Review

REVIEW ON IPQ-LWE-TCFs

Firstname Lastname [1,†,‡], Firstname Lastname [2,‡] and Firstname Lastname [2,*]

[1] Affiliation 1; e-mail@e-mail.com
[2] Affiliation 2; e-mail@e-mail.com
* Correspondence: e-mail@e-mail.com; Tel.: (optional; include country code; if there are multiple corresponding authors, add author initials) +xx-xxxx-xxx-xxxx (F.L.)
† Current address: Affiliation.
‡ These authors contributed equally to this work.

Keywords: interactive proof of quantumness, trapdoor claw-free function, lattice, lwe

1. Introduction

1.1. Overview of Post-quantum Cryptography

The need for secure communication has been a constant throughout history, driving the development of cryptographic techniques to protect sensitive information.

The security of classical cryptography largely relies on mathematical problems assumed to be computationally hard to solve. These problems, such as integer factorization and the discrete logarithm problem, form the basis of widely used algorithms like RSA and ECC. However, the development of quantum computing presents a significant challenge. Shor's algorithm [1], for example, can efficiently solve the problems as mentioned above, making these cryptographic primitives vulnerable to future quantum attacks. While not all classical cryptographic primitives are vulnerable to these attacks, RSA/ECC form the basis of contemporary public-key infrastructure, essential to the functionality of the internet.

In recent years, the rapid development of quantum computing has raised major security concerns in these cryptographic schemes. Consequently, post-quantum cryptography (PQC) is now a thriving field with the view of developing new cryptographic constructions secure against both classical and quantum attacks, often referred to as 'quantum resistance'. In the context of post-quantum cryptography, there are many approaches using different building blocks: *hash-based, code-based, multivariate-based, isogeny-based and lattice-based schemes:*

- Hash-based cryptography: Utilising the hard-to-invert property of hash functions to build digital signature schemes.
- Code-based constructions derive their security from decoding generic linear codes.
- Multivariate-based cryptography uses the NP-hard Multivariate Quadratic Problem (MQ Problem) over finite fields $\mathbb{F}_p$ to construct cryptosystems.
- Isogeny-based cryptography is the youngest field whose constructions are based on the hardness of finding special maps called isogenies between supersingular elliptic curves.
- Lattice-based cryptography is based on the computational difficulty of mathematical problems associated with lattices: short integer solution (SIS), learning with errors (LWE), and their variants.

Lattice-based cryptography is arguably the most promising branch in post-quantum cryptography today due to its versatility, supporting a wide range of symmetric and asymmetric cryptographic primitives including digital signatures and public-key encryption. Additionally, lattice-based techniques are computationally efficient with compact key sizes.

Citation: Lastname, F.; Lastname, F.; Lastname, F. Title. *Cryptography* **2024**, *1*, 0. https://doi.org/

Received:
Revised:
Accepted:
Published:

1.2. Overview of Quantum Computing

1.2.1. Quantum computing and quantum cryptography

Quantum computing

Quantum cryptography has a major impact on symmetric key encryption constructions since it can achieve information-theoretic security. However, this does not apply to asymmetric key encryption because one of the keys always remains public. For asymmetric constructions, developing post-quantum cryptography (PQC) is still the main research goal. Based on "hard-to-solve" assumptions, PQC faces the same problem as classical cryptography, i.e., computational security.

The scope of this review, however, lies in the proof of quantumness, which is distinct from these two problems. Interactive Proofs of Quantumness (IPQ) are used to establish a connection between quantum and classical devices. Note that Trapdoor Claw-free functions still base their security on hard problems like learning with errors (LWE) and ring learning with errors (RLWE), making them computationally secure, not information-theoretic secure.

1.2.2. Communication between a quantum computer and classical devices.

1.3. Notation

The following notations are used throughout this paper:

- q: prime integer;
- λ: security parameter;
- $\mathbb{N}$: the set of natural numbers;
- $\mathbb{Z}$: the set of integers;
- $\mathbb{Z}_p$: the ring of integers modulo p;
- $x \leftarrow S$: the action of sampling a uniformly random element x from the set S;
- $p(\lambda)$ a polynomial function associated with the security parameter λ. A function $\text{negl}(\lambda)$ is said to be negligible if $\text{negl}(\lambda)$ is smaller than all polynomial fractions for sufficiently large λ. We define an event to occur with overwhelming probability when its probability of occurrence is at least $1 - \text{negl}(\lambda)$.
- Vectors are represented using lowercase bold letters (e.g. $\mathbf{s}$), while matrices are represented in uppercase bold (e.g. $\mathbf{A}$).
- The inner product of vectors is denoted using angular brackets, e.g. $\langle \mathbf{a}, \mathbf{s} \rangle$.
- The transpose of a vector or matrix is denoted using a superscript 'T', e.g. $\mathbf{s}^T$ or $\mathbf{A}^T$, respectively.
- $\| \cdot \|$ denotes the Euclidean norm, $\| \cdot \|_\infty$ denotes the infinity norm (the absolute value of the largest component of the vector)
- $\mathcal{L}(\mathbf{A})$: a lattice with basis $\mathbf{A}$.
- For a function $f : X \to \mathbb{R}$ over a finite domain X, the support of f, denoted by SUPP, is the set of points in X where f is non-zero,

$$\mathsf{SUPP}(f) = \{x \in X | f(x) \neq 0\}.$$

2. Quantum computing and communications

2.1. Quantum states

In quantum mechanics, a *quantum state* is a unit vector in the complex space $\mathbb{C}^N$. As in classical computing, all computations are performed using classical bits, 0 and 1. In quantum computing, we use an equivalent term that has more interesting properties: the *quantum bit* or *qubit*. A qubit, a two-level system, can be considered the simplest unit of quantum information. Let us use Dirac notation and denote $|0\rangle$ and $|1\rangle$ as the two basis states of a qubit, similar to the classical case. The $|0\rangle$ and $|1\rangle$ are called 'kets', more explanation about the use of this notation will be proposed soon in this section.

The first special property of a quantum state is *superposition*, where a quantum system is not necessarily in one specific state but can be in a *superposition* of them. Hence, a quantum state $|\psi\rangle$ can be expressed as a linear combination of the two basis states $|0\rangle$ and $|1\rangle$,

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle . \tag{1}$$

Note that the coefficients α and β are complex numbers called amplitudes. They represent the probability amplitudes of getting the results 0 and 1 if one measures the state. We have that

$$\begin{aligned} \Pr[0] &= |\alpha|^2, \\ \Pr[1] &= |\beta|^2, \\ |\alpha|^2 + |\beta|^2 &= 1. \end{aligned} \tag{2}$$

One simple example is that if one has the

This "weird" property of qubits makes the idea of quantum computers more promising since with n classical bits, one can perform up to n computations, while with n qubits, this number could reach up to 2^n computations Not computations, rather terms in the superposition.

Recall that quantum states can be written as vectors in complex space; hence, using the Dirac notation for describing quantum states simplifies linear algebra operations while working with these states. Note that kets are column vectors, for example:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \ |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{3}$$

The superposition state (1) can be represented as the following vector:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}. \tag{4}$$

Working with row vectors in complex vector space, we also need to deal with *conjugate transpose*, ie. row vectors. In the Dirac notation, these row vectors are called "bras", for example

$$\langle 0| = \begin{bmatrix} 1 & 0 \end{bmatrix}, \ \langle 1| = \begin{bmatrix} 0 & 1 \end{bmatrix}, \tag{5}$$

and

$$\langle \psi| = \begin{bmatrix} \alpha^* & \beta^* \end{bmatrix} \tag{6}$$

where α^* and β^* are complex conjugates of α and β.

2.2. Measurement basis

2.3. No-cloning theorem

3. Lattice-based post-quantum cryptography

3.1. Lattices

3.1.1. Definitions

Definition 1. *Lattices in $\mathbb{R}^n$ are discrete subgroup of $\mathbb{R}^n$. Consider integer lattice $\mathcal{L} \subseteq \mathbb{Z}^n$, $\mathcal{L}$ can be represented by a basis $\mathbf{B} = [\mathbf{b}_1, \cdots, \mathbf{b}_n] \in \mathbb{Z}^{m \times n}$, where each vector $\mathbf{b}_i$ is written in column form as follow:*

$$\mathcal{L}(\mathbf{B}) := \left\{ \sum_{i=1}^{n} \mathbf{b}_i x_i | x_i \in \mathbb{Z} \ \forall i = 1, \cdots, n \right\} \subseteq \mathbb{Z}^m.$$

We refer to n as the rank of the lattice $\mathcal{L}$ and m as its dimension. $\mathcal{L}$ is called a full-rank lattice if $n = m$.

For a basis $\mathbf{B}$ of $\mathcal{L}$, we called $P(\mathbf{B}) = \{\mathbf{B}\mathbf{x} | \mathbf{x} \in [0,1)^n\}$ is the fundamental parallelepiped of $\mathbf{B}$. A "good" basis of $\mathcal{L}$ results in a parallelepiped that is more square-like, whereas a "bad" basis produces a very thin parallelepiped. We denote $\|\mathbf{B}\| := \max_{1 \le i \le k} \|\mathbf{b}_i\|$ the maximum l_2 length of the vectors in $\mathbf{B}$ and $\tilde{\mathbf{B}} := \{\tilde{\mathbf{b}}_1, \cdots, \tilde{\mathbf{b}}_k\}$ the Gram-Schmidt orthogonalization of the vectors $\mathbf{b}_1, \cdots, \mathbf{b}_k$ in that order.

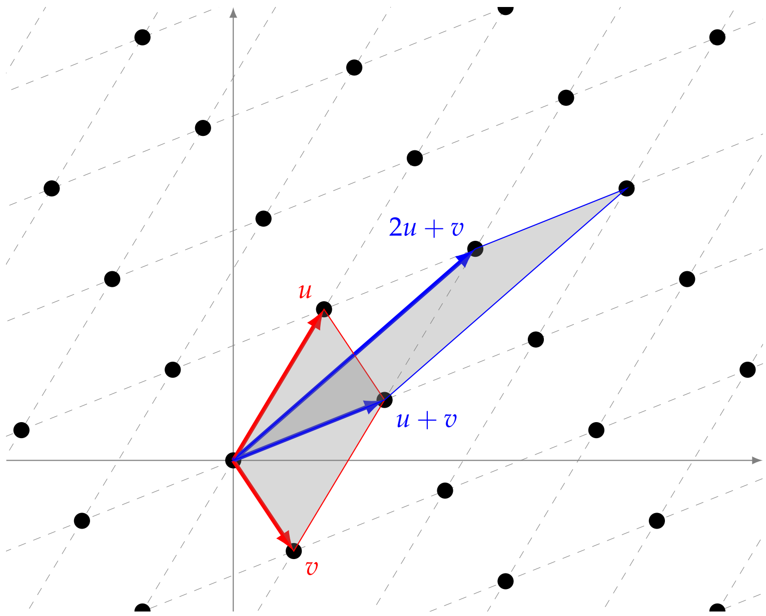


Figure 1: Consider a two-dimensional lattice $\mathcal{L}$, a basis for $\mathcal{L}$ consists of two non-zero vectors u and v such that any vector in $\mathcal{L}$ can be written as a linear combination of u and v. A good basis for $\mathcal{L}$ will have vectors with lengths that are close to each other and an angle between them that is close to 90 degrees.

