A comprehensive study of Industrial Application of Quantum Software Engineering and Post-Quantum Cryptographic techniques

Krishnamoorthy. V,
M.E. Computer Science and Engineering,
Department of Computer Science and Engineering,
Government College of Engineering, Erode

Dr. Marikkannan.M,
Assistant Professor (Sr.),
Department of Computer Science and Engineering,
Government College of Engineering, Erode

ABSTRACT

Quantum computing, an innovative field at the forefront of technology, leverages the profound principles of quantum mechanics to transform the landscape of computation and unlock new realms of possibilities. Unlike classical computers, which use bits to represent information, quantum computers utilize qubits that exist in superposition, enabling parallel processing. This extraordinary capability holds the potential to solve complex problems exponentially faster, transforming fields such as cryptography, drug discovery, optimization, and artificial intelligence. Quantum computing's power lies in its ability to exploit quantum phenomena such as entanglement and interference to perform calculations that were once deemed infeasible. As we delve into the realm of quantum computing, a new era of unprecedented computational possibilities awaits us. This paper presents the vision of the quantum software engineering (QSE) life cycle consisting of quantum requirements engineering, quantum software design, quantum software implementation, quantum software testing, and quantum software maintenance that suffice all industrial and business application software development processes. With the development in Quantum Computing, the current strength of public key cryptography is challenged. Large-scale Quantum computers will be able to break many of the public-key cryptosystems currently in use and it would threaten the confidentiality and integrity of digital communications on the public networks. The literature review studies the hypothesis based on which it is predicted that Quantum Computing will challenge current use of public key cryptography. The literature review studies the research by various scholars, NIST, and several other researchers to analyse the impact of Quantum Computing to cryptography in use by businesses, time it may take to break the current strength of cryptography and in near future would recommend a standard process on what business should do to protect data.

Keywords: Quantum Computing and applications, Quantum Software Engineering, Quantum Security, Post Quantum Cryptography, Quantum public key encryptions.

I. INTRODUCTION

Quantum computing (QC) replaces the binary digits (bits) of classical electric computing with quantum bits (qubits), which, through features of quantum physics such as quantum states and quantum entanglement, will enable information to be processed exponentially faster than classical computers can manage [1, 2]. With quantum supremacy, in which a quantum computer can be shown to process any task faster than a classical computer on the horizon, researchers and companies are beginning to see some potential applications for exponentially faster computers than those in use today. These applications can improve our quality of life shortly. Machine learning powered by quantum computers promises to improve our quality of life, which is unimaginable [3].

QC promises to solve many problems more efficiently or precisely with classical computers, e.g., simulating complex physical systems or applying machine learning techniques [4, 5]. With recent advances in developing more powerful quantum computers, developing corresponding quantum

software and applications and integrating them into existing software architectures are becoming increasingly important [6, 7]. However, the development of such quantum applications are complex and requires the knowledge of experts from various fields, e.g., physics, mathematics, and computer science [8].

Quantum software Engineering (QSE) is an emerging research area investigating concepts, principles, and guidelines to develop, maintain, and evolve quantum applications [8, 9]. It aims to increase the quality and reusability of the resulting quantum applications by systematically applying software engineering principles during all development phases, from the initial requirement analysis to the retirement of the software [10]. In classical software engineering, software development lifecycles (SDLC) are often used to document the different development phases a software artifact or application goes through [11, 12]. Furthermore, such SDLC also summarizes best practices and methods that can be applied in the various phases and corresponding tools [13-15].

Hence, industrialists, software engineers, professionals get clear picture on QSE, by providing an overview of

development process or serving as a basis for cooperating with experts from different fields [16, 17].

Building practical and real-life QC applications requires the implementation of quantum algorithms as software. Learning from the classical computing realm, developing dependable software entails following an SDLC, which typically includes requirements engineering, architecture and design, development, testing, debugging, and maintenance phases. Given that quantum software development is relatively new, a particular SDLC for quantum software does not exist. Also, the security measures with which all these Quantum applications have to safeguarded needs a greater leap of research and development.

Quantum Cryptography is a nascent and rapidly growing field. Many corporations around the world are pouring in resources to further the knowledge and practices we have with regards to Post Quantum Security. Due to the varied interests and studies spread out across the globe, a need to understand the current emphasis on Quantum Security and the current advancements in this field have been presented as a survey.

Symmetric key algorithms are both classical and quantum-resistant (AES-256 has been used to characterize the highest level of security for all new algorithms), but they are difficult to implement in quantum circuits especially considering that quantum machinery has been developed only for a very small message size (approx. 20 bits). Further advancements in Quantum Mechanics based technology might lead to an expansion of these capabilities, resulting in better and more efficient ways to implement symmetric cryptosystems such as AES. For symmetric cryptosystems, the quantum ways to crack the algorithm require a quantum oracle. As long as symmetric cryptography is not implemented along with quantum oracle, they are safe against quantum attacks. All our present-day classical data is safe. However, the implications of quantum computing on the Public Key cryptosystem are much more serious. There is no requirement for a quantum implementation of the algorithms to crack it. An adversary with local quantum resources can exploit and crack the encryption algorithms. This makes all the asymmetric enciphered data unsafe and susceptible to attacks when efficient quantum computers are built.

This paper aims to raise a voice for action concerning the importance of quantum software engineering and its development life cycle model and to explore a robust post quantum cryptographic algorithm to ensure security in quantum software.

II. BACKGROUND

Quantum software developers need novel techniques, tools, processes, and methods that explicitly focus on developing software systems based on quantum mechanics. Designing a quantum software algorithm is challenging because of fundamental quantum mechanics characteristics, including superposition and entanglement. New principles and methodologies for quantum software design are strongly demanded as the design is the most critical phase of developing the QS systems [3].

A. Importance of Quantum in Emerging Technologies

QC is an emerging area with significant potential, especially in optimization problems. Since QC works with a different mechanism than classical computing, the software approach for QC is also different. QC is primed to solve a broad spectrum of computationally expensive societal and industrial problems. Notable examples include accelerated drug discovery and vaccine development in healthcare, portfolio management, finance optimization, and complex physics simulations to understand our universe better. As a result, QC success will inevitably and significantly impact our day-to- day lives and revolutionize most industries across many domains. As examples, the implications of QC in emerging technologies are discussed below [18-20].

1) QC in Smart Cities

One application of a QC-based IoT would be a fully integrated and automated smart city of the future. This would manage energy production and distribution, waste treatment and disposal, pedestrian and vehicle traffic, lighting, and even atmospheric control [21]. Cities are proliferating with the increase in the human population, and cities of the future will need to accommodate more and more people to limit the effects of climate change on natural ecosystems. QC could enable the people living in cities to maintain a good quality of life despite the pressures of substantial local populations [22].

