e-ISSN: 2582-5208

International Research Journal of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume:06/Issue:05/May-2024 Impact Factor- 7.868 www.irjmets.com

QUANTUM COMPUTERS AND POST-QUANTUM CRYPTOGRAPHY

Neev Mehta*[1]

*[1]Witty International School, Mumbai, India.

DOI : https://www.doi.org/10.56726/IRJMETS55569

ABSTRACT

Classical computers use bits, whilst quantum computers use quantum bits (qubits). Quantum computers can perform tasks that classical computers are unable to perform. When ideal quantum computers are developed, they will be a serious threat to current encryption systems. This has prompted the development of post-quantum cryptography. The National Institute of Standards and Technology (NIST) have selected four post-quantum algorithms. Three employ lattice-based cryptography, whilst one employs hash-based cryptography. Although recent attacks on lattice-based cryptography mean that the post-quantum algorithms of the future are uncertain.

Keywords: Qubits, one-way functions, post-quantum cryptography

I. INTRODUCTION

The technological world was taken by storm when quantum computers were introduced. There was a threat to the current encryption system we use, all our encrypted private data and information could be decoded by a properly working quantum computer easily. Hence, the National Institute of Standards and Technology (NIST) encouraged the development of post-quantum cryptography methods from all around the world.

In the following sections we describe quantum computing, classical encryption and post-quantum encryption, then draw conclusions.

II. QUANTUM COMPUTING

Bits and Qubits

Traditional computers use conventional bits, which can exist in two different states: 0 or 1. This is the reason that computers can only process data in binary. These classical bits represent the value of 0 or 1, which allows the computer to process any data and instructions. Classical bits can be thought of as essentially transistors on a silicon chip, that when holding charge have the value 1, and likewise have the value 0 when not holding charge. The model of the conventional computer, made by the combination of registers, buses, RAM, etc. invented by John von Neumann in 1945, depends on these bits.[1]

However, in the 1980s, a sudden introduction of the theory behind quantum computers by David Deutsch took the technological world by storm.[2] Scientists and researchers have worked for decades, to learn more about quantum computers. Quantum computers actually don't use classical bits, but qubits instead. Qubits are special types of bits, which follow quantum mechanics instead of classical mechanics. Unlike the classical bits that can only be in one of two states at a time, qubits are different. They can exist in a superposition between both 0 and 1, which theoretically allows them to be in an infinite number of states. Two qubits can also be entangled with one another. This phenomenon is called quantum entanglement. Let's relate it with vectors to understand it better.

In Figure 1, the Y-axis denotes 1 and X-axis denotes 0. In a classical bit, the state can be either 1 or 0. This is shown by the vectors shown on the graph, which can only be in two possible directions. The length of all vectors possible is 1.

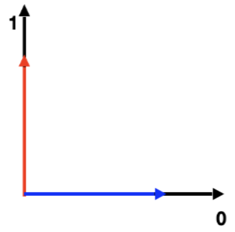


Figure 1: Classical bit

The graph in Figure 2 has the same axis as the graph in Figure 1, but rather than having two different options as states, it can stay in a superposition between the states of 0 and 1. We can say that it is in both 0 and 1 at the same time. Vectors that could represent these superpositions are shown on the left. Theoretically, a qubit could be in an infinite number of states.

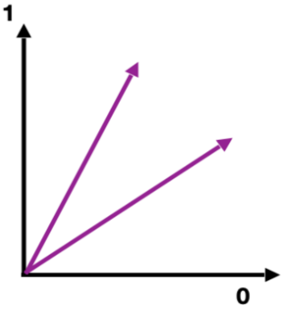


Figure 2: Qubit

The amount of information shown by n classical bits at one point in time is n. However, for qubits, this increases exponentially. The amount of information that can be displayed is 2^n, where n is the number of qubits. It is important to understand that a qubit exists in a superposition, but when it is measured, the qubit turns into either a 0 or 1, i.e. it behaves like a classical bit. This is called the collapse of the superposition. There is a certain probability of a qubit collapsing to 0 or 1, which is represented by the coefficients in the bra-ket notation.

