Survey of Quantum Computing

Sattyik Kundu
December 2017
Information Security and Assurance Department George Mason University, Fairfax, Virginia
skundu1@masonlive.gmu.edu

Abstract: Quantum computers are computers that operate under the principles of quantum mechanics whereas modern computers operate under the principles of classical physics. The concept and principles underlying quantum computers have been around for decades. Due to quantum computers being able to exploit quantum mechanics, they are able to perform superposition of values (states). This enables quantum computers to perform multiple calculations simultaneously. Due to this property, quantum computers can perform tasks much faster than believed possible with classical computers; such task are prime factorization of extremely large numbers (Shor algorithm) and faster searching of big database (Grover algorithm). This survey discusses the principles of quantum physics, quantum gates which are the basic building blocks of quantum circuits, some key quantum algorithms such Shor algorithm and Grover algorithm in detail, physical realization of qubits and quantum computers available today, and finally post-quantum cryptography due to security implications of quantum algorithms. Although D-Wave Systems builds quantum computers, but they are still laboratory tools for exploring potential applications. Quantum computers are difficult to realize on a large scale as of today. A big impact is seen today in post quantum cryptography.

Index Terms: Quantum, superposition, entanglement, qubit, Shor algorithm, Grover's algorithm, D-Wave computers, post-quantum cryptography

I. Introduction

Understanding the underlying princples is imperative to understanding quantum computing and all later section topics related to quantum computing. The underlying principles of quantum computing concerns how data on quantum compters are represented differently from data on classical computers.

A. Short History

- 1982 - Feynman proposed the idea of creating machines based on the laws of quantum mechanics instead of the laws of classical physics [1].
- 1985 - David Deutsch developed the quantum Turing machine, proving that quantum circuits are universal [1].
- 1994 - Peter Shor came up with a quantum algorithm to factor very large numbers in polynomial time [1].
- 1997 - Lov Grover develops a quantum search algorithm with $O(\sqrt{N})$ complexity [1].
- 2007 (Feb.) – D-Wave demonstrated a 16-qubit quantum annealing processor [2].
- 2017 (Jan.) – D-Wave D-Wave Announces D-Wave 2000Q Quantum Computer which has 2000 qubits. The first customer for the new system is Temporal Defense Systems Inc. (TDS), a cutting-edge cyber security firm [3].

B. Principles

A bit of data can be represented by an atom that is in one of two states denoted by |0> and |1>. In this form, it is called a qubit. The energy levels of an atom can be used for the physical implementation of the qubit. The ground state can represent |0> and the excited state can represent |1> [1].

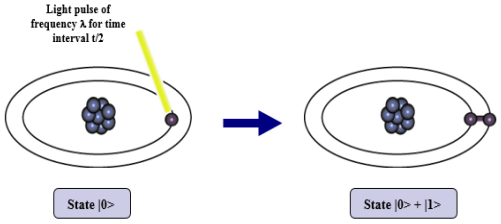


Figure 1: Atom representing a bit and superposition [1]

However, the qubit can be in a superposition of states instead of being forced into state |0> or |1> as denoted by an addition of state vectors as shown:

$$|\psi\rangle = \alpha_1|0\rangle + \alpha_2|1\rangle$$

Figure 2: Superposition of States Vectors [1]

When complex numbers α_1 and α_2 are converted to $|\alpha_1|^2$ and $|\alpha_2|^2$, they represent the probabilities that the above superposition will collapse to either |0> or |1> respectively when observed [1]. Hence, $|\alpha_1|^2 + |\alpha_2|^2 = \mathbf{1}$.

The above superposition of two states is only for one qubit. As **N** qubits can represent (2^N-1) states simultaneously in superposition, an example 3-qubit register would represent an equally weighted superposition as follows:

$$|\psi> = \frac{1}{\sqrt{8}}|000> + \frac{1}{\sqrt{8}}|001> + \ldots + \frac{1}{\sqrt{8}}|111>$$

Figure 3: 3-Qubits Superposition

Two major caveats of quantum superposition are the issues of **decoherence and entanglement**. *As long as the states remain in superposition and the system is isolated, the quantum system is in coherence*. When a system is no longer isolated and the superposition is observed, it will collapse to one of (2^N-1) states of **N** qubits; the system goes into decoherence. The reason for this is because when observing, the tiniest subtleties of heat, pressure, etc. can contact the superposition and cause it to collapse [1, 4].

II. Quantum Gates

Because quantum computers operate on a different set of physics than that of classical computers; quantum logic gates have to be made and operated differently than classical logic gates like NAND, XOR, and etc. With classical logic gates, input information is destroyed and only the output information is preserved. These gates are also called non-reversible gates because the input values are not preserved in order to reverse these gates' operations.

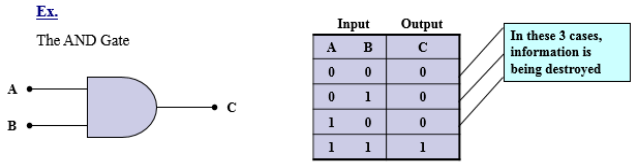


Figure 3: Destroyed Information in AND gate [1]

For example, in the above AND gate truth table the information for three of the outputs is destroyed because there is no way to know what the inputs were. For the output of 0, there are three possible input combinations. Only when the output is 1 are the exact input known; the information wasn't destroyed in this case [1].

