2 + \ldots + a_m\mathbf{b}_m : a_i \in \mathbb{Z}, i = 1, 2, \ldots, m\}$. m is called the *dimension* or *rank* of the lattice. If $m = n$, then L is said to have full rank. In this work, we will only focus on full-rank lattices. Further, $B = \{\mathbf{b}_1, \ldots, \mathbf{b}_m\}$ is called a *basis* of L. Let M be the m by n matrix with rows $\mathbf{b}_i$, $i = 1, 2, \ldots, m$. Then, the *determinant* of L (or the volume of L) is given by $\det(L) = \mathrm{vol}(L) = \sqrt{\|MM^T\|}$.

n-Ball: For $r \in \mathbb{R}$, let $B_n(r) = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| \leq r\}$ denote the n-dimensional ball centered around the origin with radius r. The volume of $B_n(r)$ is given by $V_n(r) = \frac{\pi^{n/2} r^n}{\Gamma(n/2+1)}$, where $\Gamma(\cdot)$ is the Gamma function.

Short vectors of a lattice: As a lattice is a discrete subgroup of $\mathbb{R}^n$, the set of all their Euclidean norms forms a discrete subset of $\mathbb{R}$. Hence, each lattice L has a nonzero point such that its norm is the minimum. We denote this minimum norm by $\lambda_1(L)$. More generally, for $i = 1, 2, \ldots, n$, $\lambda_i(L)$ denotes the smallest radius r such that the ball $B_n(r)$ contains i linearly independent points in L.

Gaussian heuristic: The Gaussian Heuristic estimates the number of lattice points in certain sets. Let L and S be a full-rank lattice and a connected n-dimensional subset, respectively. Then the number of lattice points in S is approximated by $\mathrm{vol}(S)/\det(L)$. This leads to the following Gaussian heuristic estimate on the shortest vector $\lambda_1(L)$ for a random lattice L: $\lambda_1(L) \approx V_n(1)^{-1/n}\det(L)^{1/n} \approx \sqrt{n/2\pi e}\det(L)^{1/n}$.

Lattice reduction: Given a basis of a lattice L, one can construct a new basis by multiplying the matrix formed by the basis vectors with unimodular integer matrices, that is, integer matrices with determinant ± 1. In general, one often looks for a basis with short vectors or nearly orthogonal vectors. There are various algorithms to reduce a basis of a lattice into a basis of better quality. Well-known reduction algorithms include the LLL algorithm [30] and the BKZ algorithm [41]. In the BKZ algorithm, one essentially tries to find short vectors in the sub-lattice formed by sub-blocks of basis vectors. In fact, the LLL algorithm can be viewed as a special case of the BKZ algorithm where we work with pairs of vectors each time. Evidently, a BKZ algorithm with a bigger block size produces a basis with shorter vectors but this is achieved at the expense of a longer running time.

In our security analysis, we consider the BKZ-2.0 algorithm [10] with better practical performance. The BKZ-β algorithm attempts to find the shortest vector by repeatedly calling an SVP oracle in a β dimensional sublattice. It was shown that after a polynomial number of calls to the SVP oracle, the quality of the resulting basis will not be much improved [22]. The basis obtained from BKZ-β reduction is called the BKZ-β reduced basis. In the Hermite-factor model, we assume

that the BKZ-β reduced basis contains a vector of length $\|\mathbf{b}_1\| = \delta^n \times Vol(L)^{1/n}$ with

$$\delta = (((\pi\beta)^{1/\beta})/(2\pi e))^{1/(2(\beta-1))}.$$

We model the lengths of the vectors in the BKZ-β reduced basis using the Geometric Series Assumption as follows.

Assumption 2.1 (Geometric Series Assumption[42]). *After lattice reduction, the Gram-Schmidt vectors of the basis satisfy the condition* $\|\mathbf{b}_i^*\| = \alpha^{i-1} \times \|\mathbf{b}_1\|$ *for some* $0 < \alpha < 1$.

Combining the root-Hermite factor $\|\mathbf{b}_1\| = \delta^n \times det(L)^{1/n}$ with the GSA assumption, we get $\alpha = \delta^{-2n/(n-1)}$ and then $\|\mathbf{b}_i^*\| \approx \delta^{n-2i-1} \times det(L)^{1/n}$.

In this work, the cost of BKZ algorithm with blocksize β in an n dimensional lattice will be taken as [2, 3]

$$8n2^{0.292\beta+16.4}. \tag{1}$$

Shortest Vector Problem (SVP): Given a lattice L, the *shortest vector problem* (SVP) seeks a nonzero point $\mathbf{v}$ in L such that $\|\mathbf{v}\| = \lambda_1(L)$. For low dimensions, some proposed approaches to solve SVP include computing the Voronoi cell of the lattice and sieving (see [22] for details) as well as enumeration methods [27, 16]. However, these methods have complexity at best exponential in the lattice dimension n. To date, SVP has been solved in a 155-dimensional lattice (https://www.latticechallenge.org/svp-challenge/index.php). Several variants of SVP have been proposed such as the *unique SVP* (uSVP). Specifically, the unique SVP with factor γ seeks for the shortest vector of a lattice given that the length of the second shortest vector $\lambda_2(L)$ is at least γ times of the length of the shortest vector $\lambda_1(L)$.

Closest Vector Problem (CVP)/Bounded Distance Decoding Problem (BDD): A closely related problem to SVP is the *closest vector problem* (CVP). Given a target vector $\mathbf{t} \in \mathbb{R}^n$, CVP seeks a vector $\mathbf{v} \in L$ that minimizes $\|\mathbf{t}-\mathbf{v}\|$. Further, when $\|\mathbf{t}-\mathbf{v}\| \leq \gamma\lambda_1(L)$ for some $\gamma < 1$, the problem is more commonly known as the bounded distance decoding problem (BDD).

In [27], an embedding technique was introduced that can transform BDD in an n-dimensional lattice to uSVP in an $(n+1)$-dimensional lattice. Essentially, suppose that we have a target vector $\mathbf{t}$ that is close to a lattice generated by a basis $B = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$. We construct another lattice $L' \subset \mathbb{Z}^{n+1}$ generated by the following matrix:

$$B' = \begin{pmatrix} \mathbf{b}_1 & 0 \\ \mathbf{b}_2 & 0 \\ \vdots & \vdots \\ \mathbf{b}_n & 0 \\ \mathbf{t} & u \end{pmatrix}.$$

Here, u is called the embedding factor.

We remark that solving BDD is one of the approaches to solve the learning with errors (LWE) problem. As such, various experiments had been performed on LWE instances via the BDD approach [33, 28]. A good discussion of the various approaches to solve BDD can be found in [4].

2.2 The McEliece Cryptosystem

The McEliece encryption scheme was proposed in [35] as a public-key cryptosystem that is based on hard problems in algebraic coding theory instead of the usual integer factoring problem or discrete

logarithm problems in groups. More specifically, its construction hinges on the difficulty to decode general linear codes over finite fields. Unlike the widely-used integer factoring problem and discrete logarithm problem which had been proven to be vulnerable to polynomial-time quantum algorithms [43], the decoding problem is touted as one of the potential candidates to be used as a basis for post-quantum cryptography.

Essentially, the McEliece encryption scheme generates two different basis of a linear code, one with an easy decoding strategy while the other is presumably difficult to decode. Concretely, let G be an $[n, k, 2t+1]$ binary Goppa code which admits an efficient decoding algorithm. Let U be a $k \times k$ invertible binary matrix and P an $n \times n$ permutation matrix. Let $G' = UGP$. Then G' represents a general linear code with no obvious way to decode. The basic structure of the McEliece encryption scheme can be described as follows:

Public key: The matrix G' and the parameters n, k, t.

Private key: The matrices G, U and P.

Encryption: Let $\mathbf{m}$ be a k-bit message. Randomly pick an n-bit error vector $\mathbf{e}$ with Hamming weight t. The encryption of $\mathbf{m}$ is given by

$$\mathbf{c} = E(\mathbf{m}) = \mathbf{m}G' + \mathbf{e}.$$

Decryption: Let $\mathbf{c}' = \mathbf{c}P^{-1}$. With the secret key G, decode $\mathbf{c}'$ to obtain the message $\mathbf{m}'$. Compute $\mathbf{m} = \mathbf{m}'U^{-1}$.