2) QC in Smart Road Networks

With smart cars and, in the near future, self-driving cars connected to a QC-based IoT, road accidents could be eradicated, fuel usage would be optimized for efficiency, and congestion could be significantly reduced [23, 24]. Thus, QC gives a potential solution for managing automatic driving in a vehicle in everything context.

3) QC in Smart Air Traffic Control

Similarly, air traffic control could be processed by quantum computers. This would deliver drastic improvements to accuracy (and, therefore, safety) and the manageable size of the ATC network [25]. This is necessary for a future where crewless aircraft (drones) will take over many of the physical delivery tasks currently performed inefficiently by humans driving on roads. It will enable quality-of-life

improvements such as the quick, individual delivery of new products, medicines, and even passengers to destinations around large cities and beyond [26].

4) QC in Smart Factories

Suppose the speed of QC can be applied within the industrial automation sector. In that case, factories of the future will become vastly more efficient and able to take over more and more menial tasks currently completed or overseen by humans. This would create economic benefits and free human labour to explore more meaningful ways of spending time.

5) QC in Smart Power Supply and Distribution

A QC-enabled electric grid linked to the IoT promises to remove inefficiencies from the power supply and distribution system, reducing human's need for energy while maintaining the modern quality of life [29]. QC-enabled modelling and forecasting. Classical computers simulating complex human and natural systems (like financial markets and the planet's climate) and using these simulations or models to predict the future accurately have already led to quality-of-life improvements in the twentieth century. This is known as 'forecasting. These can be significantly enhanced by introducing QC, gathering and taking inputs across entire complex systems, processing them, and predicting how they will interact [30].

6) QC-enabled Machine Learning

Machine learning, in which computers can learn and reprogram themselves for greater efficiency or accuracy in completing tasks, can bring unimaginable new applications for computers [31]. Enabled by the much more powerful future quantum computers, machine learning could advance rapidly. This, coupled with QC-enabled modelling and forecasting, has the potential to advance medicine and eradicate many diseases, provide treatment efficiently, and quickly diagnose and treat ailments (supported by the IoT) [31]. Further, machine learning enabled by quantum computers will improve our quality of life that cannot be imagined yet, as computers will continue to learn and develop themselves to support human goals of survival, harmony with the planet's ecosystems, and luxurious and effortless lives [32].

B. WHY QUANTUM SOFTWARE ENGINEERING?

Over the last few decades, QC has intrigued scientists, engineers, and the global public. Quantum computers use quantum superposition to perform many computations in parallel that are not possible with classical computers, resulting in tremendous computational power. By exploiting such power, QC and quantum software enable applications typically out of the reach of classical computing, such as drug discovery and faster artificial intelligence (AI) techniques.

Quantum computers are currently being developed with various technologies, such as superconducting and ion trapping. Private companies, such as Google and IBM, are building their quantum computers, while public entities invest in quantum technologies [8]. For example, the European Union Commission is spending €1 billion on quantum technologies ("EU's Quantum Flagship Project's Website). The key goal for quantum computers is to reduce hardware errors that limit their practical uses. Regardless of the eventual technology that wins the quantum hardware race, quantum software is the key enabler for building QC applications [8].

Quantum software needs to be supported with a quantum software stack, ranging from operating systems to compilers and programming language. Quantum computing's inherent characteristics, such as superposition and entanglement, and practical quantum software applications cannot be developed with classical software engineering methods. Moreover, software developers face significant challenges when coding quantum programs due to switching to an entirely different programming mindset with counter-intuitive quantum principles [14].

QSE needs to provide methods for developing quantum software. QSE requires tasks such as the design of quantum programs, implementation techniques for quantum algorithms, and testing and maintenance of quantum software [14]. Following the conventional wisdom in software programming, which started from hardware-focused, hard- wired techniques in the 1950s and then evolved into today's agile, iterative development, QSE should eventually become agile, iterative, and incremental [36]. Thus, we need to build novel QSE methodologies (with tool support) that cover different phases of QSE. Learning from the classical software engineering realm, developing trustworthy software entails following an SDLC, which typically includes requirements engineering, architecture and design, implementation, testing, and maintenance phases.

III. QUANTUM SOFTWARE DEVELOPMENT PHASES

QC is a technological revolution that demands a new software engineering paradigm to develop and conceive quantum software systems [14].

In QSE, developing a quantum software code is not the top priority problem, however, the requirements and design problems are much more common and important to correct. Therefore, the focus on quantum software development

techniques should not be limited to quantum coding issues but should overall focus on other aspects of QSE.

