Review on IPQ-LWE-TCFs

Peter P. Rohde[1, 2, 3, *] and Minh Thuy Truc Pham[1, †]
[1] *BTQ Technologies, 16-104 555 Burrard Street, Vancouver BC, V7X 1M8 Canada*
[2] *Center for Engineered Quantum Systems, School of Mathematical & Physical Sciences, Macquarie University, NSW 2109, Australia*
[3] *Hearne Institute for Theoretical Physics, Department of Physics & Astronomy, Louisiana State University, Baton Rouge LA, United States*

(Dated: May 22, 2024)

* Narrative: bridging the gap between QC and crypto researchers who aren't experts in both.
* How to write an abstract: https://www.adelaide.edu.au/writingcentre/ua/media/26/learningguide-writinganabstract.pdf

CONTENTS

* dr.rohde@gmail.com; http://www.peterrohde.org
† pm.thuytruc@gmail.com

I. INTRODUCTION

A. 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.

Classical cryptography relies on mathematical problems assumed that are computationally hard to solve. These problems, such as integer factorization or 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, for example, can efficiently solve these problems, making current cryptographic methods vulnerable under quantum attacks.

In the past few year, the rapid development of the quantum computer era lead to many concerns in the security perspective of the existing cryptography schemes. Post-quantum cryptography thrive as a field to build new cryptographic constructions that is secure under both classical and quantum computer attack, which can be sum up to the term "quantum resistance".

In the post-quantum cryptography context, there are many approaches using many difference building blocks: *hash-based, code-based, multivariate-based, isogeny-based cryptography, lattice-based*

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

Lattice-based cryptography can be considered the most promising branch in post-quantum cryptography today due to its versatility, supporting a wide range of cryptographic primitives, from symmetric to asymmetric, from digital signatures to encryption, and its efficiency with running times and key sizes that are more efficient compared to classical algorithms.

B. Overview of Quantum Computing

1. Quantum computing and quantum cryptography

2. Communication between a quantum computer and classical devices.

C. Notation

This subsection introduces the notations used throughout this note:

- q: an odd prime integer;
- λ: security parameter;
- $\mathbb{N}$: the set of nature number;
- $\mathbb{Z}$: 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 $p(\lambda)$ is said to be negligible if $p(\lambda)$ is smaller than all polynomial fractions for such a sufficiently large λ. We define an event to occur with overwhelming probability when its probability of occurrence is at least $1 - p(\lambda)$, where $p(\lambda)$ is a negligible function.
- Vectors are represented using lowercase bold letters: $\mathbf{s}$, meanwhile matrices are represented using capital bold letters: $\mathbf{A}$. The inner product of vectors is denoted using angular brackets, as $\langle \mathbf{a}, \mathbf{s} \rangle$. The transpose of a vector or matrix is denoted using a superscript "T", as $\mathbf{s}^T$ or $\mathbf{A}^T$, respectively.
- $\mathcal{L}(\mathbf{A})$: a lattice with a basis $\mathbf{A}$.
- $\| \cdot \|$ denotes the Euclidean norm.

II. QUANTUM COMPUTING AND COMMUNICATIONS

* What are the advantages of using quantum computing and the quantum computer?

A. Quantum states

- Definition
- Entanglement
- Measurement
- No-cloning theorem

B. Quantum Cryptography and the security prespective

III. LATTICE-BASED POST-QUANTUM CRYPTOGRAPHY

A. Lattices

1. Definitions

Lattice 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 $P(\mathbf{B}) = \{\mathbf{Bx} | \mathbf{x} \in [0,1)^n\}$ 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.

For a basis $\mathbf{B}$ of $\mathcal{L}$, 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.

2. Hardness assumptions

For a given lattice $\mathcal{L}(\mathbf{A})$, one basic parameter is the length of the shortest vector $\mathbf{s}$, denoted as $\lambda_1 = \|\mathbf{s}\|$. This parameter is also called *successive minima* if one consider 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 $\lambda_n(\mathcal{L}) = \inf\{r | \dim(\mathsf{span}(\mathcal{L} \cap \overline{\mathcal{B}}(0, r))) \le n\}$.