A qubit is represented as a complex vector (bra–ket notation): ψ⟩ = α|0⟩ + β|1⟩. α and β represent the amplitudes or probabilities of the qubit being in states 0 and 1. Hence, rather than holding one value, i.e. 0 or 1, at a time, a qubit can hold two values simultaneously. Again, when there are two classical bits together, we require two different values to represent both of them, for example (0,1), (0,0), (1,1) or (1,0). However, due to quantum entanglement, two different qubits become one single quantum system. Hence, there all four states that can be possible all exist together in a superposition, such as [(0,1);(0,0);(1,1);(1,0)] All these states have a certain probability of occurring when a qubit is measured. Hence, two qubits can store four classical bits worth of data. Similarly, three entangled qubits exist in eight different states at once, which all have a certain probability of the quantum system collapsing into one of them. Hence, three qubits can store eight classical bits worth of data. This pattern is exponential, the data n qubits can store is equal to 2^n classical bits.

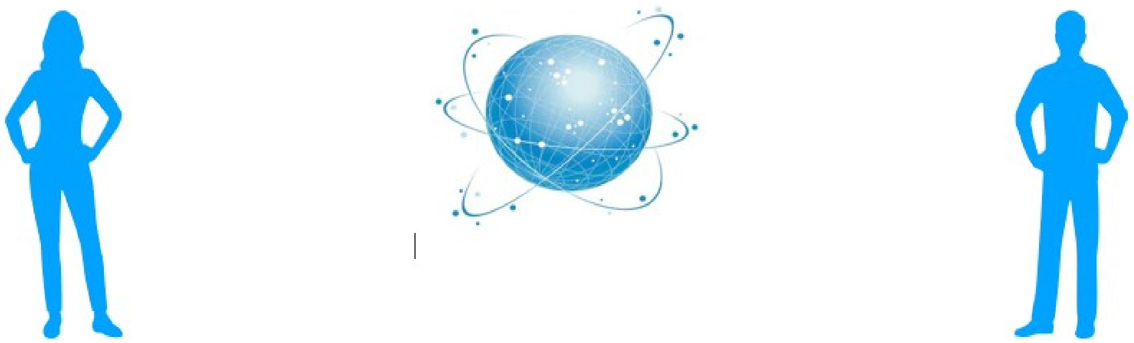
In general, we create an entangled state of the input and output registers in which every value in the domain of the function is correlated with a corresponding value in its range. However, when we make measurements, we are given just one, random, output, which can be mapped to its input. So quantum computers are only powerful when we can design algorithms that enable us to harness the aforementioned phenomena. Grover's algorithm[3] uses amplitude amplification, whilst Shor's algorithm[4] uses the quantum Fourier transform.

Grover's algorithm affects both symmetric and asymmetric encryption. With asymmetric encryption, Shor's algorithm is far more effective than Grover's algorithm, so in practice Grover's algorithm affects most symmetric cryptographic algorithms and hash functions. Basically, Grover's algorithm reduces a brute-force search algorithm from $O(n)$ steps to $O(\sqrt{n})$ steps. If we have a key of length n bits, a classical computer can brute force the key in $2n$ invocations. Whilst a quantum computer running Grover's algorithm can brute force the key in $2n/2$ invocations. Because Grover's algorithm provides only a quadratic speedup, it can be neutralised by doubling the key length in symmetric encryption or the digest length of a hash function.

Shor's algorithm can solve the integer factorisation problem, the discrete logarithm problem and the elliptic-curve discrete logarithm problem almost exponentially faster than classical computers.

III. CLASSICAL ENCRYPTION

RSA (Rivest–Shamir–Adleman) encryption allows data to be encrypted. Both sender and receiver have two keys each, a private key and public key.[5] As the name suggests, the private key is a secret key to the user, while the public key is known by the internet. Theoretically speaking, RSA encryption takes advantage of a computer's inability to factorise extremely large semi-prime numbers (product of two prime numbers). The public key is used to encrypt data into cipher text, while the private key is used to decrypt data back into plaintext. Even though private and public keys are mathematically connected, the private key cannot be calculated from the public key. Let's take an example of two people, Alice and Bob. Alice wants to send a file to Bob. In this case, Alice will use Bob's public key (available publicly) to encrypt her file, and then send the encrypted file to Bob. Bob will use his own private key to decrypt the encrypted file, and can then receive the original file. This ensures security of the data transmitted between Alice and Bob. See Figure 3.



Alice: Alice's private key Public/ Internet: Alice's public key Bob's public keBob: Bob's private key

Figure 3: Public key encryption

One-Way Functions used for encryption and cryptography

Why are semi-prime numbers used? For classical computers, it is really difficult to factorise numbers. This is because the computer has to go to each and every number before it, divide it with the actual number, and check whether it is a factor or not. For smaller values, of course, factorising is possible. However, factorising semi-prime numbers is different. This is because a semi-prime number has only two factors (disregarding the number itself and 1), hence the number cannot be broken down to its factors easily. It becomes increasingly difficult with the semi-prime number becoming larger. Two random 11-digit prime numbers can be multiplied to form an extremely large semi-prime number, but factorising this number back to two prime numbers is extremely difficult. It is considered a one-way function, since going backwards is not feasible in reality. RSA encryption takes advantage of this exact weakness of classical computers.

