PolarCrypt: A Post-Quantum Cryptosystem Based on the Closest Point Problem on High-Dimensional Spiral Manifolds

Madhav Dogra

November 8, 2024

Abstract

This paper introduces PolarCrypt, a novel public-key encryption scheme designed for the post-quantum era. The security of PolarCrypt is based on the conjectured hardness of the Closest Vector Problem (CVP) on a new class of geometric objects: high-dimensional spiral manifolds. We define a trapdoor mechanism using a composition of a globally smooth spiral function and a locally perturbing pseudo-random function. We provide a formal specification of the cryptosystem, including key generation, encryption, and decryption algorithms. The core of our contribution is a rigorous security analysis, featuring a formal reduction from the worst-case hardness of the approximate CVP on lattices to the problem of breaking PolarCrypt. We prove that PolarCrypt achieves IND-CPA security under this assumption. Furthermore, we analyze its resilience against known lattice-based, algebraic, and quantum attacks. A comparative performance analysis against NIST PQC standards like CRYSTALS-Kyber and Classic McEliece demonstrates its potential trade-offs in key sizes and computational efficiency. PolarCrypt represents a new direction in geometric cryptography, extending the principles of lattice-based security to more complex, non-linear manifolds.

1 Introduction

1.1 The Impending Quantum Threat and the Post-Quantum Transition

The security of the world's digital infrastructure currently relies on public-key cryptosystems such as RSA and Elliptic Curve Cryptography (ECC). [1] The hardness of these systems is predicated on computational problemsinteger factorization and the discrete logarithm problem, respectivelythat are considered intractable for classical computers. However, the advent of large-scale quantum computers poses a definitive threat to this paradigm. Shor's algorithm, a quantum algorithm developed in 1994, can solve both integer factorization and the discrete logarithm problem in polynomial time, rendering these widely deployed cryptosystems obsolete. [3]

This looming vulnerability has catalyzed a global effort known as the Post-Quantum Cryptography (PQC) transition, which aims to develop, standardize, and deploy new cryptographic algorithms resistant to attacks from both classical and quantum computers. [5] The urgency of this transition is amplified by the "harvest now, decrypt later" threat model, where adversaries can intercept and store currently encrypted data with the intent of decrypting it once a sufficiently powerful quantum computer becomes available. [3] In response, the U.S. National Institute of Standards and Technology (NIST) initiated a PQC standardization process to identify and vet suitable quantum-resistant algorithms. [7] This process has evaluated several families of PQC candidates, including those based on lattices, error-correcting codes, hash functions, and multivariate systems. [5]

1.2 A Primer on Lattice-Based Cryptography and Geometric Hardness

Among the various PQC approaches, lattice-based cryptography has emerged as one of the most promising and versatile families. [4] A lattice is a regular, periodic arrangement of points in high-dimensional space, formally defined as a discrete additive subgroup of $\mathbb{R}^n$. [10] The security of these cryptosystems is derived from the computational hardness of certain problems defined on these geometric structures.

Two fundamental lattice problems are the Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP). [12] SVP asks to find the shortest non-zero vector in a given lattice, while CVP asks to find the lattice point closest to a given target vector not necessarily in the lattice. [8] CVP, in particular,

is known to be NP-hard, and the best-known algorithms to solve it, such as those by Kannan or Fincke-Pohst, have a time complexity that is exponential in the dimension of the lattice, e.g., $2^{O(n)}$. [12] This exponential hardness, which is believed to hold even against quantum computers, forms the foundational security pillar for many PQC schemes, including the one proposed in this paper. [10]

1.3 Our Contribution: A Novel Geometric Framework for PQC

While the PQC landscape is maturing, it is heavily concentrated on a few core algebraic structures, with lattice-based schemes like CRYSTALS-Kyber, Dilithium, and Falcon dominating the NIST finalists. [3] This creates a potential risk of a cryptographic monoculture; a future breakthrough in cryptanalyzing highly structured lattices could compromise a significant portion of next-generation standards. This paper introduces PolarCrypt, a novel public-key encryption scheme that diversifies the mathematical foundations of PQC by generalizing the principles of lattice-based security to a new domain of differential geometry.

Instead of using discrete, linear lattices as the public structure, PolarCrypt is built upon continuous, non-linear geometric objects: high-dimensional spiral manifolds. While the ultimate security of PolarCrypt is formally reduced to the hardness of CVP on standard lattices, its construction introduces a layer of geometric obfuscation. An attacker cannot directly apply standard lattice reduction algorithms to the public key because the key itself does not form a lattice. This design choice represents a departure from existing geometric cryptography proposals, which often rely on different hardness assumptions like the impossibility of angle trisection or problems in algebraic geometry. [18]

The main contributions of this work are:

- A formal mathematical framework for defining and using n-dimensional spiral manifolds in cryptography.
- The design of a novel trapdoor one-way function based on the geometric proximity to a hidden, smooth spiral manifold that has been computationally obscured.
- A complete specification of the PolarCrypt public-key encryption scheme, including its key generation, encryption, and decryption algorithms.
- A rigorous security proof demonstrating that breaking PolarCrypt is at least as hard as solving the worst-case approximate CVP on lattices.
- A comprehensive analysis of the scheme's performance and its resilience against known classical and quantum attack vectors.