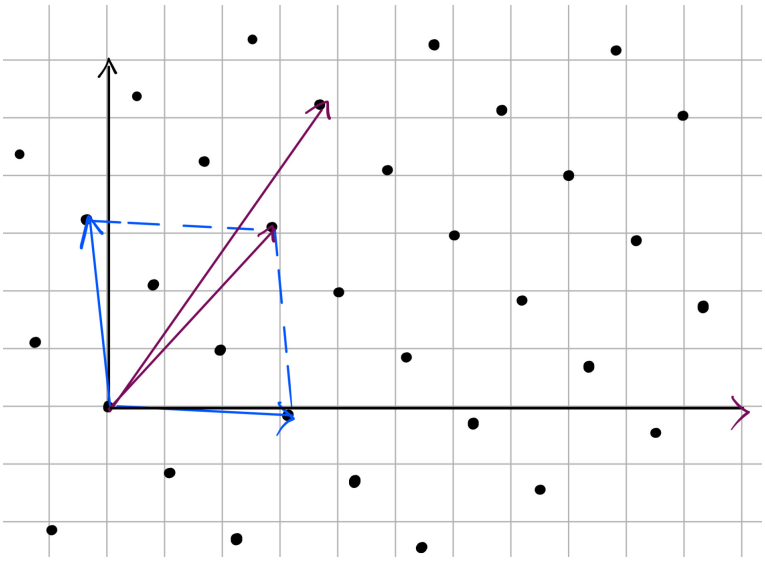

Figure 1: Consider a two-dimensional lattice $\mathcal{L}$, a basis for $\mathcal{L}$ consists of two non-zero vectors u and v such that any vector in $\mathcal{L}$ can be written as a linear combination of u and v. A good basis for $\mathcal{L}$ will have u and with lengths that are close to each other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

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_γ where the goal is to find a nonzero vector that is of length at most $\gamma = \gamma(n) \geq 1$ 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., $d = \min_{\mathbf{0} \neq \mathbf{v} \in \mathcal{L}} \|\mathbf{v}\|$. This problem is known to be **NP-hard** in the ℓ_∞ norm, meaning there is no known polynomial-time algorithm to solve it (Goldwasser, 1999). For the ℓ_2 (Euclid) norm, it remains a long standing open prorem.

Another important problem is the Shortest Independent Vectors Problem (SIVP_γ) with an approximation factor γ, a given a lattice $\mathcal{L}(\mathbf{A})$ with an arbitrary basis $\mathbf{A}$, SIVP_γ 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 III.A.3 and (Stephens-Davidowitz, 2015).

3. 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 theirs 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.

B. SIS and LWE problems

Let us first consider solving the linear equation

$$\mathbf{As} = \mathbf{y} \mod q \tag{3.1}$$

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 3.1 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 (Peikert, 2016).

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:

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 defined the hardness of the problem (the bigger n is, the harder the problem become), 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 (Peikert, 2016).

2. The learning with errors problem (LWE)

This subsection recalls the definition of the Learning With Errors (LWE) problem (Peikert, 2016). Let's revisit the equation 3.1, 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

$$\mathbf{As} + \mathbf{e} = \mathbf{y} \mod q \tag{3.2}$$

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 are mentioned in III.A.3.

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 ensuring 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 as

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

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

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

The utilization of Gaussian distributions come from its ability to effectively connect the Learning With Errors (LWE) problem to 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 showed 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 $\gamma = \tilde{O}(n/\alpha)$ (Peikert *et al.*, 2017; Regev, 2009).

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 III.A.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 (Babai and L., ????). 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. Here's the corrected version:

A sequence of works has been carried out to develop more trapdoor mechanisms. Two of the notable approaches are (Alwen and Peikert, 2009) and (Micciancio and Peikert, 2012).

(Alwen and Peikert, 2009) 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 $\mathbf{AT_A} = \mathbf{0}$ as the public matrix and its secret trapdoor.

The second approach is the trapdoor technique from (Micciancio and Peikert, 2012), which we focus on more due to its simpler, more elegant nature, and its support for more efficient cryptographic constructions.