Another weakness of classical computers is the discrete logarithm problem. The cryptography world uses this exact problem to its advantage. The discrete logarithm problem uses modulo, a function that calculates the remainder after division. For example, 19 mod 6 = 1 (which is the remainder left). Now, the discrete logarithm problem can be written as a^x mod y = r, which to solve, the computer needs to put each and every value of x from 1 to y−1 to find r. This is because for each modulus function, there is a particular exponent that will yield all different remainders. Let's take an example. In the equation 3^x mod 7 = 4, for the '7' in modulus, the exponents 3 and 5 cause values of x such that all values produced when x is from 1 to 6 yields a different remainder. The range of values that need to be input in x is 1 to (7−1). So when a remainder is given, which is unique itself, a larger modulus number, let's say, 31 digit modulus will make it very tough for the computer to use each and every number from 1 to that 31-digit number and check whether it matches the remainder or not. This is the weakness of classical computers that the cryptography world uses.

IV. POST-QUANTUM CRYPTOGRAPHY

Introduction

Chen et al. shared the National Institute of Standards and Technology (NIST)'s understanding, as of 2016, of the status of quantum computing and post-quantum cryptography, and outlined NIST's initial plan to move forward in this space.[6] Whilst Bernstein and Lange presented an excellent review of the state of post-quantum cryptography as of 2017.[7]

Due to Grover's algorithm, symmetric encryption (such as AES) will need to double key lengths and hash algorithms (e.g. SHA-2 and SHA-3) will need to double the length of the digest. Whilst all of public-key cryptography (including RSA, ECDSA, ECC, ECDH and DSA) will cease to be secure due to Shor's algorithm. However, message authentication code (MAC) will not be impacted by quantum computing.

It is expected that a quantum computer able to factorise a 2048-bit number in less than 24 hours will be available in around 2036.[8] It is clear that there is a need for post-quantum solutions. NIST initiated a process to solicit, evaluate and standardise quantum-resistant public-key cryptographic algorithms.[9] Options for post-quantum algorithms include the following families: code-based cryptography, hash-based cryptography, lattice-based cryptography and multivariate cryptography. We shall consider lattice-based cryptography and hash-based cryptography.

Lattice Problems

Lattices are used as a source of computational hardness in the construction of cryptographic functions that are at least as hard to break as solving some underlying lattice problem. The study of lattice-based cryptography essentially began with Miklós Ajtai's discovery in 1996 that certain variants of the knapsack problem are at least as hard to break on average as the worst-case instance of certain lattice problems.[10] There are currently no known quantum algorithms for solving lattice problems that perform significantly better than the best known classical (i.e. non-quantum) algorithms. It is conjectured that there is no polynomial time quantum algorithm that approximates lattice problems to within polynomial factors.

We shall define a lattice, and describe the shortest vector problem, the shortest independent vectors problem, the closest vector problem, the short integer solution problem and learning with errors.

Lattice

A lattice is an infinite set of points in a coordinate space with the properties that coordinate-wise addition or subtraction of two points in the lattice produces another lattice point, the lattice points are all separated by some minimum distance, and every point in the space is within some maximum distance of a lattice point.

The real coordinate space of dimension n, denoted R^n, is the set of all ordered n-tuples of real numbers, that is the set of all sequences of n real numbers, also known as coordinate vectors. Whilst a vector space is a set whose elements, often called vectors, may be added together and multiplied ('scaled') by numbers called scalars. A real vector space is based on the real coordinate space.

A set B of vectors in a vector space V is called a basis if every element of V may be written in a unique way as a finite linear combination of elements of B. The coefficients of this linear combination are referred to as components or coordinates of the vector with respect to B. The elements of a basis are called basis vectors.

The standard basis of a coordinate vector space (such as R^n) is the set of vectors, each of whose components are all zero, except one that equals 1. For example, in the case of the Euclidean plane R^2 formed by the pairs (x, y) of real numbers, the standard basis is formed by the vectors $\mathbf{e}_x = (1, 0)$ and $\mathbf{e}_y = (0, 1)$. Similarly, the standard basis for the three-dimensional space R^3 is formed by vectors $\mathbf{e}_x = (1,0,0)$, $\mathbf{e}_y = (0,1,0)$ and $\mathbf{e}_z = (0,0,1)$.