For a given lattice $\mathcal{L}$, let denote $\mathbf{s}$ is the shortest vector of $\mathcal{L}$; one basic parameter of a lattice is the length of this shortest vector, denoted as $\lambda_1 = \|\mathbf{s}\|$. This parameter is also called *successive minima* if one considers the scenario where λ_1 is the smallest r such that all the lattice points inside a ball of radius r $\overline{\mathcal{B}}(0,r)$ span a space of dimension 1. Similarly, we can define the n^{th} successive minimum of $\mathcal{L}(\mathbf{A})$ as the smallest number r that all the lattice points inside $\overline{\mathcal{B}}(0,r)$ span a space of dimension n. More concretely, $\lambda_n(\mathcal{L}) = \inf\{r | \dim(\mathsf{span}(\mathcal{L} \cap \overline{\mathcal{B}}(0,r))) \le n\}$.

Hence, one of the most important problems in lattice-based cryptography is the Shortest Vector Problem (SVP), which asks to find the shortest nonzero vector $\mathbf{s}$ in a given lattice $\mathcal{L}(\mathbf{A})$ with an arbitrary basis $\mathbf{A}$. There are also approximate versions of this problem called SVP$_\gamma$ where the goal is to find a nonzero vector that is of length at most $\gamma = \gamma(n) \ge 1$ times the length of the optimal solution.

The "decision" versions of the problem ask to determine whether a given number d is the minimum distance of $\mathcal{L}(\mathbf{A})$, i.e., $d = \min_{\mathbf{0} \ne \mathbf{v} \in \mathcal{L}} \|\mathbf{v}\|$. This problem is known to be NP-hard in the $\|\cdot\|_\infty$ norm, meaning there is no known polynomial-time algorithm to solve it [2]. For the $\|\cdot\|$ Euclid norm, it remains a long-standing open problem. Another important problem is the Shortest Independent Vectors Problem (SIVP$_\gamma$) with an approximation factor γ, a given a lattice $\mathcal{L}(\mathbf{A})$ with an arbitrary basis $\mathbf{A}$, SIVP$_\gamma$ asks to find find n linearly independent lattice vectors of length at most $\gamma(n) \cdot \lambda_n(\mathcal{L})$ where $\lambda_n(\mathcal{L})$ is the n^{th} successive minimum of $\mathcal{L}$. Like SVP, SIVP is also NP-hard.

Many hardness problems in lattice-based cryptography can be **"reduced"** to instances of SVP or SIVP, for more information about reductions between lattice problems, please refer to 3.1.2 and [3].

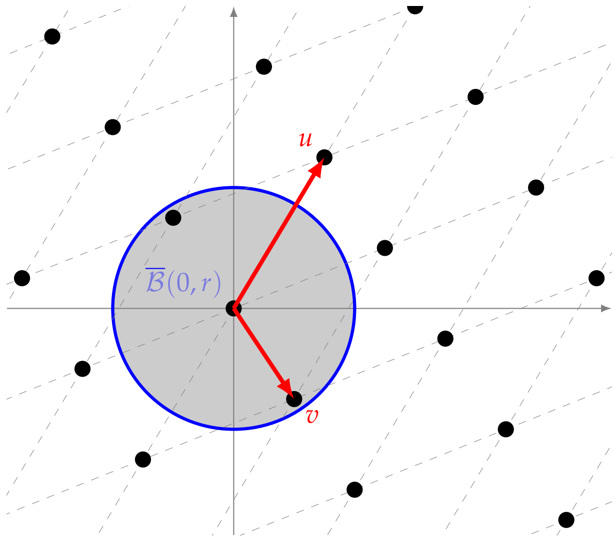


Figure 2: A lattice $\mathcal{L}$ with the successive minimum $\lambda_1 = \|v\| = r$ where all the lattice points inside a ball of radius r $\overline{\mathcal{B}}(0,r)$ span a space of dimension 1.

The above intro on lattices we might expand upon to make it more accessible to a non-specialist audience.

3.1.2. Reductions

Within the realm of computational hardness challenges, the term "reduction" denotes a methodological approach whereby one problem is transformed into another, such that solving the second problem enables solving the first. This is a formal way to compare the intrinsic difficulty of different problems. Specifically, if problem A can be reduced to problem B, and if one has a solution to B, then A can be solved. This doesn't necessarily imply that A is as hard as B, but that B is at least as hard as A.

In the analysis of problem hardness, distinctions are made between *worst-case* and *average-case* scenarios. The *worst-case* scenario is the most difficult possible instance for a given problem while the *average-case* looks at the expected performance of an algorithm across all possible instances of a problem, usually under some reasonable assumption about the distribution of these instances.

In Post-Quantum Cryptography (PQC), reductions are the main technique used to prove the security of cryptographic protocols by reducing their hardness to well-known hard problems. If such a reduction exists, then finding an efficient way to break the system would imply an efficient solution to the underlying hard problem, which is presumed not to exist, thus the security of the protocol holds.

3.2. SIS and LWE problems

Let us first consider solving the linear equation

$$\mathbf{A}\mathbf{s} = \mathbf{y} \mod q \tag{7}$$

given $n, m \in \mathbb{Z}$, a prime number q, a matrix $\mathbf{A}$ in $\mathbb{Z}_q^{m\times n}$ and $\mathbf{y}$ is a vector in $\mathbb{Z}_q^m$.

We can now view the matrix $\mathbf{A}$ as a basis for a lattice $\mathcal{L}(\mathbf{A})$. Note that solving Eq. (7) is achievable through Gaussian elimination. However, its variations, along with many other related lattice problems, are known to be NP-hard (based on the factors of polynomial approximation problems). Here, we highlight some significant lattice problems: **The shortest integer solution (SIS)** and **The Learning With Errors (LWE)** problems, further information on lattice-related problems can be found in [4].

3.2.1. The shortest integer solution (SIS)

Recall that initially, we are attempting to find a solution for the equation $\mathbf{As} = \mathbf{y} \mod q$. Consider an additional constraint that $\mathbf{s}$ is "short", i.e $\mathbf{s}$ lies in the subset $S = \{0,1\}^n \in \mathbb{Z}_q^n$; or more generally, $S = [-B, \ldots, B]^n$ ($B \ll q/2$). Since we are seeking a *short solution* to linear equations, this problem is referred to as the *Short Integer Solution* (SIS) problem. Many cryptographic protocol deal with the case $\mathbf{y} = \mathbf{0}$, the $\mathbf{SIS}_{n,m,q,B}$ problem can be stated as follows:

Definition 2. *Let $n, m, q, B \in \mathbb{N}$ be positive integers. Given a uniformly sampled matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, the shortest integer solution problem asks to find a none zero vector $\mathbf{s} \in \mathbb{Z}_q^m$ such that $\mathbf{As} = \mathbf{0} \mod q$ and $\|x\|_\infty \leq B$.*

One should think of n as the security parameter that defines the hardness of the problem (the bigger n is, the harder the problem becomes), m usually depends on the specific application, generally $m \gg n$, $q = poly(n)$ and B should be set large enough to ensure that a solution exists, but not trivial to solve. It is proven that solving the $\mathbf{SIS}_{n,m,q,B}$ problem is at least as hard as solving the SVP_γ on worst-case n-dimensional lattices with high probability for some approximation factor γ [4].

3.2.2. The learning with errors problem (LWE)

This subsection recalls the definition of the Learning With Errors (LWE) problem [4]. Let's revisit Eq. (7), we can consider a slight variant of it where the input pair ($\mathbf{A} \in \mathbb{Z}_q^{m\times n}, \mathbf{y} \in \mathbb{Z}^m$) satisfies

$$\mathbf{As} + \mathbf{e} = \mathbf{y} \mod q \tag{8}$$

where n, m, q are positive integers and $\mathbf{e}$ is an additional small noise term sample from an error distribution χ over $\mathbb{Z}^n$.

The "search" $\mathsf{LWE}_{n,m,q,\chi}$ problem asks to find $\mathbf{s} \in \mathbb{Z}^n$ and the "decision" version asks to to distinguish between the distribution $(\mathbf{A}, \mathbf{y} = \mathbf{As} + \mathbf{e} \mod q)$ and a uniformly random sample $(\overline{\mathbf{A}}, \mathbf{u})$.

The error term $\mathbf{e}$ effectively affects the exact relationship between the secret $\mathbf{s}$ and $\mathbf{y}$. Specifically, this term $\mathbf{e}$ ensures that although anyone can observe the pairs $(\mathbf{A}, \mathbf{y})$, extracting the information of $\mathbf{s}$ is computationally infeasible if the problem is properly parameterized. The hardness of the $\mathsf{LWE}_{n,m,q,\chi}$ problem (and thus the security of many LWE-based cryptosystems) relies on the assumption that finding $\mathbf{s}$ with random errors is as difficult as solving certain worst-case problems on lattices, more information is mentioned in 3.1.2.

3.2.3. Distribution over lattices

It's important to note that the choice of the error distribution χ is crucial. A widely utilized distribution in numerous lattice-based cryptosystems is the Discrete Gaussian distribution. Typically, the discrete Gaussian is chosen in a way that ensures the errors $\mathbf{e}$ are small enough to allow for the correct extraction of the secret information but large enough to prevent adversaries from solving the problem.

For any positive integer n and real $\sigma > 0$, which is taken to be 1 when omitted, the *Gaussian function* $\rho_\sigma : \mathbb{R}^n \to \mathbb{R}^+$ of parameter (or width) σ is defined

$$\rho_\sigma(\mathbf{s}) := \exp(-\pi\|\mathbf{s}||^2/\sigma^2). \tag{9}$$

A **discrete Gaussian probability distribution** $D_{\mathcal{L},\sigma}$ over a lattice $\mathcal{L}$ is given by

$$D_{\mathcal{L},\sigma}(\mathbf{s}) = \frac{\rho_\sigma(\mathbf{s})}{\rho_\sigma \mathcal{L}}, \forall \mathbf{s} \in \mathcal{L}. \tag{10}$$

The utilization of Gaussian distributions comes from its ability to effectively connect the Learning With Errors (LWE) problem to the worst-case scenario of the lattice NP-hard problems, thus providing a solid foundation for cryptographic security.

More concretely, a long sequence of works has established progressively stronger results about the worst-case lattice problems. It is shown that for any $\alpha > 0$ such that $\sigma = \alpha q \geq 2\sqrt{n}$, the $\mathsf{LWE}_{n,m,q,D_{\mathbb{Z}_q,\sigma}}$, where $D_{\mathbb{Z}_q,\sigma}$ is the discrete Gaussian distribution, is at least as hard as approximating the shortest independent vector problem (SIVP_γ) to within a factor of $\gamma = \tilde{O}(n/\alpha)$ [5,6].