Whenever information is destroyed within a classic logic gate, a little amount of heat is released each time. A quantum superposition cannot run on such gates because that heat would easily destroy of the superposition [1].

Hence, quantum gates must be reversible. To do so, the number of outputs needs to match the number of inputs; which will enable discovering any inputs for any output. Because no information is lost, both the inputs and outputs of a quantum gate will always be thermodynamically equivalent. This means that there is no change in heat and thus the superposition won't be destroyed [1, 5]. Here are the most relevant quantum gates as well as their function and purpose.

A. Hadamard Gate

The main purpose of the Hadamard gate is to transform an initial collapsed (individual) state into superposition form. Because quantum algorithm performs computations with super-positions, the Hadamard gate is used to turn initial collapsed value into a superposition that any quantum algorithm can use [1].

$$\hat{H}|0\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$
$$\hat{H}|1\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix} = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

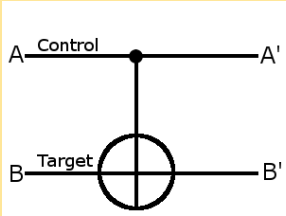
Figure 4: 1-Qubit Hadamard Gate Outputs[6]

For quantum gates, their operations can be mathematically represented via matrices. In the above diagram, the collapsed states |0> and |1> are seen mathematically transformed into superposition form. Above, the matrices [1, 0] and [0, 1] represent state |0> and |1>, respectively. The Hadamard matrix for 1 qubit is represented by $\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$.

There are additional variations of Hadamard matrices for larger number of qubits.

B. Controlled-NOT (CN) Gate

The purpose the quantum CN gate is that it can entangle and disentangle quantum states. This is an essential function given that entanglement is an inherent property of quantum physics and having control over this is essential depending on the needs of the system [7]. This quantum gate is made up of two qubits which are the control and target qubits. The CNOT gate [1] flips the target qubit's value only if the qubit of the control value is |1>.



Input		Output	
A	B	A'	B'
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Figure 5: Controlled-NOT Gate and Truth Table

From the above truth table, the relationship between inputs A and B and output A' are equivalent to the respective inputs and output of a XOR gate [1].

Lastly, the transformation of the linear supposition going through a CNOT gate can be represented by the following matrix function:

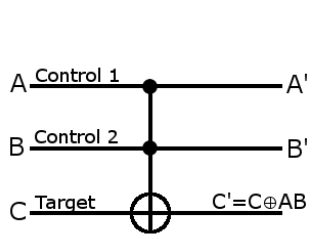
$$CNOT = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix} \bullet \begin{bmatrix}v_{00}\\ v_{01}\\ v_{10}\\ v_{11}\end{bmatrix} = \begin{bmatrix}v_{00}\\ v_{01}\\ v_{11}\\ v_{01}\end{bmatrix}$$

Figure 6: CNOT Transformation Represented via Matrices

In the above matrix transformation of the superposition, it is shown that V_{10} and V_{11} switch values in the output while V_{00} and V_{01} states values remain the same. This relates to the Figure 5 truth table where the last two values in column B input (V_{10}=0 and V_{11}=1) swap places in the output column B' where V_{10}=1 and V_{11}=0.

C. Controlled-Controlled-NOT (CCN) Gate

Last, there is the CCN gate which is also called the Toffoli-gate. The major importance of this gate is that it is a universal logic gate. This means that any reversible quantum circuit can be made only from CCN gates. This is equivalent to how NAND gates are used to create any circuit for classical computing [8].



Input			Output		
A	B	C	A'	B'	C'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	1	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Figure 7: Controlled-Controlled-NOT Gate and Truth Table

This gate has three inputs. If the first two inputs are set to one, only then the output of the third input is reversed. Otherwise, the outputs match the corresponding inputs.

Last, to represent the effect of a CCN on a superposition input, here is the matrix:

$$CCNOT = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \bullet \begin{bmatrix} v_{000} \\ v_{001} \\ v_{010} \\ v_{011} \\ v_{100} \\ v_{101} \\ v_{110} \\ v_{111} \end{bmatrix} = \begin{bmatrix} v_{000} \\ v_{001} \\ v_{010} \\ v_{011} \\ v_{100} \\ v_{101} \\ v_{111} \\ v_{110} \end{bmatrix}$$

Figure 8: CCNOT Transformation via Matrices

In the above matrix transformation of the superposition, the state values of V_{110} and V_{111} swap in the output column whereas all the other states remain the same. In relation to the Figure 7 truth table, only in the last two rows of output column C' have their values swapped just like V_{110} and V_{111} in the above matrix transformation.

III. Quantum Algorithms

Quantum algorithms are a finite set of steps for solving a problem using quantum computations. Quantum algorithms are created with the intention of taking advantage of various properties of quantum computing like **superposition or entanglement** to perform computations that would otherwise be impossible or very time consuming for classical computing. The two major quantum algorithms that are of particular interest are Shor's algorithm and Grover's algorithm. There are several other quantum algorithms are available as well.