A lattice $L \subset R^n$ is the set of all integer linear combinations of vectors from a basis $\{\mathbf{b}_1, ..., \mathbf{b}_n\}$ of R^n. In other words, $L = \{\sum a_i\mathbf{b}_i : a_i \in Z\}$. A lattice can be described as a free abelian group of dimension n which spans the vector space R^n. For any basis of R^n, the subgroup of all linear combinations with integer coefficients of the basis vectors forms a lattice, and every lattice can be formed from a basis in this way. For example, Z^n is a lattice, generated by the standard basis for R^n. The basis for a lattice is not unique. For example, the vectors (3,1,4), (1,5,9) and (2,−1,0) form an alternative basis for Z^3.

Shortest Vector Problem

The shortest vector problem (SVP) is the most famous and widely studied computational problem on lattices. It involves approximating the minimal Euclidean length of a non-zero lattice vector. SVP has been studied by mathematicians (in the equivalent language of quadratic forms) since the 19th century because of its connection to many problems in number theory. One of the earliest references to SVP in the computer science literature was a 1981 publication Peter van Emde Boas, where the problem is conjectured to be NP-hard.[11]

Formally, in the shortest vector problem a basis of a vector space V and a norm N (often L^2) are given for a lattice L, and the goal is to find the shortest non-zero vector in V, as measured by N, in L. In other words, the algorithm should output a non-zero vector v such that $\|v\|_N = \lambda(L)$. In the γ-approximation version, SVP_γ, one must find a non-zero lattice vector of length at most $\gamma \cdot \lambda(L)$ for given $\gamma \geq 1$.

When the shortest vector problem is applied to cryptography, a good basis corresponds to a private key, a bad basis corresponds to a public key and a point on the lattice corresponds to a message.

Shortest Independent Vectors Problem

The shortest independent vectors problem (SIVP) is, given a lattice L of dimension n, find an algorithm that outputs n linearly independent $v_1, v_2, \ldots, v_n$ so that max $\|v_i\| \leq \max_B \|b_i\|$, where the right-hand side considers all bases $B = \{b_1, \ldots, b_n\}$ of the lattice.

In the γ-approximate version, given a lattice L with dimension n, the goal is to find n linearly independent vectors $v_1, v_2, \ldots, v_n$ of length max max $\|v_i\| \leq \gamma\lambda_n(L)$, where $\lambda_n(L)$ is the nth successive minimum of L.

Closest Vector Problem

The closest vector problem (CVP) is a generalisation of the shortest vector problem. Given a lattice and a target point, the challenge is to find the lattice point closest to the target. CVP has been studied in mathematics (in the equivalent language of quadratic forms) since the nineteenth century. One of the first references to CVP (under the name 'nearest vector problem') in the computer science literature was, again, in the publication by Peter van Emde Boas, where the problem was shown to be NP-hard to solve exactly.[11]

Formally, in the closest vector problem, a basis of a vector space V and a metric M (often L^2) are given for a lattice L, as well as a vector v in V but not necessarily in L. The goal is to find the vector in L closest to v (as measured by M). In the γ-approximation version CVP$_\gamma$, the goal is to find a lattice vector at distance at most γ.

Short Integer Solution Problem

When Ajtai introduced lattice-based cryptography, he presented a family of one-way functions based on short integer solution problems.[10] Short integer solution (SIS) and ring-SIS problems are two average-case problems that are used in lattice-based cryptography constructions.

Let $A \in \mathbb{Z}^{n\times m}_q$ be an $n \times m$ matrix with entries in $\mathbb{Z}_q$ that consists of m uniformly random vectors $\mathbf{a_i} \in \mathbb{Z}^n_q$: $A = [\mathbf{a_1}| \cdots |\mathbf{a_m}]$. Find a nonzero vector $\in \mathbb{Z}^m$ such that for some norm $\| \cdot \|$:

- $0 < \|\| \leq \beta$,
- $f_A(\mathbf{x}) := A\mathbf{x} = 0 \in \mathbb{Z}^n_q$.
- A solution to SIS without the required constraint on the length of the solution ($\|\| \leq \beta$) is easy to compute using the Gaussian elimination technique. We also require that $\beta < q$, otherwise $= (q, 0, \ldots, 0) \in \mathbb{Z}^m$ is a trivial solution.
- In order to guarantee that $fA(\mathbf{x})$ has a non-trivial short solution, we require that:
- $\beta \geq \sqrt{(n \log q)}$, and
- $m \geq n \log q$.