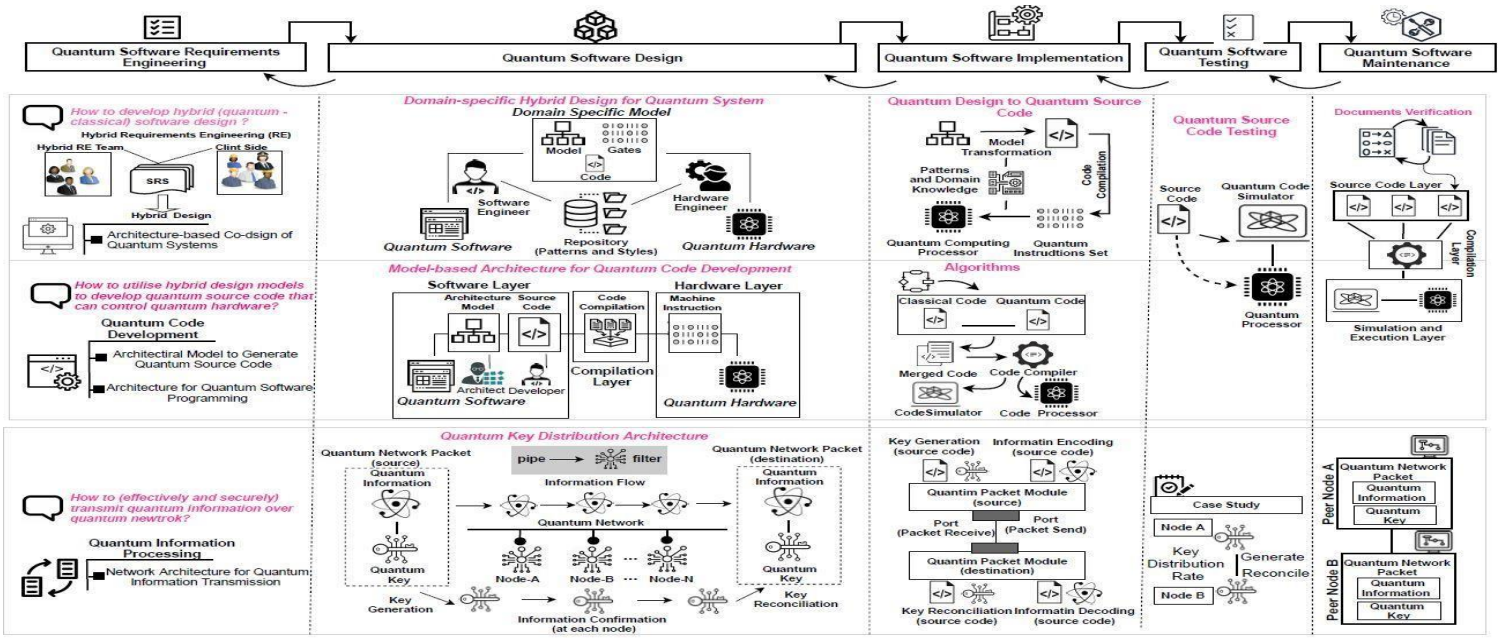

Figure 1: Quantum Software Engineering Phases and Lifecycle

QSE is the use of sound engineering principles for the development, operation, and maintenance of quantum software and the associated document to obtain economically quantum software that is reliable and works efficiently on quantum computers. Thus, QSE calls for novel techniques, tools, processes, and methods that explicitly focus on developing software systems based on quantum mechanics. In the following sections, we briefly discuss the quantum software development life-cycle phases (i.e., quantum software requirements engineering, quantum software design, quantum software implementation, quantum software testing, and quantum software maintenance) [8, 14, 36, 39]. Table 1 presents the important artifacts of QSE life cycle, and a set of quantum software development activities derived from [8] and presented in Figure 1.

A. Quantum Software Requirements Engineering

We believe that quantum requirements engineering will be like requirements engineering process done for classical computing due to focus on requirement elicitation and management aspect of the phase. However, quantum requirements engineering will need new modelling and specification techniques to model QC aspects such as login functions and state. We believe that requirements engineering research community need to extend classical use cases, user stories and goal modelling techniques to support quantum requirements engineering process [14].

B. Quantum Software Design

Like classical software development, quantum software design involves two major phases, i.e., architectural design and detail design. Architectural design is quantum software's abstract-level design, where the main components' interactions are described. The detailed design explicitly describes the data structure, algorithms, and interfaces for module interactions. Developing algorithms for quantum software design is challenging compared to classical software systems because of some fundamental quantum computing features, including superposition and entanglement [8].

C. Quantum Software Implementation

Some work has been done to develop quantum programming languages [14]. There are several quantum programming languages, i.e., C (QCL), C++ (Scaffold), C\# (Q\#), Python (ProjectQ, Qiskit, Forest), F\# (LIQUi|⟩), Scala (Chisel-Q), and Haskell (Quiper) [14]. However, research community need to focus on developing commercial integrated development environments to support quantum software implementation.

D. Quantum Software Testing

The testing phase begins to find defects and verify the system's behaviour. It might be possible that the programmers make various mistakes when performing the quantum software testing because quantum computers have different properties such as superposition, entanglement, and no cloning, which make it challenging to predict the quantum software's behaviour [14]. There is a strong need for different testing tools and techniques that consider the quantum computing characteristics such as reading intermediate states, handling probabilistic test oracles and facing the decoherence problems.

E. Quantum Software Maintenance

The maintenance phase includes updating, changing, and modifying the quantum software to meet customer needs. This is an emerging research area in quantum software

engineering, specifically focusing on re-engineering the available classical information systems and their integration with quantum algorithms.

IV SECURITY OF CLASSICAL CRYPTOGRAPHY

The advent of quantum computers has called into question the security of classical cryptographic algorithms. Symmetric cryptography's security has been halved in bits by quantum computers. Asymmetric cryptography will turn obsolete with quantum computational resources.

A) Quantum Security of DES

DES tries to achieve sufficient randomness through multiple rounds, however, due to the comparatively smaller key size of DES (56-bits), it can be broken easily in view of current computational resources. 3-DES increases the key length, albeit at the loss of ease of encryption, to 168-bits. Increasing the key space to 2168 raises the security to be resistant to modern brute force searches. Like AES, Grover's Algorithm can be used for Quantum Searches on the key, this reduces the number of operations required to 284 which is not very secure. 3DES is also prone to collisions and therefore is increasingly susceptible to Simon's Algorithm. 3DES is also very slow and computationally taxing and cannot be considered for widespread use.

Models of SDES [25] are currently being implemented to understand the Feistel Round structure in Quantum Circuits. The implementation of the S-Box lookup tables requires many complicated gate arrangements and cannot be compromised due to its core function in providing non-linearity to the Algorithm. The complexity of implementing 3DES in Quantum Circuits may suggest some viability for use however due to the ease in breaking the core DES algorithm, it is avoided by most.

Symmetric Cryptographic Algorithms are currently considered non-implementable in a Quantum Model and are a lot more resistant to Quantum Searches compared to public-key schemes.

B) Quantum Security of AES

AES has proven to be one of the most robust cryptographic schemes currently in use, proving resilient to the level of exhaustive brute force attacks that are currently computationally viable.

The recent advancements in the development of Quantum Computers have however led people to re-open the question of AES's security against Quantum Computer Brute Force Exhaustive Search attacks, Asymmetric Public-Key schemes such as RSA, ECC, etc have been known to be broken completely by Quantum Exhaustive Searches due to the immense parallel processing possible through superposition of qubits [26].

Various algorithms that exploit the principle of super-position have been put forward such as Simons Promise, Grover's Search, and Shor's Algorithm.

Grover's search algorithm reduces the exhaustive key search from O(N) to O((N/M)1/2) trials, where M can be reduced to 1 by choosing functions that give single solutions while implementing AES in a quantum circuit. A major drawback of Grover Search based cryptanalysis of symmetric cryptosystems is that the cryptosystems must be implemented as a quantum circuit.,. Many implementations of AES as a quantum circuit exist. The crack using Grover's search algorithm requires the AES algorithm to be on a quantum circuit. This limits the application of the crack only to quantum oracles. Furthermore, as mentioned above, current algorithms are only able to reduce the trials by N 1/2, where N is the length of the key. One can simply increase the key length to preserve security, for example, moving from AES-128 to AES-256 can still be considered unbreakable in the face of Brute Force Searches.