A. Shor's Algorithm

This algorithm solves the integer factorization and discrete logarithm problems in polynomial time. The true significance of this algorithm being run on a quantum computer is that it decrypts encryption keys on a relatively short time. Major encryption schemes like RSA, DSA, and Diffie-Hellman Key Exchange all rely on encryption keys which are a product of 2 high-value primes [9]. On a classical computer, figuring out the prime factors would take much longer than the key usage lifespan. On a quantum computer running Shor's algorithm, these secret prime number factors can be found in a relatively short amount of time and thus defeat several modern encryption schemes.

The general algorithm is explained as follows [1]:
Find factors **p**, **q** of **N**

1. If **N** even, prime, or a prime power, EXIT.
2. Pick random integer **x** < **N**;
   - If GCD(**x**, **N**) ≠1, factors are found.
3. Use Quantum Algorithm: Find period **P** of **F**(**a**) = $\mathbf{x}^{a}$ mod **N** where **a**=0,1,2,….,N-1.
4. If **P** is Odd, go back to step 2. If **P** is Even:
   - **p** = GCD($\mathbf{x}^{\mathbf{P}/2}$ + 1, **N**)
   - **q** = GCD($\mathbf{x}^{\mathbf{P}/2}$ - 1, **N**)
5. Regarding **p** and **q**:
   - If **N** = **p***q, → solutions found!
   - If **N** ≠ **p***q,→ redo Shor's algorithm with a new random **x** value until the solutions for **p** and **q** are found.

Most of Shor's Algorithm's steps can be done on a classical computer. However, in step 3, if **N** is very large, it can take a long time to compute each $\mathbf{x}^{a}$ mod **N** function for **a**=0,1,2,…., **N**-1 on a classical computer. Using a quantum computer would greatly reduce the time taken to calculate all the $\mathbf{x}^{a}$ mod **N** functions for **a**=0,1,2,…., **N**-1.

In the **Appendix-A**, a fully worked example of Shor's algorithm has been provided together with an in-depth explanation of how the quantum registers work in step 3.

B. Grover's Algorithm

This algorithm searches for an element from an unstructured database. For today's classical computers, the number of elements evaluated from an N-item database before reaching the solution is on average N/2 with the maximum number of possible searches being N. With Grover's algorithm, the number of average searches is $\sqrt{N}$ before reaching the solution. This amounts to a significant reduction in time for an element search in comparison to search algorithms for classical computers [10].

The general algorithm can be symbolized as follows [11]:
N→H→[OHZH] *for multiple iterations* → Solution

The above algorithm is explained in detail:

1. First is the setup of the base state 0> of the inputs. **N** represents the number of possible states. If number of qubits is **Q**, **N**=$2^{\mathbf{Q}}$ states .

2. Second is to perform the Hadamard transform **H**. The first **H** in the algorithm is used to set the **Q** qubit register into superposition. Its matrix has dimensions 2^Q x 2^Q where **Q**=total qubits. Hence, it is represented as shown in **Appendix-C**.
3. The **[OHZH]** iteration starts. **O** is an oracle gate. This black-box gate will only yield a value of f(x)=1 when the correct state(the one being searched for) is inserted as x. Otherwise, f(x)=0. Within the algorithm's implementation, the correct state within the superposition is simply has its sign flipped.
4. **H** is performed on the output of the oracle operator.
5. **Z** is the zero-state phase shift gate. This gate flips the sign of all states except the |0> (zero) state. The matrix form is in dimension size is N x N as follows:

$$Z = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & & \vdots \\ \vdots & \vdots & & -1 & 0 \\ 0 & 0 & \cdots & 0 & -1 \end{bmatrix}$$

Figure 9: Zero-state Phase Shift Gate Transform

6. **H** is performed on the output of **Z**. This ends the first iterations of **[OHZH]** operations.
7. Iterate though the **[OHZH]** operations until the quantum register collapse to the correct value.

After computing **H** for the first time, the sequence of operations [**OHZH**] is computed for multiple iterations until the solution is finally achieved. How this is achieved is that after each time the operations of sequence of [**OHZH**] are executed, the probability of the correct state (solution) of the evolving superposition is amplified. Eventually, the probability of the correct answer is increased to the point that the final superposition will easily collapse to the correct answer. A fully worked example of Grover's algorithm has been provided in the **Appendix-B.**

C. Other Quantum Algorithm

C.1: *The Deutsch Algorithm*

This algorithm answers the following question. Suppose we have a function $f:\{0,1\}\rightarrow\{0,1\}$, which can be either *constant* or *balanced*. In this case the function is constant if $f(0) = f(1)$ and it is balanced if $f(0) \neq f(1)$. Classically it would take two evaluations of the function to tell whether it is one or the other. In quantum domain, we can answer this question in a single evaluation only. The reason for this is that we can pack 0 and 1 into *x* at the same time, of course. This fact was the first breakthrough in quantum computing thanks to David Deutsch [12].

