IJHSR
ijhighschoolresearch.org
REVIEW ARTICLE

Post-Quantum Cryptography: Effects of Quantum Computing on Modern Cryptography

David Yang

University of Toronto Schools, 371 Bloor St W, Toronto, Ontario, M5S 2R7, Canada; dandave0407@gmail.com

ABSTRACT: The emergence of a new era of quantum computing is on the horizon, presenting modern cryptography with a cluster of unprecedented threats. This review delves into the cryptographic techniques that risk being challenged in a post-quantum environment and the algorithms that challenge these techniques. We first investigate aspects of traditional cryptography, covering both asymmetric and symmetric cryptography. Then, we analyze quantum computing and its impact on cryptographic techniques, notably Shor's and Grover's algorithms. Finally, we will research quantum-resistant solutions and explore our current status in post-quantum cryptography. This review aims to provide a comprehensive understanding of the threats posed by quantum computing from a cryptographic standpoint and the approaches available to mitigate these risks for a more secure future.

KEYWORDS: Systems Software; Cybersecurity; Cryptography; Quantum Computing; Shor's Algorithm.

Introduction

Amidst an ever-evolving digital communication and data security era, cryptographic techniques have long been the guardians of sensitive information. These techniques play a pivotal role in shielding data against malicious threats and ensuring data's secure and accurate transmission to their intended recipients.[1] For instance, encryption protects financial information like credit card numbers and personal details like driver's licenses and addresses.[2] Yet, no security measure will remain impenetrable in our rapidly advancing technological world forever. Quantum computing, which can significantly reduce calculation times, is advancing quickly, and businesses have already begun attempting to incorporate quantum computing into their projects.[3] For example, Mercedes-Benz is using quantum computing to simulate a more efficient car battery,[4] ExxonMobil is exploring power grid optimizations and the most efficient routes for shipping fuel,[5] and JPMorgan is investigating portfolio and derivatives pricing optimization.[6] This rapid development in quantum computing poses unprecedented challenges to the foundations of traditional cryptography, especially as quantum computing technology inches closer to practical realization. With practical quantum computers, widely used encryption schemes will become vulnerable.[7] This vulnerability will implicate individuals, businesses, and governments, compromising the overall fabric of international security.

Furthermore, there is a significant risk of malicious actors stealing currently encrypted confidential data in the hopes that a quantum computer will be able to break that encryption in the future. As such, there is a need to analyze and assess the cryptographic techniques that run the risk of being challenged in such an environment and identify suitable solutions for the future.[8]

This paper begins by analyzing traditional cryptographic techniques, offering general insight into how they function, and focusing on asymmetric and symmetric cryptography. We will also explore quantum computing and algorithms, with emphasis on the two most relevant algorithms in the industry, Shor's algorithm and Grover's algorithm, and their impacts on traditional cryptographic techniques. Finally, this paper presents potential solutions in the form of quantum-resistant algorithms, emphasizing the post-quantum cryptography that has been researched. Special attention is given to analyzing the differences and trade-offs between traditional cryptographic techniques and post-quantum alternatives such as lattice-based cryptography, hash-based cryptography, code-based cryptography, and multivariate cryptography.

With these considerations, we discuss future directions regarding post-quantum cryptography and its research.

Discussion

Traditional Cryptography:

Cryptography's historical significance spans over two millennia, with Caesar's cipher, a simple substitution cipher, being one of the most well-known historical cryptosystems.[9,10] Initially applied in warfare and diplomatic exchanges,[10] the utility of cryptography evolved alongside technology and telecommunications, rendering it indispensable for safeguarding our information across various domains. It forms the backbone of data protection, ensuring unauthorized parties cannot access or tamper with confidential data. The core objectives of cryptography include ensuring the confidentiality of information, authenticating one's identity, maintaining data integrity, guaranteeing non-repudiation, and enabling secure key exchanges between transacting parties. These objectives are achieved through a multitude of techniques revolving around data transformation, key management, and authentication. The two primary categories of traditional cryptography are asymmetric and symmetric cryptography.[9]

