An Efficient Quantum Decoder for Prime-Power Fields

Lior Eldar

September 13, 2023

Abstract

We consider a version of the nearest-codeword problem on finite fields $\mathbb{F}_q$ using the Manhattan distance, an analog of the Hamming metric for non-binary alphabets. Similarly to other lattice related problems, this problem is NP-hard even up to constant factor approximation. We show, however, that for $q = p^m$ where $p$ is small relative to the code block-size $n$, there is a quantum algorithm that solves the problem in time $\text{poly}(n)$, for approximation factor $1/n^2$, for any $p$. On the other hand, to the best of our knowledge, classical algorithms can efficiently solve the problem only for much smaller inverse polynomial factors. Hence, the decoder provides an exponential improvement over classical algorithms, and places limitations on the cryptographic security of large-alphabet extensions of code-based cryptosystems like Classic McEliece.

1 Introduction

Error correcting codes are linear subspaces of finite-field vector spaces that allow to protect information against random, and even adversarial errors. The problem of designing good, efficiently decodable, error-correcting codes is notoriously difficult, and is in fact tantamount to an art form: interestingly, it is difficult not because large minimal-distance codes are hard to find (in fact typically a random code does have a large minimal distance), but rather because it is hard to find such codes that are simultaneously efficiently decodable.

The Maximum Likelihood Decoding (MLD) Problem of error correcting codes is well known to be NP-hard since the work of Berlekamp, McEliece and Tilborg [BMT78]. Formally, for the MLD problem we are given a "syndrome" $s \in \mathbb{F}_q^m$, a "parity check matrix" $A$ and are asked to find $e \in \mathbb{F}_q^n$ of weight at most $w$ such that $Ae = s$. In a related problem, called the Nearest Codeword Problem (NCP) [Aro+97; Reg03], we are given a target vector $t$, a generator matrix $A$ and are asked to find the closest codeword to $t$, namely $s$ such that $As$ is closest to $t$, provided that this distance is at most $w$. This problem too, is known to be NP-hard even to sub-polynomial approximation factors [Aro+97] under reasonable complexity assumptions.

1.1 Defining BNCP

The NCP problem is analogous to the closest vector problem (CVP) defined on Euclidean lattices, which is also notoriously hard (see e.g. [Din+03]). Yet, as is often the case in error-correcting scenarios where the error has bounded length, and similarly to the MLD problem, one can consider a bounded error variant of NCP which we call here the Bounded NCP, namely we given a target vector $t$, a matrix $A$ we are asked to find $t$’s closest vector in the span of $A$, provided that this distance is, say, at most $1/10$ of the minimal error correcting distance.

Definition 1. Bounded Nearest Codeword Problem, Hamming Metric
Given an error correcting code \( C = [n, k, d] \subseteq \mathbb{F}_2^n \), where \( d \) is the minimal Hamming distance between any pair of distinct codewords in \( C \), and is generated by matrix \( A \in \mathbb{F}_2^{n \times k} \), and a vector \( t \) such that for some \( s \in \mathbb{F}_2^k \):

\[
\Delta(t, As) \leq \varepsilon \cdot d
\]

where \( \Delta(x, y) \) refers to the Hamming distance between \( x, y \). We are asked to find \( s \).

The definition above uses the Hamming distance between words - namely the number of positions in which two strings are different. Yet, for non-binary \( q \)-ary alphabets, it is of interest to consider different metrics that take into account the actual labels. Considering the alphabet as the additive group \( \mathbb{Z}_q \), one such distance is called the Lee Distance \(^1\) and is defined as follows:

\[
\Delta_L(x, y) = \sum_{i=1}^{n} \min\{|x_i - y_i|, q - |x_i - y_i|\}
\]

Another distance on \( \mathbb{Z}_q \) is the well-known Manhattan distance corresponding to the \( \ell_1 \)-norm of Euclidean space:

\[
\Delta_M(x, y) = \sum_{i=1}^{n} |x_i - y_i|
\]

One can then reconsider Definition 1 for large alphabets:

**Definition 2. Bounded Nearest Codeword Problem (\( \varepsilon \)-BNCP), Manhattan Distance**

Given is an error correcting code \( C = [n, k, d] \), where \( d \) is the minimal Manhattan distance between any pair of distinct codewords in \( C \). \( C \) is generated by matrix \( A \in \mathbb{F}_q^{n \times k} \). We are also given a vector \( t \) such that for some \( s \in \mathbb{F}_q^k \):

\[
\Delta_M(t, As) \leq \varepsilon \cdot d
\]

We are asked to find \( s \).

### 1.2 Hardness of BNCP

In general, BNCP has no known efficient classical algorithms, and the assumed hardness of this problem is, in fact, central to the security of the McEliece cryptosystem ([CS98], [Ber+18]), one of the finalists in the NIST effort to design quantum-secure cryptosystems [Nis].

For \( q = 2 \) the definition above generalizes the Hamming metric, hence BNCP is NP-hard to solve for general \( q \). Let us now examine the behavior of its complexity for specific values of \( q \). On one hand, for \( q \) which is a prime number, \( \mathbb{F}_q \) inherits its multiplication / addition table from \( \mathbb{Z}_q \) and in that case the problem is nearly identical \(^2\) to the Bounded Distance Decoding for \( q \)-ary lattices, a problem whose \( \varepsilon \)-approximation is known to be at least as hard as computing the (unique) shortest vector of an integer lattice up to a factor \( 1/\varepsilon \) [LM09].

As further testament for the generic hardness of this problem: the result of [Aro+97] on the hardness of approximation of the decisional version of the nearest-codeword problem w.r.t. the Hamming metric, can be readily extended to the Manhattan / Lee distances, albeit with a diminished promise gap:

---

\(^1\)In fact, the Lee distance over the ring of integers generates a metric space over the ring since it satisfies, in addition to positivity, and symmetry, the triangle inequality.

\(^2\)Up to the fact that \( q \)-ary lattices are in fact integer lattices with a shortest vector length at most \( q \), whereas \( \mathbb{F}_q \) lattices can have longer shortest vectors, for example the 1-dimensional code generated by \( (1, 2, \ldots, q - 1) \)
Theorem 3. NP-hardness of constant factor approximation of decisional BNCP
Let $C = [n, k, d]$ be some code of $\mathbb{F}_q^n$. There exists a constant $c > 0$ such that if $q = p^m$ for some integer $m$, and $c > p$ then it is NP-hard to decide whether a vector $t \in \mathbb{F}_q^n$ is at Manhattan distance at most $L$ or at distance at least $(c/p) \cdot L$ from $C$.

The proof appears in the appendix.

The resemblance of large-alphabet BNCP to $q$-ary BDD is also apparent in the behavior of random lattices: one can check that just as random $q$-ary lattices have relatively long shortest vectors, similar bounds are satisfied w.r.t. the Manhattan/Lee distance for $\mathbb{F}_q$ random lattices, for any $q$. We provide a formal statement w.r.t. the Manhattan distance.

Lemma 4. Let $A$ be a uniformly random $n \times k$ matrix over $\mathbb{F}_q$ where $n \geq k \log(q)$. Then
\[
\Pr_A \left( \min_{x \neq y, x, y \in C} \Delta_{M,q}(x, y) \geq q^{1-k/n}/2 \right) \geq 1 - 2^{-n/200}
\]

The proof appears in the appendix. Hence for a "typical" error-correcting code the shortest vector can be of length, say, $q^{0.99}$ even for linear rate codes ($k/n = 0.01$). Thus, one can define non-trivial $\varepsilon$-BNCP on random ensembles for very small values of $\varepsilon$, say $\varepsilon = 1/\sqrt{n}$. When allowing $q$ to grow with $n$ we thus achieve a setting that is similar to lattice problems used for cryptography. In this analogy: $\varepsilon$ corresponds to the security parameter of the instance (usually signified by $\alpha$ for the Learning-with-Errors cryptosystem), and an instance is considered to be "hard", or "secure", for cryptographic purposes whenever $\varepsilon \cdot q > \sqrt{n}$.