C.2: *The Deutsch-Jozsa Algorithm*

This algorithm generalizes the Deutsch algorithm to a function $f:\{0,1\}^n\rightarrow\{0,1\}$. We ask the same question: is the function is constant or balanced. Here *balanced* means that the function is 0 on half of its arguments and 1 on the other half. Of course in this case the function may be *neither* constant nor balanced. In this case the oracle does not work: it may say *yes* or *no* and the answer will be meaningless. Although deeper than Deutsch algorithm, this extension by Jozsa was still limited to particular uses [12].

C.3: *The Simon Algorithm*

Suppose there is a function $f:\{0,1\}^n\rightarrow\{0,1\}^n$. The function is supposed to be 2-to-1, i.e., for every value of *f(.)*, there are always two n-arguments such $\mathbf{x}_1$ and $\mathbf{x}_2$ that $f(\mathbf{x}_1) = f(\mathbf{x}_2)$. The function is also supposed to be such that there exists a binary vector **a** so that $f(\mathbf{x}\oplus\mathbf{a}) = f(\mathbf{x})$, where $\oplus$ is a bitwise EXOR on words, binary strings, $\mathbf{x}_1$ and $\mathbf{x}_2$. The algorithm returns the vector **a** in *O(n)* measurements. This algorithm is historically very important as it started Shor to think about using periodicity which led ultimately to the discovery of the Shor algorithm [13].

Input x	Output f(x)
000	4
001	2
010	3
011	1
100	2
101	4
110	1
111	3

In the Table above *f(x)* is 4 when **x1**=000 and **x2**=101.If **a=101**, $f(x\oplus a) = f(x)$.The steps of the algorithm are described next.

- The algorithm uses two registers, both with n qubits. The registers are initialized to base state |0⋯0>|0⋯0>. Two registers are entangled.
- Perform Hadamard transform on first register, producing uniform superposition.
- Compute ***f(x)*** and store result in second register.
- Measure second register and get result *f(z)*. Because of entanglement, input register will now collapse to only those values that led to *f(z).* For example, in above example *f(z)*=4, and two states of input register are now 000> and 101> with coefficient $1/\sqrt{2}$.
- Again, apply Hadamard transform to input register and observe input register. It can be shown mathematically that observation ***y*** is such that **y.a** = 0 (mod 2) where **y.a** gives inner product.
- If vector **a** has ***n*** components, previous step is executed (***n-1***) times to get **(*n-1*)** constraint equations.

Solution of constraint equations gives vector **a**

IV. Physical Realization of Quantum Computers

Quantum computers are difficult to realize on a large scale due to decoherence, i.e., unwanted interaction between system and environment, which introduces errors. It is also difficult to maintain lifetime of information. Observing quantum particles changes outcome that is difficult to verify. Physical realization of quantum computers is complex. Some ways to physically realize a Qubit are described first followed by a description of D-wave computers, only available and claimed Quantum computer today.

A. Physical Realization of Qubits

Altogether, to build a viable quantum computer, we need:

1. a physical qubit that is well isolated from the environment and is capable of being addressed and coupled to more than one extra qubit in a controllable manner,
2. a fault-tolerant architecture supporting reliable logical qubits, and
3. universal gates, initialization, and measurement of logical qubits

A physical quantum computer satisfying all three of these requirements is still an outstanding challenge. Here, we describe some of the ways to realize qubit physically, and mention some other ways that are being researched [14].

A.1: *Energy Levels of Hydrogen Atom*

Consider the electron in a hydrogen atom. It can be in its ground state (i.e. an *s* orbital) or in an excited state. If this were a classical system, we could store a bit of information in the state of the electron: ground= 0, excited = 1. So we can also store a qubit of information in the quantum state of the electron, i.e., in **the superposition**. Note that the electron actually has an infinite number of energy levels (indexed by quantum number *n)*, but as long as we can isolate two of them, we can use these two as a **qubi**t.

A.2: *Spin Qubits*

Electrons are natural qubits: they have a spin degree of freedom, which is automatically a quantum two-level system. They're small, so in principle they should have good coherence. The trouble is getting them to sit still and controlling them. To do this, electrons are embedded in solid crystals, often silicon. This has the downside that the electrons are therefore really close to a bunch of other atoms, so coherence becomes an issue

A.3: *Photon Polarization*

There is a qubit associated with photon - its polarization. If a photon is moving along the z-axis, it has an associated electric field in the x-y plane. The frequency of the field is determined by the frequency of the photon. However, this still leaves the x-y components of the electric field unspecified. The 2-dimensional quantity specifying this field is the polarization of the photon. Circularly polarized electromagnetic waves are composed of photons with only one type of spin, either right- or left-hand. Linearly polarized waves consist of equal numbers of right and left hand spinning photons, with their phase synchronized so they superpose to give oscillation in a plane. Thus, polarization of electromagnetic waves can represent qubits.