C) Quantum Security of RSA

Shor's factoring algorithm is a quantum circuit that can factorize big numbers in polynomial time [27]. This is a huge speedup compared to classical methods that are dependent on exponential solutions for the same. With the development of quantum computers, in the near future, the factorization problem will not be a hard problem to solve. This will make RSA cracking very easy. The first practical implementation of RSA for factorization big numbers required more than a billion qubits. This went down to 20 million qubits in late 2019 [28]. This implementation is capable of factoring 2048 RSA integers in less than 8 hours using 20 million noisy qubits. The most powerful present-day quantum computers have 50-100 qubits. In the next 25 years, the software improvements and hardware improvements will meet to make factoring prime numbers a reality.

The algorithm uses quantum Fourier transformation to take advantage of quantum parallelism. The prime number factoring problem is converted to a period finding problem. Many optimizations are done by the authors of the paper to reduce the number of qubits required for the factorization. Ekera-Hastad's [28] derivative of Shor's factorization algorithm is used to decrease the number of multiplication operations. Many other optimizations like windowed arithmetic and oblivious carry runaways are used.

The main idea behind Shor's Prime Factorisation algorithm is as follows [29].

Let $N = p * q$ such that p and q are prime numbers. To factorize N a random number $a < N$ is selected such that GCD (a, N) = 1. The period of f (x) = axmodN is found (Quantum circuit). If the period r is an odd number the

algorithm is repeated from the start. Now if the period is an even number, we have f (x) = f (x + r). The factors of N are GCD(a(r/2±1), n). The period calculation takes the help of quantum Fourier transformation.

The first part of the algorithm requires setting an input and output register where input is the superposition of all possible inputs and output is set of 0s. The input and output registers are entangled.

The period calculation takes the help of quantum Fourier transformation. Input and output registers with log2N qubits each are initialized. The input register is initialized to a superposition of all possible values from 0 to N-1.

A quantum circuit for the function f (x) = axmodN is made and the input register is passed through the circuit to initialize the output register as a superposition of f (x) for all x in input registers. Quantum Fourier Transformation is applied to the input register. A measurement is then performed in the input register which gives y. Since the input and output registers are entangled, this changed the quantum state of the output with f (x0).

Y/N is converted to an irreducible fraction and the value of the denominator is used as a potential value for the period of the function. If it is the period done else multiples of the denominator are tested. If they work the period is found else the registers are initialized again and the procedure is repeated (measurement value is random and may change resulting in the different output of y).

Shor's factoring algorithm taps at the root idea be- hind the RSA algorithm. A replacement for the same is required in the post-quantum world. NIST is looking for new post-quantum computer era encryption algorithms.

D) Quantum Security of Diffie Hellman Key Exchange Protocol

Diffie Hellman key exchange protocol is based on the discrete logarithm problem. The idea behind Shor's algorithm can be used to give an exponential speedup to solve the discrete logarithm problem.

The discrete logarithm problem can be converted to a bivariate function's period finding problem. The period finding problem can be solved using quantum Fourier transformation which takes advantage of quantum parallelism to speed up the computations.

Let p and q be a big prime number and a generator in Zp group respectively. Y = qk. The value of k has to be found to solve the problem. A bivariate function f (x1, x2) = qx1yx2 is defined. The period of the function is found using the period finding algorithm. f (x1 +w1, x2 +w2) = f (x1, x2). After finding the period, we get k = −(w1/w2) modp [30].

The above algorithm is a polynomial solution to crack the Diffie Hellman Key Exchange Protocol. The protocol is based on the discrete logarithm problem. As the discrete logarithm problem will be computable with the development of quantum computers the protocol will become unsafe.

E) Quantum Security of ECC

ECC is based on the discrete logarithm problem in elliptic groups. Shor published two algorithms in his famous paper. One of them was for factoring big numbers and the other was a solution to the discrete logarithm problem in any group.

All possible attacks to ECC require attacking the dis- crete logarithm problem. There are only exponential al- gorithms for the same. Shor's algorithm is a polynomial- time algorithm that takes advantage of quantum paral- lelism and Quantum Fourier Transformation for period finding. This can be used to solve the discrete logarithm problem and hence attack ECC in polynomial time.

As ECC requires smaller key sizes it will be com- paratively easier to crack than RSA when quantum computers are a reality. The number of qubits required to crack ECC will be significantly lesser than qubits required to crack RSA making it susceptible to attacks sooner than RSA in the future.

V POST QUANTUM CRYPTOGRAPHY

Post Quantum cryptography is the field of cryptography where encryption algorithms are developed which are secure from an adversary with quantum computers. All the present-day asymmetric algorithms (RSA, ECC, DH, DSA) are crackable using quantum computers. They are based on the Prime Factoring problem or the Discrete Logarithm Problem which are easy to solve on quantum computers using Shor's Algorithm. Mathematicians and cryptographers used these number theory problems to base the security of the

asymmetric algorithms. Now they have to search for new mathematical problems which can't be solved by quantum computers easily.

Symmetric Algorithms and hash functions are comparatively secure in a post-quantum world. Grover's Algorithm can speed up the attacks by square root complexity [31]. However, most of the algorithms can be made secure again by doubling the key size.

All the current asymmetric algorithms are based on mathematical problems for which people have searched solutions for centuries. However, the weakness they have is that quantum computers are good at parallel tasks that require one result in the end. Since the algorithms require only one result in the end a superposition of qubits can be used to parallelize all the computations and then the result can be measured. To avoid taking advantage of the parallelism of quantum computers algorithms that require

several results can be used. This way parallelism of quantum computers can't be used to its full extent.

Presently most of the post-quantum algorithms are part of 6 different families. All of them are a different family of mathematical problems which are difficult to solve even for quantum computers. These mathematical problems will form the base for the next generation of asymmetric algorithms. NIST started the Post Quantum Cryptography standardization process in 2016. It is searching for new digital signature schemes and Public Key Encryption schemes.

A) Lattice-based Cryptography

Lattice-based cryptography algorithms are based on hard mathematics lattice problems. The family of lattice problems is used in this type of cryptography. A key feature of lattice-based cryptography is that it involves security based on the worst-case problems. Most of the other cryptosystems have security based on the average case [32].

Lattice is a regularly spaced grid of points stretching out to infinity. A vector is a point in this lattice that is, a tuple representing the coordinates of a point. The origin is a tuple of all 0s. A vector is called a far vector if it is far away from the origin and a short vector if it is close to the origin. A basis is a small set of vectors that can be used to represent the entire lattice space.

These vectors can be linearly combined to represent any vector in the lattice space. For a n-dimensional lattice n vectors are selected for the basis with the condition that when a line is made with the origin and a point no other point in the set lies on the line. This way a basis can be formed.

There are many bases for a lattice. A basis is short if it consists only of short vectors and a basis is long if it consists of long vectors. A few major hard lattice problems include-

Short Vector Problem: A long basis for a lattice L is given. Find a grid point in L as close to the origin as possible.