(Micciancio and Peikert, 2012) 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 $\mathbf{A}\begin{pmatrix}\mathbf{R}\\ \mathbf{I}\end{pmatrix} = \mathbf{HG}$ 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 $\mathbf{As} + \mathbf{y}$ 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.

C. 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 offers 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.

1. The initial idea.

Let's recall that the SIS problem in equation 3.1 yields a very simple collision-resistant hash function that is provably secure if worst-case lattice problems are hard:

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

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

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

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

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

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

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

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

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

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 mention above, this corresponds to taking

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

and

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

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

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 $(y_1, \dots, y_n)^T \in \mathcal{I}$ if and only if $(-y_n, y_1 \dots, y_{n-1})^T \in \mathcal{I}$. 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. 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 $a_1, \dots, a_m \in R_q$ sampled independently and uniformly at random together with $b_1, \dots, b_m \in R_q$ where $b_i = a_i s + e_i \mod R_q$ for $s \in R_q$, and $e_i \leftarrow \chi$, find s.

4. Trapdoor algorithms in the ring setting

The trapdoor mechanism techniques from (Micciancio and Peikert, 2012) can be generalised to the ring setting where there exists an efficient randomised algorithm $\mathsf{GenTrap}(1^n, 1^m, q)$ that returns $\mathbf{a} = (a_1, \dots, a_m) \in R_q^m$ 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.

IV. TRAPDOOR CLAW FREE FUNCTIONS - TCFS

(Brakerski *et al.*, 2018; Goldwasser *et al.*, 1985)

In this section, we recall the definition of the trapdoor claw-free functions (Goldwasser *et al.*, 1985), (Brakerski *et al.*, 2018)

Trapdoor claw-free functions (TCF) is a 2-to-1 one-way function $f_{\mathcal{I}}(x)$ such that it is hard for a (quantum) polynomial-time adversary to simulatenously 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 function f_0, f_1 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 possible for one to find all the pre-images of y.

More formally, we can define the trapdoor claw-free function as follows:

Definition 1 (Trapdoor Claw-Free Function (TCF))**.** *Let λ be the security parameter, and $\mathcal{X}$, $\mathcal{Y}$ be finite sets. We require the family of functions $\mathcal{F}$ satisfies the following properties:*

- *For each public key k, there are two functions $\{f_{k,b} : \mathcal{X} \to \mathcal{Y}\}_{b\in\{0,1\}}$ 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 $y = f_{k,b}(x)$).*
- *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 $y = f_{k,0}(x_0) = f_{k,1}(x_1)$.*

However, a fully constructive Trapdoor Claw-Free (TCF) functions that satisfies our cryptographic requirements have not been explored yet. The paper (Brakerski *et al.*, 2018) considers a "relaxed" version of the TCFs, referred to as **Noisy trapdoor claw-free functions (NTCFs)**. In NTCFs, the range of functions $f_{k,0}$ and $f_{k,1}$ is now 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.

Definition 2 (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*

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

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:*

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

2. ***Trapdoor Injective Pair.***

   - *Trapdoor: There exists an efficient deterministic algorithm $\mathsf{INV}_{\mathcal{F}}$ such that with overwhelming probability over the choice of $(k, t_k) \leftarrow \mathsf{GEN}_{\mathcal{F}}(1^{\lambda})$, the following holds:*

   $$\text{for all } b \in \{0,1\},\ \mathbf{x} \in \mathcal{X} \text{ and } y \in \mathsf{SUPP}(f_{k,b}(x)),\ \mathsf{INV}_{\mathcal{F}}(t_k, b, y) = x.$$

   - *Injective pair: For all keys $k \in \mathcal{K}_{\mathcal{F}}$, there exists a perfect matching $\mathcal{R}_k \subseteq \mathcal{X}\times\mathcal{X}$ such that $f_{k,0}(x_0) = f_{k,1}(x_1)$ if and only if $(x_0, x_1) \in \mathcal{R}_k$.*