A.4: *Individual Ionized Atoms or "Ion Traps"*

This system uses the strong electromagnetic fields of an optical laser to trap ions in space. The traps keep the ions from interacting with the environment; the ions have orbiting electrons, and these electrons have various levels. We choose a pair of electron levels to act as the qubit states $|0\rangle|0\rangle$ and $|1\rangle|1\rangle$. The pair of states is selected to have a high coherence, i.e. a low rate of spontaneous emission. This is possible with atoms because they're so tiny: the electron transitions can have very small dipole moment and so they don't couple strongly to the electromagnetic field. Logic gates are done with laser or RF pulses which cause the ion's electron state to change. Two-qubit gates often use the mechanical vibrational modes of the atoms in their traps. Ion traps have been constructed using a variety of different atomic species [14].

A.5: *Superconducting Qubits*

Superconducting qubits are electronic circuits comprising lithographically defined Josephson tunnel junctions, inductors, capacitors, and interconnects. When cooled to dilution refrigerator temperatures (~20 mK, or 0.02 K), these circuits behave as quantum mechanical "artificial atoms," exhibiting quantized states of electronic charge, magnetic flux, or junction phase, depending on the design parameters of the constituent circuit elements. Their potential for lithographic scalability, compatibility with microwave control, and operability at nanosecond time scales place superconducting qubits among the leading modalities being considered for quantum information science and technology applications [14].

Nanotechnology has emerged as the most appropriate tool to realize quantum computers. Many other modalities of representing qubits physically are being investigated. Some of these modalities are [14]: Nuclear Magnetic Resonance based and Nano-mechanical Resonator based.

B. D-Wave Computers

Only close to any quantum computer today is D-Wave's current model with 2000 qubits (January 2017) while it was 512-qubit machine in 2012 [3]. It is difficult to verify if performing quantum operations or not! However, it has shown significant speed-ups but only for certain calculations. It also

has helped to advance the research in Quantum Computing. NSA is funding both quantum computing and quantum technologies for secure communication research.

D-Wave also announced the first customer for the new system, Temporal Defense Systems Inc. (TDS), a cutting-edge cyber security firm. With 2000 qubits and new control features, the new system can solve larger problems than was previously possible, with faster performance, providing a big step toward production applications in optimization, cybersecurity, machine learning, and sampling.

D-Wave's leap from 1000 qubits to 2000 qubits is a major technical achievement and an important advance for the emerging field of quantum computing [3]. D-Wave is the only company with a product designed to run quantum computing problems, and the new D-Wave 2000Q system should be even more interesting to researchers and application developers who want to explore this revolutionary new approach to computing according to the company.

V. Post-Quantum Cryptography

Public-key cryptographic algorithms based on integer factorization or discrete log problem such as RSA, DSA, Diffie-Hellman Key Exchange, ECC, ECDSA are vulnerable to Shor's and Grover's algorithm. **Thus, there is critical need of cryptographic systems that cannot be broken by quantum computers.** Main types of post quantum cryptography are:

- Code-based
- Hash-based
- Multivariate-quadratic
- Lattice-based

V.1:. *Code-Based*

Code-based cryptography is one of the few mathematical techniques that enables the construction of public-key cryptosystems that are secure against an adversary equipped with a quantum computer [15]. The McEliece public-key encryption scheme and its variants are candidates for a postquantum public-key encryption standard. McEliece's original idea was to use as ciphertext a word of a carefully chosen linear error-correcting code—a binary Goppa code, in this case—to which random errors were added [16]. An arbitrary basis of the code—a generator matrix as used in error correcting code in digital communications—is the public key, allowing anyone to encrypt. Legitimate users who know a secret trapdoor—a fast decoding algorithm for the code—can remove the errors and recover the cleartext. Adversaries are reduced to a generic decoding problem, which is believed to be hard on average, including against quantum adversaries. Because of relatively large public key sizes (65/192kBytes for 80/128-bit security), it is, as of now, rarely used in practice. However, it is the most mature PQ Cryptographic scheme.

V.2: *Hash-Based*

Security of Hash-based [17, 18] signature schemes relies on collision resistance of cryptographic hash function. Hash-based signature schemes combine a one-time signature scheme with a Merkle tree structure. Since a one-time signature scheme key can only sign a single message securely, it is practical to combine many such keys within a single, larger structure. The central idea of hash-based signature schemes is to combine a larger number of one-time key pairs into a single structure to obtain a practical way of signing more than once (yet a limited number of times). This is done using a Merkle tree structure, with possible variations. In this hierarchical data structure, a hash function and concatenation are used repeatedly to compute tree nodes. Hash-based cryptography is a type of post-quantum cryptography for its security against attacks aided by quantum computers. Additionally, hash-based signatures need no computationally expensive mathematical operations like big integer arithmetic. The only requirement is a secure cryptographic hash function. Hash-based cryptography needs relatively small public/private key sizes (e.g., 46 Bytes – 7568 Bytes). Its main advantages is it is considered the most promising PQ signature schemes at this time. Its main disadvantages is limited use of each public key.