2.3 The GGH Cryptosystem

The GGH cryptosystem was presented in [20] as a lattice analog of the McEliece cryptosystem. While the McEliece scheme exploits the difficulty to decode the received word obtained from a random code whenever errors of small weights are introduced, the GGH scheme relies on a similar phenomenon on general lattices. Indeed, the GGH scheme constructs two different bases of the same lattice, one of which allows CVP to be solved efficiently via Babai's nearest plane algorithm while the other basis is constructed as a random basis of the lattice and hence, has very poor performance with respect to CVP. More precisely, one first constructs a basis B with short highly orthogonal rows (that is a basis with small orthogonality defect) and then multiply B by random unimodular matrices to obtain a basis B' of the same lattice with much higher orthogonality defect. B is then used as the private key while B' will serve as the public key. The basic structure of the GGH encryption scheme is presented next.

Private key: The basis B and a unimodular matrix U;

Public key: The basis $B' = UB$, and the parameter n and a small positive integer σ;

Encryption: Let the message $\mathbf{m} \in \mathbb{Z}^n$. Choose an error $\mathbf{e} \in \mathbb{Z}^n$ whose entries are randomly picked to be $\pm\sigma$. The encryption of $\mathbf{m}$ is given by

$$\mathbf{c} = E(\mathbf{m}) = \mathbf{m}B' + \mathbf{e}.$$

Decryption: Using Babai's nearest plane algorithm and the basis B, determine the vector $\mathbf{v}$ closest to $\mathbf{c}$. We have $\mathbf{m} = \mathbf{v}U^{-1}$.

Remark 2.2. *We have only outlined the basic McEliece and GGH constructions. These constructions need to be further enhanced to make them semantically secure [20, 29, 24].*

3 Polynomial Lattices

In this section, we give a new construction of lattices via polynomials over a finite field. Let q be a prime power. We denote by $\mathbb{F}_q$ the finite field with q elements. Let $\mathfrak{R}$ denote the polynomial ring $\mathbb{F}_q[x]$. Fix a monic polynomial $c(x) \in \mathfrak{R}$ of degree d and let $\mathfrak{Q}_{c(x)}$ denote the quotient ring $\mathfrak{R}/c(x)$. Let $\mathfrak{Q}^*_{c(x)}$ denote the unit group of $\mathfrak{Q}_{c(x)}$, i.e., let $\mathfrak{Q}^*_{c(x)} = \{\overline{f(x)} \in \mathfrak{Q}_{c(x)} : \gcd(f(x), c(x)) = 1\}$, where $\overline{f(x)}$ denotes the equivalence class of $f(x)$ in $\mathfrak{Q}_{c(x)}$. It is easy to verify that $\mathfrak{Q}^*_{c(x)}$ forms a multiplicative group. Furthermore, the cardinality of $\mathfrak{Q}^*_{c(x)}$, denoted by $\Phi(c(x))$, is given by the following formula.

Lemma 3.1. *[32, Lemma 3.69] Let $c(x) \in \mathfrak{R}$. Recall that $c(x)$ has the canonical factorization*

$$c(x) = \prod_{i=1}^{t} c_i(x)^{e_i},$$

where $c_i(x)$'s are pairwise distinct monic irreducible polynomials over $\mathbb{F}_q$ and d_i's are the degrees of $c_i(x)$ and $e_i \geq 1$. We have

$$\Phi(c(x)) = \prod_{i=1}^{t} (q^{e_i d_i} - q^{(e_i-1)d_i}).$$

Let $\alpha_1, \ldots, \alpha_n$ be n distinct elements in $\mathbb{F}_q$ such that $c(\alpha_i) \neq 0$ for $i = 1, \ldots, n$. Denote by $\mathbf{a}$ the vector $(\alpha_1, \ldots, \alpha_n) \in \mathbb{F}_q^n$.

Define the map

$$\begin{array}{rccccc} \phi_{\mathbf{a}}: & \mathbb{Z}^n & \longrightarrow & \mathbb{F}_q(x) & \longrightarrow & \mathfrak{Q}^*_{c(x)} \\ & (u_1, \ldots, u_n) & \longmapsto & f = \prod_{i=1}^{n} (x - \alpha_i)^{u_i} & \longmapsto & f(x) \mod c(x). \end{array}$$

Observe that $\phi_{\mathbf{a}}$ is a group homomorphism from $\mathbb{Z}^n$ to $\mathfrak{Q}^*_{c(x)}$. Let $\mathcal{L}_{\mathbf{a},c(x)}$ denote the kernel of $\phi_{\mathbf{a}}$. As $\mathcal{L}_{\mathbf{a},c(x)}$ is a subgroup of $\mathbb{Z}^n$, $\mathcal{L}_{\mathbf{a},c(x)}$ is a lattice. The following lemma provides some important properties of $\mathcal{L}_{\mathbf{a},c(x)}$.

Lemma 3.2. *The lattice $\mathcal{L}_{\mathbf{a},c(x)}$ defined above satisfies the following properties:*

(i) *$\mathcal{L}_{\mathbf{a},c(x)}$ has rank n.*

(ii) *The determinant $\det(\mathcal{L}_{\mathbf{a},c(x)})$ is upper bounded by $\Phi(c(x))$. Furthermore, $\det(\mathcal{L}_{\mathbf{a},c(x)}) = \Phi(c(x))$ if $\phi_{\mathbf{a}}$ is surjective.*

(iii) *$\lambda_1(\mathcal{L}_{\mathbf{a},c(x)}) \geq \sqrt{d}$. Moreover, $\lambda_1(\mathcal{L}_{\mathbf{a},c(x)}) = \sqrt{d}$ if and only if there exists a lattice point $\mathbf{v}$ of the form $\mathbf{v} = \pm\mathbf{v}_0$, where $\mathbf{v}_0$ has d nonzero entries which are all equal to 1.*

Proof. (i) Observe that for each $i = 1, 2, \ldots, n$, we have $(0, 0, \Phi(c(x)), \ldots, 0) \mapsto (x - \alpha_i)^{\Phi(c(x))} \mapsto 1$ under $\phi_{\mathbf{a}}$. Hence, each of these points is in $\mathcal{L}_{\mathbf{a},c(x)}$. As these n points are clearly linearly independent, they form a sub-lattice of $\mathcal{L}_{\mathbf{a},c(x)}$ of rank n. Consequently, $\mathcal{L}_{\mathbf{a},c(x)}$ has rank n.

(ii) As $\mathbb{Z}^n/\mathcal{L}_{\mathbf{a},c(x)} \simeq \mathrm{Im}(\phi_{\mathbf{a}}) \leq \mathfrak{Q}^*_{c(x)}$ and $\det(\mathcal{L}_{\mathbf{a},c(x)}) = [\mathbb{Z}^n : \mathcal{L}_{\mathbf{a},c(x)}]\det(\mathbb{Z}^n) = [\mathbb{Z}^n : \mathcal{L}_{\mathbf{a},c(x)}] = |\mathrm{Im}(\phi_{\mathbf{a}})|$, we obtain the desired inequality. In addition, if $\phi_{\mathbf{a}}$ is surjective, then $\mathbb{Z}^n/\mathcal{L}_{\mathbf{a},c(x)} \simeq \mathfrak{Q}^*_{c(x)}$. Hence the equality follows.

(iii) Let $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ be a nonzero point in $\mathcal{L}_{\mathbf{a},c(x)}$. Denote by I and J the sets $\{1 \leq i \leq n : \ v_i > 0\}$ and $\{1 \leq j \leq n : \ v_j < 0\}$, respectively. By definition of $\mathcal{L}_{\mathbf{a},c(x)}$, we have $\prod_{i\in I}(x-\alpha_i)^{v_i}\prod_{j\in J}(x-\alpha_j)^{v_j} - 1 \equiv 0 \mod c(x)$, i.e., the nonzero polynomial $\prod_{i\in I}(x-\alpha_i)^{v_i} - \prod_{j\in J}(x-\alpha_j)^{-v_j}$ is divisible by $c(x)$. Hence, $\deg(\prod_{i\in I}(x-\alpha_i)^{v_i} - \prod_{j\in J}(x-\alpha_j)^{-v_j}) \geq d$, i.e., $\sum_{i\in I} v_i \geq d$ or $\sum_{j\in J} -v_j \geq d$. This gives $\sum_{i=1}^n |v_i| \geq d$. Therefore, $||\mathbf{v}|| = \sqrt{\sum_{i=1}^n v_i^2} \geq \sqrt{\sum_{i=1}^n |v_i|} \geq \sqrt{d}$ (note that each v_i is an integer). If there exists a lattice point with d nonzero entries which are either all 1 or -1, then it is clear that $\lambda_1(\mathcal{L}_{\mathbf{a},c(x)}) = \sqrt{d}$. Conversely, assume $\lambda_1(\mathcal{L}_{\mathbf{a},c(x)}) = \sqrt{d}$. Then there exists a nonzero lattice point $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ in $\mathcal{L}_{\mathbf{a},c(x)}$ such that $||\mathbf{v}|| = \sqrt{\sum_{i=1}^n v_i^2} = \sqrt{d}$. Since $\deg(\prod_{i\in I}(x-\alpha_i)^{v_i} - \prod_{j\in J}(x-\alpha_j)^{-v_j}) \geq d$, we must have that either $I = \emptyset$ & $\sum_{j\in J} -v_j = d$ or $J = \emptyset$ & $\sum_{i\in I} v_i = d$. This forces that either $v_i = 1$ for all $i \in I$ or $v_j = -1$ for all $j \in J$.