1.3 Main Results

Despite the apparent difficulty of this problem, we show, that surprisingly, for a code $C = [n, k, d]$ over $\mathbb{F}_q$ for $q$ which is a prime power, namely $q = p^m$ for prime $p$, there exists an efficient quantum algorithm that solves BNCP for $\varepsilon < 1/(p \cdot n^2)$:

Theorem 5. (sketch of Theorem 17)

A Quantum Decoder for Prime-Power Fields
There exists a quantum algorithm that for any $q = p^m$, where $p$ is prime, solves $\varepsilon$-BNCP on $\mathbb{F}_q$ w.r.t. the Manhattan distance for $\varepsilon < 1/(pm^2)$ in time $\text{poly}(n, p, \log(q))$.

We note that by a slight assumption on the distance to the lattice being an integer power of $p$ one can increase $\varepsilon$ to $1/n^2$. We note that for $q = 2$ the above does not provide a meaningful statement since the largest possible value for the minimal error correcting distance is at most $n$. It is only for relatively large values of $q$, say $q = n^3$ that this approximation provides a non-trivial statement.

1.4 Classical Algorithms

1.4.1 Direct Inversion

Consider an error correcting code $C \subseteq \mathbb{F}_q^n$ for $q = p^m$ for some integer $m$, and recall that each element of $\mathbb{F}_q$ can be regarded as an $m$-tuple of numbers in $\mathbb{F}_p$. Given a target vector $y \in \mathbb{F}_q^n$ that is close to $C$, on may be tempted to think that for a sufficiently small noise level each coordinate of $y$,
viewed as an $m$ dimensional vector in $\mathbb{F}_q^m$, has sufficiently many noise-free coordinates - namely the "most-significant" bits, that allow us to determine $y$’s closest vector precisely. In other words, instead of solving the optimization problem:

$$\min_{x \in \mathcal{C}} \Delta(Ax, y)$$

we solve the linear system of equations:

$$\tilde{y} = \tilde{A} \cdot \tilde{x}$$

where $\tilde{y}, \tilde{x}$ correspond to the top-bits in the representations of $x, y$ as vectors in $\mathbb{F}_q^m$, and $\tilde{A}$ is the corresponding $\mathbb{F}_p$ sub-matrix of $A$.

However, such a scheme fails immediately the test of invertibility: one can easily generate a matrix $A \in \mathbb{F}_q^{n \times n}$ that is invertible over $\mathbb{F}_q$, yet regarding $A$ as an $mn \times mn$ linear operator over $\mathbb{F}_p$ and taking the submatrix $A_k$ corresponding to the top $k < m$ coordinates in each tuple results in a matrix that fails to be invertible over $\mathbb{F}_p$. For example, considering $q = 4p = 2$, and constructing $\mathbb{F}_4$ via the irreducible polynomial $x^2 + x + 1$ over $\mathbb{F}_2$ one can check that the matrix

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 0 \end{bmatrix}$$

over $\mathbb{F}_4$ can be written as a linear operator over $\mathbb{F}_2$ as follows:

$$\tilde{A} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

However, extracting the sub-matrix corresponding the top coordinate of each vector results in the following matrix:

$$\tilde{A}_1 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$$

which is not invertible over $\mathbb{F}_2$.

It is plausible to hope that such examples are pathological, in the sense that they rarely appear for random codes. Yet, even for random codes this problem is prevalent. Let us consider concrete estimates: according to Lemma 4 the typical minimal distance $d$ of a random $\mathbb{F}_q$ code is at least $q^{1-k/n} / 2$. If

$$\Delta \varepsilon < q^{1-k/n}/2$$

then viewing each number $x \in \mathbb{F}_q$ as an $m$-dimensional vector $x \in \mathbb{F}_q^m$ we have that the top $mk/n \mathbb{F}_p$ coordinates of each coordinate of the target vector $y \in \mathbb{F}_q^m$ are equal to the corresponding coordinates of some codeword $c \in \mathcal{C}$. Hence, as above, we can write a linear system of equations over $\mathbb{F}_p$:

$$\tilde{y} = \tilde{A} \cdot \tilde{x}$$

corresponding to the top coordinates in the $\mathbb{F}_p$ expansion of each $\mathbb{F}_q$ number. By assumption of the random instance, the submatrix $\tilde{A}$ is in fact a random $mk \times mk$ matrix over $\mathbb{F}_p$, which is invertible with probability roughly $p^{-p}$, i.e. independently of $n$.

In order to increase the probability that $\tilde{A}$ is invertible over $\mathbb{F}_p$ to nearly 1, one would need to decrease the parameter $\varepsilon$ controlling the relative distance to the lattice further so that

$$\Delta \varepsilon \sim q^{1-\beta k/n}$$

for $\beta = \Omega(\log(p))$ but this implies increasing the promise from $\varepsilon$ to roughly $\varepsilon^{\beta-1}$.
1.4.2 Information-Set Decoding

Other classical attacks against McEliece that might also be relevant for BNCP include mainly variants of the Information-Set Decoding algorithm (see e.g. [Pet10], and in the context of the Lee metric see more recent works in [CDE21; HTW19]), but that algorithm’s run-time scales exponentially in the rate $k$. Since “good” codes, i.e. codes which have linear distance and linear rate are usually the target codes considered for both theoretic and practical applications, such algorithms are prohibitive.

1.4.3 Summary

Thus, to the best of our knowledge, efficient decoding w.r.t. the Manhattan distance is only available for random ensembles where the submatrix $\tilde{A}$ is invertible with overwhelming probability. We compute below the quantum-classical separation for $p = 16$, $q^{k/n} = n^6$, $d = q \cdot n^{-6}$ and hence $q^{\beta k/n} = n^{24}$.

|                      | Worst-case | Quantum       | Classical |
|----------------------|------------|---------------|-----------|
|                      | $1/n^2$    | $\text{poly}(n)$ | $e^{Ω(n)}$ |
|                      | $1/n^{18}$ | $\text{poly}(n)$ | $e^{Ω(n)}$ |

|                      | Average-case | Quantum       | Classical |
|----------------------|--------------|---------------|-----------|
|                      | $1/n^2$      | $\text{poly}(n)$ | $e^{Ω(n)}$ |
|                      | $1/n^{18}$   | $\text{poly}(n)$ | $\text{poly}(n)$ |

1.5 Context on the Main Result

Hard computational problems related to lattices and error correcting codes have resisted efficient quantum algorithms for nearly two decades now, despite their underlying Abelian structure that presumably makes them more susceptible to such algorithms. This resistance has given rise to the belief that quantum computers cannot outperform classical ones on problems that require any form of “bounded distance decoding” even with an inverse polynomial promise gap. Our main result suggest that this intuition may be false.