Learning With Errors

Oded Regev originally introduced the learning with errors (LWE) problem at STOC '05.[12] Learning with errors is based on the idea of representing secret information as a set of equations with errors. In other words, the value of a secret is hidden by introducing noise to it. It refers to the computational problem of inferring a linear n-ary function f over a finite ring from given samples $y_i = f(\mathbf{x}_i)$, some of which may be erroneous.

More formally, let Z_q denote the ring of integers modulo q and let Z^n_q denote the set of n-vectors over Z_q. There exists a certain unknown linear function $f : Z^n_q \to Z_q$, and the input to the LWE problem is a sample of pairs $(\mathbf{x},y)$, where $\mathbf{x} \in Z^n_q$ and $y \in Z_q$, so that with high probability $y = f(\mathbf{x})$. Furthermore, the deviation from the equality is according to some known noise model. The problem calls for finding the function f, or some close approximation thereof, with high probability.

Denote by $\mathbb{T} = R/Z$ the additive group on reals modulo one. Let $\mathbf{s} \in Z^n_q$ be a fixed vector. Let ϕ be a fixed probability distribution over $\mathbb{T}$. Denote by $A_{\mathbf{s},\phi}$ the distribution on $Z^n_q \times \mathbb{T}$ obtained as follows.

1. Pick a vector $\mathbf{a} \in Z^n_q$ from the uniform distribution over Z^n_q.
2. Pick a number $e \in \mathbb{T}$ from the distribution ϕ.
3. Evaluate $t = \langle \mathbf{a,s} \rangle/q+e$, where $\langle \mathbf{a,s} \rangle = \sum^n_{i=1} a_i s_i$ is the standard inner product in Z^n_q, the division is done in the field of reals (or more formally, this 'division by q' is notation for the group homomorphism $Z_q \longrightarrow \mathbb{T}$ mapping $1 \in Z_q$ to $1/q + Z \in \mathbb{T}$), and the final addition is in $\mathbb{T}$.
4. Output the pair $(\mathbf{a}, t)$.

The learning with errors problem LWE$_{q,\phi}$ is to find $\mathbf{s} \in \mathrm{Z}^{n}{}_{\mathrm{q}}$, given access to polynomially many samples of choice from $A_{\mathbf{s},\phi}$.

For every $\alpha > 0$, denote by D_α the one-dimensional Gaussian with zero mean and variance $\alpha^2/(2\pi)$, that is, the density function is $D_\alpha\ (x) = \rho_\alpha\ (x)/\alpha$ where $\rho_\alpha\ (x) = e^{-\pi(|x|/\alpha)2}$, and let Ψ_α be the distribution on $\mathbb{T}$ obtained by considering D_α modulo one. The version of LWE considered in most of the results would be LWE$_{q,\Psi\alpha}$.

Regev showed that the LWE problem is as hard to solve as several worst-case lattice problems. He proposed a public-key cryptosystem based on the hardness of the LWE problem. The cryptosystem and the proof of security and correctness are both completely classical. The system is characterised by two integers m and q, and a probability distribution χ on $\mathbb{T}$. The setting of the parameters used in proofs of correctness and security is

- $q \geq 2$, a prime number between n^2 and $2n^2$,
- $m = (1 + \varepsilon)(n + 1) \log q$ for an arbitrary constant ε, and
- $\chi = \Psi_{\alpha(n)}$ for $\alpha(n) \in o(1/\sqrt{n} \log n)$, where Ψ_β is a probability distribution obtained by sampling a normal variable with mean 0 and standard variation $\beta\ /\sqrt{(2\pi)}$ and reducing the result modulo 1.
- The cryptosystem is then defined by:
- Private key: The private key is an $\mathbf{s} \in \mathbb{Z}^n{}_q$ chosen uniformly at random.
- Public key: Choose m vectors $\mathbf{a}_1, \ldots , \mathbf{a}_m \in \mathbb{Z}^n{}_q$ independently from the uniform distribution and choose error offsets $e_1, \ldots , e_m \in \mathbb{T}$ independently according to χ. Then the public key consists of $(\mathbf{a}_i, b_i = \langle \mathbf{a}_i, \mathbf{s}\rangle/q + e_i)^m{}_{i=1}$.
- Encryption: The encryption of a bit $x \in \{0, 1\}$ is done by choosing a random subset S of $[m]$ and then defining Enc(x) as $(\Sigma_{i\in S}\ \mathbf{a}_i, x/2 + \Sigma_{i\in S}\ b_i)$.
- Decryption: The decryption of $(\mathbf{a}, b)$ is: if $b-\langle \mathbf{a}, \mathbf{s}\rangle/q$ is closer to 0 than to ½, then it is 0, otherwise it is 1.
- The proof of correctness follows from the choice of parameters and some probability analysis. The proof of security is by reduction to the decision version of LWE: an algorithm for distinguishing between encryptions (with above parameters) of 0 and 1 can be used to distinguish between $A_{s,\chi}$ and the uniform distribution over $\mathbb{Z}^n{}_q \times \mathbb{T}$.