Asymmetric Cryptography:

Asymmetric cryptography, also known as public-key cryptography, secures communication and information using pairs

DOI: 10.36838/v6i6.3

of keys - one public and one private. A public key functions as a lock accessible to all parties. It operates unidirectionally, permitting data encryption while preventing access without the corresponding private key. On the other hand, the private key functions as a matching key to the public lock. Knowing this key is the only way to open the public key's lock and access the desired information. For example, if your friend wants to send you an encrypted message via asymmetric cryptography, they would first use your public key to encrypt their message (plaintext). Then, they would send you the encrypted message (ciphertext), and you could decrypt it with your private key. This method enables secure communication between parties with minimum risk of the message being tampered with.[9-11]

RSA (Rivest-Shamir-Adleman):

RSA is a widely used asymmetric cryptographic algorithm that relies on the mathematical properties of prime numbers to provide secure communication, digital signatures, and key exchange without the need for a shared secret key.

The RSA algorithm involves two keys (one public and one private). The public key consists of a modulus N and exponent e, and the private key consists of a single integer d. The public key is known to everyone, but only the message recipient should know the private key. To encrypt a message, the sender first converts their message to an integer m using some reversible scheme (e.g., using ASCII bytes). This integer must be less than the public key modulus N. Otherwise, the message must be broken into chunks before transmission. Then, the sender computes the ciphertext

$$c = m^e \ (mod\ N)$$

and sends this to the recipient.[13]

To decrypt the ciphertext, the recipient computes the following using their private key

$$m = c^d \ (mod\ N)$$

But how do we get any of N, e, and d in the first place? We must select them in a very particular manner. In the original RSA algorithm, we first choose random large primes p and q.

1. Compute the modulus N as $N = pq$. N is also known as a semiprime (product of two primes). In cryptography, we often refer to the size of numbers in bits, that is, the number of binary digits needed to represent the number. In this case, N is a b-bit semiprime, where $b=[log_2]+1$.
2. Compute $\varphi(N) = (p-1)(q-1)$, where $\varphi(N)$ is the Euler Totient function.
3. Select the exponent e such that $gcd(e,\varphi(N)) = 1$.
4. Compute the private key d such that $de \equiv 1\ (mod\ \varphi(N))$.

Why exactly does this work? By Euler's Totient Theorem,[12]

$$m\varphi(N) \equiv 1\ (mod\ N)$$

Since we selected d such that $de \equiv 1\ (mod\ \varphi(N))$, we have that

$$m^{de} = m^{k.\varphi(N)+1} = m \cdot m^{k.\varphi(N)} \equiv m \cdot 1^k\ (mod\ N)$$

which proves the message decryption scheme.[13]

Symmetric Cryptography:

In contrast to asymmetric techniques, symmetric cryptography uses a single secret key for encryption and decryption. The sender and receiver share this key, which must be kept secure from unauthorized parties. For encryption, the sender takes the original message in plaintext and uses the secret key alongside an encryption algorithm to transform it into ciphertext. The receiver uses the same secret key alongside a decryption algorithm to convert the ciphertext back into the original plaintext. This process can only be done with the correct secret key shared by both parties; as a result, symmetric cryptographic methods are often difficult to scale, and secure distribution of the key to the intended recipients is essential in maintaining security.[9]

How Quantum Computers Work:

Conjectured in part by Richard Feynman in 1982,[14] the quantum computer has received massive amounts of research and development. They are known for having much faster processing speed compared to traditional computers. Quantum computing creates a new way to solve some complex issues, whereas conventional computing is often stumped due to the computational complexity of known solutions.[15]