Notably, our result, as it is, does not directly pose a threat to any known public key cryptosystem since its parameter range is quite different than those considered for established PQC systems [Nis]: for the main code-based PQC cryptosystems the alphabet size is constant with the block-size (see e.g. BIKE, HQC, Classic McEliece), whereas for lattice-based cryptosystems, namely descendants of the LWE cryptosystem [Reg09; Pei09; Bra+13], where the alphabet size is in fact allowed to grow with the lattice dimension, the underlying algebraic structure is not a finite field but rather the ring of integers.

Yet, we believe that this parameter mismatch does not capture the full story: our work here suggests that using low-order QFT’s to optimize over high-order groups, (in this case, prime-power fields) in conjunction with the recent construction of [EH22] of approximate eigenvectors of the vector shift operator, does in fact lead to an exponential quantum speed-up for lattice related problems. We hope that further study of the approach outlined here will lead to additional discoveries in this field, classical or quantum.

2 Preliminaries

2.1 Notation

$\mathbb{F}_q$ denotes the field of order $q$, $\omega_p$ denotes the $p$-th root of unity. For $x, y \in \mathbb{F}_q^n$ we use $\Delta_{M,q}(x, y)$ to denote the Manhattan distance between $x, y$ (see Definition 8). An $\mathbb{F}_q$ error-correcting code
is denoted by $C = [n, k, d] \subseteq F_q^n$ where $n$ is the block-length, $k$ is the rate, and $d$ is the minimal Manhattan distance between any pair of codewords:

$$d = \min_{x \neq y, x, y \in C} \Delta_{M,q}(x, y).$$

A code $C$ of rate $k$ is generated by a matrix $A \in F_q^{n \times k}$. Often we omit the subscript $q$ when it is clear from context. Similarly, we use $\Delta_L$ to signify the Lee distance. For a subset $S \subseteq F_q^n$$
\Delta(x, S)
$$signifies the minimal distance between $x$ and any $y \in S$. For $x \in F_q^n$ let $U_x$ denote the shift operator: $U_x|y⟩ = |x + y⟩$. The quantum Fourier Transform on the $n$-dimensional vector space (module) w.r.t. the ring of integers $\mathbb{Z}_p$ is denoted by $F_n^p$.

### 2.2 Vector Representation of Prime Power Fields

Let $p$ be a prime number, and let $F_q$ be a prime number field of order $q = p^m$ relative to some degree-$m$ irreducible polynomial $P \in F_p[x]$:

$$F_q = F_p[x]/P$$

For $a \in F_q$, $q = p^m$, let $\hat{a} \in F_p^m$ denote its $F_p$ vector representation. Likewise for a vector $a \in F_q^k$ let $\hat{a} \in F_p^{m \times k}$ denote the concatenation of the $F_p$ expansion of each of its coordinates. As an additive group, $F_q$ is equal to the $m$-dimensional vector space over $F_p$:

$$F_q = F_p \times F_p \times \ldots \times F_p$$

We assign $q$-ary labels $a \in F_q$ to the elements of $F_p^m$ as a $p$-ary expansion order:

$$\hat{0} = (0, 0, \ldots, 0, 0)$$

$m$ coordinates

$$\hat{1} = (0, 0, \ldots, 0, 1)$$

$$\vdots$$

$$\hat{p} = (0, 0, \ldots, 0, p - 1)$$

$$\vdots$$

$$\hat{q} = (p - 1, p - 1, \ldots, p - 1, 0)$$

$$\hat{q} - \hat{p} = (p - 1, p - 1, \ldots, p - 1, 1)$$

$$\vdots$$

$$\hat{q} - \hat{1} = (p - 1, p - 1, \ldots, p - 1, p - 1)$$

This corresponds to interpreting $\hat{a}$ as the $F_p$ coefficient vector of the polynomial corresponding to $\hat{a}$:

$$\hat{a}(x) = \sum_{i=1}^{m} a_i \cdot x^{i-1}$$
When we use the ordering $x < y$ on $x, y \in \mathbb{F}_q$ it means that $x < y$ as numbers in $\mathbb{Z}$. If $\sigma = p^r$ and $x < \sigma$ we will often use the notation

$$\hat{x} \in 0^{n-r}[p]^r$$

signifying that $x$’s representation as a $p$-ary vector has 0 in the first $n - r$ positions (MSB).

Unless stated otherwise, for $x, y \in \mathbb{F}_q$ the expressions $x + y, x \cdot y$ denote addition / multiplication over $\mathbb{F}_q$. The following proposition is immediately implied by definition:

**Proposition 6.**

\[ \forall x, y \in \mathbb{F}_q^n \quad \hat{x} + \hat{y} = \hat{x + y} \]

where the addition in LHS is over $\mathbb{F}_q$ and the RHS addition is over $\mathbb{F}_p^m$.

For example for $1 \in \mathbb{F}_q$ and $3 \in \mathbb{F}_q$, $q = 16 = 2^4$ we have that $1 + 3 = 2$.

### 2.3 Extending The Manhattan Distance to Prime Power Fields

The Manhattan distance was developed as an alternative to the Hamming distance for transmission of non-binary signals taken from some $q$-ary alphabet. The Manhattan distance

$$\mathbb{Z}_q^n \times \mathbb{Z}_q^n \to \mathbb{Z}^+$$

is defined as follows:

\[ \forall x, y \in \mathbb{Z}_q^n, \quad \Delta_M(x, y) := \sum_{i=1}^{n} |x_i - y_i| \]

In the context of linear codes, one considers the finite field $\mathbb{F}_q$ and not $\mathbb{Z}_q$. These objects are quite different, and are equal only for prime $q$, yet in this study we consider $q$ which itself is a prime power, i.e. $q = p^m$. To this end we define a mapping from $\mathbb{F}_q$ to the ring of integers $\mathbb{Z}_q$ using the natural $p$-ary expansion above:

**Definition 7.** $p$-ary expansion mapping.

For $a \in \mathbb{F}_q$, $q = p^m$ we define $\hat{a} \in \mathbb{Z}_q$ by writing $a$ as a vector $(\hat{a}_1, \ldots, \hat{a}_m) \in \mathbb{F}_p^m$, i.e. $\hat{a}_i \in \mathbb{F}_p$ and defining the polynomial

$$\hat{a}(x) = \sum_{i=1}^{m} \hat{a}_i \cdot x^{i-1}$$

we then set the $\mathbb{Z}_q$ representation of $a$, namely $\tilde{a}$, as the evaluation of the polynomial $\hat{a}(x)$ at point $x = p$:

$$\tilde{a} = \hat{a}(p) = \sum_{i=1}^{m} \hat{a}_i p^{i-1}$$

For $x \in \mathbb{F}_q^n$ we define $\tilde{x} \in \mathbb{Z}_q^n$ as applying the map above coordinate-wise.