3.2.4. Trapdoor mechanism

In cryptography, a **trapdoor** refers to a concealed backdoor embedded within an algorithm or a specific piece of data. The idea is that, except for the individuals authorized to hold the trapdoor, others cannot extract any information about the trapdoor. This characteristic enables secure communication between parties. This subsection revisits certain trapdoor mechanisms to provide a fundamental understanding of the relationship between lattices and their trapdoors.

The first approach to constructing trapdoors for lattices involves the use of short bases, as these lead to more efficient algorithms for solving lattice problems. Algorithms designed to tackle challenges such as the Shortest Vector Problem (SVP) or the Closest Vector Problem (CVP) become more practical with basis vectors that are short and nearly orthogonal 3.1.1. This is because the computations involved are simpler and can be executed more swiftly. Specifically, for solving the LWE problem using a short basis, one strategy employs Babai's nearest plane algorithm [7]. The initial step involves utilizing a short basis $\mathbf{T_A}$ of a lattice $\mathcal{L}(\mathbf{A})$ to identify the lattice vector closest to $\mathbf{y}$. Given that the error is sufficiently "small", it becomes feasible to extract the secret vector $\mathbf{s}$'s information. A sequence of works has been carried out to develop more trapdoor mechanisms. Two of the notable approaches are [8] and [9].

[8] presented an algorithm to sample a uniform matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ together with an associated basis $\mathbf{T}$ of $\mathcal{L}(\mathbf{A})$ with a low Gram-Schmidt norm such that $\mathbf{A}\mathbf{T_A} = \mathbf{0}$ as the public matrix and its secret trapdoor. The second approach is the trapdoor technique from [9], which we focus on more due to its simpler, more elegant nature, and its support for more efficient cryptographic constructions.

[9] first defined a "primitive vector" $\mathbf{g} = \{g_1, \ldots, g_k\} \in \mathbb{Z}_q^k$ where $\gcd(g_1, \ldots, g_k, q) = 1$. Then a matrix $\mathbf{G} \in \mathbb{Z}_q^{n\times m}$ is called "primitive matrix" if its columns generate all of $\mathbb{Z}_q^n$, i.e., $\mathbf{G}\mathbb{Z}^m = \mathbb{Z}_q^n$. We can now define the $\mathbf{G}$-trapdoor of a lattice $\mathcal{L}$ is a matrix $\mathbf{R} \in \mathbb{Z}^{(m-\omega)\times\omega}$ such that $\mathbf{A}\begin{pmatrix}\mathbf{R}\\ \mathbf{I}\end{pmatrix} = \mathbf{H}\mathbf{G}$ for some invertible matrix $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$. We refer to $\mathbf{H}$ as the **tag** or **label** of the trapdoor.

The paper also presented an efficient algorithm $\mathsf{GenTrap}(1^n, 1^m, q)$ that returns a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a trapdoor $\mathbf{T_A} = \mathbf{R}$ such that the distribution of $\mathbf{A}$ is negligibly (in n) close to the uniform distribution.

Moreover, there is an efficient algorithm Invert that, on input $\mathbf{A}$, $t_\mathbf{A}$ and $\mathbf{A}\mathbf{s} + \mathbf{y}$ where $\mathbf{s} \in \mathbb{Z}_q^n$ is arbitrary and $\mathbf{y} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\alpha q}$ for $1/\alpha \geq \sqrt{n\log q}\cdot\omega(\sqrt{\log n})$, outputs $\mathbf{s}$ and $\mathbf{y}$ with overwhelming probability over $(\mathbf{A}, \mathbf{t_A}) \leftarrow \mathsf{GenTrap}(1^n, 1^m, q)$. By using the "strong" trapdoor $\mathbf{R}$, one also can efficiently generate the "weak" trapdoor $\mathbf{T_A}$ or $\mathbf{s}$. Since in lattice-based PQC, these trapdoors $\mathbf{R}, \mathbf{T_A}, \mathbf{s}$ can be used as the secret keys of hierarchical parties in some hierarchical cryptosystems.

The above we could also expand to make more accessible.

3.3. RLWE problem

There is a variant of the LWE problem that has been widely used in recent years called the Ring Learning With Error problem (RLWE). The primary benefit of using polynomial rings in RLWE offer more efficient and practical implementations of cryptographic schemes with significantly smaller keys and signature sizes while maintaining strong security guarantees.

Here, let's first recall some basic ideas of the RLWE problem.

3.3.1. The initial idea.

Let's recall that the SIS problem in Eq. (7) yields a very simple collision-resistant hash function that is provably secure if worst-case lattice problems are hard:

$$h_{\mathbf{A}}(\mathbf{s}) = \mathbf{A}\mathbf{s} \mod q \tag{11}$$

where $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ is uniformly random and $\mathbf{s} \in \{0,1\}^n$.

However, this is inefficient, since just reading the public description of $\mathbf{A}$ takes time roughly $mn \log q > n^2$. The goal is now to show a variant of $h_{\mathbf{A}}$ whose running time is roughly linear in n.

A trivial idea is taking some short, uniformly random seed r and then setting $\mathbf{A} = H(\mathsf{seed})$ for some suitable expanding function H. However, we need to describe H so that it can be efficiently used in cryptography constructions.

Let's first take the random seed to be $\ell = m/n$ uniformly random vectors $\mathbf{a}_1, \ldots, \mathbf{a}_\ell \in [\mathbf{q}]^{\mathbf{n}}$. Consider the "cyclic rotations" of $\mathbf{a}_i$, i.e. for a single vector $\mathbf{a} = (a_1, \ldots, a_n)^T \in \mathbb{Z}^n$, we define

$$\mathsf{Rot}(\mathbf{a}) = \begin{pmatrix} a_1 & a_n & \ldots & a_3 & a_2 \\ a_2 & a_n & \ldots & a_4 & a_3 \\ a_3 & a_n & \ldots & a_5 & a_4 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-2} & a_{n-3} & \ldots & a_n & a_{n-1} \\ a_{n-1} & a_{n-2} & \ldots & a_1 & a_n \\ a_n & a_{n-1} & \ldots & a_2 & a_1 \end{pmatrix} \in \mathbb{Z}^{n\times n}, \tag{12}$$

where each column is a simple cyclic permutation of the previous column. $\mathbf{A}$ can now be rewritten as

$$\mathbf{A} = \begin{pmatrix} \mathsf{Rot}(\mathbf{a}_1) \\ \mathsf{Rot}(\mathbf{a}_2) \\ \vdots \\ \mathsf{Rot}(\mathbf{a}_\ell) \end{pmatrix} \in \mathbb{Z}^{m\times n}. \tag{13}$$

Due to the property of $\mathsf{Rot}(\mathbf{a}$, we can write $\mathsf{Rot}(\mathbf{a}) = (\mathbf{a}), \mathbf{X}\mathbf{a}, \ldots, \mathbf{X}^{\mathbf{n-1}}\mathbf{a})$ where

$$\mathbf{X} = \begin{pmatrix} 0 & 0 & \ldots & 0 & 1 \\ 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 \end{pmatrix} \in \{\mathbf{0},\mathbf{1}\}^{\mathbf{n}\times\mathbf{n}} \tag{14}$$

is the "cyclic shift" matrix.
Consider the set of all integer cyclic matrices, $\tilde{R} := \{\mathsf{Rot}(\mathbf{a}) : \mathbf{a} \in \mathbb{Z}^n\}$. Hence we have that $\tilde{R}$ is a lattice with rank n and basis $\{\mathbf{I_n}, \mathbf{X}, \mathbf{X^2}, \ldots, \mathbf{X^{n-1}}\}$. We have that $\tilde{R}$ is a commutative ring. We can instead think of $\mathsf{Rot}(\mathbf{a}) \in \tilde{R}$ as the corresponding polynomial $a \in R = \mathbb{Z}[x]/(x^n - 1)$. Hence, we can therefore identify the matrix $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ with a tuple of ring elements $(a_1, \ldots, a_\ell)^T \in R_q^\ell = R = \mathbb{Z}_q^\ell[x]/(x^n - 1)$. Similarly, $\mathbf{s}$ is a tuple of ring elements $(s_1, \ldots, s_\ell)^T \in R_{\{0,1\}}^\ell$. Therefore, $h_{\mathbf{A}}$ can be written as

$$h_a(s) = h_{a_1,\ldots,a_\ell}(s_1, \ldots, s_\ell) = a_1 s_1 + \cdots + s_\ell s_\ell \mod qR. \tag{15}$$

The Ring-SIS problem, which is the analogue of SIS in this setting, can be defined as follows: For a ring R, integer modulus $q \geq 2$, and integer $\ell \geq 1$. Given $a_1, \ldots, a_\ell \in R_q$ sampled independently and uniformly at random. The search Ring-SIS problem asks to output $e_1, \ldots e_\ell \in R_{\{-1,0,1\}}$ not all zero such that $h_{a_1,\ldots,a_\ell}(e_1, \ldots, e_\ell) = a_1 e_1 + \cdots + a_\ell e_\ell = 0 \mod qR$.

However, for security reasons, there exist attacks on h_a over $\mathbb{Z}[x]/(x^n - 1)$ since $(x^n - 1)$ has a nontrivial factor over the integers. So, it is natural to try replacing $(x^n - 1)$ with an irreducible polynomial $p(x)$.

If n is a power of 2, we have that $x^n + 1$ is an irreducible polynomial and $R = \mathbb{Z}[x]/(x^n + 1)$ is an integral domain. From the matrix perspective mentioned above, this corresponds to taking

$$\mathsf{Rot}(\mathbf{a}) = \begin{pmatrix} a_1 & -a_n & \ldots & -a_3 & -a_2 \\ a_2 & a_n & \ldots & -a_4 & -a_3 \\ a_3 & a_n & \ldots & -a_5 & -a_4 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-2} & a_{n-3} & \ldots & -a_n & -a_{n-1} \\ a_{n-1} & a_{n-2} & \ldots & -a_1 & -a_n \\ a_n & a_{n-1} & \ldots & a_2 & a_1 \end{pmatrix} \in \mathbb{Z}^{n \times n}, \tag{16}$$

and

$$\mathbf{X} = \begin{pmatrix} 0 & 0 & \ldots & 0 & -1 \\ 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 \end{pmatrix} \in \{-\mathbf{1}, \mathbf{0}, \mathbf{1}\}^{\mathbf{n} \times \mathbf{n}}. \tag{17}$$

Matrices of the form $\mathsf{Rot}(\mathbf{a})$ as above are occasionally called "*anti-cyclic*". Ring-SIS is hard over this ring, under a reasonable worst-case complexity assumption.