The main property that enables this is known as superposition. In traditional computers, information is represented as binary bits (a series of 0s and 1s); quantum computers, on the other hand, use qubits. Qubits have binary states (0 and 1) just like regular bits but can also exist in superposition states between the two. This means that while sixteen bits on a traditional computer can only represent one number between 0 and $2^{16}-1=65,535$, a superposition of sixteen qubits can simultaneously represent all 65,536. In particular, the state of a quantum system with 16 qubits is effectively a 2^{16}-dimensional vector representing a superposition of all the possible classical states. Note that this is not exactly akin to classical parallel computing - the real power in quantum computing lies in being able to manipulate these superposition states through quantum interference and entanglement.[16]

Quantum interference allows us to constructively or destructively interfere with amplitudes of certain states in the superposition, effectively reinforcing certain states while diminishing others. On the other hand, entanglement allows qubits to be "linked" with other qubits, regardless of the distance between them (often referred to as 'spooky action at a distance'). These properties enable complex correlations between qubits that cannot easily be replicated in classical systems.[16]

To ensure their stability, quantum computers operate at temperatures just above absolute zero, approximately -273 degrees Celsius, often using supercooled superfluids for cooling.[17] While these extreme requirements have limited the practical construction of quantum computers, this has continued the theoretical development of quantum algorithms, which has seen much activity over the past two decades. Currently, two fundamental algorithms expected to improve their respective best-known classical solutions are Shor's and Grover's algorithms, which we will discuss in detail in this section.

Shor's Algorithm:

Recall RSA, as described earlier. The crux of the security assured by RSA is in the current inability of classical computers to efficiently retrieve p and q from N (i.e., prime factor N). Since N is a part of the public key, we have direct access to it, and prime factoring allows us to calculate the private key d, which is used for decryption.[13] Shor's algorithm greatly speeds up the process of finding the prime factors p and q of a large semiprime N by taking advantage of the properties of quantum superposition.

The mathematical underpinnings of Shor's algorithm can be described as follows: Suppose that, for some positive integer a, we find an exponent p such that

$$a^p \equiv 1 \ (mod\ N)$$

Then,

$$a^p - 1 = (a^{p/2} - 1)\ (a^{p/2} + 1) \equiv 0 \ (mod\ N),$$

which means that

$$(a^{p/2} - 1)\ (a^{p/2} + 1) = k \cdot N \text{ for some } k \in Z_+$$

Then, we can use the Euclidean algorithm to identify if either $a^{p/2} \pm 1$ share nontrivial factors with N (i.e., not 1 or a multiple of N itself). Recall that the Euclidean algorithm allows us to find the greatest common divisor between two numbers in logarithmic time.[18] If they do share nontrivial factors, then we would have successfully prime factored N. Finding such an exponent p is quite difficult classically. However, we can make use of the fact that the sequence

$$a_n(k) = k^n \ (mod\ N)$$

repeats with a period that is a factor of $\varphi(N)=(p-1)(q-1)$. In particular, if $gcd(k,N) = 1$, this period p (also known as the multiplicative order of k mod N) satisfies $g^p \equiv 1 \ (mod\ N)$ and then we may attempt the above factorization.[19]

We have to introduce the (Quantum) Fourier Transform to see why this is useful. Classically, the Fourier Transform takes a set of points and decomposes them into frequencies - it is often used in applications like signal processing and solving differential equations. The Quantum Fourier Transform instead operates on qubits. Essentially, it computes a new superposition state representing the input frequencies.[20,21] This is important as we had just reduced the factorization problem to finding a cycle length. A high-level overview of Shor's algorithm goes as follows:

1. First, we pick a random starting guess X and initialize a uniform superposition of exponents A.
2. Then, we compute the remainder modulo N as our random guess is raised to each power. At this point, we have a uniform superposition of pairs $(A, X^A \ mod\ N)$.
3. Now, we observe the second register (say, R), which collapses the state into a uniform superposition of states with the remainder R. The key is that now, due to entanglement, the superposition in the first register is of states that correspond to the observed state in the second register.
4. Next, using the Quantum Fourier Transform on the first register, we can extract the frequency (and thus, period p) of the exponent cycles.
5. If p is even, we can check if $gcd(N, a^{p/2}-1)$ is nontrivial (i.e., not equal to 1 or N). If so, we have successfully factored N.