Short Basis Problem: A long basis for a lattice L is given. Find a short basis for the same lattice space L. Closest Vector Problem: A long basis for a lattice L is given. Also, a challenge point P in the lattice space is given. Find the closest point to P in the lattice space L.

Short Integer Solution Problem: m vectors of n dimensions are given such that the vectors Vi ∈ Zn. Find a coefficient vector y ∈ Zn such that the linear combination of y and vi gives 0. Here y are small integers like y ∈ {0, 1} m.

All these problems seem elementary in a small space. However, in cryptography, the lattice space has huge dimensions. The problems are also simple if a short basis is given but for a long basis, it's a hard mathematics problem. Mathematicians as far back as the 1800s have worked on lattices. These serve as deep insights into what can and cannot be done with lattices. This gives confidence in using the lattice problems as the base for asymmetric algorithms. A major attack against them involves using lattice reduction algorithms like LLL. LLL algorithm is a polynomial algorithm used to find a relatively small basis (not smallest) for a lattice space in which a long basis has been specified.

Ajtai-Dwork [32] came up with a cryptosystem using the Shortest Vector Problem in 1997. It was cracked in 1998 by Nguyen and Ster [33]. Goldreich-Goldwasser- Halevi algorithm [34]as published based on the Closest Vector Problem. This was cracked by Nguyen in 1999. NTRU [36] was published in 1996. Over the years it has been modified and improved and the NTRU encryption system is a final candidate for the NIST Standardization process. The present NTRU is a merger of two different second-round NTRU candidates.

1) GGH Encryption Scheme

A receiver's private key is a short base and the public key is a long base. The number of dimensions in the lattice is equal to the bit size of the message. The sender uses the bad base to find a challenge point that is close to a lattice point which is the message. This message is easily decrypted by the receiver as the receiver has a short base available.

For adversaries, it is difficult to crack as they have only long bases. It is believed to be quantum-proof but was broken by a classical computer due to certain vulnerabilities. After the collection of many encrypted messages, partial information of the plaintext could be recovered. Similar ideas with reduced vulnerabilities are used in many of the post-quantum cryptography candidates.

2) Learning With Errors

Learning With Errors is a subset of lattice cryptography. It takes advantage of a new trapdoor function that is easy to compute but hard to invert. AX = B. If A and B are given computing X is easy using Gaussian Elimination [37]. Now if random errors e are added to AX and the only information available is B and A finding X becomes a hard mathematics problem. AX + e = B. This is used as the base for cryptographic algorithms.

Rings Learning with Error and Module Learning with Errors are modifications of LWE that do not require big key sizes like LWE.

The equation AX = B can have more equations than variables. However, equations are written so that the system is solvable. The private key is the value of all the variables. The public key is the matrices A and

B. Now if a sender wants to encrypt a message the sender randomly selects a subset of the equations and adds them. In this new equation, the sender adds a big error to encode 1 and a small error to encode 0. The receiver can use the private key (value of variables) to check if the error added is big or small to decode to 1 or 0 accordingly. This problem is a hard mathematics problem making it computationally secure.

27 of the 69 algorithms submitted to NIST are lattice-based algorithms. Google implemented Learning with errors to google chrome browser recently.

B) Hash-based Digital Signatures

Hash-based digital signature schemes are an alternative to present day digital signature schemes which use asymmetric algorithms like RSA. They depend on 2 properties of the hash function's collision resistance and preimage resistance.

Preimage resistance of a hash function H implies that given an output y of the hash function it is difficult to find any input x such that y = H(x). Weak collision resistance of a hash function H implies that given arbitrary message m1, it is difficult to find another message m2 such that H(m1) = H(m2). Strong collision resistance of a hash function H implies finding messages m1 and m2 such that H(m1) = H(m2). Note: Strong collision resistance is easier to exploit because of the birthday paradox. Unlike weak collision resistance and preimage resistance which require a O(2n−1) (where n is the number of bits of the output of the hash function) search, strong collision resistance requires o(2n/2) search as a consequence of the birthday paradox.

Finding collisions and preimages is a difficult problem if the underlying hash function is good. Finding quantum algorithms to perform these tasks will be hard if not impossible. Hence these hash-based digital sig- nature schemes can be used for authentication in the post-quantum world. However, they suffer from a serious disadvantage that each digital signature can be used only once.

2 of the 69 schemes in the NIST competition are hash-based cryptosystems. Lamport introduced the hash digital signature scheme in 1979 [38] . Witernitz described a One Time Signature scheme which was significantly more efficient than Lamport's scheme. It has the smaller key size and signature size. Merkle introduced a new scheme that combined the Witernitz approach with binary trees and called it Merkle Signature Scheme. SPHINCS+ an alternative candidate for digital signatures uses a combination of the Witernitz One-Time Signature Plus Scheme and Merkle hash trees in the Forest of Random Subsets signature scheme.

1) Lamport Digital Signature Scheme

The Lamport Digital Signature Scheme is a one-time signature scheme. It requires a secure hash function. For b parameter of security [4] we require a hash function that produces 2b output. Consider we require 128-bit security hence any secure hash function with 256 bits can be used.

The private key is produced using a random number generator. 256 pairs of random numbers are generated. Each number is 256 bits. This serves as the private key. Hence the size of the private key is 8b2.

The public key is the 512 hashes of all the random numbers generated. Hence it is also of the size 8b2. These values are published by the user who wants to digitally sign a document. The signing algorithm [3] requires the hash of the message. Now as the hash of the message consists of 256 bits. For all the 256 bits (depending on the value 0 or 1) we select a number from the 256 pairs and publish it. So, a sequence of 256 numbers is the digital signature. This is published along with the message.

The verification algorithm requires the verifier to hash the message. Then for each bit of the hashed message, the corresponding hash is selected from the public key. The recipient then hashes each number of the sender's private key ad verifies if they match with the selected public key. This scheme can only be used for one-time signature after which the private key and public key pair is discarded.

C) Code-based Cryptography

Code-based cryptography is based on error-correcting codes. Computer scientists have been working on these for over 40 years. Error correction codes are codes used widely in communications to correct transmission errors. To send a message the text is sent into an error correction code. Then to the output, a few errors are randomly introduced and sent.

An example of a well-known code-based cryptosystem is the McEliece algorithm [39]. It takes the help of linear error correction codes (matrix multiplication). The receiver has a good error correction code as the private key. This is multiplied by 2 blinding matrices to produce a bad error correction code, the public key. The public key is shared with everyone. A sender sends the plaintext through the bad error correction code. Then according to the overhead, the sender adds errors. This is the final ciphertext that is transmitted. The receiver uses her good error correction code to decrypt the ciphertext.