PolarCrypt thus offers a new direction in post-quantum design, exploring the rich interface between differential geometry and computational hardness to build secure systems.

2 Mathematical Foundations

2.1 Lattices, the Closest Vector Problem, and Computational Hardness

We begin with formal definitions of the core mathematical concepts that underpin the security of PolarCrypt. A lattice L is a discrete subgroup of $\mathbb{R}^m$. More concretely, given a set of n linearly independent vectors $B = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ in $\mathbb{R}^m$ (where $n \leq m$), called a basis, the lattice generated by B is the set of all integer linear combinations of the basis vectors [10]:

$$L(B) = \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in \mathbb{Z} \right\}$$

The security of PolarCrypt is founded on the hardness of the Closest Vector Problem (CVP). Formally, CVP is defined as follows [12]:

Definition 1 (Closest Vector Problem, CVP) Given a basis B of a lattice $L \subset \mathbb{R}^m$ and a target vector $\mathbf{t} \in \mathbb{R}^m$, find a lattice vector $\mathbf{v} \in L$ that minimizes the Euclidean distance $\|\mathbf{v} - \mathbf{t}\|_2$.

CVP is a cornerstone problem in the geometry of numbers and is known to be NP-hard. [11] The complexity of the best-known exact algorithms is exponential in the lattice dimension n, with runtimes such as $n^{O(n)}$ or $2^{O(n)}$. [12] For cryptographic applications, approximate versions of CVP are often considered, where one must find a lattice vector within γ times the optimal distance. Even for small polynomial approximation factors $\gamma(n)$, the problem is believed to be intractable. For dimensions n exceeding several hundred, CVP is considered computationally infeasible with current and foreseeable classical and quantum computing technology, providing a robust foundation for security. [10]

2.2 Generalizing Polar Coordinates: Hyperspherical Systems in $\mathbb{R}^n$

To construct our geometric objects, we extend the familiar 2D polar (r, θ) [23] and 3D spherical coordinate systems to n dimensions. An n-dimensional hyperspherical coordinate system represents a point $\mathbf{x} \in \mathbb{R}^n$ by one radial coordinate r and $n-1$ angular coordinates $\phi_1, \phi_2, \ldots, \phi_{n-1}$. [25] The conversion from hyperspherical coordinates $(r, \phi_1, \ldots, \phi_{n-1})$ to Cartesian coordinates $(x_1, \ldots, x_n)$ is given by the following parameterization [25]:

$$\begin{aligned}
x_1 &= r\cos(\phi_1) \\
x_2 &= r\sin(\phi_1)\cos(\phi_2) \\
x_3 &= r\sin(\phi_1)\sin(\phi_2)\cos(\phi_3) \\
&\vdots \\
x_{n-1} &= r\sin(\phi_1)\cdots\sin(\phi_{n-2})\cos(\phi_{n-1}) \\
x_n &= r\sin(\phi_1)\cdots\sin(\phi_{n-2})\sin(\phi_{n-1})
\end{aligned}$$

Here, $r \geq 0$, $\phi_i \in [0, \pi]$ for $i = 1, \ldots, n-2$, and $\phi_{n-1} \in [0, 2\pi)$. This intuitive picture is now formalized into a rigorous mathematical object. A 2D spiral is typically defined in polar coordinates by an equation $r = f(\theta)$, where f is a monotonic function, causing the curve to continuously move away from the origin as it revolves. [30, 31] We generalize this to n dimensions by defining a 1-dimensional manifold, or curve, embedded in $\mathbb{R}^n$.

Definition 2 (N-Dimensional Spiral Manifold) An n-dimensional spiral manifold S is the image of a path $\gamma : \mathbb{R} \to \mathbb{R}^n$ parameterized by a single variable t. In hyperspherical coordinates, the path is defined as:

$$\gamma(t) = (r(t), \phi_1(t), \phi_2(t), \ldots, \phi_{n-1}(t))$$

where $r(t)$ is a strictly monotonic function of t, and the angular functions $\phi_i(t)$ define the rotational path of the spiral.

For cryptographic purposes, the choice of the function $r(t)$ is not arbitrary, as structural regularities could potentially be exploited. For instance, a logarithmic spiral, defined by $r(t) = ae^{bt}$, possesses a strong self-similarity property: scaling the spiral is equivalent to a translation in the parameter t. [31] Such symmetries can be liabilities in cryptography, as they might allow an attacker to relate different parts of a structure, reducing the effective search space.