Shor's algorithm reduces the time complexity of factoring N, a b-bit semiprime, to $O(b^3)$, which is an exponential improvement over the current best-known classical algorithm, the general number field sieve. This renders RSA useless in the face of a good practical quantum computer, at least for general applications with reasonable key sizes.[22,23]

Shor's algorithm can also solve the discrete logarithm problem (given a generator $g \in G$ and element $h \in G$ of a finite group, finding some integer x such that $g^x = h$) in polynomial time (in $log|G|$). As such, it can be used to break other asymmetric cryptographic schemes, such as the Diffie-Hellman key exchange and Elliptic-curve cryptography, which both depend on the hardness of the discrete logarithm problem.[22,23]

Currently, Shor's algorithm is not practically implementable in any quantum computer. The factorization record for a quantum computer using Shor's algorithm has remained at 21 over the past decade.[24,25] A major bottleneck for integer factorization on quantum computers is the stability of large quantum systems and susceptibility to errors - it is estimated that factoring a 2048-bit RSA integer would take on the order of millions of qubits, many orders of magnitude greater than state-of-the-art quantum computers today.[26]

Grover's Algorithm:

Grover's algorithm aims to solve unstructured search problems, where you are trying to find a particular item from a set without any information about the structure of the set. Suppose you have a set of N identical closed boxes, each containing a different book. Now, if you were to go looking for a specific book, you would have to open, on average, $N/2$ of these boxes. On a quantum computer, however, such a search can be performed in $O(\sqrt{N})$ steps using Grover's algorithm; in fact, it has been proved that this is the optimal time complexity possible on a quantum computer.[27]

To apply Grover's algorithm, we first need to create an "oracle" function, which identifies the desired element. In particular, the oracle flips the amplitude of the solution state and leaves other states unchanged. Applying this oracle function allows us to take advantage of interference between quantum states and destructively interfere with the non-solution states.

We then begin with a superposition over all function input states. Then we repeatedly perform the following steps around $\pi\sqrt{N}/4$ times:

1. Apply the oracle function, which only flips the amplitude of the solution state.
2. Apply the Grover Diffusion Operator, which inverts the superposition state about the mean of the amplitudes. This amplifies the flipped solution state and weakens all the non-solution states, as shown in Figure 1.

At this point, measuring the superposition collapses it to the desired solution state with extremely high probability, giving us an $O(\sqrt{N})$ overall search complexity

DOI: 10.36838/v6i6.3

Grover's algorithm allows quadratic speedup for function inversion problems, which is particularly relevant to symmetric encryption algorithms and hashing methods such as AES or SHA. However, since this is not an exponential improvement, it can be easily mitigated by increasing key lengths.[28,29] Currently, Grover's algorithm has only been tested in low search spaces ($N \leq 32$), with results that deviated significantly from simulations due to noise - as a result, tackling search problems many orders of magnitude larger (like current hashing methods) is by far computationally infeasible unless significant advances are made in quantum computing architecture (Figure 1).[30]

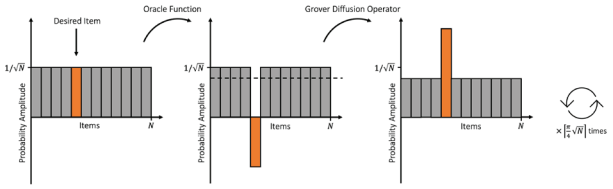

Figure 1: Plots of superposition probability amplitudes during the initial steps of Grover's algorithm. The desired solution state (marked in orange) is thoroughly amplified after multiple iterations (Figure inspired by IBM's Qiskit Textbook and made in Microsoft PowerPoint).[29]