3. ***Efficient Range Superposition.*** *For all keys $k \in \mathcal{K}_{\mathcal{F}}$ and $b \in \{0,1\}$ there exists a function $f'_{k,b} : \mathcal{X} \to \mathcal{D}_{\mathcal{Y}}$ such that the following hold:*

   - *For all $(x_0, x_1) \in \mathcal{R}_k$ and $y \in \mathsf{SUPP}(f'_{k,b})(x_b)$, $\mathsf{INV}_{\mathcal{F}}(t_k, b \oplus 1, y) = x_{x\oplus 1}$.*
   - *There exists an efficient deterministic procedure $\mathsf{CHK}_{\mathcal{F}}$ that, on input k, $b \in \{0,1\}$, $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, returns 1 if $y \in \mathsf{SUPP}(f'_{k,b}(x))$ and 0 otherwise. Note that $\mathsf{CHK}_{\mathcal{F}}$ is not provided the trapdoor t_k.*
   - *For every k and $b \in \{0,1\}$,*

   $$E_{x\leftarrow_U\mathcal{X}}[H^2(f_{k,b}(x), f'_{k,b}(x))] \leq 1/50.$$

   Here H^2 is the Hellinger distance. Moreover, there exists an efficient procedure $\mathsf{SAMP}_{\mathcal{F}}$ that on input k and $b \in \{0,1\}$ prepares the state

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

4. ***Claw-free Property.*** *For any PPT adversary $\mathcal{A}$, there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that the following holds:*

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

1. TCFs from LWE

Given a LWE sample $(\mathbf{A}, \mathbf{y} = \mathbf{As} + \mathbf{e})$, we can define the TCF family by letting $f_0(\mathbf{x}) = \mathbf{Ax} + \mathbf{e}_0$ and $f_1(\mathbf{x}) = \mathbf{Ax} + \mathbf{e}_0 + \mathbf{y}$. Substituting $\mathbf{y} = \mathbf{As} + \mathbf{e}$, we have that $f_1(\mathbf{x}) = \mathbf{A}(\mathbf{x} + \mathbf{s}) + \mathbf{e}_0 + \mathbf{e}$.

- If $\mathbf{e}_0 = \mathbf{0}$, then $f_1(\mathbf{x}) = f_0(\mathbf{x} + \mathbf{s})$; however, we need to hide the information of $\mathbf{x}$, it is essential to sample $\mathbf{e}_0$ from some random distribution.
- By sampling $\mathbf{e}_0$ from a Gaussian distribution much wider than $\mathbf{e}$, we can ensure that the distribution of $f_1(\mathbf{x})$ and $f_0(\mathbf{x} + \mathbf{s})$ are statistically close.

Definition 3 (TCFs based on LWE (Brakerski *et al.*, 2018))**.** *Let λ be a security parameter. All other parameters are functions of λ. Let $q \geq 2$ be a prime and $\ell, n, m \geq 1$ be polynomially bounded functions of λ, and B_L, B_V, B_P be positive integers such that the following conditions hold:*

- $n = \Omega(\ell \log q + \lambda)$,
- $n = \Omega(n \log q)$,
- $B_P = \dfrac{q}{2C_T\sqrt{mn \log q}}$, *for C_T is a universal constant in the* $\mathsf{GenTrap}$ *algorithm.*
- *We have $B_L < B_V < B_P$ so that the ratios $\frac{B_P}{B_V}$ and $\frac{B_V}{B_L}$ are both super-polynomial in λ.*

1. ***Efficient Function Generation****:*
   - *The algorithm* $\mathsf{GenTrap}(1^n, 1^m, q)$ *returns a matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ and a trapdoor $t_{\mathbf{A}} = \mathbf{R}$, $m \geq \Omega(n \log q)$ 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 $D^m_{\mathbb{Z}_q, B_V}$.*
   - *Compute $\mathbf{As} + \mathbf{e}$.*
   - $\mathsf{GEN}_{\mathcal{F}}(1^\lambda)$ *returns the key pair as*

$$(k, t_k) = ((\mathbf{A}, \mathbf{As} + \mathbf{e}), t_{\mathbf{A}} = (\mathbf{R}, \mathbf{s})).$$