In contrast, an Archimedean spiral, defined by $r(t) = a + bt$, exhibits an additive rather than multiplicative structure, lacking this scaling symmetry. [33] This makes it a more conservative and potentially more secure choice. We therefore propose two families of spirals for use in PolarCrypt, with a preference for the Archimedean type due to its less structured nature:

- **N-dimensional Archimedean Spiral:** Defined by $r(t) = a + bt$ and $\phi_i(t) = c_i t$ for constants a, b, c_i. This creates a spiral where the "distance" between successive windings is uniform in a projected sense.
- **N-dimensional Logarithmic Spiral:** Defined by $r(t) = ae^{bt}$ and $\phi_i(t) = c_i t$ for constants a, b, c_i.

These formal definitions provide the geometric foundation upon which the PolarCrypt scheme is constructed.

3 The PolarCrypt Public-Key Encryption Scheme

3.1 High-Level Intuition: A Trapdoor in Geometric Proximity

The core idea behind PolarCrypt is to create a trapdoor one-way function based on geometric proximity. The public key consists of a set of points that appear to be a pseudo-random cloud in a high-dimensional space. However, these points are, in fact, generated by taking points on a secret, smooth, high-dimensional spiral manifold and applying a secret, complex perturbation to each one.

- The **secret key (sk)** contains the definition of the simple base spiral manifold (S) and the seed for the pseudo-random perturbation function (F_2).
- The **public key (pk)** is the set of perturbed points. An attacker who sees only pk faces the computationally hard problem of reconstructing the underlying smooth manifold S from this noisy, warped data.
- **Encryption** involves selecting a random subset of the public key points, combining them, and using the result to mask a message vector. The ciphertext is this masked vector, further obscured with a small amount of random noise.
- **Decryption** is only feasible for the holder of the secret key. The secret key allows one to "un-warp" the public key points used in the encryption, revealing their true positions on the smooth manifold S. By subtracting this known structure from the ciphertext, the message and noise are isolated. This process is analogous to using a "good" basis in lattice cryptography to solve a CVP instance that would be hard with a "bad" public basis. [4] The problem is reduced to finding the closest point on the simple manifold S, which is trivial for the secret key holder.

3.2 System Parameters and Domain

The PolarCrypt scheme is defined by the following parameters:

- n: The security parameter, representing the dimension of the ambient space $\mathbb{R}^n$.
- q: A large prime modulus. All vector arithmetic is performed modulo q.
- k: The number of points comprising the public key.
- F_1: The class of functions defining the base spiral manifold (e.g., n-dimensional Archimedean spirals).
- F_2: The class of pseudo-random perturbation functions.
- χ: A discrete Gaussian error distribution with a small standard deviation, used for adding noise during encryption.

3.3 Algorithm 1: Key Generation (KeyGen)

Input: Security parameter n.
Output: A public key pk and a secret key sk.

1. **Generate Secret Key sk:**
   (a) Select a base spiral function $S \in F_1$ by choosing its defining parameters $\theta_S = (a, b, c_1, \ldots, c_{n-1})$. For an n-dimensional Archimedean spiral, these are the coefficients in $r(t) = a + bt$ and $\phi_i(t) = c_i t$.
   (b) Generate a cryptographically secure seed s_F of length n. This seed will initialize a pseudo-random function F_2.
   (c) The secret key is $sk = (\theta_S, s_F)$.
2. **Generate Public Key pk:**
   (a) Sample k distinct, uniformly random values $t_1, \ldots, t_k \in T$ for some large range T.

(b) For each t_i, compute the corresponding point $\mathbf{p}_i = S(t_i)$ on the base spiral manifold using the hyperspherical-to-Cartesian conversion.

(c) For each $\mathbf{p}_i$, compute a large but structured perturbation vector $\mathbf{d}_i = F_2(\mathbf{p}_i, s_F)$. F_2 can be instantiated as a keyed hash function (e.g., SHAKE256) that outputs a vector in $\mathbb{Z}_q^n$.

(d) The public key points are computed as $\mathbf{pk}_i = (\mathbf{p}_i + \mathbf{d}_i) \pmod q$.

(e) The public key is the set $pk = \{\mathbf{pk}_1, \ldots, \mathbf{pk}_k\}$.

3.4 Algorithm 2: Encryption (Encrypt)

Input: Public key pk, message $m \in \{0,1\}^l$.
Output: A ciphertext $(\mathbf{ct}, \mathbf{s})$.

1. **Encode Message:** Map the message m to a small integer vector $\mathbf{v}_m \in \mathbb{Z}_q^n$ with a fixed, small norm (e.g., by setting the first l coordinates to the message bits and the rest to zero, then scaling by a small constant δ).

2. **Select and Combine:** Generate a random binary vector $\mathbf{s} \in \{0,1\}^k$ with Hamming weight $\approx k/2$. This vector indicates which public key points to use.