Lyubashevsky developed the first ring learning with errors-based signature.[13] He demonstrated how the framework that is used for creating efficient number-theoretic identification and signature schemes can be transferred into the setting of lattices.

Hash-Based Cryptography

A one-time key is when the encryption scheme is set up such that only one message may be encrypted with a key. Whilst a Merkle tree is a tree in which every 'leaf' node is labelled with the cryptographic hash of a data block, and every node that is not a leaf (called a branch) is labelled with the cryptographic hash of the labels of its child nodes.[14, 15]. A hash tree allows efficient and secure verification of the contents of a large data structure.

A Lamport signature or Lamport one-time signature scheme is a method for constructing a digital signature.[16] Lamport signatures can be built from any cryptographically secure one-way function; usually a cryptographic hash function is used. Whilst a Merkle signature scheme is a digital signature scheme based on Merkle trees and one-time signatures such as the Lamport signature scheme. It was developed by Ralph Merkle in the late 1970s.[17]

The central idea behind hash-based signature schemes is to combine a number of one-time key pairs into a single structure to obtain a practical way of signing more than once (but a limited number of times). This is done using a Merkle tree, with possible variations. One public and one private key are constructed from the numerous public and private keys of the underlying one-time scheme. The global public key is the node at the top of the Merkle tree. Its value is an output of the selected hash function, so a typical public key size is 32 bytes. The validity of this global public key is related to the validity of a given one-time public key using a sequence of tree nodes. This sequence is called the authentication path. It is stored as part of the signature, and allows a verifier to reconstruct the node path between those two public keys.

In 2011 Buchmann, Dahmen, and Hülsing presented the hash-based signature scheme XMSS.[18] It was the first provably (forward) secure and practical signature scheme with minimal security requirements: a pseudorandom and a second preimage resistant (hash) function family. Its signature size was reduced to less than 25% compared to the best provably secure hash-based signature scheme.

NIST's Post-Quantum Cryptography Selected Algorithms

The National Institute of Standards and Technology (NIST) has been at the forefront of the transition to post-quantum cryptography. In 2022, they selected four algorithms designed to withstand attack by quantum computers.[19] One, CRYSTALS-Kyber, for general encryption purposes such as creating secure websites, and three for digital signatures. Three of them, employ lattice problems, and one, SPHINCS+, employs a hash-based signature scheme. See Table 1.

Table 1. NIST's post-quantum cryptography selected algorithms

Type	Name	Algorithm type	Algorithm
Public-key encryption and key-establishment algorithms	CRYSTALS-Kyber	Lattice-based	Learning with errors
Digital signature algorithms	CRYSTALS-Dilithium	Lattice-based	Short integer solution problem
	Falcon	Lattice-based	Short integer solution problem
	SPHINCS+	Hash-based	Hash-based signature scheme

CRYSTALS-Kyber[20] uses the learning-with-errors (LWE) problem over module lattices. Whilst the design of Dilithium[21] is based on Lyubashevsky's 'Fiat–Shamir with aborts' technique,[22] which employs sample rejection to make lattice-based Fiat–Shamir schemes compact and secure. Falcon's[23] security scheme is based on Gentry, Peikert and Vaikuntanathan (GPV),[24] NTRU lattices,[25] fast Fourier sampling, short integer problems and Gaussian sampling floating-point arithmetic. SPHINCS+ is a stateless hash-based signature scheme.[26] It uses many components from XMSS but works with larger keys and signatures to eliminate the state.