2. ***Trapdoor Injective Pair***
   - *For any key pair $(k, t_k) = ((\mathbf{A}, \mathbf{As}+\mathbf{e}), t_{\mathbf{A}} = (\mathbf{R}, \mathbf{s}))$, a bit $b \in \{0,1\}$, the density function $f_{\mathbf{A},b}(\mathbf{x})$ is defined as follows:*

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

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

$$\mathsf{SUPP}(f_{\mathbf{A},b}(\mathbf{x})) = \{\mathbf{y} \in \mathcal{Y} : \|\mathbf{y} - \mathbf{Ax} - b\mathbf{As}\| \leq B_P\sqrt{nm}\}.$$

     For $k = (\mathbf{A}, \mathbf{As} + \mathbf{e})$, the inversion algorithm $\mathsf{INV}_{\mathcal{F}}(t_k, b, \mathbf{y} \in \mathcal{Y}) \equiv \mathsf{INVERT}(t_{\mathbf{A}}, \mathbf{A}, \mathbf{y}) - b\mathbf{s}$. We have that, for all $\mathbf{y} \in \mathsf{SUPP}(f_{\mathbf{A},b}(\mathbf{x}))$, $\mathsf{INVERT}(t_{\mathbf{A}}, \mathbf{A}\,\mathbf{y}) = \mathbf{x} + b\mathbf{s}$. Hence, it follows that $\mathsf{INV}_{\mathcal{F}}(t_k, b, \mathbf{y}) = \mathbf{x}$.
   - *Injective pair: Let $k = (\mathbf{A}, \mathbf{As} + \mathbf{e})$. From the construction, it follows that $f_{\mathbf{A},0}(\mathbf{x}_0) = f_{\mathbf{A},1}(\mathbf{x}_1)$ if and only if $\mathbf{x}_1 = \mathbf{x}_0 + \mathbf{s}$. Denote the set $\mathcal{R}_k = \{(\mathbf{x}, \mathbf{x} + \mathbf{s}) | \mathbf{x} \in \mathbb{Z}_q^n\}$.*

3. ***Claw-free Property*** *Suppose there exists an adversary $\mathcal{A}$, on input $\mathbf{y} = \mathbf{As} + \mathbf{e}$ 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 $\mathbf{x}_1 - \mathbf{x}_0 = \mathbf{s}$.*

2. TCFs from RLWE

(Brakerski *et al.*, 2020) used the trapdoor claw-free function based on the Ring-LWE assumption, 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.

Definition 4 (TCFs from RLWE problem (Brakerski *et al.*, 2020))**.** *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.
  Each function key $k = (\mathbf{a}, \mathbf{a}\cdot s+\mathbf{e})$, 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 $b \in \{0, 1\}$, $x \in R_q$, $k = (\mathbf{a}, \mathbf{a} \cdot s + \mathbf{e})$, the density function $f_{\mathbf{a},b}(x)$ is defined as follows:

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

  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 $t_{\mathbf{a}}$ 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 the key pair as*

$$(k, t_{\mathbf{a}}) = ((\mathbf{a}, \mathbf{a} \cdot s + \mathbf{s}), t_{\mathbf{a}}).$$

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

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

     For $k = (\mathbf{a}, \mathbf{a} \cdot s + \mathbf{e})$, the inversion algorithm $\mathsf{INV}_{\mathcal{F}}(t_{\mathbf{a}}, b, \mathbf{y} \in R_q^m) \equiv \mathsf{INVERT}(t_{\mathbf{a}}, \mathbf{a}, \mathbf{y}) - b \cdot s$. We have that, for all $\mathbf{y} \in \mathsf{SUPP}(f_{\mathbf{a},b}(x))$, $\mathsf{INVERT}(t_{\mathbf{a}}, \mathbf{a}\, y) = x + b \cdot s$. Hence, it follows that $\mathsf{INV}_{\mathcal{F}}(t_{\mathbf{a}}, b, \mathbf{y}) = x$.
   - *Injective pair: Let $k = (\mathbf{a}, \mathbf{a} \cdot s + \mathbf{s})$. From the construction, it follows that $f_{\mathbf{a},0}(x_0) = f_{\mathbf{a},1}(x_1)$ if and only if $x_1 = x_0 + s$. Denote the set $\mathcal{R}_k = \{(x, x + s) | x \in R_q\}$.*

3. ***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 $x_1 - x_0 = s$.*

A. Adaptive hardcore bit property

1. One-way function and the hardcore bit property

This section introduces the definition of the hardcore bit property.