McEliece algorithm was introduced in 1978 [40] and nobody has found a weakness in it till now. The major reason why it is not practically implemented is the size of the public key. It is much larger than its asymmetric counterparts like RSA.

D) Non-Commutative Cryptography

Non-Commutative cryptography takes the help of non-commutative groups(A + B /= B + A). A simple example is

a Rubik's cube where a move sequence is a set of moves applied on the Rubik's cube [41]. The addition is the concatenation of move sequences. The addition of move sequences is non-commutative. Move elimination is a move that does the opposite move. This way the previous move is eliminated and the move sequence can be written without the two moves. Move sequence negation of a move sequence consists of all the oppo- site moves in reverse order. This way an entire move sequence is reversed. The negation sequence of a move A is represented as -A.

Conjugacy Problem: 2 move sequences A and B are given. Find X such that X + A − X = B

Since it is non-commutative it is very hard to solve. The one-way function is easy to set up but difficult to reverse.

1) Stickel Key Exchange Protocol

As Diffie-Hellman key exchange protocol is not quantum- proof an alternative called Stickel Key Exchange is used. This is quantum-proof and based on non-commutative cryptography.

In Stickel key exchange 2 sequences, A and B are defined. Also, both Alice and Bob have 2 natural numbers each which serve as their private key. Let the numbers be n and m for Alice and r and s for Bob. Alice generates a public key PKa = mA + nB and sends it to Bob. Bob generates a public key PKb = rA + sB and sends it to Alice. Then Alice computes Ka = mA + PKb + nB = (m + r)A + (s + n)B and Bob computes Kb = rA + PKa + sB = (r + m)A + (n + s)B The 2 keys are the same and cannot be computed by an adversary due to non-commutative property of the group. For an eavesdropper, the conjugacy problem has to be solved.

In the NIST competition, only 1 algorithm out of 69 is a non-commutative cryptosystem. This was broken hence no non-commutative cryptosystem will be standardized in the competition.

E) Multivariate Cryptography

Multivariate cryptography is based on the hard math- ematics problem of solving a system of multivariate polynomials. Multivariate cryptosystems are all based on the multivariate quadratic map [44]. The quadratic map takes a sequence x = (x1, ..., xn) ∈ Fn and returns an output y = (p1(x), ..., pm(x)) ∈ Fm where pi(x) are multivariate quadratic polynomials for i = 1, .., m and the coefficients of the polynomials are in Fq . The map is called a multivariate quadratic map P with m compo- nents and n variables.

MQ Problem: Given P : Fn '→ Fm a multivariate quadratic map and a target t ∈ F find a value s such that P (s) = t. Here s is not unique as map P is not an injective map. This problem is considered a hard problem even for quantum computers. There are methods like the Grobner basis that help solve the problem. Recently many Grobner basis-like algorithms such as F4/F5 and XL are used for solving the MQ Problem.

Mainly digital signature schemes are designed on top of the MQ problem. The most widely known digital signature algorithm in multivariate cryptography is the Oil and Vinegar Scheme. Rainbow is a digital signature scheme based on the Unbalanced Oil and Vinegar scheme and a finalist in the NIST competition.

1) Oil and Vinegar Scheme

The digital signature scheme is based on the hard mathematics MQ problem. It has a combination of Oil and Vinegar variables in the polynomials. A random quadratic map P : Fn '→ Fm is selected. The number of variables in this map is n (variables) and the number of polynomials is m (components). The number of variables is greater than the number of polynomials. There are m oil variables and n-m vinegar variables [46].

The idea behind the scheme is that all the polynomials have quadratic terms which combined vinegar- vinegar variables and oil-vinegar variables. There are no oil-oil quadratic terms. These polynomials are dispersed using an invertible linear map to avoid differentiation between oil and vinegar variables.

If Alice wants to digitally sign a document, she first makes her complete quadratic map R = P ◦T (combination of linear maps and quadratic map) public and keeps P and T as her private key. To digitally sign a document d she computes a hash of d using any secure hash function H. y = H(d). Then she computes s′ = T −1(y) as T is invertible linear map. Then she finds s such that s = P −1 (s') and shares it. This is her signature. Anyone who receives the document can use her public key R to compute y =R(s) and match it with the hash of the document.

For Alice computation of s = P −1(s′) is easy as she knows the vinegar and oil variables. She randomly selects values for vinegar variables and inserts them. As only the vinegar variables have quadratic terms she is left with a linear system of oil variables. As there are m oil variables and m equations it is easily solved using Gaussian elimination. However, due to the linear map it becomes difficult for the adversary to differentiate between oil and vinegar variables. He is stuck with an NP-hard MQ problem to solve.

The oil and Vinegar scheme [47] has an equal number of oil and vinegar variables. It was easily cracked by Kipnis and Shamir [48]. A new scheme called Un- balanced Oil and Vinegar scheme was [46] proposed where the number of vinegar variables is not equal to the number of oil variables. After a lot of deliberation, correct parameter values for n and m have been decided which are less vulnerable to attacks.

The Rainbow digital signature scheme is a scheme that has multiple UOV layers built into it [49]. This was done to make it less vulnerable to the MQ Problem attack. However, these complications gave rise to a new attack called the MinRank Problem attack.

MinRank Problem: k Matrices Mi{i = 1, 2, .., k} are given with n rows and m columns and a target rank r is given. The problem requires finding coefficient vector y such that the linear combination of the matrices with coefficient vector y gives rank at most r.

The increase in layers increases its vulnerability against the MinRank Problem. Optimized size of 2 layers is used in the NIST Rainbow Scheme. Beullens [50] recently came up with a new intersection attack that reduces the security of the rainbow by a few bits.

F) Isogeny-based Cryptosystems

The isogeny-based cryptosystems are based on elliptic curves. Curves are defined by the solution of a polynomial equation in 2 variables. Elliptic curves are of the form y2 = x3 + ax + b, where 4a3 + 27b2 /= 0. They have a property that when any two points on the elliptic curve are joined a line is formed that intersects the curve at a third point. This third point is reflected over the X-axis to find a new point that is called addition in the elliptic curve. The condition 4a3 + 27b2 /= 0 ensures no singular points are there. To double a point a tangent is drawn on the elliptic curve with that point. The tangent intersects at another point whose reflection over X-axis is taken. This gives its double. Also, scalar multiplication with n of a point P involves adding P, n times i.e., [n].P = P + P + .. + P (ntimes).