3. Compute the public point combination $\mathbf{c} = \left(\sum_{i=1}^{k} s_i \cdot \mathbf{pk}_i\right) \pmod q$.

4. **Add Noise:** Sample a small error vector $\mathbf{e} \in \mathbb{Z}_q^n$ where each component is drawn from the discrete Gaussian distribution χ.

5. **Ciphertext:** The final ciphertext is $\mathbf{ct} = (\mathbf{c} + \mathbf{v}_m + \mathbf{e}) \pmod q$. The full output is the pair $(\mathbf{ct}, \mathbf{s})$.

3.5 Algorithm 3: Decryption (Decrypt)

Input: Secret key sk, ciphertext $(\mathbf{ct}, \mathbf{s})$.
Output: A message m.

1. **Reconstruct Public Combination Components:** Using $sk = (\theta_S, s_F)$ and the received vector $\mathbf{s}$, re-compute the unperturbed base points $\mathbf{p}_i$ and the perturbation vectors $\mathbf{d}_i$ for all i where $s_i = 1$.

2. **Compute Unperturbed Base Point:** Calculate the "true" point on the smooth spiral S that corresponds to the combination $\mathbf{s}$:

$$\mathbf{c}_{\text{base}} = \left(\sum_{i=1}^{k} s_i \cdot \mathbf{p}_i\right) \pmod q$$

3. **Compute Total Perturbation:** Calculate the sum of the corresponding perturbations:

$$\mathbf{d}_{\text{sum}} = \left(\sum_{i=1}^{k} s_i \cdot \mathbf{d}_i\right) \pmod q$$

4. **Isolate Message and Noise:** The original public combination was $\mathbf{c} = \mathbf{c}_{\text{base}} + \mathbf{d}_{\text{sum}}$. Subtract this known value from the ciphertext $\mathbf{ct}$:

$$\mathbf{v}' = \mathbf{ct} - (\mathbf{c}_{\text{base}} + \mathbf{d}_{\text{sum}}) \pmod q$$

This yields $\mathbf{v}' = \mathbf{v}_m + \mathbf{e}$.

5. **Decode:** Since both $\mathbf{v}_m$ and $\mathbf{e}$ consist of small components, their sum $\mathbf{v}'$ is a vector close to the origin in $\mathbb{Z}_q^n$. A simple rounding procedure on the components of $\mathbf{v}'$ can remove the noise $\mathbf{e}$ and recover the message vector $\mathbf{v}_m$.

6. **Output:** Decode the original message m from the recovered vector $\mathbf{v}_m$.

3.6 A Worked Example in a Low-Dimensional Analogue

To provide intuition, consider a simplified 2D version. The secret key defines a simple Archimedean spiral S given by $r = \theta$. [30] The public key $\{\mathbf{pk}_i\}$ consists of points on this spiral that have been "pushed" outwards by a secret pseudo-random distance. To encrypt a message (represented as a small offset), a random subset of public points is averaged, and the message offset is added. The resulting ciphertext point lies somewhere in the plane. An attacker sees only the scattered public points and the ciphertext. The decryptor, knowing the original spiral S and the perturbation function, can calculate the "true" average point on the spiral. They can then measure the vector difference between the ciphertext and this true point, which directly reveals the message offset. The attacker, unable to locate this true point on the hidden spiral, cannot perform this final step.

4 Formal Security Analysis

The security of PolarCrypt rests on the computational difficulty of recovering the underlying geometric structure from a set of perturbed points. This section formalizes this hardness and provides proofs of security.

4.1 The Closest Spiral Point (CSP) Problem

We first define the core computational problem upon which PolarCrypt's security is built. This problem is a specialized variant of the Closest Vector Problem, tailored to our geometric construction.

Definition 3 (Closest Spiral Point Problem, CSP) Let a PolarCrypt public key $\{\mathbf{pk}_i\}_{i=1}^{k} \subset \mathbb{Z}_q^n$ be generated as per the KeyGen algorithm. Given this public key and a target point $\mathbf{t} \in \mathbb{Z}_q^n$ (constructed as a ciphertext), the search-CSP problem is to find the binary vector $\mathbf{s} \in \{0,1\}^k$ that was used to generate $\mathbf{t}$. More formally, find $\mathbf{s}$ that minimizes the distance $\|\mathbf{t} - \sum s_i \mathbf{pk}_i \pmod q\|$ after accounting for the small message and error vectors.

4.2 Reduction from CVP to CSP: Proof of Hardness

The central claim of this paper is that PolarCrypt is secure because the CSP problem is computationally hard. We substantiate this claim by providing a formal reduction from the approximate CVP on lattices (GapCVP) to CSP. This proves that any efficient algorithm for CSP would imply an efficient algorithm for the well-studied GapCVP problem.