Definition 5 (One-way function)**.** *A function $f : \{0, 1\}^* \to \{0, 1\}^*$ is called one way if two condition 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 $\mathcal{A}'$, every polynomial $p(\cdot)$,*

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

Before defining the hardcore bit property, a question is raised: does the output of a one-way function, $y = f(x)$, truly conceal all information of the preimage x?

An example: Assuming that $g(x)$ is a one-way function, it follows that $f(x) = x_1||g(x)$ remains a one-way function. This is because an adversary must still invert $g(x)$ to determine x. The first bit of x 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.

The hardcore bit property (or hard-core predicate) ensures that even with the information of $y = f(x)$, guessing a bit of x is as challenging as finding x.

Definition 6 (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 $\mathcal{A}$*

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

2. Adaptive hardcore bit property for TCFs from LWE

Consider the TCFs f from LWE problem, we have that $f(x_0) = f(x_1) = y$ and we have the superposition:

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

Note that when measuring the above sate by the X basis, we obtain a string $d \in \{\{0,1\}^n$ such that $d \cdot (x_0 \oplus x_1) = c$ where x_0, x_1 are preimage of some y .

Here, d provides an equation of x_0 and x_1; to be more specific, this equation provides some information about both preimages together. (Brakerski *et al.*, 2018) used the LWE-based hardness assumption, where $x_0 + x_1 = s$, where s is some fixed secret independent of y.

Hence, measuring the first register in the Hadamard basis yields a uniformly random d such that $d \cdot s = c$. If we repeat the entire procedure $O(n)$ times yield linearly independent such d's, which suffices to recover s with high probability.

We may starting from the above state, measure in the X basis, obtaining as before an equation d such that $d \cdot (x_0 \oplus x_1) = c$. However, we also saw that if x_0 and $x_1 = x_0 - s$ contain the information of s for some fixed secret s; then, the Hadamard measurements alone allow us to recover s. Hence, **we need the structure of the function f is a little more complicated**, so that even $x_0, x_1 = x_0 - s$, and given many "equations" d 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 satisfies the hardcore bit property: if the adversary is able to find a tuple $(y, b, x_b, (d, c))$ with $b, c \in 0, 1$ that satisfies two checks with a probability significantly higher than $1/2$, i.e., $f_{k,b}(x_b) = y$, $d \cdot (x_0 \oplus x_1) = c$, and $d \neq 0$, it could find the whole claw (x_0, x_1).

Why does IPOF (Brakerski *et al.*, 2018) construction use the term "adaptive" for the hardcore bits property? Combining to 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 a hardcore bit for f is a 1-bit function h such that given $f(x)$ (but not x) it is hard to predict $h(x)$. Here, the hardcore bit that underlies the assumption is the function $c = h(x) = d \cdot (x_0 + x_1)$ for any d. (Brakerski *et al.*, 2018) showed that this equation can be rewrite as $d' \cdot s = c$. The property can be stated as it is hard to produce the string d' satisfies the above equation.

Moreover, we allow the adversary $\mathcal{A}$ to ***adaptively choose the string*** d'. For example, $\mathcal{A}$ may always return the same d'. 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, the adversary $\mathcal{A}$ can adaptively choose d after being given the access to the LWE sample $\mathbf{As}+\mathbf{e}$, i.e. the choice of d' depend upon the LWE sample. Due to the *leakage resilience* property of the LWE problem. 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 of the proof could be found in section 4.4.2 (Brakerski *et al.*, 2018).