V.3: *Multivariate-Quadratic*

Multivariate-quadratic (MQ) [19] is based on difficulty in solving a set of nonlinear MQ equations. A multivariate public key cryptosystem (MPKCs for short) have a set of (usually) quadratic polynomials over a finite field as its public map. Its main security assumption is backed by the NP-hardness of the problem to solve nonlinear equations over a finite field. This family is considered as one of the major families of PKCs that could resist potentially even the powerful quantum computers of the future. There has been fast and intensive development in Multivariate Public Key Cryptography in the last two decades. Other than Matsumoto-Imai constructions, there are 3 other known constructions: Oil and Vinegar [17, 18], Rainbow, and Quartz/HFE [18]. Some constructions are not as secure as was claimed initially, but others are still viable. It has large public and private key sizes (up to 75kBytes). It is not suitable for embedded devices due to large key sizes.

V.4: *Lattice-Based Cryptography*

Lattice-based cryptography is the generic term for constructions of cryptographic primitives that involve lattices, either in the construction itself or in the security proof. Lattice-based constructions are currently important candidates for post-quantum cryptography as some lattice-based constructions appear to be resistant to attack by both classical and quantum computers. Furthermore, many lattice-based constructions are known to be secure under the assumption that certain well-studied computational lattice problems cannot be solved efficiently.

A lattice is the set of all integer linear combinations of k basis vectors b1, b2, .., bk. Crucially, the basis for a lattice is not unique. The most important lattice-based computational problem is the Shortest Vector Problem (SVP), which asks us to approximate the minimal Euclidean length of a non-zero lattice vector [20]. This problem is thought to be hard to solve efficiently even with a quantum computer. Many (though not all) lattice-based cryptographic constructions are known to be

secure if SVP is in fact hard in this regime [21, 22]. The main disadvantage of is large public private key sizes (up to 732kBytes); the main advantages are that underlying operations can be implemented efficiently and it is the most promising post-qunatum crypto, attracting most interest in research community.

VI. Conclusions

We make the following conclusions based on our study:

1. Quantum-computing research is now decades old, but it still a young field, one should remember that decades of development separated original concept of general-purpose digital computers from what is available today.
2. Quantum computing is based on manipulating qubits, quantum-mechanical superposition of two states, such as two orthogonal polarizations of a photon.
3. Potential power of such quantum processing increases with number of qubits that can maintain coherence with each other; **Q** qubits can represent $\mathbf{N}=2^{\mathbf{Q}}$ quantum states.
4. Because qubits are quantum-mechanical objects, their values remain uncertain until superposition of states collapses during final readout stage.
5. Most research focuses on essential building blocks such as qubits, the quantum counterparts of the hardware bits in digital computers, which perform quantum operations.
6. Other groups are developing paradigms and algorithms for quantum computing, testing concepts for quantum error correction and quantum sensors.
7. D-Wave builds quantum computers, but they are still laboratory tools for exploring potential applications.
8. Quantum computers are difficult to realize on a large scale as of today. A big impact is seen today in post quantum cryptography.

VII. References

[1] J. Stelmach, 'Quantum Computing', University of Delaware, 2003 [Online]. Available: https://www.eecis.udel.edu/~saunders/courses/879-03s/quantumComputers.ppt

[2] A. Vance, 'D-Wave qubits in the era of Quantum Computing', *The Register*, 2007 [Online]. Available: https://www.theregister.co.uk/2007/02/13/dwave_quantum/

[3] 'D-Wave Annnounces D-Wave 2000Q Quantum Computer and First System Order', 2017. [Online]. Available: https://www.dwavesys.com/press-releases/d-wave%C2%A0announces%C2%A0d-wave-2000q-quantum-computer-and-first-system-order

[4] The Audiopedia, *What is QUANTUM DECOHERENCE? What does QUANTUM DECOHERENCE mean?*. 2017 [Online]. Available: https://www.youtube.com/watch?v=SKWI5CjbqnQ

[5] A. Muthukrishnan, 'Classical and Quantum Logic Gates: An Introduction to Quantum Computing', Rochester Center for Quantum Information, 1999 [Online]. Available: http://www2.optics.rochester.edu/~stroud/presentations/muthukrishnan991/LogicGates.pdf

[6] M. Mondal, 'Shor's Algorithm', University of Calcutta, 2014 [Online]. Available: https://www.slideshare.net/imrinalmondal/shors-algorithm-the-ppt

[7] A.G. White et al., 'Measuring Controlled-NOT and two-qubit gate operation', University of Queensland, 2003 [Online]. Available: https://arxiv.org/pdf/quant-ph/0308115.pdf

[8] 'Toffoli gate', Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Toffoli_gate

[9] D.J. Bernstein, 'Introduction to post-quantum cryptography', University of Illinois, 2009 [Online]. Available:www.springer.com/cda/content/document/cda_downloaddocument/9783540887010-c1.pdf

[10] D. Suter, 'Chapter 8: Quantum Algorithms', Dortmund University, 2014 [Online]. Available: https://e3.physik.uni-dortmund.de/~suter/Vorlesung/QIV_WS09/8_QAlgorithms.pdf

[11] M. A. Perkowski, 'Grover Part 2', Portland State University, 2005 [Online]. Available: www.ee.pdx.edu/~mperkows/QS/SLIDES-C-2005-q-0062-grover2.ppt

[12] R. Josza, "Quantum factoring, discrete logarithms, and the hidden subgroup problem," IEEE Computing in Science & Engineering, no. 2, pp. 34–43.