We then extend the Manhattan Distance to finite fields $\mathbb{F}_q$ by setting:

**Definition 8.** Manhattan Distance for $\mathbb{F}_q$.

\[ \forall x, y \in \mathbb{F}_q^n \quad \Delta_{M,q}(x, y) = \Delta_M(\tilde{x}, \tilde{y}) \]

and define the length of a vector $x \in \mathbb{F}_q^n$ as its distance from 0:

$$\|x\|_M = \Delta_{M,q}(x, 0) = \Delta_M(\tilde{x}, 0)$$
When we consider \( \mathbb{F}_q \)-linear codes we would like to consider the "minimal distance" of a code, or the distance of a given word from an \( \mathbb{F}_q \) codespace, however, since the Manhattan distance is not a metric, then in particular it is not shift invariant on \( \mathbb{Z}_q \): for example, setting \( q = p \) for prime \( p \) and \( n = 1 \) we have:

\[
p - 1 = \Delta_M(p - 1, 0) \neq \Delta_M(p - 1 + 1, 0 + 1) = \Delta_M(0, 1) = 1
\]

Similarly, the shift-invariance property does not hold for the distance \( \Delta_{M,q} \). However what one can show, is that specifically for the \( p \)-ary expansion mapping the following gap-presevation property does hold:

**Proposition 9. Gap Preserving Property**

For all \( x, y \in \mathbb{F}_q^n \) if for some \( r < m \)

\[
\Delta_{M,q}(x, y) < p^r
\]

then

\[
\Delta_{M,q}(x - y, 0) \leq n \cdot p^r
\]

In particular, if \( C \subseteq \mathbb{F}_q^n \) is such that for some \( r < m \)

\[
d = \min_{x \neq y, x, y \in C} \Delta_{M,q}(x, y) < p^r
\]

then the shortest vector \( x \) of \( C \) satisfies:

\[
\|x\|_M < n \cdot p^r
\]

**Proof.** If \( \Delta_{M,q}(x, y) < p^r \) then

\[
\forall i \in [n] \quad \Delta_M(\tilde{x}_i, \tilde{y}_i) < p^r
\]

This implies that for each \( i \in [n] \) the respective \( p \)-ary expansions of \( \tilde{x}_i, \tilde{y}_i \in \mathbb{F}_p^m \) are identical on indices \( m, m - 1, \ldots, r + 1 \). Thus \( z = x - y \) (subtraction over \( \mathbb{F}_q \)) is such that for each \( i \in [n] \) the \( p \)-ary expansion of \( z_i \), i.e. \( \hat{z}_i \) is 0 for the top \( m - r \) coordinates. In particular \( \hat{z}_i < p^r \) so

\[
\|z\|_M \equiv \sum_{i \in [n]} \hat{z}_i < n \cdot p^r
\]

We note that if one replaces the Manhattan distance with the Lee distance, which is in fact a metric on \( \mathbb{Z}_q \) one obtains a metric space on \( \mathbb{F}_q^n \) via the \( p \)-ary expansion mapping defined above, and under the partial order on elements of \( \mathbb{F}_q \) defined above for the \( p \)-ary expansion. This would imply, in particular that one would be able to improve the performance of the proposed quantum algorithm by a factor of \( n \). Still, we decided to develop this study using the Manhattan distance and not the Lee distance, losing the property of a metric space, for a more general statement.

### 2.4 Invertibility of Random Matrices over Finite Fields

The well-established theory of random matrices over finite fields characterizes the probability that a uniformly random matrix over a finite field is invertible as follows:

**Lemma 10. Theorem 1.1 in [Map10]**

Let \( A \sim U[\mathbb{F}_p^{n \times k}] \). There exists a constant \( c > 0 \) such that:

\[
\Pr_A (A \text{ is invertible}) \geq \prod_{k=1}^{\infty} (1 - p^{-k}) - e^{-cT}
\]
3 Quantum PCS States on Finite Fields

In [EH22] the authors define the "Phased Coset State" (or PCS) on \( q \)-ary lattices as a certain superposition on the lattice, comprised of copies of a bounded function - each centered around an individual lattice point and multiplied by a phase that depends on that lattice point. Here we redefine the PCS on finite fields:

**Definition 11. PCS on Finite Fields**

For \( \sigma \in \mathbb{F}_q \) define the set \( \{0, \ldots, \sigma - 1\} \subseteq \mathbb{F}_q \) and \( [\sigma]^n \) as the \( n \)-th fold product thereof.

1. Define the cube state anchored at a point \( y \in \mathbb{F}_q^n \) by
   \[
   |C(y)\rangle = \sigma^{-n/2} \cdot \sum_{z \in [\sigma]^n} |y + z\rangle.
   \]

2. Let \( C = [n, k, d] \subseteq \mathbb{F}_q^n \) with \( q = p^n \). The phased cube state with label \( \hat{a} \in \mathbb{F}_p^{n-k} \) is the following state:
   \[
   |\psi_{\hat{a}}\rangle = q^{-k/2} \cdot \sum_{c \in \mathbb{F}_q^k} \omega_{p}^{\hat{a} \cdot c} |C(G_c)\rangle = q^{-k/2} \cdot \sigma^{-n/2} \cdot \sum_{c \in \mathbb{F}_q^k} \omega_{p}^{\hat{a} \cdot c} \sum_{z \in [\sigma]^n} |G_c + z\rangle.
   \]

Note that the quantum state is defined on a register with numbers in \( \mathbb{F}_q \) whereas the phase that multiplies each basis element is a power of the primitive root \( \omega_p \).

**Lemma 12** (Cube state properties).

1. \( \forall x, y \in \mathbb{F}_q^n, U_x |C(y)\rangle = |C(x + y)\rangle \), and the transformation \( |C(y)\rangle|x\rangle \) to \( |C(x + y)\rangle|x\rangle \) is computable in time \( \text{poly}(n, \log q) \).

2. Let \( |C(y)\rangle \) be a cube state of side length \( \sigma = p^r \) for some \( r > 0 \), \( y \in \mathbb{F}_q^n \) and let \( \Delta \in \mathbb{F}_q^n \).
   
   (a) If \( \|\Delta\|_{M,q} \leq \sigma \) then \( |C(y)\rangle = |C(y + \Delta)\rangle \).
   
   (b) If \( \|\Delta\|_{M,q} > n \cdot \sigma \) then \( \langle C(y)|C(y + \Delta)\rangle = 0 \).

Consider the implication of Item 2a: it implies that the PCS is not "periodic" on the code-space in the usual sense of having symmetric support around each codeword (/lattice point). The function of symmetric support around each codeword is a different function which is the convolution of the Hamming ball and the code-space. Rather, the support of the cube-shaped super-position starts at the point which is the original codeword with all the right-most \( r \) coordinates erased. Note that the "erased" information is encoded in the phase that multiplies each cube. For example, the cube anchored at a codeword \( c \) such that \( \hat{c} \in [p]^{n-r}0^r \) is situated to the "bottom-right" of the codeword, whereas if \( \hat{c} \in [p]^{n-r}1^r \) it is situated on the "top-left" of the codeword.

**Proof.**

**Item: 1**

\[
\forall x \in \mathbb{F}_q^n \quad U_x |C(y)\rangle = \sigma^{-n/2} \cdot \sum_{x \in [\sigma]^n} U_x |y + z\rangle
\]

\[
= \sigma^{-n/2} \cdot \sum_{x \in [\sigma]^n} |x + y + z\rangle
\]

\[
= |C(x + y)\rangle
\]

Therefore, given \( |C(y)\rangle|x\rangle \), one addition from the second register into the first register results in \( |C(x + y)\rangle|x\rangle \).
Item: 2a
Start with
\[ \| |C(y)\rangle - |C(y + \Delta)\rangle \|^2 = 2 \cdot (1 - \Re(\langle C(y) | C(y + \Delta) \rangle)). \]

We have:
\[ \langle C(y) | C(y + \Delta) \rangle = \langle C(0) | C(\Delta) \rangle \quad \text{Item 1 for the shift by } y \] (4)

Since \( \| \Delta \|_M \leq \sigma \) then
\[ \forall i \in [n] \quad \tilde{\Delta}_i \leq \sigma. \]