V. PROOF OF QUANTUMNESS

A. 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 $L \subseteq \{0,1\}^*$. Then, the definition of an interactive proof for a statement x in a language L is as follows:

Definition 7 (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 $\mathcal{V}$.*

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

B. Proof of quantumness

In recent years, under the efforts of building quantum computers, the verification of their 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 (Brakerski *et al.*, 2018; Zhu *et al.*, 2023) 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.

Definition 8 (Proof of Quantumness (Liu *et al.*, 2022))**.** *A proof of quantumness is an interactive protocol with an efficient classical verifier satisfying:*

- ***Completeness.** There exists a polynomial-time quantum prover that can convince the verifier about its "quantum" ability.*
- ***Soundness** ϵ. Any polynomial-time classical prover convinces the verifier with probability at most $\epsilon + \mathsf{negl}$ for some negligible function* negl.

A common recent approach is to build the Interactive Proof of Quantumness protocol using TCF.

C. LWE

The basic idea of employing the TCF function $f_{k,b}(\cdot)$ is as follows:

1. The verifier samples a public key k together with a trapdoor t_k, sends k, $f_{k,0}(\cdot)$, $f_{k,1}(\cdot)$ to the prover.

2. The prover:
   - prepares a uniform superposition over $\{0,1\}^n$
   $$\sum_{b\in\{0,1\}}\sum_{x\in\{0,1\}^n}|b\rangle|x\rangle|f_{k,b}(x)\rangle,$$
   - computes $f_{k,b}(\cdot)$ on the superposition, measures the image register to obtain $y\in\mathcal{Y}$,
   - sends y to the verifier.
3. The remaining state of the prover now is
   $$|0\rangle|x_0\rangle+|1\rangle|x_1\rangle$$
   where x_0, x_1 are two preimages of y.
4. The verifier asks the prover to measure on either the standard basis or the Hadamard basis.
5. Measurement outcome:
   - Standard basis: either $(0,x_0)$ or $(1,x_1)$
   - Hadamard basis: $(c\in\{0,1\},d\in\{0,1\}^n)$ such that $d\cdot(x_0\oplus x_1)=c$.
6. The verifier checks these tests:
   - Standard basis: simply check that $f_{k,b}(x_b)=y$.
   - Hadamard basis: the verifier can recover x_0, x_1 from y using the trapdoor t_k, and check that $d\cdot(x_0\oplus x_1)=c$.

   The protocol proceeds for λ rounds. At the end of each round, the verifier samples a new pair of functions from the NTCF family and sends the new public key to the prover. In a recent paper by Zhu (Zhu *et al.*, 2023), 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. 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.

Recall about the adaptive hardcore bit property. Suppose there exists a prover capable of successfully navigating both verifier challenges. Specifically, given the public key k, this prover can generate a tuple $(y,b,x_b,(d,c))$ with $b,c\in 0,1$ that satisfies two checks with a probability significantly higher than $1/2$, i.e., $f_{k,b}(x_b)=y$, $d\cdot(x_0\oplus x_1)=c$, and $d\neq 0$.

It is important to note that in this scenario, the prover does not discover the entire claw (x_0,x_1) but rather identifies one preimage x_b along with a linear function of the other preimage x_{1-b}. Therefore, the **hardcore bit** property is indispensable, asserting that it is computationally challenging to find even a single bit of x_{1-b}—meaning the prover cannot determine a single bit of x_{1-b}, let alone the entire claw.

D. 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. (Brakerski *et al.*, 2020) 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 $H(\cdot)$ modelled as a quantum random oracle. 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 parameterised by a hash function $H : \{0,1\}^n \to \{0,1\}$.

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 $\{(y_i, c_i, d_i)\}_{i\in[\lambda]}$, 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 outputs 1, else it output $\perp$.

E. QROM

Why can QROM replace the hardcore bit property? Recall that in the LWE-based proof of quantumness construction (Brakerski *et al.*, 2018), the prover applies the trapdoor claw-free function f on a uniform superposition of inputs and then measures the image register. It then obtains the value y, which will be sent to the verifier. The remaining state of the prover now is $|0\rangle|x_0\rangle + |1\rangle|x_1\rangle$ where x_0, x_1 are two preimages of y. In other words, the prover now has a superposition over $(0, x_0)$ and $(1, x_1)$. 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 at the same time, which violates the *hardcore bit property* and the hardness assumption.