3.3.2. Ideal lattices and computational problems

Recall that a lattice is an additive subgroup of $\mathbb{Z}^n$,i.e., a subset of $\mathbb{Z}^n$ closed under addition and subtraction. An ideal $\mathcal{I} \subset R = \mathbb{Z}[x]/(x^n + 1)$ is an additive subgroup of R that is closed under multiplication by any ring element. Concretely, $\mathcal{I}$ is closed under addition and subtraction, and for any $y \in \mathcal{Y}$ and $r \in R$, we have that $ry \in \mathcal{I}$.

For the choice of ring, we can view $\mathcal{I}$ as a lattice by embedding R in $\mathbb{Z}^n$ via the trivial embedding that maps x_i to the unit vector $\mathbf{e}_i$. We have that $\mathcal{I} \subset \mathbb{Z}^n$ is a lattice such that $(y_1, \ldots, y_n)^T \in \mathcal{I}$ if and only if $(-y_n, y_1 \ldots, y_{n-1})^T \in \mathcal{I}$. Such lattices are sometimes called "anti-cyclic" lattices.
This view of ideals as lattices is useful since we can extend computational lattice problems to ideals. Concretely, for some fixed ring R, we can define the computational problems IdealSVP, IdealSIVP, GapIdealSVP, etc., similar to SVP, SIVP, etc. as the corresponding computational problems restricted to ideal lattices.

3.3.3. The Ring-LWE problem

Here we define the Ring-LWE in a similarly natural way as the Ring-SIS problem.

For integers $q \geq 2$, a power of two n, and an error distribution χ over short elements in $R_q = \mathbb{Z}_q[x]/(x^n + 1)$, the (average-case, search) Ring-LWE problem (RLWE) is defined as follows: Given $a_1, \ldots, a_m \in R_q$ sampled independently and uniformly at random together with $b_1, \ldots, b_m \in R_q$ where $b_i = a_i s + e_i \mod R_q$ for $s \in R_q$, and $e_i \leftarrow \chi$, find s.

3.3.4. Trapdoor algorithms in the ring setting

The trapdoor mechanism techniques from [9] can be generalised to the ring setting where there exists an efficient randomised algorithm $\mathsf{GenTrap}(1^n, 1^m, q)$ that returns $\mathbf{a} = (a_1, \ldots, a_m) \in R_q^m$ and a trapdoor $t_{\mathbf{a}}$ such that the distribution of $\mathbf{a}$ is negligibly (in n) close to the uniform distribution. Moreover, there is an efficient algorithm Invert and a universal constant C_T that, on input $\mathbf{a}$, $t_{\mathbf{a}}$ and $\mathbf{a} \cdot s + \mathbf{e}$ where $s \in R_q$ is arbitrary and $\|\mathbf{e}\| \leq q/(C_T\sqrt{n \log q})$, outputs s with overwhelming probability.

4. Trapdoor Claw Free Functions - TCFs

4.1. One-way function

Informally, one-way functions are functions that are "easy" to compute but "hard" to invert in a "computational sense." With these special properties, one-way functions are among the most fundamental tools for building classical and post-quantum cryptographic protocols, ranging from authentication, and signature to public key encryption schemes.

Definition 3 (One-way function)**.** *A function $f : \{0,1\}^* \to \{0,1\}^*$ is called one-way if two conditions hold:*

- ***Easy to compute**: There exists a polynomial-time algorithm that, on input x, outputs $f(x)$.*
- ***Hard to invert**: For every probabilistic polynomial-time algorithm $\mathcal{A}$, given $f(x)$, the probability that $\mathcal{A}$ finds a preimage of $f(x)$ is negligible, i.e.,*

$$\Pr[\mathcal{A}(f(x)) \in f^{-1}(f(x))] = \mathsf{negl}(\cdot). \tag{18}$$

Given a one-way function $f(x)$, one question is raised: *does the output of a one-way function, $y = f(x)$, truly conceal all information of the preimage $x \in f^{-1}(f(x))$?* The answer is NO! Consider $x = x_1 \ldots x_n$ and assuming that $g(x)$ is another one-way function, it follows that $f(x) = x_1||g(x)$ remains a one-way function since an adversary $\mathcal{A}$ must still invert $g(x)$ to determine the whole x. The first bit x_1 does not reveal information about other bits of x, as x is sampled uniformly at random. This example illustrates the fact that the output of a one-way function could reveal some information about a certain bit of x.

Hence, we need a Boolean predicate (a Boolean function) $\mathcal{B} : \{0,1\}^* \to \{0,1\}$ that is completely hidden given the output of a one-way function. This leads us to the definition of the hardcore bit property.
More formally, the hardcore bit property (or hard-core predicate) ensures that even with the information of $y = f(x)$, guessing $\mathcal{B}(x) \in \{0,1\}$ is as challenging as finding x.

Definition 4 (Hardcore bit property)**.** *$\mathcal{B} : \{0,1\}^* \to \{0,1\}$ is a hardcore bit predicate of a one way function $f : \{0,1\}^* \to \{0,1\}^*$ if*

- *$\mathcal{B}$ is efficiently computable,*
- *It is "hard" to compute a bit $\mathcal{B}(x)$ of x given $f(x)$. Formally, for all adversaries $\mathcal{A}$*

$$\Pr_{x \leftarrow \{0,1\}^*}[\mathcal{A}(f(x)) = \mathcal{B}(x)] \leq \frac{1}{2} + \mathsf{negl}(\cdot).$$

4.2. Trapdoor Claw Free Functions

In this section, we recall the definition of the trapdoor claw-free functions [10], [11].

Trapdoor claw-free functions (TCF), Fig. 3, is a 2-to-1 one-way function $f_{\mathcal{I}}(x)$ such that it is hard for a (quantum) polynomial-time adversary to simultaneously find the preimage pair (x_0, x_1) *"a claw"*, given any image $y = f_{\mathcal{I}}(x_0) = f_{\mathcal{I}}(x_1)$ where $\mathcal{I}$ is some problem instance. Sometimes, TCFs can also be defined as a pair of functions $f_{\mathcal{I},0}(x)$, $f_{\mathcal{I},1}(x)$ that are both injective and have the same image space and it is cryptographically hard to find two inputs mapping to the same output. One special property of TCFs is that given a problem instance $\mathcal{I}$ sampled together with a suitable trapdoor t_k, it is possible for one to find all the pre-images of y. More formally, we can define the trapdoor claw-free function as follows:

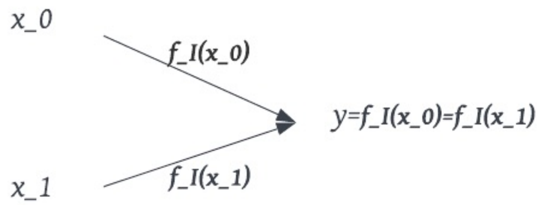


Figure 3: "A claw" of a TCF function $f_{\mathcal{I}}(x)$ is a pair of (x_0, x_1) such that $y = f_{\mathcal{I}}(x_0) = f_{\mathcal{I}}(x_1)$

Definition 5 (Trapdoor Claw-Free Function (TCF))**.** *Let $\mathcal{X}$, $\mathcal{Y}$ be finite sets, we require the family of functions $\mathcal{F}$ satisfies the following properties:*

- *For each public key k, there are two functions $\{f_{\mathcal{I},b} : \mathcal{X} \to \mathcal{Y}\}_{b\in\{0,1\}}$ that are both injective and have the same range, and are invertible given a suitable trapdoor t_k (i.e. t_k can be used to efficiently compute x given b and $y = f_{\mathcal{I},b}(x)$).*
- *The pair of functions should be claw-free: it is hard to find the pair $(x_0, x_1) \in \mathcal{X}^2$ such that $y = f_{k,0}(x_0) = f_{k,1}(x_1)$.*

Note that a quantum device can not invert the function a find the claw (x_0, x_1), it can hold these two preimages together with y in a superposition

$$\frac{1}{\sqrt{2}}(|x_0\rangle + |x_1\rangle)\,|y\rangle\,. \tag{19}$$

This is obtained by first preparing a uniform superposition over all $x \in \mathcal{X}$ via Hadamard transform

$$\frac{1}{\sqrt{\mathcal{X}}}\sum_{x\in\mathcal{X}} |x\rangle\,, \tag{20}$$

then evaluating $f_{\mathcal{I},b}(x)$ into an output register to yield

$$\frac{1}{\sqrt{\mathcal{X}}}\sum_{x\in\mathcal{X}} |x\rangle\,|f_{\mathcal{I},b}(x)\rangle\,. \tag{21}$$

If one measures the output register and obtains the result y, the state would collapse to the (19) state such that $y = f_{k,0}(x_0) = f_{k,1}(x_1)$. Measuring the (19) state in the standard basis (Z basis), one would obtain a random preimage as the measurement result, i.e. $m = x_0$ or $m = x_1$, but not both, since measuring a quantum state means collapsing that state to one possible outcome. Besides, measuring (19) state in the X basis yields a measurement result m and a bit c that satisfy $m \cdot (x_0 \oplus x_1) = c$. This special property of TCFs could be used to construct a Proof of Quantumness protocol, the details of which will be discussed in section 5.

However, a fully constructive Trapdoor Claw-Free (TCF) function that satisfies our cryptographic requirements has not been explored yet. The paper [11] considers a "relaxed" version of the TCFs, referred to as **Noisy trapdoor claw-free functions (NTCFs)**.

4.2.1. TCFs from LWE

For a matrix $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ and vectors $\mathbf{s}, \mathbf{e} \in \{0,1\}^n$, consider the LWE instance $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ with the trapdoor $t_k = \{\mathbf{s}, \mathbf{e}\}$. We can define the TCFs family by letting $f_{\mathcal{I},0}(\mathbf{x}) = \mathbf{Ax} + \mathbf{e}_0$ and $f_{\mathcal{I},1}(\mathbf{x}) = \mathbf{A}(\mathbf{x} + \mathbf{s}) + \mathbf{e}_0 + \mathbf{e}$. If $\mathbf{e}_0 = \mathbf{0}$, then $f_{\mathcal{I},1}(\mathbf{x}) = f_{\mathcal{I},0}(\mathbf{x} + \mathbf{s})$; however, we need to hide the information of $\mathbf{x}$, it is essential to sample $\mathbf{e}_0$ from a Gaussian distribution much wider than $\mathbf{e}$ such that it ensures that the distribution of $f_{\mathcal{I},1}(\mathbf{x})$ and $f_{\mathcal{I},0}(\mathbf{x} + \mathbf{s})$ are statistically close as per the requirements of the NTCFs defined in the previous subsection.

4.2.2. TCFs from RLWE

Two years later from the first definition of the NTCFs from LWE was published, Brakerski continued his work in [12] and showed that using the ring version of the LWE problem yeild better results since RLWE-based primitives exhibit greater efficiency and yield smaller key sizes than their LWE-based counterparts, due to the ring dimension. This dimension plays a crucial role in determining the size of the challenge space. The larger the challenge space, the less the soundness error of the scheme has and the fewer times the underlying protocol has to repeat.