Hence, for each \( i \) the set \( [\sigma] \) is invariant under shift by \( \Delta_i \):
\[ \Delta_i + [\sigma] = \{ x + \Delta_i, x \in [\sigma] \} \]
\[ = \{ x + \Delta_i, \hat{x} \in 0^n - [p]^r \} \quad \text{By the assumption that } \sigma = p^r \] (6)
\[ = \{ x + \Delta_i, x + \tilde{\Delta}_i + \hat{\Delta}_i \in 0^n - [p]^r \} \quad \text{By Proposition 6} \] (7)
\[ = \{ x, \hat{x} + \tilde{\Delta}_i \in 0^n - [p]^r \} \quad \text{Re-indexing} \] (8)
\[ = \{ x, \hat{x} \in 0^n - [p]^r \} \quad \text{By the assumption that } \Delta_i \leq \sigma \] (9)
\[ = \{ x, x \in [\sigma] \} \quad \text{By the assumption that } \Delta_i \leq \sigma \] (10)
\[ = [\sigma] \quad \text{By the assumption that } \Delta_i \leq \sigma \] (11)

It follows that:
\[ \langle C(0) | C(\Delta) \rangle = 1 \]

Substituting in Equation 4 implies: \( \| |C(y)\rangle - |C(y + \Delta)\rangle \| = 0. \)

Item: 2b
If \( \| \Delta \|_M \geq n \cdot \sigma + 1 \) there exists at least one coordinate \( i \in [n] \) such that
\[ \tilde{\Delta}_i > \sigma \]
in that case \( \tilde{\Delta}_i \notin 0^n - [p]^r \), and together with the assumption \( \sigma = p^r \) we have:
\[ \forall x \in [\sigma] \quad \hat{x} + \tilde{\Delta}_i \notin 0^n - [p]^r \]
so
\[ \langle C(0) | C(\Delta) \rangle = 0 \]

We conclude from the lemma above that \( |\psi_\hat{a}\rangle \) is an eigenvector of \( U_t \) for \( t \) that is \( \sigma \)-close to a word \( s \) with eigenvalue \( \omega_p^{a \cdot \hat{a} \cdot \hat{\delta}} \).

Lemma 13. Let \( |\psi_\hat{a}\rangle \) denote a PCS state with label \( \hat{a} \in \mathbb{F}_p^{m-k} \) and parameter \( \sigma = p^r \) for integer \( r < m \), and let \( t \in \mathbb{F}_q^n \) such that
\[ \Delta_{M,q}(t, As) \leq \sigma/n \]
for some \( s \in \mathbb{F}_q^k \). Then
\[ U_t |\psi_\hat{a}\rangle = \omega_p^{-a \cdot \hat{a} \cdot \hat{\delta}} \cdot |\psi_\hat{a}\rangle. \]
Proof. Since \( \Delta_{M,q}(t, As) \leq \sigma/n \), for \( \sigma = p^r, r < m \) then by Proposition 9 we can write:

\[
t = As + \Delta
\]

where \( \| \Delta \|_{M,q} \leq \sigma \). Therefore

\[
U_t|\psi_{\hat{a}}\rangle = q^{-k/2} \cdot U_t \cdot \sum_{c \in \mathbb{F}_q^k} \omega_p^{\hat{a}c} |C(Ac)\rangle
\]

\[
= q^{-k/2} \cdot U_{\Delta} \cdot U_{As} \cdot \sum_{c \in \mathbb{F}_q^k} \omega_p^{\hat{a}c} |C(Ac)\rangle
\]

\[
= q^{-k/2} \cdot U_{\Delta} \cdot \sum_{c \in \mathbb{F}_q^k} \omega_p^{\hat{a}c} |C(A(c + As))\rangle \quad \text{definition of shift over } \mathbb{F}_q
\]

\[
= q^{-k/2} \cdot U_{\Delta} \cdot \sum_{c \in \mathbb{F}_q^k} \omega_p^{\hat{a}c} |C(A(c + s))\rangle \quad \text{linearity over } \mathbb{F}_q
\]

\[
= q^{-k/2} \cdot \omega_p^{\hat{a}s} \sum_{c \in \mathbb{F}_q^k} \omega_p^{\hat{a}(c + s)} |C(A(c + s))\rangle
\]

\[
= q^{-k/2} \cdot U_{\Delta} \cdot \omega_p^{\hat{a}s} \sum_{c \in \mathbb{F}_q^k} \omega_p^{\hat{a}c} |C(A(c + s))\rangle \quad \text{Proposition 6}
\]

\[
= \omega_p^{\hat{a}s} \cdot q^{-k/2} \cdot U_{\Delta} \sum_{c \in \mathbb{F}_q^k} \omega_p^{\hat{a}c} |C(Ac)\rangle \quad \text{Re-indexing } c + s \rightarrow c
\]

\[
= \omega_p^{\hat{a}s} \cdot q^{-k/2} \sum_{c \in \mathbb{F}_q^k} \omega_p^{\hat{a}c} |C(Ac)\rangle \quad \text{Item 2a since } \| \Delta \|_{M,q} \leq \sigma
\]

\[
= \omega_p^{-\hat{a}s} |\psi_{\hat{a}}\rangle
\]

We now show an efficient algorithm for sampling a PCS state \( |\psi_{\hat{a}}\rangle \) for random \( \hat{a} \):

**Lemma 14. An efficient quantum PCS sampler**

Let \( C = [n, k, d] \) be a code of \( \mathbb{F}_q^n \) generated by matrix \( A \in \mathbb{F}_q^{n \times k}, q = p^m \), i.e.

\[
d = \min_{x \neq y, x, y \in \Delta_{M,q}(x, y)}
\]

There exists a quantum algorithm that samples \( |\psi_{\hat{a}}\rangle \) for \( \hat{a} \sim U [\mathbb{F}_p^{m \times k}] \) in time \( \text{poly}(n, \log(q)) \), whenever \( \sigma < d/n \).

**Proof.** Consider the following evolution according to the computational steps specified in each
equation:

\[ |0\rangle_1 \otimes |0\rangle_2 \rightarrow |0\rangle_1 \otimes |C(0)\rangle_2 \quad \text{QFT: } I \otimes \mathcal{F}_q^n \]
\[ \rightarrow q^{k/2} \cdot \sum_{c \in \mathbb{F}_q^k} |c\rangle \otimes |C(c)\rangle \quad \text{QFT: } \mathcal{F}_q^k \otimes I \]
\[ \rightarrow q^{k/2} \cdot \sum_{c \in \mathbb{F}_q^k} |c\rangle \otimes |C(Ac)\rangle \quad \text{controlled shift by } Ac \]
\[ \rightarrow q^{k/2} \cdot \sum_{c \in \mathbb{F}_q^k} |c\rangle \otimes |C(c)\rangle \quad \text{Change rep.: } \mathbb{F}_q^k \text{ to } \mathbb{F}_p^{m \cdot k} \]
\[ = q^{-k} \sum_{\hat{a} \in \mathbb{Z}_p^{mk}} |\hat{a}\rangle \otimes \left( \sum_{c \in \mathbb{F}_q^k} \omega_p^{\hat{a}c} |C(Ac)\rangle \right) \quad \text{QFT: } \mathcal{F}_p^{m \cdot k} \otimes I \]

By definition we have

\[ \forall c \in \mathbb{F}_q^k \quad \|Ac\|_M = \Delta_{M,q}(Ac,0) \geq d > \sigma \cdot n. \]

Then by Item 2b it follows that the set \{ |C(Ac)\rangle \}_{c \in \mathbb{F}_q^k} forms an orthonormal set. Hence

\[ \forall \hat{a} \in \mathbb{F}_p^{mk} \quad \Pr(\hat{a}) = q^{-k} \]

which is independent of \( \hat{a} \), i.e. \( \hat{a} \) is sampled uniformly from \( \mathbb{Z}_p^{mk} \). The running time of the procedure is determined by the complexity of the Fourier transform over \( \mathbb{F}_p^{mk} \), which is at most

\[ \log(p) \cdot m \cdot k = \text{poly}(n, \log(q)). \]

\[ \Box \]

4 An Algorithm for BNCP for Prime-Power Fields

We now define the following quantum bounded-distance decoder: We first define the algorithm in terms of \( q = 2^m \) for simplicity of exposition, and later we’ll generalize it to any \( q = p^m \) for prime \( p \).