Harvest Now, Decrypt Later:

Even though practical quantum computers have yet to be developed, the mere concepts of algorithms such as Shor's and Grover's algorithms pose significant threats to our data today. The harvest now, decrypt later strategy describes the collection of currently encrypted data with the idea of potentially decrypting this data using quantum computers.[31]

Lots of data have long-term value (i.e., passwords, identification information, national secrets, financial transactions, etc.) and would be valuable in the hands of an adversary. This is part of the reason for the push for post-quantum cryptographic standards. These standards are being designed not only to protect new communications but also to ensure the safety of historical data. It is important to note that little can be done to secure data that has already been taken. Therefore, the danger the harvest now, decrypt later strategy poses makes it of utmost importance to shift to post-quantum cryptographic techniques as soon as possible.[31]

Limitations of Quantum Computing:

Despite its promise, developing a working large-scale quantum computer requires the resolution of many technical challenges. One of the critical aspects of a reliable quantum computer is stability—quantum decoherence results in alterations in the information in a quantum system as a result of environmental interactions, so the external environment of a quantum computer must be very accurately maintained. On top of this, increasing the computational power of a quantum computer (i.e., entangling additional qubits into a quantum system) increases its fragility, making it challenging to scale quantum computers. Furthermore, the probabilistic and unstable nature of quantum computing necessitates using an error correction method, such as additional "error-correcting" qubits, as shown in Figure 2.[32]

In classical information transmission, errors may arise from random bits flipping (from 0 to 1 and vice versa). For classical systems, the simplest way to correct these errors involves redundancy - sending multiple copies of the same bit and then having the receiver take the majority element it receives. However, quantum error correction faces a few additional challenges: phase errors may corrupt the information stored in qubits instead of bit flips.[33] Furthermore, quantum error correction cannot involve copying quantum states, as this is impossible by the no-cloning theorem.[34] However, it is still possible to spread that state of a qubit onto an entangled state of other qubits; for instance, in 1995, Shor presented a method to store an arbitrary state of n qubits with $9n$ qubits in an error-resistant manner.[35]

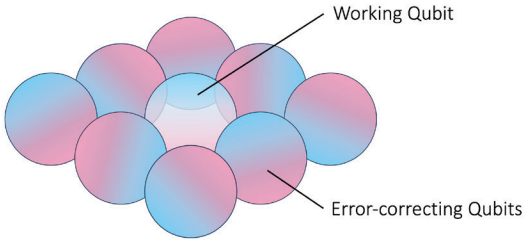

Figure 2: Depiction of the use of auxiliary qubits in error correction for quantum computers. Such a system for error correction ensures increased correctness but reduces scalability (Figure inspired by Trabesinger, 2017, and made in Microsoft PowerPoint).[32]

Solutions: Post-Quantum Cryptography:

Post-quantum cryptography refers to a new field of cryptographic algorithms focused on resistance to quantum algorithms such as Shor's and Grover's.[36] Various organizations are currently researching and testing quantum-resistant cryptographic techniques, and various groups are implementing standardization initiatives to ensure these schemes' applicability in industrial settings.[37] One of the best-known organizations involved is the National Institute of Standards and Technology (NIST). Currently, NIST is leading the standardization process for post-quantum cryptography. After calling for submissions and establishing a standard for traditional cryptography in the 20th century, they have once again called for submissions to tackle the new threat of quantum cryptographic algorithms.[38] The NIST standardization process has undergone three rounds of judging and has selected the four most secure algorithms: CRYSTALS-KYBER, CRYSTALS-Dilithium, Falcon, and SPHINCS+. We are now in the fourth round, and four alternative algorithms are being judged.[39,40]

Lattice-based Cryptography:

Of the current post-quantum cryptographic techniques researched, lattice-based cryptography is the most used.[37] Lattice-based cryptography, as the name suggests, is based upon lattices, which are generated from the set of all possible points reached by an integral linear combination of a set of basis vectors. An example of a simple two-dimensional lattice is shown in the Figure 3 below.