□

According to Lemma 3.2 (iii), we see that $\mathcal{L}_{\mathbf{a},c(x)}$ has minimum norm $\lambda_1(\mathcal{L}_{\mathbf{a},c(x)}) = \sqrt{d}$ when there exist $i_1, \ldots, i_d \in [n]$ such that $\prod_{j=1}^d (x - \alpha_{i_j}) = 1 + c(x)$ or $\prod_{j=1}^d (x - \alpha_{i_j})^{-1} = 1 + c(x)$. It follows that there are at most $2\binom{n}{d}$ different $c(x) \in \mathfrak{R}$ of degree d out of a total of q^d such polynomials such that $\mathcal{L}_{\mathbf{a},c(x)}$ satisfies $\lambda_1(\mathcal{L}_{\mathbf{a},c(x)}) = \sqrt{d}$. In other words, given a polynomial $c(x)$ of degree d and an $(\alpha_1, \ldots, \alpha_n) \subset \mathbb{F}_q^n$, the probability that the lattice $\mathcal{L}_{\mathbf{a},c(x)}$ has minimum norm $\sqrt{d}$ is less than $2\frac{\binom{n}{d}}{q^d} < 1/d!$ and we can expect the minimum norm of the lattice $\mathcal{L}_{\mathbf{a},c(x)}$ to be bigger. In particular, we will use the Gaussian heuristic to estimate the minimum norm of the lattices. Assume that the map $\phi_{\mathbf{a}}$ is surjective. By Lemma 3.2 (ii) and Lemma 3.1, the determinant of $\mathcal{L}_{\mathbf{a},c(x)}$ is approximately q^d. The Gaussian heuristic suggests that a random lattice of dimension n and determinant q^d has minimum norm approximately $\sqrt{n/2\pi e}q^{d/n}$.

Next, we describe how to construct the set $\mathbf{a} = (\alpha_1, \ldots, \alpha_n) \subset \mathbb{F}_q^n$ for which $\mathcal{L}_{\mathbf{a},c(x)}$ admits a nice basis for a class of $c(x)$. In the following, randomness will always refer to picking an element randomly from a uniform distribution. Further, we assume that $c(x)$ is of the form $c(x) = c_1(x)\ldots c_t(x)$, where $c_i(x)$'s are distinct irreducible polynomials over $\mathbb{F}_q$, each having degree d_0. Hence, $\mathfrak{Q}_{c(x)} \cong \oplus_{i=1}^t \mathbb{F}_{q^{d_0}}$. Let β denote a generator of $(\mathbb{F}_{q^{d_0}})^*$.

Let $\alpha_{n-t+1}, \ldots, \alpha_n$ be t distinct elements in $\mathbb{F}_q$. For $i = 1, \ldots, t$ and $j = 1, \ldots, t$, let $\gamma_{ij} = x - \alpha_{n-t+i} \mod c_j(x)$. Let $m_{ij} = \log_\beta \gamma_{ij}$ and $M = (m_{ij})_{i=1,\ldots,t, j=1,\ldots,t}$.

Suppose that M is invertible over the ring $\mathbb{Z}_{q^{d_0}-1}$. For each $\alpha \in \mathbb{F}_q$ with $\alpha \neq \alpha_{n-t+1}, \ldots, \alpha_n$, let $\mathbf{y} = (y_1, \ldots, y_t)$, where $y_j = \log_\beta((x - \alpha) \mod c_j(x))$, $j = 1, \ldots, t$. Let $\mathbf{g}_\alpha = \mathbf{y}M^{-1} \mod q^{d_0} - 1$. Write $\mathbf{g}_\alpha = (g_{\alpha,n-t+1}, \ldots, g_{\alpha,n})$. Note that for each $j = 1, \ldots, t$,

$$y_j = \sum_{i=1}^t g_{\alpha,n-t+i} m_{ij} \mod q^{d_0} - 1. \tag{2}$$

For $j = 1, \ldots, t$, we have

$$\prod_{i=1}^{t}(x-\alpha_{n-t+i})^{g_{\alpha,n-t+i}} \mod c_j(x) \equiv \prod_{i=1}^{t}({\gamma_{ij}}^{m_{ij}})^{g_{\alpha,n-t+i}} \mod c_j(x)$$
$$\equiv \prod_{i=1}^{t}(\beta^{m_{ij}})^{g_{\alpha,n-t+i}} \mod c_j(x)$$
$$\equiv \beta^{\sum_{i=1}^{t} g_{\alpha,n-t+i}m_{ij}} \mod c_j(x)$$
$$\equiv \beta^{y_j} \mod c_j(x) \equiv x-\alpha \mod c_j(x).$$

Note that the first equality follows from the definition of γ_{ij}, the second follows from the definition of the entry m_{ij} of the matrix M, the fourth is derived from the relation of $\mathbf{y}$ and $\mathbf{g}_\alpha$ as shown in 2, and the last follows from the definition of y_j. Since it holds for any $c_j(x)$, it follows that $x-\alpha \equiv \prod_{i=1}^{t}(x-\alpha_{n-t+i})^{g_{\alpha,n-t+i}} \mod c(x)$. Consequently, the point $(0,\ldots,1,0,\ldots,-g_{\alpha,n-t+1},\ldots,-g_{\alpha,n})$, where 1 is in the entry indexed by α is a point in $\mathcal{L}_{\mathbf{a},c(x)}$ for any $\alpha \in (\alpha_1,\ldots,\alpha_{n-t})$, which follows the definition of $\mathcal{L}_{\mathbf{a},c(x)}$ and the map $\phi_{\mathbf{a}}$ in Lemma 3.1.

Proposition 3.3. *Let $\alpha_{n-t+1},\ldots,\alpha_n$ be t distinct elements in $\mathbb{F}_q$ with the matrix M as above. Suppose that M is invertible over $\mathbb{Z}_{q^{d_0}-1}$. Pick $\alpha_1,\ldots,\alpha_{n-t}$ randomly from $\mathbb{F}_q$ such that $\mathbf{a}=(\alpha_1,\ldots,\alpha_n)$ contains n distinct elements. Define G as the $(n-t)\times t$ matrix with rows given by $\mathbf{g}_{\alpha_i}$, for $i=1,\ldots,n-t$. A basis of the lattice $\mathcal{L}_{\mathbf{a},c(x)}$ is given by:*

$$B_{\mathbf{a},c(x)} = \begin{pmatrix} I_{n-t} & -G \\ 0_{t\times(n-t)} & (q^{d_0}-1)I_t \end{pmatrix},$$

where I_r denotes the identity matrix of rank r.