Algorithm 15. A Quantum Decoder for Finite Field BNCP

Input: \( A \in \mathbb{F}_q^{n \times k}, \ t \in \mathbb{F}_q^n \), \( q = 2^m \), and parameter \( \sigma > 0 \).

1. Sample \( T = k \cdot m \) quantum PCS states with parameter \( \sigma \):

\[ |\psi_{\hat{a}_1}\rangle \otimes \ldots \otimes |\psi_{\hat{a}_T}\rangle \]

2. Let \( \hat{A} \) denote the matrix whose columns are the labels of the sampled PCS states:

\[ \hat{A} = [\hat{a}_1, \ldots, \hat{a}_T] \]

Assume w.l.o.g. that \( \hat{A} \) is invertible over \( \mathbb{F}_2 \).
3. Tensor with the uniform superposition
\[ q^{-k/2} \cdot \sum_{\hat{z} \in \mathbb{F}_q^k} |\hat{z}\rangle \]

4. Apply $\hat{A}^{-1}$ to the register:
\[ q^{-k/2} \cdot \sum_{\hat{z} \in \mathbb{F}_q^k} |\hat{A}^{-1} \cdot \hat{z}\rangle \]

5. Apply a controlled-shift operation where bit $j \in [T]$ of $\hat{A}^{-1} \hat{z} \in \mathbb{F}_2^T$ controls whether or not we apply $U_t$ to the $j$-th PCS state:
\[ q^{-k/2} \cdot \sum_{\hat{z} \in \mathbb{F}_q^k} |\hat{A}^{-1} \cdot \hat{z}\rangle \otimes U_t^{(\hat{A}^{-1} \hat{z})_1} |\psi_{\hat{a}_1}\rangle \otimes \ldots \otimes U_t^{(\hat{A}^{-1} \hat{z})_T} |\psi_{\hat{a}_T}\rangle \]

6. Apply $\hat{A}$ to the first register:
\[ q^{-k/2} \cdot \sum_{\hat{z} \in \mathbb{F}_q^k} |\hat{z}\rangle \otimes U_t^{(\hat{A}^{-1} \hat{z})_1} |\psi_{\hat{a}_1}\rangle \otimes \ldots \otimes U_t^{(\hat{A}^{-1} \hat{z})_T} |\psi_{\hat{a}_T}\rangle \]

7. Apply the $\mathbb{F}_2^T$ quantum Fourier transform on the first register, and measure in the standard basis. Denote as output $O$.

Using this algorithm we solve an instance of $\varepsilon$-BNCP to factor $\varepsilon = 1/(2n^2)$.

**Theorem 16.** Let $C = [n, k, d]$ be an error correcting code over $\mathbb{F}_q^n$ for $q = p^m$, $p = 2$. Let $(A \in \mathbb{F}_q^{n \times k}, t \in \mathbb{F}_q^n)$ be an instance of $\varepsilon$-BNCP where
\[ \Delta(t, As) \leq d/(2n^2) \]
for some $s \in \mathbb{F}_q^k$. Then upon input $(A, t)$ and parameter $\sigma = 2^r, r < m$ that satisfies:
\[ (*) \quad d/(2n) \leq \sigma < d/n \]
Algorithm 15 runs in expected time $\text{poly}(n, \log(q))$ and returns an outcome $O = \hat{s}$.

We note that the theorem above assumes a-priori knowledge of $d$. This is reasonable in the error-correction setting, but in the computational theory of lattices knowledge of the minimal distance amounts to an oracle to the GapSVP problem which is also known to be a hard problem. However, by initializing $\sigma = 1$ and executing the algorithm on sequential doubling of the parameter there will be at least one iteration such that $\sigma = 2^r$ satisfies the condition $(*)$. Since the correct answer can be easily checked this essentially removes the need to know $d$ in advance.

**Proof.** Assume for now that $\hat{A}$ is invertible over $\mathbb{F}_2$ and consider the output of step 6
\[ |\psi\rangle = q^{-k/2} \cdot \sum_{\hat{z} \in \mathbb{F}_q^k} |\hat{z}\rangle \otimes U_t^{(\hat{A}^{-1} \hat{z})_1} |\psi_{\hat{a}_1}\rangle \otimes \ldots \otimes U_t^{(\hat{A}^{-1} \hat{z})_T} |\psi_{\hat{a}_T}\rangle \]

By our choice of parameters we have:
\[ \sigma \geq n \cdot \Delta_{M,q}(t, As). \]
Since in addition $\sigma = 2^r, r < m$, we can invoke Lemma 13 which implies:

$$\forall \tilde{a} \in \mathbb{F}_2^T \quad U_t |\psi_{\tilde{a}}\rangle = (-1)^{\hat{s}\tilde{a}} |\psi_{\tilde{a}}\rangle.$$ 

Observe that:

$$U_t^1 = U_t \quad U_t^0 = I.$$ 

Hence each PCS state $|\psi_{\tilde{a}}\rangle$ above is multiplied by a phase $(-1)^{\hat{s}\tilde{a}}$ if $(\hat{A}^{-1}\hat{z})_i = 1$ and by phase 1 if $(\hat{A}^{-1}\hat{z})_i = 0$:

$$|\psi\rangle = q^{-k/2} \sum_{z \in \mathbb{F}_q^k} |\hat{z}\rangle \otimes (-1)^{\hat{s}\hat{A}^{-1}\hat{z}} |\psi_{\hat{a}}\rangle \otimes \ldots \otimes (-1)^{\hat{s}\hat{A}^{-1}\hat{z}_T} |\psi_{\hat{a}_T}\rangle \quad (27)$$

$$= q^{-k/2} \sum_{z \in \mathbb{F}_q^k} |\hat{z}\rangle \otimes (\hat{s}\hat{A}^{-1}\hat{z}) |\psi_{\hat{a}}\rangle \otimes \ldots \otimes |\psi_{\hat{a}_T}\rangle \quad (28)$$

$$= \left(q^{-k/2} \sum_{z \in \mathbb{F}_q^k} (-1)^{\hat{s}\hat{z}} |\hat{z}\rangle \right) \otimes |\psi_{\hat{a}}\rangle \otimes \ldots \otimes |\psi_{\hat{a}_T}\rangle \quad (29)$$

In this case measuring register 1 in the $\mathbb{F}_2^T$ Fourier basis results in outcome $\hat{s}$ with probability 1.