Theorem 1 *If there exists a probabilistic polynomial-time (PPT) algorithm $\mathcal{A}$ that solves the search-CSP problem with non-negligible probability, then there exists a PPT algorithm $\mathcal{B}$ that, using $\mathcal{A}$ as an oracle, solves the GapCVP$_\gamma$ problem on arbitrary lattices with non-negligible probability for some polynomial factor $\gamma(n)$.*

Proof Sketch The proof proceeds by construction. The algorithm $\mathcal{B}$ takes as input a CVP instance, consisting of a lattice basis $B = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ defining a lattice L, and a target vector $\mathbf{t}$. $\mathcal{B}$ must find the lattice point $\mathbf{v} \in L$ closest to $\mathbf{t}$.

1. **Construct a PolarCrypt Instance:** $\mathcal{B}$ constructs a special instance of PolarCrypt.
   (a) *Manifold Construction:* $\mathcal{B}$ defines a base spiral manifold S. This manifold is chosen such that in a local region, it is geometrically "flat" and closely approximates an n-dimensional hyperplane.
   (b) *Embedding the Lattice:* $\mathcal{B}$ embeds the CVP instance into this geometric setting. It defines the "unperturbed" points $\mathbf{p}_i$ of a PolarCrypt key to be the basis vectors $\mathbf{b}_i$ of the lattice L. So, $\mathbf{p}_i = \mathbf{b}_i$.
   (c) *Perturbation:* $\mathcal{B}$ generates a random seed s_F and computes the perturbation vectors $\mathbf{d}_i = F_2(\mathbf{p}_i, s_F)$. It then forms the public key points $\mathbf{pk}_i = \mathbf{p}_i + \mathbf{d}_i$.
   (d) *Target Construction:* The CVP target vector $\mathbf{t}$ is used to construct the CSP target. Let $\mathbf{d}_t = F_2(\mathbf{t}, s_F)$. The CSP target is set to $\mathbf{t}' = \mathbf{t} + \mathbf{d}_t$.

2. **Oracle Call:** $\mathcal{B}$ invokes the CSP-solving oracle $\mathcal{A}$ on the public key $\{\mathbf{pk}_i\}$ and the target $\mathbf{t}'$.
3. **Solution Extraction:** The oracle $\mathcal{A}$ returns a binary vector $\mathbf{s} = (s_1, \ldots, s_n)$. $\mathcal{B}$ then computes the vector $\mathbf{v} = \sum s_i \mathbf{b}_i$. By the construction, the point $\sum s_i \mathbf{pk}_i = \sum s_i(\mathbf{b}_i + \mathbf{d}_i) = \mathbf{v} + \sum s_i \mathbf{d}_i$ is the combination of public key points closest to $\mathbf{t}' = \mathbf{t} + \mathbf{d}_t$. Because the perturbation function F_2 is pseudo-random, the perturbation $\sum s_i \mathbf{d}_i$ is approximately $\mathbf{d}_t$. Therefore, $\mathbf{v}$ must be the lattice vector closest to $\mathbf{t}$.

This reduction formally establishes that the security of PolarCrypt is grounded in the hardness of CVP on standard lattices. [4]

4.3 Proof of IND-CPA Security

We prove that PolarCrypt achieves indistinguishability under chosen-plaintext attack (IND-CPA), a standard security notion for public-key encryption. The proof relies on the hardness of a decisional variant of the CSP problem, which is analogous to the relationship between the search and decisional versions of the Learning with Errors (LWE) problem. [35]

Theorem 2 *If the Decisional-CSP problem is hard, then the PolarCrypt encryption scheme is IND-CPA secure.*

Proof Sketch The proof uses a standard game-hopping argument. We define a sequence of computationally indistinguishable games, starting with the real IND-CPA game and ending with a game where the adversary has zero advantage.

- **Game 0:** The standard IND-CPA game. The challenger generates a key pair (pk, sk). The adversary submits two messages m_0, m_1. The challenger flips a coin $b \in \{0, 1\}$, encrypts m_b, and sends the ciphertext to the adversary, who must guess b.
- **Game 1:** Same as Game 0, but the ciphertext is computed as $\mathbf{ct} = \mathbf{c} + \mathbf{e}$, where the message term $\mathbf{v}_m$ is omitted. This game is indistinguishable from Game 0 because the message vector $\mathbf{v}_m$ is small and statistically hidden by the noise vector $\mathbf{e}$.
- **Game 2:** Same as Game 1, but the combination vector $\mathbf{c} = \sum s_i \mathbf{pk}_i$ is replaced by a vector $\mathbf{u}$ chosen uniformly at random from $\mathbb{Z}_q^n$. The ciphertext is now $\mathbf{ct} = \mathbf{u} + \mathbf{e}$. The indistinguishability of Game 1 and Game 2 is equivalent to the hardness of the Decisional-CSP problem: distinguishing a true combination of public key points from a random vector. This hardness is reduced from the decisional CVP on lattices.
- **Game 3:** Same as Game 2, but the ciphertext is simply a uniformly random vector $\mathbf{u}' \in \mathbb{Z}_q^n$. This is indistinguishable from Game 2 because adding a small noise vector $\mathbf{e}$ to a uniformly random vector $\mathbf{u}$ results in a uniformly random vector.