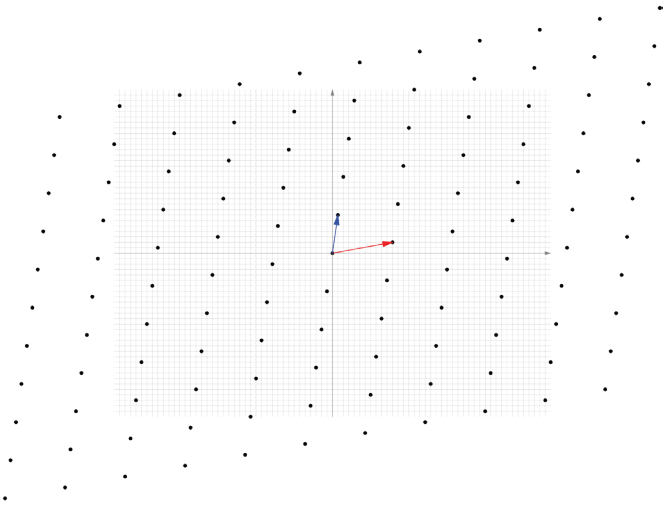

Figure 2: Graph of a two-dimensional lattice generated by the basis vectors (11,2) (red) and (1,7) (blue) (Figure made using Asymptote in LaTeX).

In Figure 3, the red and blue vectors are known as the basis vectors, which are the key to generating our lattice. One important fact to note is that different sets of basis vectors can generate the same lattice. Furthermore, note that lattices can exist in arbitrarily many dimensions as we increase the dimension of our basis vectors. Lattice-based cryptography is based on the hardness of certain lattice problems - one of the most basic is the shortest vector problem (SVP). To solve SVP, we want to find the closest nonzero vector on the lattice to a given point.[41] Many lattice-based encryption schemes, such as Goldreich-Goldwasser-Halevi (GGH), use this problem. The GGH lattice-based algorithm was broken in 1999,[42] but it highlights the methods used by most lattice-based algorithms. It uses 'good' and 'bad' basis vectors to form the same lattice plot. A good basis is one where the vectors are closer to perpendicular, and a bad basis is one where they are closer to parallel. SVP is much easier to solve on a good than a basis because the parallel vectors nearly cancel out, making calculations more difficult.

To explain how the GGH encryption scheme works, we can assume a hypothetical scenario. Let's say Alice wants to send Bob a message. Bob will set up a lattice with two bases, one with the good basis and one with the bad basis. Bob will keep the good basis to himself as his private key while telling Alice the bad basis as the public key. Alice will embed a message using the public key (bad basis). A point on the lattice plot will be chosen, and when Alice sends the message, another point nearest to that first point will be selected. Since Bob knows the good basis, it will be easy for him to decrypt the message, allowing him to access the information while others are in the dark.[43]

Although the GGH encryption scheme is not secure, lattice-based cryptography has been proven to be quantum-resistant. Three of the four algorithms set for standardization by NIST are based on it: Kyber, Dilithium, and Falcon. Kyber is valuable for generic encryption when accessing secure online platforms and is based on the shortest vector problem. On the other hand, Dilithium and Falcon demonstrate their utility in digital signature applications by confirming identities during digital transactions. These are all based on structured lattice problems.[39] In addition to these standardized algorithms, homomorphic encryption (HE) is another quantum-resistant cryptographic technique. Lattice-based cryptography forms the basis of many HE schemes, a type of encryption that allows specific computations to be performed directly on encrypted data without needing first to decrypt it. When decrypted, the results of these computations match those that would have been obtained if they were performed on the unencrypted data.[44,45]

Other Cryptographic Approaches:

Apart from lattice-based cryptography, some of the cryptographic algorithms being researched for their quantum resistance also include hash-based cryptography, code-based cryptography, multivariate cryptography, and isogeny-based cryptography.[46]