Proof. According to the preceding arguments, the first $n-t$ rows of $B_{\mathbf{a},c(x)}$ are points in $\mathcal{L}_{\mathbf{a},c(x)}$. Since $(x-\alpha_i)^{q^{d_0}-1} \equiv 1 \mod c(x)$ for $i=n-t+1,\ldots,n$, the last t rows of $B_{\mathbf{a},c(x)}$ are also in $\mathcal{L}_{\mathbf{a},c(x)}$. Clearly, the rows of the matrix are linearly independent. It remains to show that the rows span $\mathcal{L}_{\mathbf{a},c(x)}$. Let $\mathbf{u}=(u_1,\ldots,u_n)$ be a point in $\mathcal{L}_{\mathbf{a},c(x)}$ so that $\prod_{i=1}^{n}(x-\alpha_i)^{u_i} \equiv 1 \mod c(x)$. Consider the point $\mathbf{v}=\mathbf{u}-\sum_{i=1}^{n-t}u_iB_i$, where B_i denotes the i-th row of $B_{\mathbf{a},c(x)}$. Hence, $\mathbf{v}\in\mathcal{L}_{\mathbf{a},c(x)}$ and we can write $\mathbf{v}=(0,\ldots,0,v_{n-t+1},\ldots,v_n)$. It is sufficient to show that $v_{n-t+i}\equiv 0 \mod q^{d_0}-1$ for $i=1,\ldots,t$. In other words, $\prod_{i=1}^{t}(x-\alpha_{n-t+i})^{v_{n-t+i}} \equiv 1 \mod c(x)$, equivalently, $\prod_{i=1}^{t}(x-\alpha_{n-t+i})^{v_{n-t+i}} \equiv 1 \mod c_j(x)$ for $j=1,\ldots,t$. Now,

$$\prod_{i=1}^{t}(x-\alpha_{n-t+i})^{v_{n-t+i}} \equiv \prod_{i=1}^{t}(\beta^{m_{ij}})^{v_{n-t+i}} \equiv \beta^{\sum_{i=1}^{t} v_{n-t+i}m_{ij}} \equiv 1 \mod c_j(x)$$

which gives $(v_{n-t+1},\ldots,v_n)M=0 \mod q^{d_0}-1$. Since M is invertible, we conclude that $v_{n-t+i}\equiv 0 \mod q^{d_0}-1$ for $i=1,\ldots,t$. □

Remark 3.4. *Note that the lattices $\mathcal{L}_{\mathbf{a},c(x)}$ for different pairs of $\mathbf{a}$ and $c(x)$ are not all distinct. For instance, let $\mathbf{a}=(\alpha_1,\ldots,\alpha_n)$ and let $\gamma\neq 0\in\mathbb{F}_q$. Let $\mathbf{a}'=(\alpha_1+\gamma,\ldots,\alpha_n+\gamma)$ and $c'(x)=c(x-\gamma)$. Then, it is easy to check that $\mathcal{L}_{\mathbf{a},c(x)}=\mathcal{L}_{\mathbf{a}',c'(x)}$.*

Next, we analyze the complexity of constructing $B_{\mathbf{a},c(x)}$. First, one needs to compute about tn discrete logs in the field $\mathbb{F}_{q^{d_0}}$. The discrete logarithm problem over finite fields is one of the fundamental hard problems widely used in cryptography. Extensive studies have been done in this area and various methods have been proposed to solve the discrete logarithm problem over finite fields. In particular, it is adequate for us to employ Pollard's rho method to compute discrete logarithm with time complexity $O(\sqrt{q^{d_0}})$. Please refer to the survey paper [26] for the state-of-the-art results on the discrete logarithm problem. For our construction, we have $r = q^{d_0}$. For $q = O(n)$ and $d_0 = O(1)$, it follows that solving the discrete log is efficient.

Remark 3.5. *To achieve a more practical scheme, we suggest that the user chooses d_0 as a small constant. Otherwise, computing the discrete logarithm in the field $\mathbb{F}_{q^{d_0}}$ may not be so efficient. For our concrete construction in Section 6, we choose $d_0 = 1$.*

Second, one needs to pick $\alpha_{n-t+1}, \ldots, \alpha_n$ so that the matrix M is invertible. Now, each entry m_{ij} is the discrete log of $x - \alpha_{n-t+i} \mod c_j(x)$. Since α_{n-t+i} is random, we may assume that the matrix M is a random matrix in the ring $\mathbb{Z}_{q^{d_0}-1}$.

Lemma 3.6. *[38, Theorem 2] Let $s = q^{d_0} - 1$ be a positive integer. Let $p_1, \ldots, p_m$ be the distinct prime divisors of s. The probability that a random $t \times t$ matrix in $\mathbb{Z}_s$ is invertible is*

$$P_s = \prod_{i=1}^{m} \prod_{j=1}^{t} (1 - p_i^{-j}).$$

It can be seen from the formula that the probability of a random $t \times t$ matrix being invertible is non-negligible.

Remark 3.7. *One sees that our construction bears some similarities to the Goppa code construction. Recall that the binary Goppa code with Goppa support $\mathbf{a}$ and Goppa polynomial $c(x)$ consists of $(c_1, \ldots, c_n) \in \mathbb{F}_2^n$ such that*

$$\sum_{i=1}^{n} \frac{c_i}{x - \mathbf{a}_i} \equiv 0 \mod c(x).$$

Our construction in this paper is defined over the finite ring $\mathbb{Z}_{q^{d_0}-1}$ instead of over finite fields. As seen in Theorem 4.2, the error decodability of our construction is upto $\deg(c(x)) - 1$, independent of the shortest length of the lattice or the Hamming weight of the corresponding code.

Remark 3.8. *Note that distinct degrees of $c_i(x)'s$ are still possible. If so, the discrete logarithm should be computed in different fields when computing the matrix M, which also makes the notation too complex there. For simplicity, we just assume that all the $c_i(x)$'s are of the same degree. In particular, in our concrete construction, we let the degree of every $c_i(x)$ be 1.*

4 Construction of Our Trapdoor Functions

In this section, we describe new trapdoor functions where inverting the function amounts to solving the BDD for the associated lattices. Unlike the GGH construction, we do not generate two different bases of a lattice. Instead, we require only one basis of our polynomial lattice as the trapdoor involves information to construct the polynomial lattice. Recall that a trapdoor function

encompasses four different sub-algorithms, namely, *generate*, *sample*, *evaluate* and *invert*. We will now present each of these in detail.

Generate: Set the public parameters q, n, d according to the desired security level (see the next section for details) such that $\sqrt{n/2\pi e}q^{d/n} > 2\sqrt{d-1}$. Let $d = d_0 t$. Choose t irreducible polynomials $c_i(x)$ of degree d_0 and let $c(x) = c_1(x)\dots c_t(x)$. Choose a set $\mathbf{a} = (\alpha_1, \dots, \alpha_n) \subset \mathbb{F}_q^n$ such that $c(\alpha_i) \neq 0$ for $i = 1, 2, \dots, n$ and the elements $\alpha_{n-t+1}, \dots, \alpha_n$ satisfy the conditions in Proposition 3.3. Construct the basis $B_{\mathbf{a},c(x)}$ of the lattice $\mathcal{L}_{\mathbf{a},c(x)}$ as described in Proposition 3.3. Write $B_{\mathbf{a},c(x)} = \begin{pmatrix} I_{n-t} & -G \\ 0_{t\times(n-t)} & (q^{d_0}-1)I_t \end{pmatrix}$.

Let $H = B'_{\mathbf{a},c(x)} = \begin{pmatrix} I_{n-t}, & -G \end{pmatrix}$. The trapdoor for our function includes the polynomial $c(x)$ and the set $\mathbf{a} = (\alpha_1, \dots, \alpha_n)$.

Sample: Randomly sample $\mathbf{m} \in \mathbb{Z}_{q^{d_0}-1}^{n-t}$ and the error $\mathbf{e} = (e_1, \dots, e_n) \in \{0,1\}^n$ satisfying: $\sum_{i=1}^n e_i = d - 1$.

Evaluate: For each input $\mathbf{m} \in \mathbb{Z}_{q^{d_0}-1}^{n-t}$, the function f is evaluated on $\mathbf{m}$ as

$$\mathbf{c} = f(\mathbf{m}, \mathbf{e}) = \mathbf{m}H + \mathbf{e} \mod q^{d_0} - 1.$$

Invert: Suppose that we are given a valid output $\mathbf{c} = (c_1, \dots, c_n)$ of the function f. The inversion process is as follows.

Step 1: Compute

$$r(x) = \prod_{i=1}^{n} (x - \alpha_i)^{c_i} \mod c(x).$$

Step 2: Factorize $r(x)$ over the ring $\mathbb{F}_q[x]$ as $r(x) = \prod_{i=1}^n (x-\alpha_i)^{u_i}$. Let $\mathbf{u} = (u_1, u_2, \dots, u_n)$.

Step 3: Compute $\mathbf{v}' = \mathbf{c} - \mathbf{u}$. Write $\mathbf{v}' = (v'_1, \dots, v'_n)$.

Step 4: Let $\mathbf{m}' = (v'_1, \dots, v'_{n-t})$.

Without knowledge of the trapdoor, observe that inverting the function will require us to find the error $\mathbf{e}$ or equivalently, a point $\mathbf{t}$ in $\mathcal{L}_{\mathbf{a},c(x)}$ such that $\|\mathbf{c} - \mathbf{t}\| = \sqrt{d-1}$. Concretely, one will use the basis formed by the rows of the matrix $\begin{pmatrix} H \\ 0 \quad (q^{d_0}-1)I_t \end{pmatrix}$

Thus, one needs to be able to solve BDD with respect to this basis. We will discuss more about this in the next section.