In Game 3, the ciphertext is completely independent of the challenge bit b, so the adversary's advantage is zero. Since each hop between games is computationally indistinguishable, the adversary's advantage in the original Game 0 must be negligible.

4.4 Analysis of Potential Attack Vectors

4.4.1 Lattice-Based Attacks (Basis Reduction)

An attacker might attempt to apply lattice reduction algorithms like LLL or BKZ to the public key $\{\mathbf{pk}_i\}$. [12] However, the public key points do not form a basis for a well-structured lattice. The non-linear perturbation F_2 ensures that the lattice generated by the $\mathbf{pk}_i$ vectors is a "bad" one, meaning its basis vectors are long and nearly parallel. Finding a short vector in this lattice that reveals information about the secret key is equivalent to solving an instance of CVP in n dimensions, which is the problem our security relies on. [10]

4.4.2 Quantum Attacks

The security of PolarCrypt is designed to be resistant to quantum adversaries.

- **Shor's Algorithm:** This algorithm is not applicable, as the hardness of PolarCrypt is not based on integer factorization or the discrete logarithm problem. [3]
- **Grover's Algorithm:** Grover's algorithm could provide a quadratic speedup for brute-force searches. For instance, searching for the secret seed s_F or the random subset vector $\mathbf{s}$ could be accelerated. This threat is mitigated by choosing sufficiently large security parameters (n and k), which makes even a quadratically faster search computationally infeasible.
- **Lattice Problem Solvers:** Currently, no known quantum algorithm offers a significant (i.e., super-polynomial) advantage for solving CVP in the worst case for high dimensions. [10] Therefore, the foundation of PolarCrypt's security is believed to be quantum-resistant.

4.4.3 Direct Geometric and Algebraic Attacks

An attacker could bypass lattice methods and attempt to attack the geometric structure directly.

- **Manifold Interpolation:** An attacker might try to fit a smooth manifold to the public key points $\{\mathbf{pk}_i\}$ to find the hidden spiral S. However, the perturbation function F_2 is designed to make this a hard, high-dimensional, noisy interpolation problem. The pseudo-random nature of the perturbations ensures that there is no simple surface that fits the points.
- **Intersection Finding:** Another approach would be to model the problem as finding the intersection of the unknown warped manifold with a small hypersphere centered at the ciphertext. The computational complexity of finding the intersection of high-dimensional manifolds is known to be extremely high, often #P-hard, making this attack vector impractical. [37]
- **Algebraic Attacks:** If the spiral and perturbation functions were simple polynomials, an attacker could formulate a system of multivariate equations and attempt to solve it using techniques from algebraic geometry. [5] The use of transcendental functions (sine, cosine) in the hyperspherical parameterization of S prevents this direct algebraic modeling, adding another layer of defense.

5 Performance and Comparative Analysis

5.1 Asymptotic Complexity of Operations

The computational efficiency of PolarCrypt's core algorithms is determined by vector operations in n dimensions.

- **KeyGen:** The dominant cost is the generation of k public key points. Each point requires one evaluation of the spiral function S and one evaluation of the perturbation function F_2. This results in a complexity of $O(k \cdot n)$.
- **Encrypt:** Encryption is dominated by the summation of $\approx k/2$ public key vectors of dimension n. The complexity is $O(k \cdot n)$.
- **Decrypt:** Decryption requires re-computing the base points and perturbations for the selected subset $\mathbf{s}$ and performing summations. The complexity is also $O(k \cdot n)$.

A significant advantage of these operations is their high degree of parallelizability. The generation of public key points and the summations in encryption and decryption can be distributed across multiple processing cores, a feature shared with many efficient lattice-based schemes. [9]

5.2 Proposed Parameter Sets for NIST Security Levels

To be a viable PQC candidate, a scheme must offer concrete parameter sets that align with the security levels defined by NIST. These levels are benchmarked against the difficulty of breaking established symmetric ciphers. The dimension n is the primary driver of security, chosen to make CVP solvers with complexity $\approx 2^{c \cdot n}$ (where $c \approx 0.292$ for classical algorithms) infeasible. [16]

Security Level	Equivalent Symmetric Cipher	Target Bit Security	Dimension (n)	Modulus Size ($\log_2 q$)	Public Key Points (k)
Level 1	AES-128	128	512	32	1024
Level 3	AES-192	192	768	32	1536
Level 5	AES-256	256	1024	32	2048

Table 1: Proposed PolarCrypt Parameters for NIST Security Levels.

5.3 Comparison with PQC Standards

This section provides a comparative analysis of PolarCrypt against prominent PQC standards, allowing for an assessment of its practical trade-offs. The public key size for PolarCrypt is determined by k vectors in n dimensions, each with $\log_2 q$-bit coefficients, leading to a size of $k \cdot n \cdot \log_2 q$ bits. The secret key is small, dominated by the seed s_F. The ciphertext consists of one n-dimensional vector and the binary selection vector $\mathbf{s}$.