**Running time:** Since

$$\sigma < d/n$$

then by Lemma 14 we can sample PCS states $|\psi_{\hat{a}}\rangle$ such that $\hat{a} \sim U[\mathbb{F}_2^T]$ in time $\text{poly}(n, \log(q))$. By independence of sampling this implies that the entries of $A$ are i.i.d. uniform on $\mathbb{F}_2$. By Lemma 10 this implies that

$$\Pr(\hat{A} \text{ is invertible}) \geq \prod_{k=1}^{\infty} (1 - 2^{-k}) - e^{-cT} \geq 1/10$$

It follows that after $O(1)$ iterations of Step 1 the matrix $\hat{A}$ is invertible. The rest of the computational steps: namely the Quantum Fourier Transform, the controlled shift operation, and multiplication by $\hat{A}, \hat{A}^{-1}$ all take time at most $\text{poly}(n, \log(q))$.

\[\boxed{}\]

### 4.1 Generalization to Arbitrary Characteristic

In the previous section we have shown an algorithm to solve BNCP on fields $\mathbb{F}_q$ of characteristic 2, namely $q = 2^m$ for some integer $m$. In this section we’ll generalize this algorithm to arbitrary characteristic: $q = p^m$ for prime $p$.

We consider again Algorithm 15 previously stated for $p = 2$. For general $p$ we require in Step 1 that $\hat{A}$ is invertible over $\mathbb{F}_p$, and in Step 1 we consider operators of the form $U_t^\ell$ where now $\ell$ can assume any number in $\mathbb{F}_p$ (instead of a binary value) and $U_t^\ell$ is then interpreted as taking the $\ell$-th power of $U_t$ where:

$$U_t^\ell = U_t \cdot \ldots \cdot U_t$$

$\ell$ times

We now restate Theorem 16 for prime-power fields $q = p^m$. We note that the distance to the lattice for which the theorem holds is now decreased by a factor of $p$, i.e. we can solve the problem when the distance $\Delta(t, C)$ is at most $d/(p \cdot n^2)$. This extra condition is set in order to allow the existence
of a value \( \sigma = p^r, r < m \) that is at least \( \Delta(t, \mathcal{C}) \) and at most \( d/n \). As before, this condition can be omitted by making a numerical assumption on the distance from \( \mathcal{C} \).

**Theorem 17.** Let \( \mathcal{C} = [n, k, d] \) be an error correcting code over \( \mathbb{F}_q^n \) for \( q = p^m \) for prime \( p \). Let \( (A \in \mathbb{F}_{q}^{n \times k}, t \in \mathbb{F}_q^n) \) be an instance of \( \varepsilon \)-BNCP where

\[
\Delta_M,q(t, As) \leq d/(pn^2)
\]

for some \( s \in \mathbb{F}_{q}^k \). Then upon input \((A, t)\) and parameter \( \sigma = p^r, r < m \) that satisfies:

\[
(*) \quad d/(pn) < \sigma < d/n
\]

Algorithm 15 runs in expected time \( \text{poly}(n, p, \log(q)) \) and returns results an outcome \( O = \hat{s} \).

**Proof.** Assume for now that \( \hat{A} \) is invertible over \( \mathbb{F}_p \) and consider the output of step 6

\[
|\psi\rangle = q^{-k/2} \sum_{z \in \mathbb{F}_q^k} |\hat{z}\rangle \otimes U_t^\dagger(\hat{A}^{-1}\hat{z}) |\psi_\hat{a}\rangle \otimes \cdots \otimes U_t^\dagger(\hat{A}^{-1}\hat{z})^T |\psi_{\hat{a}T}\rangle
\]

By our choice of parameters we have:

\[
\sigma > n \cdot \Delta(t, As)
\]

and \( \sigma = p^r \) for \( r < m \). Thus we can invoke Lemma 13 which implies:

\[
\forall \hat{a} \in \mathbb{F}_p^n, \quad U_t |\psi_\hat{a}\rangle = \omega_p^{-\hat{a}\hat{s}} |\psi_\hat{a}\rangle.
\]

Therefore

\[
\forall \ell \in \mathbb{F}_p, U_t^\ell = U_t \cdots U_t |\psi_\hat{a}\rangle = \omega_p^{-\hat{a}\hat{s}} |\psi_\hat{a}\rangle
\]

Hence each PCS state \( |\psi_\hat{a}\rangle \) above is multiplied by a phase \( \omega_p^{-\ell\hat{a}\hat{s}} \) where \( \ell = (\hat{A}^{-1}\hat{z})_i \in \mathbb{F}_p \):

\[
|\psi\rangle = q^{-k/2} \sum_{z \in \mathbb{F}_q^k} |\hat{z}\rangle \otimes \omega_p^{-\hat{a}\hat{s}} \hat{A}^{-1} |\psi_\hat{a}\rangle \otimes \cdots \otimes \omega_p^{-\hat{A}^{-1}\hat{z}} |\psi_{\hat{a}T}\rangle \tag{30}
\]

\[
= q^{-k/2} \sum_{z \in \mathbb{F}_q^k} |\hat{z}\rangle \otimes \omega_p^{-\hat{a}\hat{s}} \hat{A}^{-1} |\psi_\hat{a}\rangle \otimes \cdots \otimes |\psi_{\hat{a}T}\rangle \tag{31}
\]

\[
= \left(q^{-k/2} \sum_{z \in \mathbb{F}_q^k} \omega_p^{-\hat{a}\hat{s}} |\hat{z}\rangle\right) \otimes |\psi_\hat{a}\rangle \otimes \cdots \otimes |\psi_{\hat{a}T}\rangle \tag{32}
\]

In this case measuring register 1 in the \( \mathbb{F}_p^T \) Fourier basis results in outcome \( \hat{s} \) with probability 1. Taking the negation of the answer yields \( O = \hat{s} \).

**Running time:** Since

\[
\sigma < d/n
\]

then by Lemma 14 we can sample PCS states \( |\psi_\hat{a}\rangle \) such that \( \hat{a} \sim U[\mathbb{F}_p^n] \) in time \( \text{poly}(n, \log(q)) \). So by independence of sampling this implies that the entries of \( \hat{A} \) are i.i.d. uniform on \( \mathbb{F}_p \). By Lemma 10 this implies that

\[
\Pr(\hat{A} \text{ is invertible}) \geq \prod_{k=1}^{\infty} (1 - p^{-k}) - e^{-cT} \geq 1 - \sum_{k=1}^{\infty} p^{-k} - e^{-cT} = 1 - \frac{1/p}{1-1/p} - e^{-cT}
\]
where the last inequality follows from assuming $p \geq 3$ and sufficiently large $T$. It follows that after $O(1)$ iterations of Step 1 the matrix $\hat{A}$ is invertible. The rest of the computational steps: namely the Quantum Fourier Transform, the controlled shift operation, and multiplication by $\hat{A}$, $\hat{A}^{-1}$ all take time at most $\text{poly}(n, p, \log(q))$, where the extra factor of $p$ comes from the fact that $U^p_t$ is implemented as $p$ sequential applications of $U_t$.

\[ \frac{1-2/p}{1-1/p} - e^{-cT} \geq 1/4 \]

5 Acknowledgements

The author thanks Léo Ducas, Saeed Mehraban, Peter Shor, Nicolas Sendrier, and an anonymous reviewer for their useful comments and suggestions.

References

[BMT78] E. Berlekamp, R. McEliece, and H. van Tilborg. “On the inherent intractability of certain coding problems (Corresp.)” In: IEEE Transactions on Information Theory 24.3 (1978), pp. 384–386. DOI: 10.1109/TIT.1978.1055873.

[Aro+97] Sanjeev Arora, László Babai, Jacques Stern, and Z Sweedyk. “The Hardness of Approximate Optima in Lattices, Codes, and Systems of Linear Equations”. In: Journal of Computer and System Sciences 54.2 (1997), pp. 317–331. ISSN: 0022-0000. DOI: https://doi.org/10.1006/jcss.1997.1472.