[13] D. R. Simon, "On the Power of Quantum Computation," SIAM Journal on Computing, no. 5, p. 1474.

[14] S. Gildert, "Experimental Quantum Computing: A technology overview", University of Birmingham, 2010 [Online]. Available: https://physicsandcake.files.wordpress.com/2010/02/experimental_qc_overview1.pdf

[15] N. Sendrier, "Code-Based Cryptography: State of the Art and Perspectives," *IEEE Security & Privacy*, vol.15, no.4, pp. 44-50, July 2017.

[16] R.J. McEliece, 'A Public-Key Cryptosystem Based on Algebraic Coding Theory,' *Deep Space Network progress report*, vol.44, pp. 114–116, 1978.

[17] L. Lamport, "Constructing digital signatures from a one-way function," SRI-CSL, Menlo Park, CA., USA, op.52, July, 2017, pp. 44-50. [Online]. Available: http://citeseerx.ist.psu.

edu/viewdoc/download?doi=10.1.1.228.2958&rep=rep1&type=pdf

[18]R. Merkle, “A certified digital signature,” in *Crypto’ 89 Proceedings on Advances in Cryptology*, vol.435 of *Lecture Notes in Computer Science*, July 1, 1989, Gilles Brassard, New York: Springer-Verlag, 1989, pp.218-238.

[19] T. Matsumoto and H. Imai, "Public quadratic polynomial-tuples for efficient signature-verification and message-encryption," in *Advances in cryptology—EUROCRYPT 1988*, vol.330 of *Lecture Notes in Computer Science*, May 25-27, 1998, Berlin: Springer-Verlag, 1988, pp.419-453.

[20] M. Ajtai et al., “A sieve algorithm for the shortest lattice vector problem,” in *STOC ’01 Proc. 33rd ACM Symposium on Theory of Computing*, July 6, 2001, New York: ACM, 2001, pp.601-610

[21] O. Regey, “On Lattices, Learning with Errors, Random Linear Codes, and Cryptography,” in *STOC ’05 Proc. 37rd ACM Symposium on Theory of Computing*, May 22-24, 2005, New York: ACM, 2005, pp.84-93

[22] O. Goldreich et al., “Public-key cryptosystems from lattice reduction problems,” in *CRYPTO ’97 Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology*, Sept. 12, 1997, Burton Kaliski Jr, New York: ACM, 1997, pp.112-131.

APPENDIX-A: SHOR'S ALGORITHM EXAMPLE

Example & Steps: Find factors p,q of **N**=15

1. If **N** even, prime, or a prime power (Ex: $49=7^2$), EXIT.
2. Pick random integer **x** < **N**; **x** =7
   - If GCD(**x**, **N**) ≠1, factors found. However, GCD(7,15)=1.
3. Quantum Algorithm*****: Find period **P** of **F**(**a**) = $\mathbf{x}^{\mathbf{a}}$ mod **N** where **a**=0,1,2,….,N-1

   7^0 **mod 15 = 1**
   7^1 **mod 15 = 7**
   7^2 **mod 15 = 4**
   7^3 **mod 15 = 13**
   7^4 **mod 15 = 1**

   Hence, period **P=4**
4. If **P** is Odd, go back to step 2. If **P** is Even:
   - **p** = GCD($\mathbf{x}^{P/2}$ + 1, **N**) →**p**=GCD($7^{4/2}$ + 1, 15) →**p**=GCD(50,15) →**p**=5
   - **q** = GCD($\mathbf{x}^{P/2}$ - 1, **N**) → **q**=GCD($7^{4/2}$ - 1, 15) →**q**=GCD(48,15) →**q**=3
5. Hence, N=p*q because 15=5*3.

*****How the Quantum Algorithm Section is Done with Quantum Computer Registers:

- **N**=15 and **x**=7
- Pick **u** as smallest power of 2 with $\mathbf{N}^2 \le \mathbf{u} < 2*\mathbf{N}^2$
  - If **N**=15 → 225 ≤ **u** < 450 → **u**=2^8 =256
- Quantum Algorithm: Find period **P** of ($\mathbf{X}^{\mathbf{a}}$ mod **N**) where **a**=0,1,2,….,**N**-1
  1. Input Register(**IR**) = [0,……..,**u**-1] = [0, 1, 2, 3, 4,……..,255]
  2. Output Register(**OR**) = [$\mathbf{X}^0$ mod **N**, $\mathbf{X}^1$ mod **N**,...., $\mathbf{X}^{(u-1)}$ mod **N**] all done at once
     = [7^0 mod 15, 7^1 mod 15,...., 7^{255} mod 15]
     = [1, 7, 4, 13, 1, 7, 4, 13,……..,13]
  3. **OR** is observed→ m=1(could be any of repeating 1,7,4,13):
     via entanglement, **IR** → [0,4,8,12,…..,252] collapses from 256 to 64 values
  4. QFT on the **IR**=[0,4,8,12,…..,252] → superposition with a "favored" value when observed
     - QFT(**IR**) →favored value **m**=64; or multiple like 0, 128, 192; QFT-Quantum Fourier transform
  5. Finally, **m**/**u** ≈ **d**/**P** by reduced fraction where **d** ϵ $\mathbf{Z}^+$, set of all positive integers.
     - **m**/**u** = 64/256 → **d**/**P** = 1/4 → **P**=4
  6. If **P** is wrong → Redo Shor's Algorithm with new random **x** value. However, in this example, P=4 is correct.