Remark 4.1. *Here, we have chosen the parameters such that the expected shortest vector in the lattice $\mathcal{L}_{\mathbf{a},c(x)}$ is greater than $2\sqrt{d-1}$ so we are in fact solving BDD with $\gamma < 1/2$. Thus, a solution of the BDD will yield the error vector.*

The following theorem shows that the inversion process indeed recovers $\mathbf{m}$.

Theorem 4.2. *Let $\mathbf{m}$ be a random element in $\mathbb{Z}_{q^{d_0}-1}^n$ and let $\mathbf{c}$ be the output produced by the* **Evaluate** *algorithm. Let $\mathbf{m}'$ be the output of the* **Invert** *algorithm. Then $\mathbf{m}' = \mathbf{m}$.*

Proof. First, we have $\mathbf{c} = \mathbf{m}B'_{\mathbf{a},c(x)} + \mathbf{e}$. We claim that $\mathbf{v} = \mathbf{v}'$, where $\mathbf{v} = \mathbf{m}B'_{\mathbf{a},c(x)}$. To see this, note that

$$\begin{aligned}\prod_{i=1}^{n}(x-\alpha_i)^{c_i} =& \prod_{i=1}^{n}(x-\alpha_i)^{v_i+e_i} = \prod_{i=1}^{n}(x-\alpha_i)^{v_i}\prod_{i=1}^{n}(x-\alpha_i)^{e_i} \\ \equiv& 1\cdot\prod_{i=1}^{n}(x-\alpha_i)^{e_i} \mod c(x) \equiv \prod_{i=1}^{n}(x-\alpha_i)^{e_i} \mod c(x)\end{aligned}$$

Since $\sum_{i=1}^{n} e_i = d-1 < d$, we must have $r(x) = \prod_{i=1}^{n}(x-\alpha_i)^{e_i}$, so $u_i = e_i$. Therefore, $\mathbf{v}' = \mathbf{c} - \mathbf{u} = \mathbf{m}B'_{\mathbf{a},c(x)} + \mathbf{e} - \mathbf{u} = \mathbf{v}$ and the claim is proved.

Note that we have $\mathbf{v} = \mathbf{m}B'_{\mathbf{a},c(x)} = (\mathbf{m}, -\mathbf{m}G)$. Therefore, we have $(v_1, \ldots, v_{n-t}) = \mathbf{m} \mod q^{d_0} - 1$. □

Remark 4.3.

- *In general, we like to have as many nonzero entries of the error as possible. Hence, we choose e_i to take small values. In particular, we typically let $e_i = 1$.*
- *Instead of letting all the error entries be positive, we can equivalently let them be all negative. In this case, in the inversion process, one needs to check if $r(x)$ or $1/r(x) \mod c(x)$ can be factorized. In the former case, we have the usual case where $e_i \geq 0$. In the latter case, it is easy to verify that we have $e_i \leq 0$, that is all the nonzero entries of the error are -1.*
- *For the inversion process, one can simply check if $r(\alpha_i) = 0$ to check if $u_i = 0$ or 1.*

Remark 4.4.

- *Here, only the right part $-G$ of the matrix $H = \begin{pmatrix} I_{n-t}, & -G \end{pmatrix}$, which is used to evaluate the function, is undetermined. It is an $(n-t) \times t$ matrix over $\mathbb{Z}_{q^{d_0}-1}$ and thus, has size $(n-t)td_0 \log_2 q$ bits.*
- *Unlike the GGH scheme, inversion does not require solving the CVP. Instead, inversion is carried out using properties of polynomials and remainders.*
- *In the above scheme, the first $n-t$ positions of $\mathbf{c}$ may contain some information about $\mathbf{m}$. This is because we have only introduced error to $d-1$ positions, and thus, at least $n-t-d+1$ positions will be in the clear. In Section 6, we present a practical encoding scheme to mask the original message m.*
- *Apart from the above scheme, other modifications are possible. Randomly pick an $(n-t) \times (n-t)$ unimodular matrix T with small entries and an $n \times n$ permutation matrix P. Construct $H = TB'_{\mathbf{a},c(x)}P \mod q^{d_0} - 1$. The left part of the new matrix H will hide all the information of message m. This is analogous to the classic McEliece scheme where two secret matrices are used to hide the structure of the underlying code used. However, in situations where the inputs are completely random, the roles of T and P will not be so critical. Indeed, multiplying by a permutation matrix P serves to permute the elements in the set $\mathbf{a}$, which can be achieved if the elements in $\mathbf{a}$ are chosen randomly. Similarly, the unimodular matrix T will not improve the security of the scheme as given a matrix H, one can efficiently transform it such that the left portion is a scalar matrix.*

5 Security Analysis of Our Trapdoor Functions

In deciding the parameters for our scheme, we will like to achieve the following:

- The public key size should be reasonably small;
- Key generation, encryption and decryption should be efficient;
- The scheme is resistant against all existing attacks.

We now discuss some possible attacks on our scheme to help us decide the appropriate parameters. First, suppose that $d \geq n/2 + 1$. Let $\mathbf{c}$ be a valid output with error $\mathbf{e}$. Then, $\mathbf{e}$ has $d-1$ entries $= 1$. Consider $\mathbf{c}' = \mathbf{c} - (1, \ldots, 1)$. It is $\mathbf{c}' = \mathbf{m}H + \mathbf{e} - (1, \ldots, 1) = \mathbf{m}H + \mathbf{e}'$, where $\mathbf{e}'$ has $< n/2$ entries $= -1$. Hence, one may decrypt using $\mathbf{c}'$ instead. (refer to Remark 4.3 on how one can decrypt when the entries are negative.) It follows that we may assume that $d \leq n/2$. Therefore, we have $t \leq d \leq n/2$ and $n \leq q$.

5.1 Error search

At first glance, it appears that one needs to search through all $\binom{n}{d-1}$ entries to find the error. However, one can in fact reduce the search in the following way. It is obvious that the first $n-t$ columns of H are linearly independent. Let $I = \{1, \ldots, n-t\}$. Search through all possible error positions in I. Recall that there are at most $d-1$ such positions. Assuming that the error bits are uniformly distributed, the number of nonzero error bits in these positions is roughly $l = \frac{(n-t)(d-1)}{t}$. Consequently, the number of tries is around $\binom{n-t}{l}$.

Let $\mathbf{c_I}$ denote the vector formed by the entries in $\mathbf{c}$ indexed by I. For each guess $\mathbf{e_I}$, define $\mathbf{x} = \mathbf{c_I} - \mathbf{e_I} \mod q^{d_0} - 1$. To verify if our guess is correct, check if $\mathbf{c} - \mathbf{x}H$ is of the correct error form, that is, contains exactly $d-1$ 1's and all other entries are 0.

Consequently, the complexity of this attack is $O(\binom{n-t}{l}(n-t)t)$.

5.2 Search for the trapdoor

One obvious way to attack the function is to find the trapdoor information. We will need to search for $c(x)$ and $\mathbf{a} = (\alpha_1, \ldots, \alpha_n)$. One way to do this is as follows:

- Exhaustively search for the polynomial $c(x)$. There are $O(q^d)$ different $c(x)$ of degree d of the form $c(x) = c_1(x) \ldots c_t(x)$.
- For each $c(x)$, guess the ordered set $(\alpha_{n-t+1}, \ldots, \alpha_n)$. For each such set, determine if there exist $\alpha_1, \ldots, \alpha_{n-t}$ that satisfy the matrix $B'_{\mathbf{a},c(x)} = \begin{pmatrix} I_{n-t}, & -G \end{pmatrix}$. Let $-G = (b_{i,n-t+j})_{i=1,\ldots,t, j=1,\ldots,t}$. Specifically, from the definition of $\mathcal{L}_{\mathbf{a},c(x)}$ and $B'_{\mathbf{a},c(x)}$, we can construct α_i by checking for α_i such that $(x - \alpha_i) \times \prod_{j=1}^{t} (x - \alpha_{n-t+j})^{b_{i,n-t+j}} \equiv 1 \pmod{c(x)}$ for each $i \in \{1, \ldots, n-t\}$. Our guesses of $c(x)$ and α_i's are correct if we can reconstruct $(\alpha_1, \ldots, \alpha_{n-t})$ by the preceding procedure. There are ${}^{n-t}P_n \approx n^{n-t}$ possible ordered sets $(\alpha_{n-t+1}, \ldots, \alpha_n)$.