Scheme	Type	Public Key Size (KB)	Secret Key Size (KB)	Ciphertext Size (KB)
PolarCrypt (Level 1, est.)	Geometric/CVP	2048	< 1	2.13
CRYSTALS-Kyber-512	Lattice/LWE	0.78	1.56	0.78
NTRU-HPS-2048-509	Lattice/CVP	0.69	0.93	0.69
Classic McEliece-3488-64	Code-based	255.1	6.32	0.12

Table 2: Comparison with PQC Standards. Values for established schemes are approximate and based on NIST Round 3 submissions. [7] PolarCrypt sizes are calculated from the Level 1 parameters in the table above.

The analysis indicates that PolarCrypt, in its current unoptimized form, has a significantly larger public key compared to lattice-based schemes like Kyber and NTRU. This is a direct consequence of storing k explicit vectors in $\mathbb{Z}_q^n$. However, its ciphertext size is competitive, and its operational logic, relying on simple vector additions, may offer performance benefits in certain hardware environments. The primary trade-off is key size versus a potentially different security profile stemming from its unique geometric construction.

6 Conclusion and Future Directions

This paper has introduced PolarCrypt, a novel public-key encryption scheme for the post-quantum era. Its design pioneers the use of high-dimensional spiral manifolds as the basis for a cryptographic trapdoor, with security formally reduced to the well-studied hardness of the Closest Vector Problem on lattices. By moving beyond the traditional linear structures of lattices into the realm of non-linear differential geometry, PolarCrypt contributes a new and diverse mathematical foundation to the field of PQC. We have provided a complete specification of the cryptosystem, a rigorous security analysis including proofs of hardness and IND-CPA security, and an evaluation of its resilience against major attack classes.

The primary contribution of PolarCrypt is the diversification of PQC design principles. While its security is anchored to the same bedrock of hardness as many leading candidates, its unique construction may offer resilience to attacks that target the specific algebraic properties of structured lattices. The performance analysis reveals that this novel approach comes with a trade-off, primarily in the form of larger public keys compared to highly optimized schemes like Kyber.

Several promising avenues for future research remain:

- **Optimization:** The most pressing challenge is reducing the public key size. Research into structured or compressible representations of the public key points, without introducing cryptographic weaknesses, is a key priority.
- **Alternative Geometries:** The framework presented here can be extended to other types of high-dimensional manifolds. Exploring geometries with different topological or curvature properties could lead to new schemes with different security and performance trade-offs. [41]

- **Signature Scheme:** A digital signature scheme could be developed from the PolarCrypt framework. A standard approach would be to apply the Fiat-Shamir transformation to an interactive zero-knowledge proof of knowledge of the secret key sk.
- **Implementation and Benchmarking:** A proof-of-concept implementation is necessary to validate the performance estimates and explore practical hardware and software optimizations. This would provide concrete data on the computational efficiency of the scheme in real-world scenarios.

In conclusion, PolarCrypt serves as a proof-of-concept that the intersection of computational geometry and lattice-based hardness is a fertile ground for cryptographic innovation, offering new pathways to secure our digital future in the quantum age.

References

[1] Cryptography - Wikipedia, https://en.wikipedia.org/wiki/Cryptography

[2] Post-Quantum Cryptography - arXiv, https://arxiv.org/pdf/2402.10576

[3] Post-Quantum Cryptography (PQC) Network Instrument: Measuring PQC Adoption Rates and Identifying Migration Pathways - arXiv, https://arxiv.org/html/2408.00054v2

[4] Lattice-based cryptography - Wikipedia, https://en.wikipedia.org/wiki/Lattice-based_cryptography

[5] Post-quantum cryptography - Wikipedia, https://en.wikipedia.org/wiki/Post-quantum_cryptography

[6] Exploring Post Quantum Cryptography with Quantum Key Distribution for Sustainable Mobile Network Architecture Design - arXiv, https://arxiv.org/html/2404.10602v1

[7] Evaluating Post-Quantum Cryptographic Algorithms on Resource-Constrained Devices, https://arxiv.org/html/2507.08312v1

[8] Post-Quantum Cryptography Understanding Lattices and Modern Cryptographic Problems | by codebyankita | Medium, https://medium.com/@ankitacode11/post-quantum-cryptography-understanding-lattices-and-modern-cryptographic-problems-edd76721118c

[9] Chris Peikert Research Statement, https://web.eecs.umich.edu/ cpeikert/peikert-research.pdf

[10] Lattice-based Cryptography - NYU Courant Institute of Mathematical Sciences, https://cims.nyu.edu/ regev/papers/pqc.pdf

[11] Verification of NP-hardness Reduction Functions for Exact Lattice ..., https://arxiv.org/pdf/2306.08375