4.3. Adaptive hardcore bit property for TCFs from LWE

Recall that in the TCFs from LWE, measuring the output register of the (A13) state, obtaining the measurement outcome $\mathbf{y}$ and the superposition

$$\frac{1}{\sqrt{2}}(|\mathbf{x}_0\rangle + |\mathbf{x}_1\rangle)|\mathbf{y}\rangle. \tag{22}$$

Recall that when measuring the $\mathbf{x}$ register in the X basis, we obtain the measurement result $m \in \{0,1\}^n$ and a bit $c \in \{0,1\}$ such that $m \cdot (\mathbf{x}_0 + \mathbf{x}_1) = c$. This equation provides some information about both preimages. Note that [11] used the LWE-based hardness assumption, where $\mathbf{x}_0 + \mathbf{x}_1 = \mathbf{s}$, where $\mathbf{s}$ is some fixed secret determined by the problem instance $\mathcal{I}$. Hence, measuring the first register in the Hadamard basis yields a uniformly random m such that $m \cdot \mathbf{s} = c$. If we repeat the entire procedure $O(n)$ times yield linearly independent such m, which suffices to recover $\mathbf{s}$ with high probability. Hence, we need the structure of the function f to be a little more complicated so that even with $\mathbf{x}_0, \mathbf{x}_1 = \mathbf{x}_0 - \mathbf{s}$, and given many "equations" m could be useless since without additional knowledge of a preimage, one cannot determine what the equation is about. This is why we need the trapdoor claw-free function to satisfy the **hardcore bit**, which is similar to a one-way function. Specifically, $m \cdot (\mathbf{x}_0 + \mathbf{x}_1) = c$ is the hardcore bit that ensures it is "hard" to compute a bit of $\mathbf{x}_0$ or $\mathbf{x}_1$ given $\mathbf{y}$. Moreover, if the adversary can simultaneously find this predicate and a preimage $\mathbf{x}_b$, it could find the whole claw $(\mathbf{x}_0, \mathbf{x}_1)$.

Why does IPOF [11] construction use the term "adaptive" for the hardcore bits property? Combining all the mentioned requirements, we can write the hardcore bit property for the LWE-based trapdoor claw-free function as follows: Given a trapdoor claw-free function $f_{\mathcal{I},b}(\mathbf{x})$, a hardcore bit for f is a 1-output-bit function h such that given $f_{\mathcal{I},b}(\mathbf{x})$ (but not $\mathbf{x}$) it is hard to predict h. Here, the hardcore bit that underlies the assumption is the function $c = h(x) = m \cdot (\mathbf{x}_0 + \mathbf{x}_1)$ for any m. Since this equation can be rewritten as $m' \cdot \mathbf{s} = c$. The property can be stated as it is hard to produce the string m' that satisfies the above equation.

Moreover, we allow the adversary $\mathcal{A}$ to ***adaptively choose the string*** m'. This also means that more power is given to the adversary. It is in this sense that the hardcore bit property is **adaptive**. Concretely, in the LWE-based construction, even the adversary $\mathcal{A}$ can adaptively choose the string m after being given access to the LWE sample $\mathbf{As} + \mathbf{e}$, i.e. the choice of m' depend upon the LWE sample, due to the *leakage resilience* property of the LWE problem, the information of the $\mathbf{s}$ remains secret. The idea is that instead of sampling a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ together with its trapdoor, we replace $\mathbf{A}$ by a "lossy mode" $\tilde{\mathbf{A}}$, where $\tilde{\mathbf{A}}$ is sampled from a special distribution so that $\tilde{\mathbf{A}} = \mathbf{BC} + \mathbf{E}$ for a skinny matrix $\mathbf{B}$, a wide matrix $\mathbf{C}$ and $\mathbf{E} \leftarrow \chi^{m \times n}$. It holds that $\tilde{\mathbf{A}}$ is indistinguishable from $\mathbf{A}$. Now, when we multiply $\tilde{\mathbf{A}}$ by $\mathbf{s}$, the matrix $\mathbf{C}$ compresses the information of $\mathbf{s}$. To be more specific, provided the matrix $\mathbf{C}$ is a uniformly random matrix with sufficiently few rows, the distribution $(\mathbf{C}, \mathbf{Cs})$ for an arbitrary secret $\mathbf{s}$ does not reveal any parity of $\mathbf{s}$. More information on the proof can be found in [11].

5. Proof of Quantumness

5.1. Interactive proof

A proof serves as a means of persuading someone that a specific statement is true. We typically involve two parties when discussing proofs: *a prover* and *a verifier*. The prover aims to convince the verifier of the validity of a given statement, while the verifier's goal is to accept only accurate assertions. Before giving the definition of the interactive proof, we first recall that a language is simply a set of strings $L \subseteq \{0,1\}^*$. The definition of an interactive proof for a statement x in a language L is defined as follows:

Definition 6 (Interactive proof)**.** *An interactive proof system for a language L is a protocol between a computationally unrestricted prover $\mathcal{P}$ and a polynomial-time verifier $\mathcal{V}$ such that on input a statement x:*

- ***Completeness**: If $x \in L$, then an honest prover $\mathcal{P}$ that follows protocol should be able to convince the verifier $\mathcal{V}$ about the validity of the statement.*

$$\forall x \in L, \Pr[(\mathcal{V} \leftrightarrow \mathcal{P})(x) accepts\,] = 1.$$

- ***Soundness ϵ:** If $x \notin L$, then no malicious prover $\mathcal{P}'$ should be able to convince $\mathcal{V}$ with a success probability greater than ϵ.*

$$\forall x \notin L, \forall \mathcal{P}', \Pr[(\mathcal{V} \leftrightarrow \mathcal{P}')(x) accepts] \leq \epsilon.$$

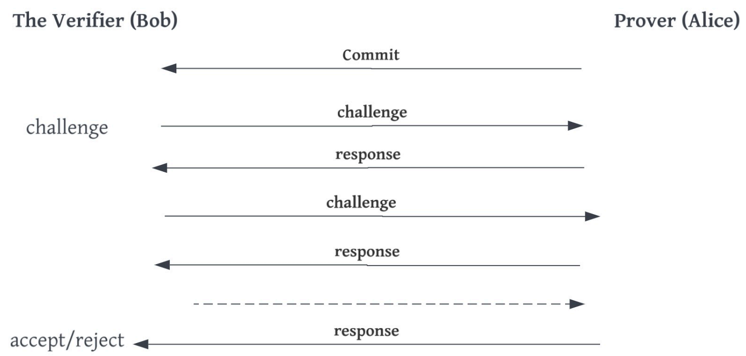


Figure 4: An interactive proof between a prover (Alice) and a verifier (Bob) typically has three main phases: commit, challenge, and response. The protocol may repeat the challenge-response phase multiple times to ensure the soundness property.

5.2. Proof of quantumness

In recent years, under the efforts of building quantum computers, the verification of quantum behaviour has become significantly crucial. This underscores the necessity for a protocol in which a classical party can efficiently certify the quantum advantage of an untrusted device - the Proof of Quantumness. An interactive proof of quantumness [11,13] is an interactive quantum verification protocol between two parties: a quantum prover $\mathcal{P}$ and a classical verifier $\mathcal{V}$. The verifier $\mathcal{V}$'s goal is to test the "quantumness" of the prover through an exchange of classical information. A common recent approach is to build the Interactive Proof of Quantumness protocol using TCFs.

5.3. LWE

The basic idea of employing the TFCs from LWE (Definition A2) to build the interactive proof of quantumness protocol is presented as follows:

1. The verifier samples an LWE instance $\mathcal{I} = (\mathbf{A}, \mathbf{y} = \mathbf{As} + \mathbf{e})$ together with a trapdoor $t_k = (\mathbf{s}, \mathbf{e})$, and sends $\mathcal{I}$ and the description of the NTCFs $f_{\mathcal{I}}(\cdot)$ to the prover.

2. The prover then:
   - prepares a uniform superposition over $\mathbf{x} \in \{0,1\}^n$

$$\frac{1}{\sqrt{2}^n} \sum_{\mathbf{x} \in \{0,1\}^n} |\mathbf{x}\rangle, \tag{23}$$

   - evaluates $f_{\mathcal{I}}(\cdot)$ into an output register yields the state

$$\sum_{\mathbf{x} \in \{0,1\}^n} |\mathbf{x}\rangle \, |f_{\mathcal{I}}(\mathbf{x})\rangle, \tag{24}$$

   - measures the output register to obtain $\mathbf{w} \in \mathcal{Y}$. The prover sends $\mathbf{w}$ to the verifier as the commitment for its quantum power.
3. The remaining state after measuring on the output register of the prover now is

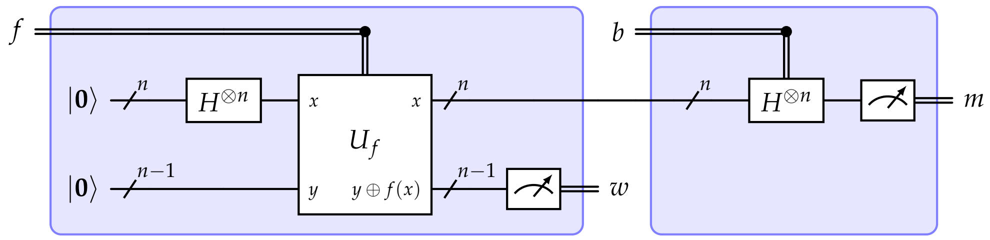
$$\frac{1}{\sqrt{2}}(|\mathbf{x}_0\rangle + |\mathbf{x}_1\rangle) \tag{25}$$

   where $\mathbf{x}_0$, $\mathbf{x}_1$ are two preimages of $\mathbf{w}$.
4. The verifier challenge by asking the prover to measure on either the standard basis (Z basis) $b = 0$ or the Hadamard basis (X basis) $b = 1$.
5. According to the verifier's challenge, the measurement outcome of the prover's state is:
   - Standard basis: either $m = \mathbf{x}_0$ or $m = \mathbf{x}_1$,
   - Hadamard basis: $(c \in \{0,1\}, m \in \{0,1\}^n)$ such that $m \cdot (\mathbf{x}_0 + \mathbf{x}_1) = c$.
6. The verifier checks these tests:
   - Standard basis: simply check that $f_{\mathcal{I}}(m) = \mathbf{w}$.
   - Hadamard basis: the verifier can recover $\mathbf{x}_0$, $\mathbf{x}_1$ from $\mathbf{y}$ using the trapdoor $t_k = (\mathbf{s}, \mathbf{e})$, and check whether the pair (m, c) satisfies $m \cdot (\mathbf{x}_0 + \mathbf{x}_1) = c$.

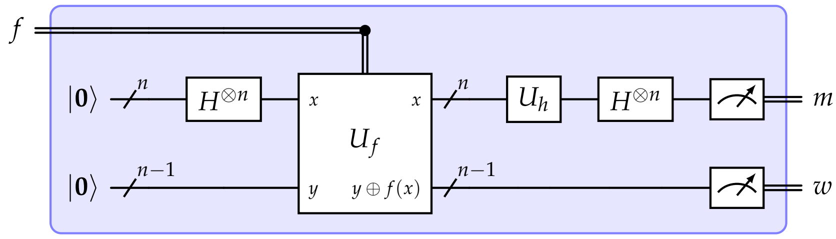
   The protocol proceeds for λ rounds. At the end of each round, the verifier samples a new problem instance and sends the new public key to the prover. In a recent paper by Zhu [13], for each of the verifier's possible choices, the experiment is repeated approximately 10^3 times to collect statistics.