[CS98] Anne Canteaut and Nicolas Sendrier. “Cryptanalysis of the Original McEliece Cryptosystem”. In: Advances in Cryptology — ASIACRYPT’98. Ed. by Kazuo Ohta and Dingyi Pei. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 187–199.

[Din+03] I. Dinur, Guy Kindler, R. Raz, and S. Safra. “Approximating CVP to Within Almost-Polynomial Factors is NP-Hard”. In: Combinatorica 23 (Apr. 2003), pp. 205–243. DOI: 10.1007/s00493-003-0019-y.

[Reg03] O. Regev. “Improved inapproximability of lattice and coding problems with preprocessing”. In: 18th IEEE Annual Conference on Computational Complexity, 2003. Proceedings. 2003, pp. 363–370. DOI: 10.1109/CCC.2003.1214435.

[LM09] Vadim Lyubashevsky and Daniele Micciancio. “On Bounded Distance Decoding, Unique Shortest Vectors, and the Minimum Distance Problem”. In: Advances in Cryptology - CRYPTO 2009. Ed. by Shai Halevi. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 577–594.

[Pei09] Chris Peikert. “Public-key cryptosystems from the worst-case shortest vector problem: extended abstract”. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009. ACM, 2009, pp. 333–342. DOI: 10.1145/1536414.1536461.URL: https://doi.org/10.1145/1536414.1536461.

[Reg09] Oded Regev. “On lattices, learning with errors, random linear codes, and cryptography”. In: J. ACM 56.6 (2009), pp. 1–40. ISSN: 0004-5411.

[Map10] Kenneth Maples. “Singularity of Random Matrices over Finite Fields”. In: arXiv: Combinatorics (2010).
A Proof of Technical Lemmas

A.1 Proof of Lemma 4

Proof. Consider $a_1, \ldots, a_k$ random vectors in $\mathbb{F}_q^n$ that generate $C$, and a vector of coefficients $x = (x_1, \ldots, x_k) \in \mathbb{F}_q^k$. We have

$$
\Pr \left( \exists c \in C, \ c \neq 0 \ | |c||_M \leq n \cdot q^{1-k/n}/2 \right) \leq \Pr_{a_1, \ldots, a_k} \left( \exists x \in \mathbb{F}_q^k, x \neq 0, \ \left| \sum_{i=1}^k a_i x_i \right|_M \leq n \cdot q^{1-k/n}/2 \right)
$$

$$
\leq q^k \cdot \Pr_{a_i \sim U[\mathbb{F}_q], x \neq 0} \left( \left| \sum_{i=1}^k a_i x_i \right|_M \leq n \cdot q^{1-k/n}/2 \right)
$$

Considering the above, for any nonzero $x \in \mathbb{F}_q^k$ each coordinate $z_j \in \mathbb{F}_q, j \in [n]$ of $z = \sum_i a_i x_i$ is a uniformly random variable on $\mathbb{F}_q$ that is independent of all other coordinates. We have

$$
\Pr(|z_j| \leq q^{1-k/n}) \leq q^{-k/n}.
$$

Thus, by Chernoff:

$$
\Pr \left( \sum_i |z_i| \leq n \cdot q^{1-k/n}/2 \right) \leq 2^{-n \cdot (q^{-k/n})^2/16} \leq 2^{-n/200}
$$
where the last inequality follows from \( n \geq k \log(q) \). Applying the contrapositive of Proposition 9 we conclude that

\[
\Pr \left( \min_{x \neq y, x, y \in C} \Delta_{M,q}(x, y) < q^{1-k/n}/2 \right) < 2^{-n/200}
\]

\[\blacksquare\]

A.2 Proof of Theorem 3

Consider the problem of \( c \)-approximate set-cover: a ground-set \( \mathcal{U} \), and a collection of subsets \( S_1, \ldots, S_m \). A cover is a sub-collection of the \( S_i \)'s whose union is \( \mathcal{U} \). The cover is exact if the sets in the cover are disjoint. The size of a cover is the number of sets that comprise it.

The construction of [Aro+97] defines \( m+1 \) vectors \( b_0, \ldots, b_m \in \mathbb{F}_q, r = L|\mathcal{U}| + m, L = cK \) as follows: for each set \( S_i \) we define a vector \( b_i \) on \( L \cdot |\mathcal{U}| + m \). The first \( L|\mathcal{U}| \) coordinates are considered as \( |\mathcal{U}| \) tuples of \( L \) coordinates each, where each tuple corresponds to an element of \( \mathcal{U} \). \( b_i \) is zero except for the \( L \cdot |S_i| \) coordinates corresponding to the characteristic vector of \( S_i \). The last \( m \) coordinates are zero except at the \( i \)-th position which is 1. The vector \( b_0 \) is the all-ones vector on the first \( L|\mathcal{U}| \) coordinates, and 0 on the last \( m \). We now claim similarly to [Aro+97]:

**Lemma 18.** Let \( q = p^m \) and suppose that \( c > p \). Let \( L = c \cdot K \). Define:

\[
\text{OPT} = \min_{\alpha} \Delta_M \left( b_0, \sum_i \alpha_i b_i \right)
\]

and let \( C \) denote the linear span of the vector \( b_1, \ldots, b_m \) over \( \mathbb{F}_q \). If there exists an exact cover of size \( K \) then

\[
\Delta_M(b_0, C) \leq p \cdot K
\]

and if any cover is of size at most \( c \cdot K \) then

\[
\Delta_M(b_0, C) \geq c \cdot K
\]

**Proof.** Let \( p \) denote the characteristic of \( \mathbb{F}_q \), i.e. \( q = p^m \) for some integer \( m \geq 0 \). If there exists an exact cover \( S_{i_1}, \ldots, S_{i_K} \) then choosing \( \alpha_{j} = p - 1 \) for all \( j \in [K] \) and 0 otherwise has that \( b_0 + \sum_i \alpha_i b_i \) is equal to 0 on the first \( |\mathcal{U}| \cdot L \) bits. On the last \( m \) bits the Manhattan weight of \( b_0 + \sum_i \alpha_i b_i \) is precisely \( \sum_i \alpha_i = (p - 1) \cdot K \). Hence

\[
\Delta_M(b_0, C) \leq (p - 1) \cdot K
\]

Suppose now that any cover has size at least \( L = c \cdot K \). Let \( \alpha = (\alpha_1, \ldots, \alpha_r) \) denote an assignment vector \( \alpha_i \in \mathbb{F}_q \). First, suppose that \( \sum_i \alpha_i b_i \) has non-zero coordinates on each of the first \( |\mathcal{U}| \) tuples of \( L \) bits. Then each tuple is "covered" by at least one vector \( b_i \), that corresponds to set \( S_i \), and \( b_i \) is multiplied by a non-zero coefficient \( \alpha_i \). Thus \( \sum_i \alpha_i \geq c \cdot K = L \) this is manifested in the last \( m \) coordinates, implying

\[
\Delta_M(b_0, \sum_i \alpha_i b_i) \geq L
\]

On the other hand, if not all tuples are covered, i.e. there is at least one \( L \)-tuple that is all zeros, then the Manhattan distance on the first \( |\mathcal{U}| \cdot L \) coordinates is at least \( L \), implying

\[
\Delta_M(b_0, \sum_i \alpha_i b_i) \geq L
\]

\[\blacksquare\]

The proof of Theorem 3 follows by applying the lemma in conjunction with the fact that there exists a constant \( c > 0 \) such that it is NP-hard to approximate exact set-cover to factor at most \( c \).