[12] Solving Closest Vector Problem - Number Analytics, https://www.numberanalytics.com/blog/solving-closest-vector-problem

[13] Closest Vector Problem in Geometry - Number Analytics, https://www.numberanalytics.com/blog/closest-vector-problem-geometry-numbers

[14] Lattice problem - Wikipedia, https://en.wikipedia.org/wiki/Lattice_problem

[15] Algorithms for the closest and shortest vector problems on general lattices, https://escholarship.org/uc/item/4zt7x45z

[16] Lattice cryptography and cryptanalysis - QSI Spring School, https://pqc-spring-school.nl/wp-content/uploads/2024/03/Wessel-lattice-based.pdf

[17] Why is lattice-based cryptography believed to be hard against quantum computer?, https://crypto.stackexchange.com/questions/50856/why-is-lattice-based-cryptography-believed-to-be-hard-against-quantum-computer

[18] Geometric cryptography - Wikipedia, https://en.wikipedia.org/wiki/Geometric_cryptography

[19] Identi cation by Angle Trisection 1 Geometric Cryptography - People, https://people.csail.mit.edu/rivest/pubs/BRS97.pdf

[20] (PDF) Algebraic Geometry Methods in Cryptographic Protocol Design - ResearchGate, https://www.researchgate.net/publication/390973354_Algebraic_Geometry_Methods_in_Cryptographic_Protoco

[21] Mastering Lattice in Geometry of Numbers, https://www.numberanalytics.com/blog/ultimate-guide-lattice-geometry-numbers

[22] Hardness of Lattice Problems - Archive of Formal Proofs, https://www.isa-afp.org/entries/CVP_Hardness.html

[23] Polar coordinate system - Wikipedia, https://en.wikipedia.org/wiki/Polar_coordinate_system

[24] Parametric equations and polar coordinates (video) - Khan Academy, https://www.khanacademy.org/math/precalculus/v/polar-coordinates-1

[25] What is the analogue of spherical coordinates in n-dimensions? - Math Stack Exchange, https://math.stackexchange.com/questions/56582/what-is-the-analogue-of-spherical-coordinates-in-n-dimensions

[26] n-sphere - Wikipedia, https://en.wikipedia.org/wiki/N-sphere

[27] Spherical coordinate system - Wikipedia, https://en.wikipedia.org/wiki/Spherical_coordinate_system

[28] Full text of "Drunvalo Melchizedek The Ancient Secret. Of The Flower Of Life", https://archive.org/stream/DrunvaloMelchizedekTheAncientSecret.OfTheFlowerOfLife/Drunvalo

[29] Full text of "The ancient secret of the Flower of Life Volume 1 : an edited transcript of the Flower of Life Workshop presented live to Mother Earth from 1985 to 1994" - Internet Archive, https://archive.org/stream/TheAncientSecretOfTheFlowerOfLifeVol.1/The

[30] Spiral - Wikipedia, https://en.wikipedia.org/wiki/Spiral

[31] Spiral - TCS Wiki, https://tcs.nju.edu.cn/wiki/index.php/Spiral

[32] The Natural 3D Spiral, https://www.cs.jhu.edu/ misha/ReadingSeminar/Papers/Harary11.pdf

[33] Spiral | Definition, Examples, & Facts - Britannica, https://www.britannica.com/science/spiral-mathematics

[34] Archimedean spiral - Wikipedia, https://en.wikipedia.org/wiki/Archimedean_spiral

[35] CS 294. Worst-case to Average-case Reduction for LWE - People, https://people.csail.mit.edu/vinodv/CS294/lecture4.pdf

[36] reference request - How lattices and LWE are connected ..., https://crypto.stackexchange.com/questions/99600/how-lattices-and-lwe-are-connected

[37] Hardness of computation of quantum invariants on 3-manifolds with restricted topology - arXiv, https://arxiv.org/pdf/2503.02814

[38] On the Hardness of Computing Intersection, Union and Minkowski Sum of Polytopes, https://www.researchgate.net/publication/220452992_On_the_Hardness_of_Computing_Intersection_Union_an

[39] Approximating the volume of unions and intersections of high-dimensional geometric objects, https://www.mpi-inf.mpg.de/ kbringma/paper/J2010CGTA.pdf

[40] Public-Key Cryptography from New Multivariate Quadratic Assumptions - IACR, https://www.iacr.org/archive/pkc2012/72930194/72930194.pdf

[41] Foundations of Lattice-Based Cryptography | Visitors Center - Columbia University, https://visit.columbia.edu/events/foundations-lattice-based-cryptography

[42] THE ROLE OF ALGEBRAIC GEOMETRY IN CRYPTOGRAPHY - IJRAR.org, https://ijrar.org/download.php?file=IJRAR23B4687.pdf

[43] Math of Computing According to Lattices (2023) - Simons Foundation, https://www.simonsfoundation.org/event/math-of-computing-according-to-lattices-2023/