Elliptic Curve Diffie-Hellman (ECDH) was the first elliptic curve key exchange algorithm introduced [51]. It involved standardizing an elliptic curve in a field over a prime number p and a point P with special properties for communication. Then both Alice and Bob would select random numbers na and nb which functioned as their private key. They calculated Pa = [na]. P and Pb = [nb].P and shared it with the other party. They then multi- plied their private key to the shared point to arrive at a shared secret ie Alice calculated [na].Pb = [nanb].P and Bob calculated [nb].Pa = [nanb].P . This is the shared secret. The security of this algorithm is based on the discrete logarithm problem over elliptic curves. This was theoretically cracked by Shor's algorithm and hence is no longer safe in a world with quantum computers. After this people further studied elliptic curves and came up with isogeny-based systems.

1) Super singular Isogeny-based Diffie Hellman (SIDH)

An isogeny is a homomorphic rational map between two elliptic curves Φ : E0 '→ E1. In SIDH Alice and Bob generate private isogenies as their private key. They apply their isogenies to a common elliptic curve E and then share their results to the other party. The other party uses the new elliptic curve and finds another elliptic curve with the help of their private isogenies. Unlike ECDH, the final curves are not necessarily the same curve, however, they are structurally identical ie the curves are isomorphic. For elliptic curves, a number called the j-invariant can be calculated. This is the same for isomorphic curves. Hence the j-invariant is calculated individually by Alice and Bob and serves as the shared secret. An n-torsion group of an elliptic curve E[n] is a set of all points P in E that satisfy [n].P = 0.(where 0 is the additive identity of E).

The SIDH was introduced in 2011 [52]. It involves fixing a supersingular elliptic curve E that is defined on the field Fq where q = p2 and p = 2a.3b − 1. Alice and Bob share the super singular elliptic curve E along with a basis Pa, Qa for E[2a] and Pb, Qb for E[3b](A basis is a set of points whose linear combinations can generate the entire set of points in the group).

Alice and Bob select random numbers ra and rb between 0 and 2a−1 and 0 and 3b−1 respectively. Al- ice calculates her isogeny ΦA using the kernel generated by RA = PA + [rA].QA. Alice then shares EA = ΦA(E), ΦA(Pb) and Φa (Qb) as her public key. This helps Bob compute his second isogeny ΨB . Bob calculates ΨB using the kernel generated by ΦA(PB) + [rB].ΦA(QB). He then finds the j-invariant of ΨB(EA). Alice uses a similar procedure to compute her j-invariant which becomes the shared secret.

The security of the scheme is based on the le-isogeny hard problem. The le isogeny problem is given two elliptic curves E1 and E2 and an isogeny Φ exists between the two with kernel size le, find the kernel of Φ. For SIDH it involves the 2a-isogeny and 3b-isogeny hard problems. Mathematicians have worked for over 20 years but have not found any significant attacks against it for a big power e and any natural number l. A key encapsulation mechanism SIKE (Supersingular Isogeny-Based Key En- capsulation) is based on the SIDH and is part of the 3rd round alternate candidates for Key Encapsulation Mechanisms in the Post Quantum Cryptography Standardization process.

VI CONCLUSION AND FUTURE WORKS

Quantum computing is primed to solve a broad spectrum of computationally expensive societal and industrial problems. Notable examples include accelerated drug discovery and vaccine development in healthcare, portfolio management, finance optimization, and complex physics simulations to better understand the universe [43]. As a result, QC's success will inevitably and significantly impact our day- to-day lives and revolutionize most industries across different domains. Such impact must be realized via quantum software, the development of which should be systematically powered by quantum software engineering [14]. QSE opens new research areas to develop real applications by fostering research communities across

disciplines (such as computer science, software engineering, mathematics, and physics) and interactions with other fields, such as medicine, chemistry, and finance. QC is on the rise and will revolutionize many areas of life. It will transform our understanding of and deal with complex problems and challenges. QSE is key to the systematic and cost-effective creation of tomorrow's robust, reliable, and practical QC applications.

Despite recent advances in quantum programming tools, there are two challenges as of now, in Quantum algorithms exist for all major Public Key Cryptosystems and it is only a matter of time before they are broken completely. Researchers have been trying to find ways to either in- crease the hardness of the problems that are currently being used (RSA, ECC) or come up with new problems that are sufficiently difficult enough even for a Quan- tum Computer. Quantum cryptography has been more extensively utilized, with more research being encouraged into the fields of Lattice Problems, Error-Correcting Codes, Noncommutative cryptosystems, and Hash-Based Cryptosystems. Many algorithms being proposed however are difficult to implement and their performance must be optimized for widespread public use.

VII REFERENCES

[1] C. Outeiral, M. Strahm, J. Shi, G. M. Morris, S. C. Benjamin, and C. M. J. W. I. R. C. M. S. Deane, "The prospects of quantum computing in computational molecular biology," vol. 11, no. 1, p. e1481, 2021.

[2] M. De Stefano, F. Pecorelli, D. Di Nucci, F. Palomba, A. J. J. o. S. De Lucia, and Software, "Software engineering for quantum programming: How far are we?," vol. 190, p. 111326, 2022.

[3] A. Ahmad, A. A. Khan, M. Waseem, M. Fahmideh, and T. Mikkonen, "Towards Process Centered Architecting for Quantum Software Systems," in 2022 IEEE International Conference on Quantum Software (QSW), 2022, pp. 26-31: IEEE.

[4] T. Häner, D. S. Steiger, K. Svore, M. J. Q. S. Troyer, and Technology, "A software methodology for compiling quantum programs," vol. 3, no. 2, p. 020501, 2018.

[5] A. Ahmad, A. A. Khan, M. Waseem, M. Fahmideh, and T. Mikkonen, "Towards Process Centered Architecting for Quantum Software Systems."

[6] L. Nita, L. Mazzoli Smith, N. Chancellor, H. J. R. i. S. Cramman, and T. Education, "The challenge and opportunities of quantum literacy for future education and transdisciplinary problem-solving," pp. 1-17, 2021.

[7] E. Altman et al., "Quantum simulators: Architectures and opportunities," vol. 2, no. 1, p. 017003, 2021.

[8] A. A. Khan et al., "Software Architecture for Quantum Computing Systems-A Systematic Review," 2022.

[9] M. A. Akbar, S. Rafi, and A. A. J. a. p. a. Khan, "Classical to Quantum Software Migration Journey Begins: A Conceptual Readiness Model," 2022.

[10] R. J. Q. LaRose, "Overview and comparison of gate level quantum software platforms," vol. 3, p. 130, 2019.

[11] M. A. Akbar et al., "Improving the quality of software development process by introducing a new methodology–AZ- model," vol. 6, pp. 4811-4823, 2017.

[12] N. M. A. Munassar and A. J. I. J. o. C. S. I. Govardhan, "A comparison between five models of software engineering," vol. 7, no. 5, p. 94, 2010.