The overall complexity of this attack is $O(q^d n^{n-t}(n-t)d^2(\log q)^2)$ if we assume that the complexity of polynomial multiplication in the ring $\mathbb{F}_q[x]/c(x)$ is $O(d^2(\log q)^2)$.

Remark 5.1. *In Remark 3.4, we have discussed that for each $c(x)$ and ordered set $\mathbf{a}$, there are at least $q-1$ different choices of pairs of polynomials and ordered sets that give rise to the same lattice. Hence, for a more precise complexity, one will need to take into account these equivalent lattices. However, as this number is relatively small and has a negligible impact on the complexity, we will omit these details in our complexity analysis.*

5.3 Embedding attack to solve BDD

As mentioned in the preceding section, one way to recover the message is to solve the BDD problem on the lattice given the output. A common way to solve BDD is via the embedding technique (see Section 2.1). We now investigate the cost of this attack.

There are two typical estimates for the costs of the embedding attack to recover the error vector when employing BKZ to solve the uSVP. The first was proposed in [15] and the second was sketched in [5]. The predicted costs of the two estimates are quite different. [3] investigated these two estimates and suggested that the latter estimate is more indicative of the cost of the attack. As such, we will use the second estimate in our analysis of the embedding attack.

Briefly, this estimate forecasts that the the attack outputs the correct error $\mathbf{e}$ if

$$\sqrt{\beta/n}\|(\mathbf{e},1)\| \leq \delta^{2\beta-n} det(L)^{1/n} \tag{3}$$

under the assumption that GSA holds, where n is the dimension of the full rank lattice L, β is the block size when running the BKZ reduction. A concise explanation of the condition is that the projection of the error vector $\mathbf{e}$ onto the last β vectors is shorter than the $b^*_{n-\beta+1}$ under the GSA assumption. Then an SVP oracle call on the last block of size β can find the error vector $\mathbf{e}$.

It follows that in our situation, we will require β to satisfy

$$\sqrt{\beta/n}\|(\mathbf{e},1)\| \geq \delta^{2\beta-n}(q-1)^{d/n}.$$

To achieve a desired level λ, one will need the to ensure that the cost of running BKZ-β given by Equation 1 to be at least 2^λ.

The embedding attack can be enhanced by combining with partial search of the error bits. Specifically, if k nonzero error bits are guessed correctly, the remaining error norm will be reduced to $\sqrt{d-k}$. thereby making the gap from $\lambda_1(\mathcal{L}_{\mathbf{a},c(x)})$ bigger. In fact, the dimension of the embedding lattice is now reduced from n to $n-k$. Hence, we need to ensure that $\binom{n}{k}$ times of each single BKZ execution will be infeasible to carry out.

5.4 Attack when $c(x)$ is irreducible

Note that with knowledge of the public information G, one can easily construct the matrix $B_{\mathbf{a},c(x)}$. The question is whether this matrix will leak information about the polynomial $c(x)$ as well as $\mathbf{a}$. Here, we discuss a possible attack when $c(x)$ is irreducible over $\mathbb{F}_q$, that is, $t=1$.

In this case, the matrix $B_{\mathbf{a},c(x)}$ takes a very simple form, namely,

$$B_{\mathbf{a},c(x)} = \begin{pmatrix} I_{n-1} & -G \\ 0_{1\times(n-1)} & q^d-1 \end{pmatrix},$$

where G is a $(n-1) \times 1$ column. Write G as $G = \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_{n-1} \end{pmatrix}$, where each $g_i \in \mathbb{Z}_{q^d-1}$. Note that g_i satisfies $x - \alpha_i \equiv (x - \alpha_n)^{g_i} \mod c(x)$. Without any loss of generality, we may assume that $\alpha_n = 0$ (by substituting x by $x - \alpha_n$ in the whole system). It follows that $c(x)$ is a common factor of the polynomials $x - \alpha_i - x^{g_i}$, $i = 1, 2, \ldots, n-1$. Since $c(x)$ is irreducible over $\mathbb{F}_q$ of degree d, it is a factor of $x^{q^d} - x$.

We can now perform the following steps to recover $c(x)$ and the α_i's.

- Randomly guess α_1. For each α_1, compute the gcd $h(x) = \gcd(x - \alpha_1 - x^{g_1}, x^{q^d} - x)$. Find all pairs $\alpha_1, d(x)$ such that $d(x)$ is irreducible of degree d and divides the polynomial $h(x)$.
- For each pair $\alpha_1, d(x)$ found above, test for α_2 such that $d(x)$ is also a factor of $x - \alpha_2 - x^{g_2}$.
- Continue the process until one $d(x)$ is left. Let $c(x)$ to be this $d(x)$.
- Find the remaining α_i's by direct computation of $\alpha_i = x - x^{g_i} \mod d(x)$.

We remark that with high probability, the set of possible $d(x)$ after the first step will be very small. It follows that the main complexity of the above attack comes from performing the gcd computations to find gcd of polynomials of the form $x - \alpha - x^g$ and $x^{q^d} - x$. In general, such a gcd computation has complexity polynomial in g. Furthermore, with high probability, g is of the order of q^d. Consequently, in general, the above attack has complexity polynomial in q^d. However, the above attack works if g is small or is of a special form that makes the gcd computation easy.

The above attack easily generalizes to the case when $t > 1$ but the complexity increases as well. Specifically, we will need to guess t different values of α in the first step. This has complexity $n!/(n-t)! \approx n^t$. In view of these considerations, we will choose t to be as big as possible, say $c(x)$ is a product of linear or quadratic polynomials.

5.5 Attack based on Chor-Rivest scheme

When $c(x)$ is irreducible, our scheme can be regarded as a variant of the Chor-Rivest scheme [11], which is a knapsack based cryptosystem. The public key in Chor-Rivest scheme consists of q integers, which are the discrete logarithms of q field elements in the extension field $\mathbb{F}_q[x]/(c(x))$, shifted by a random number v and then permuted by a random permutation. When $v = 1$, these integers in fact constitute the matrix G described in the preceding attack. In the final version of Chor-Rivest scheme [12, 7.1.e], they showed a sub-exponential attack attributed to Ernest Brickell by Chor and Rivest. The attack consists of two major steps. The first step computes a very small weight and small Euclidean norm vector in the dual space of the public key such that the sum of the entries in the vector is also zero by employing a tweaked Laarias-Odlyzko lattice. The second step uses the newfound small vector to facilitate guessing the private permutation and the private shift number to proceed a successful attack.

We next consider the situation when $c(x)$ is a product of t small degree polynomials, say $c(x)$ is a product of d linear polynomials. In this case, the public key will no longer be one column (as in the case of the Chor-Rivest scheme) but comprises d columns. In the following, a *short vector* refers to a vector of small Hamming weight and small Euclidean norm. Following the attack amounts to

find a short coefficient vector for the public key such that the linear combination of the public key is a zero vector. By considering the lattice formed by the rows of the public key, this problem is equivalent to finding short vectors in the dual lattice. It is not clear that very short vectors exist and our experiments for different choices of parameters have not produced such vectors.

Even if short vectors exist, finding any one of them is believed to be a difficult problem. As such, we believe that in the case when $c(x)$ is a product of linear polynomials, the scheme will not suffer from the attack on the Chor-Rivest scheme.

6 A Practical Encryption Scheme

6.1 Description

Similar to the GGH encryption scheme and the McEliece encryption scheme, in order to transit from the one-way trapdoor function to an encryption scheme, one needs a method to encode the message before passing to the trapdoor function. In particular, the chosen encoding scheme should ensure that the encryption scheme is semantically secure.

Recall that a cryptosystem is said to achieve CCA2 security if an attacker has no advantage to decipher a given ciphertext given the plaintexts of other (possibly selected) ciphertexts. It is indistinguishable in the CCA2-model if the attacker, when provided with two plaintexts and a ciphertext, has no advantage to determine which plaintext correctly corresponds to the ciphertext. Different proposals were presented in [20, 29, 24] to achieve semantic (or CCA2) security for the GGH scheme and the McEliece scheme. We first show why the scheme employed in [20] will not work for our construction.

Recall that for the GGH scheme, it was suggested to encode the plaintext bits as the least significant bits of the input message to the trapdoor function and the other bits are allowed to be picked randomly. We now show how this will make our scheme vulnerable to the related message attack.