The basic idea is to authenticate a huge number of few-time signature (FTS) key pairs using a so-called hypertree. FTS schemes are signature schemes that allow a key pair to produce a small number of signatures, e.g. in the order of ten for their parameter sets. For each new message, a (pseudo)random FTS key pair is chosen to sign the message. The signature then consists of the FTS signature and the authentication information for that FTS key pair. The authentication information is roughly a hypertree signature, i.e. a signature using a certification tree of Merkle tree signatures. Vidaković and Miličević provided a comparative evaluation of the three digital signature algorithms selected by NIST.[27] Their findings indicated that CRYSTALS-Dilithium offers advantages in low-power scenarios, Falcon excels in signature verification speed and SPHINCS+ provides robust security at the cost of computational efficiency.

Attacks

Mujdei et al. used correlation power analysis-based side-channel analysis methodologies to target every polynomial multiplication strategy for all lattice-based post-quantum key encapsulation mechanisms in the final round of the NIST post-quantum standardization procedure.[28] They were able to extract the secret key from all lattice-based post-quantum key encapsulation mechanisms.

The proposed attacks can be mitigated by using masking and hiding countermeasures. Chen gave a polynomial time quantum algorithm for solving the learning with errors problem (LWE) with certain polynomial modulus-noise ratios.[29] Combining with the reductions from lattice problems to LWE shown by Regev, he obtained polynomial time quantum algorithms for solving the decisional shortest vector problem (GapSVP) and the shortest independent vector problem (SIVP) for all n-dimensional lattices within approximation factors of $\Omega(n^{4.5})$. Bambury and Nguyen showed how blockwise reduction can exploit lattices with special geometric properties, effectively reducing the required blocksize to solve the shortest vector problem to half of the lattice's rank, and in the case of the hypercubic lattice $\mathbb{Z}^n$, further relaxing the approximation factor of blocks to $\sqrt{2}$.[30]

V. CONCLUSION

When ideal (i.e. large-scale and fault-tolerant) quantum computers are developed, public-key algorithms will cease to be secure. It is expected that a quantum computer able to factorise a 2048-bit number in less than 24 hours will be available in around 2036.

The results of the NIST competition suggest that lattice-base algorithms are the most successful algorithms for post-quantum cryptography. However, there have been recent attacks on lattice-based algorithms, so it remains to be seen which post-quantum algorithms will be regarded as safe in the longer term. Cryptographers should continue to publish successful attacks on existing post-quantum algorithms, and research new post-quantum algorithms.

ACKNOWLEDGEMENTS

Dr Martin Sewell, BSc (Hons), MSc, PhD

VI. REFERENCES

[1] John von Neumann, "First Draft of a Report on the EDVAC", University of Pennsylvania, Jun. 1945.

[2] David Deutsch, "Quantum Theory, the Church–Turing Principle and the Universal Quantum Computer", Proceedings of the Royal Society of London A, Vol. 400, Issue 1818, Jul. 1985, pp. 97-117.

[3] Lov K. Grover, "A Fast Quantum Mechanical Algorithm for Database Search", STOC '96: Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Gary L. Miller (Ed.), ACM, New York, Jul. 1996, pp. 212-219.

[4] Peter W. Shor, "Algorithms for Quantum Computation: Discrete Logarithms and Factoring", Proceedings 35th Annual Symposium on Foundations of Computer Science, Shafi Goldwasser (Ed.), pp. 124-134, IEEE, Los Alamitos, CA, Nov. 1994.

[5] R. L. Rivest, A. Shamir, L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM, Vol. 21, Issue 2, Feb. 1978, pp. 120-126.

[6] Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, Daniel Smith-Tone, "Report on Post-Quantum Cryptography", NIST, Gaithersburg, NISTIR 8105, 2016.

[7] Daniel J. Bernstein and Tanja Lange, "Post-Quantum Cryptography", Nature, Vol. 549, Issue 7671, Sep. 2017, 188-194.

[8] Michele Mosca and Marco Piani, "2021 Quantum Threat Timeline Report", Global Risk Institute, Toronto, Jan. 2022.

[9] NIST, "Post-Quantum Cryptography", Gaithersburg, Feb. 2024. https://csrc.nist.gov/projects/post-quantum-cryptography

[10] M. Ajtai, "Generating Hard Instances of Lattice Problems", In Gary L. Miller (Ed.), STOC '96: Proceedings of the twenty-eighth annual ACM symposium on theory of computing, ACM, New York, Jul. 1996, pp. 99-108.

[11] P. van Emde Boas, "Another NP-Complete Problem and the Complexity of Computing Short Vectors in a Lattice", Report 81-04, University of Amsterdam, 1981.