[13] B. Weder, J. Barzen, F. Leymann, M. Salm, and D. Vietz, "The quantum software lifecycle," in Proceedings of the 1st ACM SIGSOFT International Workshop on Architectures and Paradigms for Engineering Quantum Software, 2020, pp. 2-9.

[14] J. J. a. p. a. Zhao, "Quantum software engineering: Landscapes and horizons," 2020.

[15] M. A. Akbar, K. Smolander, S. Mahmood, A. J. I. Alsanad, and S. Technology, "Toward successful DevSecOps in software development organizations: A decision-making framework," vol. 147, p. 106894, 2022.

[16] M. A. Akbar et al., "Statistical analysis of the effects of heavyweight and lightweight methodologies on the six- pointed star model," vol. 6, pp. 8066-8079, 2018.

[17] M. A. Akbar, A. A. Khan, Z. J. I. T. Huang, and Management, "Multicriteria decision making taxonomy of code recommendation system challenges: a fuzzy-AHP analysis," pp. 1-17, 2022.

[18] T. Zülsdorf et al., "Quantum engagements: social reflections of nanoscience and emerging technologies," 2011.

[19] M. O. Alas, F. B. Alkas, A. Aktas Sukuroglu, R. Genc Alturk, and D. J. J. o. M. S. Battal, "Fluorescent carbon dots are the new quantum dots: an overview of their potential in emerging technologies and nanosafety," vol. 55, no. 31, pp. 15074-15105, 2020.

[20] R. Wille, B. Li, U. Schlichtmann, and R. Drechsler, "From biochips to quantum circuits: Computer-aided design for emerging technologies," in 2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2016, pp. 1- 6: IEEE.

[21] R. S. Nuvvula et al., "Multi-objective mutation-enabled adaptive local attractor quantum behaved particle swarm optimisation based optimal sizing of hybrid renewable energy system for smart cities in India," vol.

49, p. 101689, 2022.

[22] V. S. Naresh, M. M. Nasralla, S. Reddi, and I. J. S. García- Magariño, "Quantum Diffie–Hellman Extended to Dynamic Quantum Group Key Agreement for e-Healthcare Multi- Agent Systems in Smart Cities," vol. 20, no. 14, p. 3940, 2020.

[23] T. Q. Duong, L. D. Nguyen, B. Narottama, J. A. Ansere, D. Van Huynh, and H. J. I. O. J. o. t. C. S. Shin, "Quantum- inspired Real-time Optimisation for 6G Networks: Opportunities, Challenges, and the Road Ahead," 2022.

[24] Y. Cao et al., "The evolution of quantum key distribution networks: On the road to the qinternet," vol. 24, no. 2, pp. 839- 894, 2022.

[25] N. Wu and J. J. A. S. Sun, "Fatigue Detection of Air Traffic Controllers Based on Radiotelephony Communications and Self-Adaption Quantum Genetic -Algorithm Optimization Ensemble Learning," vol. 12, no. 20, p. 10252, 2022

[26] T. S. Humble, "Consumer Applications of Quantum Computing: A Promising Approach for Secure Computation, Trusted Data Storage, and Efficient Applications," IEEE Consumer Electronics Magazine, 2018.

[27] L. K. Grover, "A Fast Quantum Mechanical Algorithm for Database Search," ArXiv:quant-ph/9605043, 1996.

[28] D. J. Bernstein, "Introduction to Post-Quantum Cryptography," Springer, p. pp. 1–14, 2009.

[29] K. V. M. D. Z. A. J. Vasileios Mavroeidis, "The Im- pact of Quantum Computing on Present," International Journal of Advanced Computer Science and Applications 2018. "Cryptography in Everyday Life," February 2021. [Online]. Available: laits.utexas.edu/ anorman/BUS.FOR/ course.mat/SSim/life.html.

[30] P. J. Paar C., "The Data Encryption Standard (DES) and Alternatives," in Understanding Cryptog- raphy, Springer, 2010, pp. 55-86.

[31] G. Karthikeyan Bhargavan and Leurent, "On the Practical (In-)Security of 64-Bit Block Ciphers: Collision Attacks on HTTP over TLS and OpenVPN," in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Aus- tria, 2016.

[32] NIST, "Advanced Encryption Standard(AES)," Fed- eral Information Processing Standards Publications, 2001.

[33] December 2020. [Online]. Available: https://www.sans. org/reading-room/whitepapers/vpns/paper/1006.

[34] P. J. Paar C., "Public-Key Cryptosystems Based on the Discrete Logarithm Problem," in Understanding Cryptography, Springer, 2010.

[35] P. J. Paar C., "Elliptic Curve Cryptosystems," in Understanding Cryptography, Springer, 2010.

[36] L. Wagner, "Basic Intro To Elliptic Curve Cryp- tography," December 2020. [Online]. Available: https:// qvault.io/2020/09/17/very-basic-intro-to-elliptic-curve- cryptography/.

[37] O. L.D., "Side-Channel Attacks in ECC: A General Technique for Varying the Parametrization of the Ellip- tic Curve," in Lecture Notes in Computer Science, vol 3156, Berlin, 2004.

[38] M. Gilles, 2019. [Online]. Available: https://www.technologyreview.com/2019/01/ 29/66141/what-is-quantum-computing/.

[39] V. Feldman, "Basics of Quantum Mechanics," 2003. [Online]. Available: https://ocw.mit.edu/courses/ math- ematics/ 18-435j-quantum-computation-fall-2003/lecture- notes/qc lec02.pdf.

[40] N. S. Yanofsky, "An Introduction to Quantum Computing," arXiv:0708.0261 [Quant-Ph], 2007. M. G. Kuzyk, "Quantum no-cloning theorem and entangle- A. I. a. N. R. S. Nurhadi, "Quantum Key Distribu- tion (QKD) Protocols: A Survey," in 2018 4th Interna-tional Conference on Wireless and Telematics (ICWT).

[41] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," SIAM Journal on Computing, p. 1484–509, 1997.

[42] J. Hui, "QC — Simon's Algorithm," April 2019. [On- line]. Available: https://jonathan-hui.medium.com/qc- simons-algorithm-be570a40f6de.

[43] E. a. V. U. Bernstein, "Quantum Complexity The- ory," Association for Computing Machinery, p. 11–20, 1993.

[44] D-WAVE, [Online]. Available: https://www. dwavesys.com/resources/publications?type=technology.

[45] [Online]. Available: https://www.cbinsights.com/ research/report/quantum-computing/.

[46] IBM, [Online]. Available: https://www.ibm.com/blogs/ research/2020/09/ibm-quantum-roadmap.