To this end, let $\mathbf{p}$ be an $(n-t)$-bit plaintext to be encrypted. Suppose that $\mathbf{p}$ is encrypted twice, that is, encoded into $\mathbf{m_1}$ and $\mathbf{m_2}$ with $\mathbf{p}$ occupying the least significant bits of $\mathbf{m_1}$ and $\mathbf{m_2}$. Thus, $\mathbf{m}_1 + \mathbf{m}_2 \mod 2 = \mathbf{0}$. This gives $\mathbf{c_1} = \mathbf{m_1}H + \mathbf{e_1} \mod q^{d_0} - 1$ and $\mathbf{c_2} = \mathbf{m_2}H + \mathbf{e_2} \mod q^{d_0} - 1$. Summing up, this yields $(\mathbf{m_1} + \mathbf{m_2})H + \mathbf{e_1} + \mathbf{e_2} \mod q^{d_0} - 1 = \mathbf{c_1} + \mathbf{c_2}$. If q is odd, we can consider the equation modulo 2 to get $0 \cdot H + \mathbf{e_1} + \mathbf{e_2} = \mathbf{c_1} + \mathbf{c_2} \mod 2$ or $\mathbf{e_1} + \mathbf{e_2} \mod 2 = \mathbf{c_1} + \mathbf{c_2}$. If d is small relative to n, the number of entries which are 1 in both $\mathbf{e_1}$ and $\mathbf{e_2}$ will be very small. Hence, we can guess the positions in which $\mathbf{e_1}$ or $\mathbf{e_2}$ is 1 from the non-zero entries in $\mathbf{c_1} + \mathbf{c_2} \mod 2$ and use the attacks in Section 5 to recover $\mathbf{m}$.

In view of the above, we modify the encoding scheme to work for our trapdoor function. Suppose that the parameters q, n, d, t are fixed. Our input to our trapdoor function is a vector in $\mathbb{Z}^{n-t}_{q^{d_0}-1}$. Thus, each entry is an s-bit string, where $s = 1 + \lfloor \log_2(q^{d_0} - 1) \rfloor$. We will encode an $(n-t)$-bit plaintext message $\mathcal{P}$ into the input $\mathbf{m}$ for the trapdoor function f. The ciphertext will be the output of f in $\mathbb{Z}^{n}_{q^{d_0}-1}$. The entire encryption and decryption processes are described as follows.

Let $\mathbf{m} = (m_1, \ldots, m_{n-t})$ be in $\mathbb{Z}^{n-t}_{q^{d_0}-1}$ and let $m_i^{(j)}$ denote the j-th least significant bit of m_i, $j = 0, 1, \ldots, s-1$. Further, let $\mathbf{m^{(j)}} = (m_1^{(j)}, \ldots, m_{n-t}^{(j)})$. Suppose the plaintext message is $\mathcal{P} = (\mathcal{P}_1, \ldots, \mathcal{P}_{n-t})$. In the following, let `Hash` denote a cryptographic hash function from $\{0,1\}^*$ to $\{0,1\}^{n-t}$. Let f be the trapdoor function with all the notations in Section 4.

Private key: The degree d polynomial $c(x)$, the n elements $\alpha_1, \ldots, \alpha_n$, the unimodular matrix T and the permutation matrix P.

Public key: The parameters n, q, d, t and the matrix $-G$.

Encryption:

- Randomly select $n - t$ bits $\mathbf{z} = (z_1, \ldots, z_{n-t})$.
- Randomly select the error string $\mathbf{e} = (e_1, \ldots, e_n)$ satisfying the desired properties.
- Set $\mathbf{m^{(0)}} = \mathcal{P} \oplus \mathbf{z}$.
- Set $\mathbf{m^{(1)}} = \mathbf{z}$.
- Set $\mathbf{m^{(2)}} = \texttt{Hash}(\mathcal{P}||\mathbf{z}||\mathbf{e})$.
- For $j = 3, \ldots, s-1$, set $\mathbf{m^{(j)}}$ randomly.
- Let $H = \begin{pmatrix} I_{n-t}, & -G \end{pmatrix}$. Then, the ciphertext $\mathbf{c}$ is $\mathbf{c} = f(\mathbf{m}, \mathbf{e}) = \mathbf{m}H + \mathbf{e}$.

Decryption: Given a ciphertext $\mathbf{c}$, the decryption proceeds as follows:

- Compute $\mathbf{m} = f^{-1}(\mathbf{c})$ using the private key. Let $\mathbf{e}$ be the corresponding error. If $\mathbf{e}$ contains only 0 or 1 entries with exactly $d - 1$ 1's, then continue. Otherwise, decryption fails.
- Write $\mathbf{m} = (m_i^{(j)})_{i=1,\ldots,n-t, j=0,1,\ldots,s-1}$.
- Set $\mathcal{P}' = \mathbf{m^{(0)}} \oplus \mathbf{m^{(1)}}$.
- If $\text{Hash}(\mathcal{P}'||\mathbf{m^{(1)}}||\mathbf{e}) = \mathbf{m^{(2)}}$, then $\mathcal{P} = \mathcal{P}'$ and the decryption is successful. Otherwise, decryption fails.

Remark 6.1.

- *Like the GGH scheme [20], we encode our plaintext bits in the least significant bits of the input to our trapdoor function.*
- *In our scheme, the input includes not only the plaintext bits but the error bits and random bits as well. By including the error bits to the input, changing bits of the ciphertext will likely make the decryption process fail. This helps to prevent reactive attacks where attackers try to guess the error bits by sending modified ciphertexts.*
- *Similar to the conversion schemes suggested in [29] for the McEliece encryption scheme, random bits and the hash of the plaintext bits are added to the input to ensure semantic security and to prevent other attacks such as related message attacks.*

6.2 Choosing the Parameters

In view of the attacks presented in Section 5, we will choose the parameters q, n, d, t to resist all the possible attacks. Concretely, the following choices will be made.

- We let $t = d$, that is, $c(x)$ is a product of linear polynomials.
- We set q to be the smallest prime bigger than $n + d$.

- We set d to satisfy $20 \leq d \leq n/2$.
- For a security level λ, we set n and d so that $\binom{n-d}{l} \geq 2^{\lambda}$, where $l = \frac{(n-d)(d-1)}{n}$.

With $d = t$, we have $d_0 = 1$ so all our operations are done modulo $q-1$. Our public key size is $(n-t)t(1+\lfloor \log_2(q-2) \rfloor)$ bits. Since encryption only involves matrix multiplication modulo $q-1$, encryption is efficient with complexity $O(n^2)$.

We now provide possible sets of values of n and d for our encryption scheme. For each pair of n and d, we compute the smallest β such that Equation 3 holds for specified parameters, which gives the corresponds BKZ cost by Equation 1.

Table 1: Possible n and d for Practical Encryption Scheme

n	d	q	$\log_2 \binom{n-d}{l}$	β	$\log_2(BKZ_{cost})$	public key size	P_s
285	41	2819	138	180	80.1	120048bits= 14.66KB	0.289
500	43	29599	184	342	128.2	314416bits= 38.39KB	0.162
729	42	152003	208	518	180.1	519372bits= 63.40KB	0.289

Remark 6.2. *Note that for the parameters in Table 1, the discrete logarithm is computed over the field $\mathbb{F}_q$ rather than $\mathbb{F}_{q^d}$. Thus, Pollard rho method suffices to accomplish this task. As commented in [14], our construction can be viewed as an improvement of the Chor-Rivest scheme in terms of efficiency. The improvement is indeed that the discrete logarithm computation is done in the small field $\mathbb{F}_q/(c_i(x))$ rather than in the big field $\mathbb{F}_q/(c(x))$. In particular, for the parameters in Table 1, we choose $deg(c_i(x)) = 1$.*

Remark 6.3. *As mentioned in Remark 3.7, our construction is analogous to the Goppa code construction. One may use techniques to reduce the key size of Goppa codes in our construction as well. In particular, the techniques in [6, 31] based on automorphism groups can be adapted to reduce the public key size of our construction.*

7 Conclusion

In this paper, we proposed a new lattice-based trapdoor function that inherits the nice properties of the McEliece encryption scheme. Based on this trapdoor function, we designed an encryption scheme that achieves CCA2-security. the public key size of our scheme is around $d(n-d)\log_2(q-1)$. For future research, one may investigate the worst-case to average case hardness of the problem we considered. Another interesting direction is to consider hardness problems for lattices where the ratio between its shortest length and the degree of the polynomial is smaller than 2.

Acknowledgment

We would like to thank Léo Ducas for interesting discussions and for beneficial comments on an earlier version of this work.