[12] Oded Regev, "On Lattices, Learning With Errors, Random Linear Codes, and Cryptography", Journal of the ACM, Vol. 56, Issue 6, Sep. 2009, 1-40.

[13] Vadim Lyubashevsky, "Fiat-Shamir With Aborts: Applications to Lattice and Factoring-Based Signatures", In Mitsuru Matsui (Ed.), Advances in cryptology – ASIACRYPT 2009, Vol. 5912, Springer, Berlin, Nov. 2009, pp. 598-616.

[14] Ralph C. Merkle, "Method of Providing Digital Signatures", United States Patent Nos. 4,309,569, Jan. 1982.

[15] Ralph C. Merkle, "A Digital Signature Based on a Conventional Encryption Function", In Carl Pomerance (Ed.), Advances in cryptology – CRYPTO '87, Vol. 293, Springer, Berlin, Feb. 1988, pp. 369-378.

[16] Leslie Lamport, "Constructing Digital Signatures From a One Way Function", Tech. Rep. No. CSL – 98, SRI International, Menlo Park, CA, Oct. 1979.

[17] Ralph Charles Merkle, "Secrecy, Authentication, and Public Key Systems", Technical Report No. 1979-1, Stanford University, Stanford, CA, PhD Thesis, Jun. 1979.

[18] Johannes Buchmann, Erik Dahmen, Andreas Hülsing, "XMSS - A Practical Forward Secure Signature Scheme Based on Minimal Security Assumptions", In Bo-Yin Yang (Ed.), Post-quantum cryptography, Vol. 7071, Springer, Berlin, Nov. 2011, pp. 117-129.

[19] NIST, "Post-quantum cryptography: Selected Algorithms 2022", Gaithersburg, Feb. 2024. https://csrc.nist.gov/projects/post-quantum-cryptography/selected-algorithms-2022

[20] Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, Damien Stehlé, "CRYSTALS-Kyber: Algorithm Specifications and Supporting Documentation", Aug. 2021.

[21] Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, Damien Stehlé, "CRYSTALS-Dilithium Algorithm Specifications and Supporting Documentation", Feb. 2021.

[22] Vadim Lyubashevsky, "Lattice Signatures Without Trapdoors", In David Pointcheval and Thomas Johansson (Eds.) Advances in Cryptology – EUROCRYPT 2012, LNCS, Vol. 7237, Springer, Berlin, Apr. 2012.

[23] Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, Zhenfei Zhang, "FALCON: Fast-Fourier Lattice-Based Compact Signatures Over NTRU", Oct. 2020.

[24] Craig Gentry, Chris Peikert, Vinod Vaikuntanathan, "Trapdoors for Hard Lattices and New Cryptographic Constructions", STOC '08: Proceedings of the fortieth annual ACM symposium on theory of computing, May 2008, 197-206.

[25] Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman, "NTRU: A Ring-Based Public Key Cryptosystem", In J. P. Buhler (Ed.) Algorithmic Number Theory, LNCS, Vol. 1423, Springer, Berlin, Jun. 1998.

[26] Jean-Philippe Aumasson, Daniel J. Bernstein, Ward Beullens, Christoph Dobraunig, Maria Eichlseder, Scott Fluhrer, Stefan-Lukas Gazdag, Andreas Hülsing, Panos Kampanakis, Stefan Kölbl, Tanja Lange, Martin M. Lauridsen, Florian Mendel, Ruben Niederhagen, Christian Rechberger, Joost Rijneveld, Peter Schwabe, Bas Westerbaan, "SPHINCS+", Jun. 2022.

[27] Marin Vidaković and Kruno Miličević, "Performance and Applicability of Post-Quantum Digital Signature Algorithms in Resource-Constrained Environments", Algorithms, Vol. 16, Issue 11, Nov. 2023, 518.

[28] Catinca Mujdei, Lennert Wouters, Angshuman Karmakar, Arthur Beckers, Jose Maria Bermudo Mera, Ingrid Verbauwhede, "Side-Channel Analysis of Lattice-Based Post-Quantum Cryptography: Exploiting Polynomial Multiplication", ACM Transactions on Embedded Computing Systems, Vol. 23, Issue 2, Mar. 2024, 1-23.

[29] Yilei Chen, "Quantum Algorithms for Lattice Problems", Cryptology ePrint Archive, Paper 2024/555, Apr. 2024.

[30] Henry Bambury and Phong Q. Nguyen, "Improved Provable Reduction of NTRU and Hypercubic Lattices", Cryptology ePrint Archive, Paper 2024/601, Apr. 2024.