Let's assume that $H : \{0,1\}^n \to \{0,1\}$ is a one-bit hash function (H is modelled as a quantum random oracle in the security proof). Using the hash function H, (Brakerski *et al.*, 2020) 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 the Hadamard basis only and send the outcome, including a bit c and a vector d, to the verifier. It holds that (c, d) satisfies the equation $c = d \cdot (x_0 \oplus x_1) \oplus H(0, x_0) \oplus H(1, x_1)$. 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 that 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.

F. QC implementation resource tradeoffs

VI. EXPERIMENTAL RESULTS

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

REFERENCES

Alwen, Joël, and Chris Peikert (2009), "Generating shorter bases for hard random lattices," *26th International Symposium on Theoretical Aspects of Computer Science STACS 2009*, , 75–86.

Babai„ and L. (????), Combinatorica **6** (1), 1.

Brakerski, Zvika, Paul Christiano, Urmila Mahadev, Umesh Vazirani, and Thomas Vidick (2018), "A cryptographic test of quantumness and certifiable randomness from a single quantum device," J. ACM **68** (5), 10.1145/3441309.

Brakerski, Zvika, Venkata Koppula, Umesh Vazirani, and Thomas Vidick (2020), "Simpler proofs of quantumness," arXiv:2005.04826 [quant-ph].

Goldwasser, Shafi (1999), "On the hardness of the shortest vector problem," .

Goldwasser, Shafi, Silvio Micali, and Ronald L. Rivest (1985), "A "paradoxical" solution to the signature problem," in *Advances in Cryptology*, edited by George Robert Blakley and David Chaum (Springer Berlin Heidelberg, Berlin, Heidelberg) pp. 467–467.

Table I: Comparison between the Proof of Quantumness protocols

	Problem	Adaptive hardcore bit	IPOQ	Random oracle	Gate count	Adversary type
IPoQ	LWE	✓	✓	✗	$n^2 \log^2 n$	classical
textE-IPoQ	LWE	✓	✓	✗	$n^2 \log^2 n$	classical
NIPoQ	RLWE	✗	✗	✓	$n \log^2 n$	quantum
textE-NIPoQ	LWE	✗	✗	✓	$n^2 \log^2 n$	quantum

Liu, Jiahui, Qipeng Liu, and Luowen Qian (2022), "Beating classical impossibility of position verification," arXiv:2109.07517 [quant-ph].

Micciancio, Daniele, and Chris Peikert (2012), "Trapdoors for lattices: Simpler, tighter, faster, smaller," in *Advances in Cryptology – EUROCRYPT 2012*, edited by David Pointcheval and Thomas Johansson (Springer Berlin Heidelberg, Berlin, Heidelberg) pp. 700–718.

Peikert, Chris (2016), "A decade of lattice cryptography," Found. Trends Theor. Comput. Sci. **10** (4), 283–424.

Peikert, Chris, Oded Regev, and Noah Stephens-Davidowitz (2017), "Pseudorandomness of ring-lwe for any ring and modulus," in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017 (Association for Computing Machinery, New York, NY, USA) p. 461–473.

Regev, Oded (2009), "On lattices, learning with errors, random linear codes, and cryptography," J. ACM **56** (6), 10.1145/1568318.1568324.

Stephens-Davidowitz, Noah (2015), "Dimension-preserving reductions between lattice problems,".

Zhu, Daiwei, Gregory Kahanamoku-Meyer, Laura Lewis, Crystal Noel, Or Katz, Bahaa Harraz, Qingfeng Wang, Andrew Risinger, Lei Feng, Debopriyo Biswas, Laird Egan, Alexandru Gheorghiu, Yunseong Nam, Thomas Vidick, Umesh Vazirani, Norman Yao, Marko Cetina, and Christopher Monroe (2023), "Interactive cryptographic proofs of quantumness using mid-circuit measurements," Nature Physics **19**, 1–7.