References

[1] Miklós Ajtai and Cynthia Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, pages 284–293, 1997.

[2] Martin R. Albrecht. On dual lattice attacks against small-secret LWE and parameter choices in helib and SEAL. In *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part II*, pages 103–129, 2017.

[3] Martin R. Albrecht, Florian Göpfert, Fernando Virdia, and Thomas Wunderer. Revisiting the expected cost of solving usvp and applications to LWE. In *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I*, pages 297–322, 2017.

[4] Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *J. Mathematical Cryptology*, 9(3):169–203, 2015.

[5] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - A new hope. In *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016.*, pages 327–343, 2016.

[6] Magali Bardet, Elise Barelli, Olivier Blazy, Rodolfo Canto-Torres, Alain Couvreur, Philippe Gaborit, Ayoub Otmani, Nicolas Sendrier, and Jean-Pierre Tillich. Big quake: Binary goppa quasi-cyclic key encapsulation, Apr 2018. https://bigquake.inria.fr/files/2018/04/corrected_proposal.pdf.

[7] Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen. *Post Quantum Cryptography*. Springer Publishing Company, Incorporated, 1st edition, 2008.

[8] Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-lwe and security for key dependent messages. In *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, pages 505–524, 2011.

[9] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. *SIAM J. Comput.*, 43(2):831–871, 2014.

[10] Yuanmi Chen. *Réduction de réseau et sécurité concrete du chiffrement completement homomorphe*. PhD thesis, Paris 7, 2013.

[11] Benny Chor and Ronald L. Rivest. A knapsack type public key cryptosystem based on arithmetic in finite fields. In *Advances in Cryptology, Proceedings of CRYPTO '84, Santa Barbara, California, USA, August 19-22, 1984, Proceedings*, pages 54–65, 1984.

[12] Benny Chor and Ronald L. Rivest. A knapsack-type public key cryptosystem based on arithmetic in finite fields. *IEEE Trans. Information Theory*, 34(5):901–909, 1988.

[13] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Trans. Information Theory*, 22(6):644–654, 1976.

[14] Léo Ducas and Cécile Pierrot. Polynomial time bounded distance decoding near minkowski's bound in discrete logarithm lattices. *Des. Codes Cryptogr.*, 87(8):1737–1748, 2019.

[15] Nicolas Gama and Phong Q. Nguyen. Predicting lattice reduction. In *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, pages 31–51, 2008.

[16] Nicolas Gama, Phong Q. Nguyen, and Oded Regev. Lattice enumeration using extreme pruning. In *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*, pages 257–278, 2010.

[17] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 169–178, 2009.

[18] Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. A simple bgn-type cryptosystem from LWE. In *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*, pages 506–522, 2010.

[19] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 197–206, 2008.

[20] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, pages 112–131, 1997.

[21] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from LWE. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, pages 503–523, 2015.

[22] Guillaume Hanrot, Xavier Pujol, and Damien Stehlé. Algorithms for the shortest and closest lattice vector problems. In *Coding and Cryptology - Third International Workshop, IWCC 2011, Qingdao, China, May 30-June 3, 2011. Proceedings*, pages 159–190, 2011.

[23] Jeffrey Hoffstein, Jill Pipher, and Joseph 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*, pages 267–288, 1998.

[24] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the fujisaki-okamoto transformation. In *TCC (1)*, volume 10677 of *Lecture Notes in Computer Science*, pages 341–371. Springer, 2017.

[25] Heeralal Janwa and Oscar Moreno. Mceliece public key cryptosystems using algebraic-geometric codes. *Des. Codes Cryptography*, 8(3):293–307, 1996.

[26] Antoine Joux and Cécile Pierrot. Technical history of discrete logarithms in small characteristic finite fields - the road from subexponential to quasi-polynomial complexity. *Des. Codes Cryptography*, 78(1):73–85, 2016.

[27] Ravi Kannan. Improved algorithms for integer programming and related lattice problems. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*, pages 193–206, 1983.

[28] Elena Kirshanova, Alexander May, and Friedrich Wiemer. Parallel implementation of BDD enumeration for LWE. In *Applied Cryptography and Network Security - 14th International Conference, ACNS 2016, Guildford, UK, June 19-22, 2016. Proceedings*, pages 580–591, 2016.

[29] Kazukuni Kobara and Hideki Imai. Semantically secure mceliece public-key cryptosystems-conversions for mceliece PKC. In *Public Key Cryptography, 4th International Workshop on Practice and Theory in Public Key Cryptography, PKC 2001, Cheju Island, Korea, February 13-15, 2001, Proceedings*, pages 19–35, 2001.

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

[31] Zhe Li, Chaoping Xing, and Sze Ling Yeo. Reducing the key size of mceliece cryptosystem from goppa codes via permutations. To appear in PKC 2019.

[32] Rudolf Lidl and Harald Niederreiter. *Finite fields.* Encyclopedia of mathematics and its applications: v. 20. Cambridge ; New York : Cambridge University Press, 1997., 1997.

[33] Mingjie Liu and Phong Q. Nguyen. Solving BDD by enumeration: An update. In *Topics in Cryptology - CT-RSA 2013 - The Cryptographers' Track at the RSA Conference 2013, San Francisco,CA, USA, February 25-March 1, 2013. Proceedings*, pages 293–309, 2013.

[34] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. *J. ACM*, 60(6):43:1–43:35, 2013.

[35] Robert J McEliece. A Public-Key Cryptosystem Based On Algebraic Coding Theory. *Deep Space Network Progress Report*, 44:114–116, January 1978.

[36] Daniele Micciancio and Shafi Goldwasser. *Complexity of Lattice Problems: a cryptographic perspective*, volume 671 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Boston, Massachusetts, March 2002.

[37] Rafael Misoczki, Jean-Pierre Tillich, Nicolas Sendrier, and Paulo S. L. M. Barreto. Mdpc-mceliece: New mceliece variants from moderate density parity-check codes. In *ISIT*, pages 2069–2073. IEEE, 2013.

[38] Phong Q. Nguyen. Cryptanalysis of the goldreich-goldwasser-halevi cryptosystem from crypto '97. In *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, pages 288–304, 1999.

[39] H Niederreiter. Knapsack type cryptosystems and algebraic coding theory. 15(2):159–166, 01 1986.

[40] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, 2009.

[41] C. P. Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Math. Program.*, 66(2):181–199, September 1994.

[42] Claus-Peter Schnorr. Lattice reduction by random sampling and birthday methods. In Helmut Alt and Michel Habib, editors, *STACS*, volume 2607 of *Lecture Notes in Computer Science*, pages 145–156. Springer, 2003.

[43] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*, pages 124–134, 1994.

[44] Vladimir Michilovich Sidelnikov. A public-key cryptosystem based on binary reed-muller codes. *Discrete Mathematics and Applications*, 4(3):191–208, 1994.

Zhe Li is currently a third year PhD student in School of Physical and Mathematical Sciences, Nanyang Technological University. He is broadly interested in theoretical computer science and in particular in coding theory.

San Ling received the B.A. degree in mathematics from the University of Cambridge and the Ph.D. degree in mathematics from the University of California, Berkeley. He is currently President's Chair in Mathematical Sciences, at the Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore, which he joined in April 2005. Prior to that, he was with the Department of Mathematics, National University of Singapore. His research fields include: arithmetic of modular curves and application of number theory to combinatorial designs, coding theory, cryptography and sequences.

Chaoping Xing received his PhD degree in 1990 from University of Science and Technology of China. From 1990 to 1993 he was a lecturer and associate professor in the same university. He joined University of Essen, Germany as an Alexander von Humboldt fellow from 1993 to 1995. After this he spent most time in Institute of Information Processing, Austrian Academy of Sciences until 1998. From March of 1998 to November of 2007, he was working in National University of Singapore as an assistant/associate/full professor. From December of 2007 to October of 2019, he was working in Nanyang Technological University as a full professor. Since November 2019, he has been with Shanghai Jiao Tong University as a chair professor. Dr. Xing has been working on the areas of coding theory, cryptography, algebraic curves over finite fields and quasi-Monte Carlo methods, etc.

Sze Ling Yeo received the Ph.D. degree in 2006 from the National University of Singapore in mathematics. Since then, she has been working as a research scientist with the Department of Cryptography and Security, Institute for Infocomm Research, under the Agency for Science, Technology, and Research. Her research interests are primarily in the areas of function field theory and their applications, coding theory,as well as the applications of computational number theory to public key cryptography.