Analysis. For the proof of quantumness protocol, the post-quantum cryptography hardness assumption here is employed in a different way, where we do not require the prover to break something that the classical machine cannot. The protocol is constructed in a way that the security is preserved even against both *classical* and *quantum* prover. We use the *rewinding* technique to distinguish between the classical and quantum cases. If a classical prover can cause the verifier to accept the statement, it can rewind and thus break the underlying hardness assumption which is computationally infeasible due to the hardness of the LWE problem. However, if the prover is "quantum," it can persist with the verifier without breaking the underlying cryptographic assumption while performing measurements that cannot be reversed due to the no-cloning theorem.

Note the adaptive hardcore bit property plays an important role in this construction. Suppose there exists a prover capable of successfully navigating both verifier challenges. Specifically, given the problem instance $\mathcal{I}$, this prover can generate a tuple $(\mathbf{w}, b, \mathbf{x}_b, (m, c))$ with $b, c \in 0, 1$ that satisfies two checks with a probability significantly higher than $1/2$, i.e., $f_{\mathcal{I}}(\mathbf{x_b}) = \mathbf{w}$, $m \cdot (\mathbf{x}_0 \oplus \mathbf{x}_1) = c$, and $m \neq \mathbf{0}^n$. It is important to note that in this scenario, the prover does not discover the entire claw $(\mathbf{x}_0, \mathbf{x}_1)$ but rather identifies one preimage $\mathbf{x}_b$ along with a linear function of the other preimage $\mathbf{x}_{1-b}$. Therefore, the **hardcore bit** property is indispensable, asserting that it is computationally challenging to find even a single bit of $\mathbf{x}_{1-b}$—meaning the prover cannot determine a single bit of $\mathbf{x}_{1-b}$, let alone the entire claw.



(a) Two-round IPQ protocol, where f is the TCF problem instance specified by the verifier. The prover executes the first round of the protocol, measuring the $\mathbf{y}$ register to obtain classical outcome $\mathbf{w}$. After receiving $\mathbf{w}$ from the prover the verifier communicates a random measurement basis Z or X for the prover to measure the $\mathbf{x}$ register in, yielding outcome $m \in \{0,1\}^n$. With knowledge of the problem instance's secret trapdoor, the verifier can efficiently verify the prover's response.



(b) Single-round IPQ protocol, replacing the randomly chosen challenge b specified by the verifier with a hash-function-based quantum random oracle.

Figure 5: Quantum circuits for interactive proofs-of-quantumness (IPQ). Blue boxes contain all quantum operations within the protocol with classical inputs and outputs.

5.4. Non-interactive proof of quantumness from the RLWE problem

Up till now, the hardcore bit property has only been shown for TCFs based on the LWE problem. However, it is still possible to construct a **non-interactive proof of quantumness** protocol based on other hardness assumptions. [12] employs the **random oracle heuristic** as a tool to reduce the round complexity of the proof of quantumness protocol which also makes it possible to implement the protocol in a single round and eliminates the need for the hard-core bit property. The quantum device must evaluate the random oracle on a quantum superposition.

Assume we have a hash function $h(\cdot)$. In the unitary model of quantum computing, consider this means that there exists a unitary U_h that performs the mapping

$$U_h : |x\rangle\,|y\rangle \mapsto |x\rangle\,|y \oplus h(x)\rangle$$

on the basis states.

For an element x, $\mathsf{BitDecomp}(x)$ is the binary representation of x. The proof of quantumness protocol is now parameterised by a hash function $h : \{0,1\}^n \to \{0,1\}$ as follows:

1. The verifier generates the pair of public key and trapdoor (k, t_k) and sends k to the prover.
2. The prover sends λ tuple $\{(y_i, c_i, m_i)\}_{i\in[\lambda]}$, where λ is the security parameter. The verifier initialises $\mathsf{count} = 0$ and performs the following checks:
   - It checks that all value $\{y_i\}_{i\in[\lambda]}$ are distinct.
   - It uses the trapdoor to compute $x_{i,b}$ for each $i \in [\lambda]$ and $b \in \{0,1\}$. The verifier then checks $c_i = d_i^T \cdot (\mathsf{BitDecomp}(x_{i,0}) + \mathsf{BitDecomp}(x_{i,1})) + h(x_{i,0}) + h(x_{i,1})$. If (c_i, d_i) satisfies this equation, it increases the value of count by 1.
3. If $\mathsf{count} > 0.75\lambda$, the verifier accepts, else it rejects.

The second figure in 5 represents the non-interactive proof of quantumness, which removes the interaction between the prover and the verifier during the challenge phase using the hash function $h(\cdot)$, as demonstrated above. Using the RLWE problem, we can significantly reduce the size of keys and the commitment of the construction.

Why can using the hash function can replace the hardcore bit property? Recall that in the LWE-based proof of quantumness construction [11], the prover applies the trapdoor claw-free function $f_{\mathcal{I}}$ on a uniform superposition of inputs and then measures the image register. It then obtains the value w, which will be sent to the verifier. The prover now has a state that is a superposition over x_0 and x_1. The prover then has to measure this state according to the challenge of the verifier: measure using the standard basis or Hadamard basis. A malicious prover that can break the protocol must be able to answer both challenges by the verifier simultaneously, which violates the *hardcore bit property* and the hardness assumption. Using a one-bit hash function $h : \{0,1\}^n \to \{0,1\}$, [12] generates the superposition over $(0, x_0, h(0, x_0))$ and $(1, x_1, h(1, x_1))$. The prover is then asked to measure the resulting state on a Hadamard basis only and send the outcome to the verifier, including a bit c and the measurement outcome m. It holds that (c, m) satisfies the hardcore predicate $c = m \cdot (x_0 \oplus x_1) \oplus h(0, x_0) \oplus h(1, x_1)$. The verifier employs the trapdoor to recover y, x_0, x_1 and verifies whether the above equation is satisfied. The security of the protocol is upheld because an adversary cannot query the random oracle for both $(0, x_0)$ and $(1, x_1)$; thus, at least one value $h(0, x_0)$ or $h(1, x_1)$ remains random. This implies the malicious prover cannot compute (c, d) satisfying the equation with a probability greater than $1/2$. Consequently, this also eliminates the need for the hardcore bit property.

6. Comparison on experimental results

This section gives the connection between result measurements in the original and experimental papers.

Table 1. Comparison between the Proof of Quantumness protocols.

Paper	Problem	Adaptive hardcore bit	Interactive	Using hash function	Gate count	Adversary type
[11]	LWE	✓	✓	✗	$n^2 \log^2 n$	classical
[13]	LWE	✓	✓	✗	$n^2 \log^2 n$	classical
[12]	RLWE	✗	✗	✓	$n \log^2 n$	quantum
[14]	RLWE	✗	✗	✓	$n \log^2 n$	quantum

7. Discussion

8. Conclusions

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used "Conceptualization, X.X. and Y.Y.; methodology, X.X.; software, X.X.; validation, X.X., Y.Y. and Z.Z.; formal analysis, X.X.; investigation, X.X.; resources, X.X.; data curation, X.X.; writing—original draft preparation, X.X.; writing—review and editing, X.X.; visualization, X.X.; supervision, X.X.; project administration, X.X.; funding acquisition, Y.Y. All authors have read and agreed to the published version of the manuscript.", please turn to the CRediT taxonomy for the term explanation. Authorship must be limited to those who have contributed substantially to the work reported.

Funding: Please add: "This research received no external funding" or "This research was funded by NAME OF FUNDER grant number XXX." and and "The APC was funded by XXX". Check carefully that the details given are accurate and use the standard spelling of funding agency names at https://search.crossref.org/funding, any errors may affect your future funding.

Institutional Review Board Statement: In this section, you should add the Institutional Review Board Statement and approval number, if relevant to your study. You might choose to exclude this statement if the study did not require ethical approval. Please note that the Editorial Office might ask you for further information. Please add "The study was conducted in accordance with the Declaration of Helsinki, and approved by the Institutional Review Board (or Ethics Committee) of NAME OF INSTITUTE (protocol code XXX and date of approval)." for studies involving humans. OR "The animal study protocol was approved by the Institutional Review Board (or Ethics Committee) of NAME OF INSTITUTE (protocol code XXX and date of approval)." for studies involving animals. OR "Ethical review and approval were waived for this study due to REASON (please provide a detailed justification)." OR "Not applicable" for studies not involving humans or animals.

Informed Consent Statement: Any research article describing a study involving humans should contain this statement. Please add "Informed consent was obtained from all subjects involved in the study." OR "Patient consent was waived due to REASON (please provide a detailed justification)." OR "Not applicable" for studies not involving humans. You might also choose to exclude this statement if the study did not involve humans.

Written informed consent for publication must be obtained from participating patients who can be identified (including by the patients themselves). Please state "Written informed consent has been obtained from the patient(s) to publish this paper" if applicable.

Data Availability Statement: We encourage all authors of articles published in MDPI journals to share their research data. In this section, please provide details regarding where data supporting reported results can be found, including links to publicly archived datasets analyzed or generated during the study. Where no new data were created, or where data is unavailable due to privacy or ethical restrictions, a statement is still required. Suggested Data Availability Statements are available in section "MDPI Research Data Policies" at https://www.mdpi.com/ethics.

Acknowledgments: In this section you can acknowledge any support given which is not covered by the author contribution or funding sections. This may include administrative and technical support, or donations in kind (e.g., materials used for experiments).

Conflicts of Interest: Declare conflicts of interest or state "The authors declare no conflicts of interest." Authors must identify and declare any personal circumstances or interest that may be perceived as inappropriately influencing the representation or interpretation of reported research results. Any role of the funders in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; or in the decision to publish the results must be declared in this section. If there is no role, please state "The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results".

Abbreviations

The following abbreviations are used in this manuscript:

PQC	Post-quantum cryptography
RSA	Rivest-Shamir-Adleman algorithm
ECC	Elliptic Curve cryptography
IPQ	Interactive proof of quantumness
SVP	Shortest vector problem
SIVP	Shortest independent vectors problem
CVP	Closest vector problem
SIS	Shortest integer solution
LWE	Learning with error
RLWE	Ring-Learning with error
TFCs	Trapdoor claw-free functions
NTCFs	Noisy trapdoor claw-free functions

Appendix A Noisy Trapdoor Claw-free Functions (NTCFs)

In NTCFs, the range of functions $f_{k,0}$ and $f_{k,1}$ is represented by a distribution $\mathcal{D}$ over $\mathcal{Y}$, denoted as $\mathcal{D}_{\mathcal{Y}}$. In other words, each function returns a density rather than a point. Instead of considering the range of fk, b, we consider the support of the output densities, $\mathsf{SUPP}(f_{\mathcal{I},b}(x))$. If y lies in this support, a party that holds the trapdoor t_k can use it to invert the function and find the preimages of y.