Note: The Quantum Fourier transform has the effect of taking a state **|c>** and transforming it into a state **|a>** given by:

$$\frac{1}{\sqrt{q}} \sum_{c=0}^{q-1} |c\rangle * e^{2\pi i a c / q}$$

Note: A is the set of all values that 7^a mod 15 yielded 1.
In our case A = {0, 4, 8, …, 252}

So the final state of the input register after the QFT is:

$$\frac{1}{\sqrt{64}} \sum_{a \in A} \frac{1}{\sqrt{256}} \sum_{c=0}^{255} |c\rangle * e^{2\pi i a c / 256}; \; |1\rangle$$

APPENDIX-B: GROVER'S ALGORITHM EXAMPLE

Example: Q=2 qubits → N=4 values (00, 01, 10, 11)

1. With function *f(.)*, determine index with correct solution
   - f(00)=f(10)=f(11)=0; f(01)=1
2. Initial State: |00>
3. Hadamard Transform(H_2) on State |00>:

$$H_2 * |00\rangle = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{2}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \frac{1}{2}\left(\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}\right) = ½(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$$

4. Oracle Operator makes "01" register negative (OHZH):

$$\text{Hence, } \frac{1}{2}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 \\ -1 \\ 1 \\ 1 \end{pmatrix} = ½(|00\rangle - |01\rangle + |10\rangle + |11\rangle)$$

5. Do $\mathbf{H_2}$ on ½(|00> - |01> + |10> + |11>) **(**OHZH**):**

$$\mathbf{H_2}*\begin{pmatrix}1\\-1\\1\\1\end{pmatrix} = \frac{1}{2}\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix} * \left(\begin{bmatrix}1\\0\\0\\0\end{bmatrix} + \begin{bmatrix}0\\-1\\0\\0\end{bmatrix} + \begin{bmatrix}0\\0\\1\\0\end{bmatrix} + \begin{bmatrix}0\\0\\0\\1\end{bmatrix}\right) = \frac{1}{2}\begin{pmatrix}1\\1\\-1\\1\end{pmatrix} = \text{½(|00> + |01> - |10> + |11>)}$$

6. Do **Z** on ½(|00> - |01> + |10> + |11>) **(**OHZH**):**

$$\mathbf{Z}*\frac{1}{2}\begin{pmatrix}1\\1\\-1\\1\end{pmatrix} = \begin{bmatrix}1&0&0&0\\0&-1&0&0\\0&0&-1&0\\0&0&0&-1\end{bmatrix} * \frac{1}{2}\begin{pmatrix}1\\1\\-1\\1\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1\\-1\\1\\-1\end{pmatrix} = \text{½(|00> - |01> +|10> - |11>)}$$

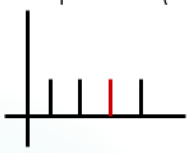
7. Do $\mathbf{H_2}$ on ½(|00> - |01> +|10> - |11>) (OHZH):

$$\mathbf{H_2}*\begin{pmatrix}1\\-1\\1\\-1\end{pmatrix} = \frac{1}{2}\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix} * \frac{1}{2}\begin{pmatrix}1\\-1\\1\\-1\end{pmatrix} = \frac{1}{4}\begin{pmatrix}0\\4\\0\\0\end{pmatrix} = |01\rangle$$

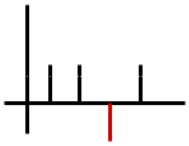
Note: Even though it was completed in 1 iteration of [OHZH] operations, it may take several iterations for larger databases.

Graphical Understanding

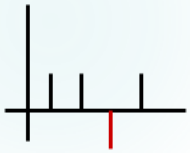
- Original Amplitudes (of each state)



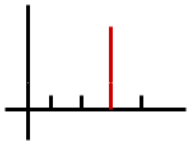
- Average of all amplitudes



- Negate Amplitude (of chosen number or state)



- Flip all amplitudes around average



Once these steps are done iteratively, state that we are searching for becomes increasing more enhances. After required number of iterations, when observed, the quantum superposition collapses to correct state.

APPENDIX-C: A NOTE ON HADAMARD MATRIX

PROP. If H is a Hadamard matrix of order n, then $\begin{bmatrix} H & H \\ H & -H \end{bmatrix}$ is a Hadamard matrix of order $2n$.

Examples of Hadamard Matrices:

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$H_4 = \left[\begin{array}{cc|cc} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ \hline 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{array}\right]$$

$$H_8 = \left[\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ \hline 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{array}\right]$$