Definition A1 (Noisy Trapdoor Claw-Free family)**.** *Let λ be a security parameter. Let $\mathcal{X}$ and $\mathcal{Y}$ be finite sets. Let $\mathcal{K}_{\mathcal{F}}$ be a finite set of keys. A family of functions*

$$\mathcal{F} = \{f_{\mathcal{I},b} : \mathcal{X} \to \mathcal{D}_{\mathcal{Y}}\}_{b \in \{0,1\}} \tag{A1}$$

is called a noisy trapdoor claw-free (NTCF) family if the following conditions hold:

1. ***Efficient Function Generation.*** *There exists an efficient probabilistic algorithm $\mathsf{GEN}_{\mathcal{F}}$ which generates problem instance $\mathcal{I}$ together with a trapdoor t_k:*

$$\mathsf{GEN}_{\mathcal{F}}(1^{\lambda}) \to (\mathcal{I}, t_k). \tag{A2}$$

2. ***Trapdoor Injective Pair.***
   - *Trapdoor: There exists an efficient deterministic algorithm Invert $\mathsf{INV}_{\mathcal{F}}$ such that with overwhelming probability over the choice of $(\mathcal{I}, t_k)$, the following holds:*

     for all $b \in \{0,1\}$, $\mathbf{x} \in \mathcal{X}$ and $y \in \mathsf{SUPP}(f_{\mathcal{I},b}(x))$, $\mathsf{INV}_{\mathcal{F}}(t_k, b, y) = x$.
   - *Injective pair: There exists a perfect matching set for an instance $\mathcal{I}$ denoted $\mathcal{R}_{\mathcal{I}} \subseteq \mathcal{X} \times \mathcal{X}$ such that $f_{\mathcal{I},0}(x_0) = f_{\mathcal{I},0}(x_1)$ if and only if $(x_0, x_1) \in \mathcal{R}_{\mathcal{I}}$.*
3. ***Efficient Range Superposition.*** *For all keys $k \in \mathcal{K}_{\mathcal{F}}$ and $b \in \{0,1\}$ there exists a function $f'_{\mathcal{I},b} : \mathcal{X} \to \mathcal{D}_{\mathcal{Y}}$ such that the following hold:*
   - *For all $(x_0, x_1) \in \mathcal{R}_k$ and $y \in \mathsf{SUPP}(f'_{\mathcal{I},b})(x_b)$, $\mathsf{INV}_{\mathcal{F}}(t_k, b \oplus 1, y) = x_{x \oplus 1}$.*
   - *There exists an efficient deterministic procedure $\mathsf{CHK}_{\mathcal{F}}$ that given set of input $\mathcal{I}$, $b \in \{0,1\}$, $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, check whether y is an element of the support of f or not. This procedure returns 1 if $y \in \mathsf{SUPP}(f'_{\mathcal{I},b}(x))$ and 0 otherwise.*
   - *There exists an efficient procedure $\mathsf{SAMP}_{\mathcal{F}}$ that on input the problem instance $\mathcal{I}$ and $b \in \{0,1\}$, prepares the state*

$$\frac{1}{\sqrt{|\mathcal{X}|}} \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \sqrt{(f'_{\mathcal{I},b}(x)(y))} |x\rangle |y\rangle. \tag{A3}$$

4. ***Claw-free Property.*** *There no probabilistic polynomial time adversary $\mathcal{A}$ can find the "claw" (x_0, x_1, y). More formally, there exists a negligible function $\mathsf{negl}(\cdot)$ such that the following holds:*

$$\Pr[(x_0, x_1) \in \mathcal{R}_{\mathcal{I}} : (\mathcal{I}, t_k) \leftarrow \mathsf{GEN}_{\mathsf{F}}(1^{\lambda}), (x_0, x_1) \leftarrow \mathcal{A}(\mathcal{I})] \leq \mathsf{negl}(\lambda). \tag{A4}$$

Appendix A.1 NTCFs from LWE

[11] built a NTCFs family from the LWE problem as follows:

Definition A2 (NTCFs based on LWE [11])**.** *Let λ be a security parameter. Let $q \geq 2$ be a prime and $\ell, n, m \geq 1$ are polynomially bounded functions of λ, and B_L, B_V, B_P be positive integers such that the following conditions hold:*

- $n = \Omega(\ell \log q + \lambda)$,
- $m = \Omega(n \log q)$,
- $B_P = \dfrac{q}{2C_T\sqrt{mn \log q}}$, *for C_T is a universal constant in the $\mathsf{GenTrap}$ algorithm.*

1. ***Efficient Function Generation:*** *The $\mathsf{GEN}_{\mathcal{F}}(1^{\lambda})$ is constructed as follows:*
   - *Using the trapdoor mechanism $\mathsf{GenTrap}(1^n, 1^m, q)$ to return a matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ and a trapdoor $t_k = \mathbf{R}$, $m \geq \Omega(n \log q)$ such that the distribution of $\mathbf{A}$ is negligibly (in n) close to the uniform distribution on $\mathbb{Z}_q^{m \times n}$.*
   - *Sample a uniformly random $\mathbf{s} \leftarrow \{0,1\}^n$ and a vector $\mathbf{e} \leftarrow \mathbb{Z}_q^m$ from the distribution $D^m_{\mathbb{Z}_q, B_V}$.*

- $\mathsf{GEN}_{\mathcal{F}}(1^{\lambda})$ *returns*

$$(\mathcal{I}, t_k) = ((\mathbf{A}, \mathbf{As} + \mathbf{e}), t_k = \mathbf{R}. \tag{A5}$$

Note that the trapdoor $t_k = \mathbf{R}$ can be referred to as a "strong" trapdoor since it can be used to reconstruct $\mathbf{s}$ and $\mathbf{e}$. In most existing cryptographic constructions using NTCFs, for simplicity, the trapdoor is usually set as $t_k = (\mathbf{s}, \mathbf{e})$.

2. ***Trapdoor Injective Pair***

- *For any pair $(\mathcal{I}, t_k) = ((\mathbf{A}, \mathbf{As} + \mathbf{e}), t_k = (\mathbf{s}, \mathbf{e})$, a bit $b \in \{0, 1\}$, the density function of $f_{\mathcal{I},b}(\mathbf{x})$ is defined as follows:*

$$\forall \mathbf{y} \in \mathcal{D}_{\mathcal{Y}}, (f_{\mathcal{I},b}(\mathbf{x}))(\mathbf{y}) = \mathcal{D}^m_{\mathbb{Z}_q, B_P}(\mathbf{y} - \mathbf{A}(\mathbf{x} + b\mathbf{s})) \tag{A6}$$

where $\mathbf{y} \in \mathbb{Z}_q^m$. The support of the density function $f_{\mathcal{I},b}(\mathbf{x})$ is

$$\mathsf{SUPP}(f_{\mathcal{I},0}(\mathbf{x})) = \{\mathbf{Ax} + \mathbf{e}_0 : \|\mathbf{e}_0\| \leq B_P\sqrt{m}\}. \tag{A7}$$

$$\mathsf{SUPP}(f_{\mathcal{I},0}(\mathbf{x})) = \{\mathbf{A}(\mathbf{x} + \mathbf{s}) + \mathbf{e}_0 : \|\mathbf{e}_0\| \leq B_P\sqrt{m}\}. \tag{A8}$$

The inversion algorithm $\mathsf{INV}_{\mathcal{F}} \equiv \mathsf{INVERT}(t_k, \mathbf{A}, \mathbf{y})$ finds $(\mathbf{s}_0, \mathbf{e}_0)$ such that $\mathbf{y} = \mathbf{As}_0 + \mathbf{e}_0$ and returns $\mathbf{s}_0 - b \cdot \mathbf{s} \in \mathcal{X}$. Due to the parameter selections and the choice of $\mathbf{e}_0$, we have that $\mathsf{INV}_{\mathcal{F}}$ returns a preimage of the NTCFs.
We have that, for all $\mathbf{y} \in \mathsf{SUPP}(f_{\mathcal{I},b}(\mathbf{x}))$, $\mathsf{INVERT}(t_k, \mathbf{A}, \mathbf{y}) = \mathbf{x} + b\mathbf{s}$. Hence, it follows that $\mathsf{INV}_{\mathcal{F}}(t_k, b, \mathbf{y}) = \mathbf{x}$.

- *Injective pair: Let $\mathcal{I} = (\mathbf{A}, \mathbf{As} + \mathbf{e})$. From the construction, it follows that $f_{\mathcal{I},0}(\mathbf{x}_0) = f_{\mathcal{I},0}(\mathbf{x}_1)$ if and only if $\mathbf{x}_1 = \mathbf{x}_0 + \mathbf{s}$. Also, the matching set is defined as $\mathcal{R}_{\mathcal{I}} = \{(\mathbf{x}, \mathbf{x} + \mathbf{s}) | \mathbf{x} \in \mathbb{Z}_q^n\}$.*

3. ***Efficient Range Superposition***

On input $(\mathbf{A}, \mathbf{As} + \mathbf{e})$, $b \in \{0, 1\}$, $\mathbf{x} \in \mathbb{Z}_q^n$ and $\mathbf{y} \in \mathbb{Z}_q^m$, the procedure $\mathsf{CHK}_{\mathcal{F}}$ operates as follows.

- *If $b = 0$, it computes $\mathbf{e}' = \mathbf{y} - \mathbf{Ax}$. If $\|\mathbf{e}'\| \leq B_P\sqrt{m}$, the procedure outputs 1, else output 0.*
- *If $b = 1$, it computes $\mathbf{e}' = \mathbf{y} - \mathbf{Ax} - (\mathbf{As} + \mathbf{e})$. If $\|\mathbf{e}'\| \leq B_P\sqrt{m}$, the procedure outputs 1, else output 0.*

The procedure $\mathbf{SAMP}_{\mathcal{F}}$ prepares the quantum state as per the following steps:

- *Create the following superposition:*

$$\sum_{\mathbf{e}_0 \in \mathbf{Z}_q^m} \sqrt{\mathcal{D}_{\mathbb{Z}_q^m, B_P}(\mathbf{e}_0)}\, |\mathbf{e}_0\rangle\,. \tag{A9}$$

- *Create a uniform superposition over $\mathbf{x} \in \mathbf{Z}_q^n$ to obtain the state:*

$$\frac{1}{\sqrt{q^m}} \sum_{\mathbf{x} \in \mathbb{Z}_q^n, \mathbf{e}_0 \in \mathbf{Z}_q^m} \sqrt{\mathcal{D}_{\mathbb{Z}_q^m, B_P}(\mathbf{e}_0)}\, |\mathbf{x}\rangle\, |\mathbf{e}_0\rangle\,. \tag{A10}$$

- *Now, using the problem instance $\mathcal{I} = (\mathbf{A}, \mathbf{As} + \mathbf{e})$ and the bit b to create the state*

$$\frac{1}{\sqrt{q^m}} \sum_{\mathbf{x} \in \mathbb{Z}_q^n, \mathbf{e}_0 \in \mathbf{Z}_q^m} \sqrt{\mathcal{D}_{\mathbb{Z}_q^m, B_P}(\mathbf{e}_0)}\, |\mathbf{x}\rangle\, |\mathbf{e}_0\rangle\, |\mathbf{Ax} + \mathbf{e}_0 + b \cdot (\mathbf{As} + \mathbf{e})\rangle \tag{A11}$$

- *Since* $\mathbf{e}_0$ *can be computed from* x *and the last register,* b *and the problem instance* $\mathcal{I}$*, one can uncompute the register containing* $\mathbf{e}_0$ *and obtain the state*

$$\frac{1}{\sqrt{q^m}} \sum_{\mathbf{x}\in\mathbb{Z}_q^n, \mathbf{e}_0\in\mathbf{Z}_q^m} \sqrt{\mathcal{D}_{\mathbb{Z}_q^m, B_P}(\mathbf{e}_0)}\, |\mathbf{x}\rangle\, |\mathbf{A}\mathbf{x} + \mathbf{e}_0 + b\cdot(\mathbf{A}\mathbf{s}+\mathbf{e})\rangle \tag{A12}$$

$$= \frac{1}{\sqrt{q^m}} \sum_{\mathbf{x}\in\mathbb{Z}_q^n, \mathbf{e}_0\in\mathbf{Z}_q^m} \sqrt{(f'_{\mathcal{I},b})(\mathbf{y})}\, |\mathbf{x}\rangle\, |\mathbf{y}\rangle\,. \tag{A13}$$

4. ***Claw-free Property*** *Suppose there exists an adversary* $\mathcal{A}$*, on input* $\mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{e}$ *can output the pair* $(\mathbf{x}_0, \mathbf{x}_1) \in \mathcal{R}_k$*. This adversary can be used to break the* LWE *problem since* $\mathbf{x}_1 - \mathbf{x}_0 = \mathbf{s}$.

Appendix A.2 NTCFs from RLWE

Definition A3 (TCFs from RLWE problem [12])**.** *Let* λ *be the security parameter,* $n = 2^{\log\lambda}$*. Set*

- *Ring* $R = \mathbb{Z}[x]/(x^n+1)$.
- *Modulus* $q = \mathsf{poly}(n)$, $R_q = R/qR$.
- $m = \Omega(\log q)$*: determines the dimension of range space* $\chi = D_{\mathbb{Z}_q^n, B_V}$*: the noise distribution.*
- B_P*: the noise bound for function evaluation satisfies the constraints:*
  - $B_P \geq \Omega(nmB_V)$,
  - $2B_P\sqrt{nm} \leq q/(C_T\sqrt{n\log q})$ *for some constant* C_T.

The domain is $R_q = \mathbb{Z}_q[x]/(x^n+1)$ *and the range is* R_q^m.
The problem instance $\mathcal{I} = (\mathbf{a}, \mathbf{a}\cdot s + \mathbf{e})$*, where* $s \in R_q$*,* $a_i, e_i \in R_q$ *for all* $i \in [m]$*,* $\mathbf{a} = [a_1, \ldots, a_m]^T$*,* $\mathbf{e} = [e_1, \ldots, e_m]^T$.
For $b \in \{0,1\}$*, the density function* $f_{\mathcal{I},b}(x)$ *is defined as follows:*

$$\forall \mathbf{y} \in R_q^m, (f_{\mathcal{I},b}(x))(\mathbf{y}) = D_{\mathbb{Z}^{nm}, B_P}(\mathbf{y} - \mathbf{a}\cdot x - b\cdot\mathbf{a}\cdot s), \tag{A14}$$

where $\mathbf{y} = [y_1, \ldots, y_m]^T$*, and each* y_i *can be represented as element in* $\mathbb{Z}_q^n$*; similarly for* $\mathbf{a}\cdot x$ *and* $\mathbf{a}\cdot s$.

1. ***Efficient Key Generation***
   - *The algorithm* $\mathsf{GenTrap}(1^n, 1^m, q)$ *returns* $\mathbf{a} \in R_q^m$ *and a trapdoor* t_k *such that the distribution of* $\mathbf{a}$ *is negligibly (in* n*) close to the uniform distribution on* R_q^m.
   - *Sample a uniformly random* $\mathbf{s} \leftarrow R^q$ *and a vector* $\mathbf{s} \leftarrow \chi$.
   - *Compute* $\mathbf{a}\cdot s + \mathbf{s}$.
   - $\mathsf{GEN}_{\mathcal{F}}(1^\lambda)$ *returns*

$$(\mathcal{I}, t_k) = ((\mathbf{a}, \mathbf{a}\cdot s + \mathbf{s}), (t_k, s)). \tag{A15}$$

2. ***Trapdoor Injective Pair***
   - *The support of the density function of* $f_{\mathcal{I},b}(x)$ *is*

$$\mathsf{SUPP}(f_{\mathcal{I},b}(x)) = \{\mathbf{y} \in R_q^m : \|\mathbf{y} - \mathbf{a}\cdot x - b\cdot\mathbf{a}\cdot s\| \leq B_P\sqrt{nm}\}. \tag{A16}$$

   For $\mathcal{I} = (\mathbf{a}, \mathbf{a}\cdot s + \mathbf{e})$*, the inversion algorithm* $\mathsf{INV}_{\mathcal{F}}(t_k, b, \mathbf{y} \in R_q^m) \equiv \mathsf{INVERT}(t_k, \mathbf{a}, \mathbf{y}) - b\cdot s$*. We have that, for all* $\mathbf{y} \in \mathsf{SUPP}(f_{\mathcal{I},b}(x))$*,* $\mathsf{INVERT}(t_k, \mathbf{a}\, y) = x + b\cdot s$*. Hence, it follows that* $\mathsf{INV}_{\mathcal{F}}(t_k, b, \mathbf{y}) = x$.
   - *Injective pair: From the construction, it follows that* $f_{\mathcal{I},0}(x_0) = f_{\mathcal{I},0}(x_1)$ *if and only if* $x_1 = x_0 + s$*. Denote the matching set* $\mathcal{R}_{\mathcal{I}} = \{(x, x+s) | x \in R_q\}$.
3. ***Efficient Range Superposition*** *Perform similar operations as NTCFs from LWE, on input* $(\mathbf{a}, \mathbf{a}\cdot s + \mathbf{s})$*,* $b \in \{0,1\}$*,* $x \in R^q$ *and* $\mathbf{y}' \in R_q^m$*, the procedure* $\mathsf{CHK}_{\mathcal{F}_{\mathsf{RLWE}}}$ *outputs* 1 *if* $\mathbf{y} \in \mathsf{SUPP}(f'_{\mathcal{I},b}(x))$ *and* 0 *otherwise. Note that the condition that* $\mathsf{CHK}_{\mathcal{F}_{\mathsf{RLWE}}}$ *needs to check*

is $\|\mathbf{y}' - \mathbf{a} \cdot x - b \cdot \mathbf{y}\| \leq B + P\sqrt{n \cdot m}$*. The procedure* $\mathbf{SAMP}_{\mathcal{F}}$ *prepares the quantum states similar to the one built on LWE in the previous section.*

4. ***Claw-free Property*** *Suppose there exists an adversary* $\mathcal{A}$*, on input* $\mathbf{y} = \mathbf{a} \cdot s + \mathbf{s}$ *can output the pair* $(x_0, x_1) \in \mathcal{R}_k$*. This adversary can be used to break the* RLWE *problem since* $x_1 - x_0 = s$*.*

Appendix B

All appendix sections must be cited in the main text. In the appendices, Figures, Tables, etc. should be labelled, starting with "A"—e.g., Figure A1, Figure A2, etc.

References

1. Shor, P. Algorithms for quantum computation: discrete logarithms and factoring. In Proceedings of the Proceedings 35th Annual Symposium on Foundations of Computer Science, 1994, pp. 124–134. https://doi.org/10.1109/SFCS.1994.365700.
2. Goldwasser, S. On the Hardness of the Shortest Vector Problem **1999**.
3. Stephens-Davidowitz, N. Dimension-preserving reductions between lattice problems.
4. Peikert, C. A Decade of Lattice Cryptography. *Found. Trends Theor. Comput. Sci.* **2016**, *10*, 283–424. https://doi.org/10.1561/0400000074.
5. Regev, O. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. *J. ACM* **2009**, *56*. https://doi.org/10.1145/1568318.1568324.
6. Peikert, C.; Regev, O.; Stephens-Davidowitz, N. Pseudorandomness of Ring-LWE for Any Ring and Modulus. In Proceedings of the Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, New York, NY, USA, 2017; STOC 2017, p. 461–473. https://doi.org/10.1145/3055399.3055489.
7. Babai, L. On Lovász' lattice reduction and the nearest lattice point problem. *Combinatorica* **1986**, *6*, 1–13. https://doi.org/10.1007/BF02579403.
8. Alwen, J.; Peikert, C. Generating shorter bases for hard random lattices **2009**. pp. 75–86.
9. Micciancio, D.; Peikert, C. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In Proceedings of the Advances in Cryptology – EUROCRYPT 2012; Pointcheval, D.; Johansson, T., Eds., Berlin, Heidelberg, 2012; pp. 700–718.
10. Goldwasser, S.; Micali, S.; Rivest, R.L. A "Paradoxical" Solution to The Signature Problem. In Proceedings of the Advances in Cryptology; Blakley, G.R.; Chaum, D., Eds., Berlin, Heidelberg, 1985; pp. 467–467.
11. Brakerski, Z.; Christiano, P.; Mahadev, U.; Vazirani, U.; Vidick, T. A Cryptographic Test of Quantumness and Certifiable Randomness from a Single Quantum Device. *J. ACM* **2018**, *68*. https://doi.org/10.1145/3441309.
12. Brakerski, Z.; Koppula, V.; Vazirani, U.; Vidick, T. Simpler Proofs of Quantumness, 2020, [arXiv:quant-ph/2005.04826].
13. Zhu, D.; Kahanamoku-Meyer, G.; Lewis, L.; Noel, C.; Katz, O.; Harraz, B.; Wang, Q.; Risinger, A.; Feng, L.; Biswas, D.; et al. Interactive cryptographic proofs of quantumness using mid-circuit measurements. *Nature Physics* **2023**, *19*, 1–7. https://doi.org/10.1038/s41567-023-02162-9.
14. Lewis, L.; Zhu, D.; Gheorghiu, A.; Noel, C.; Katz, O.; Harraz, B.; Wang, Q.; Risinger, A.; Feng, L.; Biswas, D.; et al. Experimental implementation of an efficient test of quantumness. *Phys. Rev. A* **2024**, *109*, 012610. https://doi.org/10.1103/PhysRevA.109.012610.