Hash-based cryptography utilizes secure hash functions, which transform data into a fixed-size hash, a process that is easy to compute but hard to reverse, ensuring data integrity. Code-based cryptography constructs cryptographic systems using the principles of error-correcting codes, exploiting the hardness of decoding a general linear code.[47] Multivariate cryptography secures data by relying on the computational difficulty of solving multivariate polynomial equations over finite fields.[48] Isogeny-based cryptography employs the properties of isogenies between elliptic curves to create secure cryptographic protocols, a method considered resistant to quantum computing attacks.[49]

■ Conclusion

Despite already exploring post-quantum cryptography in the past years, more research is needed to be fully prepared for when cryptographically relevant quantum computers become available. The current highly considered algorithms should be continually improved, and ways should be regarded to make them more robust and flexible. This will allow us to remain adaptable to changes and potentially substitute algorithms in situations where they no longer ensure the targeted level of security. Moreover, the resistance of many cryptosystems to attacks and their implementation security has yet to be thoroughly examined. For example, there is a need for comparisons of the security provided by structured and unstructured lattices. This further research and testing will ensure the reliability of our post-quantum cryptographic techniques.[7]

It is quite clear that the cryptographic techniques we have relied on in the past few decades are becoming increasingly vulnerable in the evolving era of quantum computing. This paper has examined the impact quantum computing has on traditional cryptography and highlighted the need to address the challenges posed. While our conventional cryptographic methods may be on the verge of obsolescence, the emergence of post-quantum cryptography offers a promising avenue for sustaining the security of our world. This paper understands the present state of quantum-resistant algorithms and their research in the face of quantum. It suggests that tech companies transition to quantum-safe cryptography in the foreseeable fu-

DOI: 10.36838/v6i6.3

ture. We will be able to achieve a fully secure digital future for everyone as long as research continues to be done to strengthen software-based approaches like post-quantum cryptography so organizations can easily migrate to a fully quantum-safe cryptographic state.

Acknowledgments

I would like to thank the 'Iris Research Program' for allowing me to write this research paper. I would also like to express sincere gratitude to my two mentors in the program, Dr. Sakk and Tim Adamson, who guided me through this process. Finally, I would like to extend my gratitude to Sara Ahmed, who helped in the editing process.

References

1. Sharma, D. K.; Singh, N. C.; Noola, D. A.; Doss, A. N.; Sivakumar, J. A Review on Various Cryptographic Techniques & Algorithms. *Materials Today: Proceedings* **2022**, *51*, 104–109. DOI: 10.1016/j.matpr.2021.04.583.
2. Banasode, P.; Padmannavar, S. Protecting and Securing Sensitive Data in a Big Data Using Encryption. *EAI Endorsed Transactions on Smart Cities* **2020**, *4* (11), e5. DOI: 10.4108/eai.13-7-2018.163991.
3. Shipilov, A. *Making quantum computing a reality*. Harvard Business Review. https://hbr.org/2022/04/making-quantum-computing-a-reality (accessed 2023-12-23).
4. IBM. *Envisioning a new wave in power*. Daimler | IBM. https://www.ibm.com/case-studies/daimler (accessed 2023-12-23).
5. Olatunji, O. O.; Adedeji, P. A.; Madushele, N. Quantum Computing in Renewable Energy Exploration: Status, Opportunities, and Challenges. In *Elsevier eBooks*; 2021; pp 549–572. DOI: 10.1016/b978-0-12-824555-2.00019-8.
6. Kanamori, Y.; Yoo, S.-M. Quantum Computing: Principles and Applications. *Journal of International Technology and Information Management* **2020**, *29* (2), 43–71. DOI: 10.58729/1941-6679.1410.
7. Barker, W.; Polk, W.; Souppaya, M. *Getting ready for post-quantum cryptography: Exploring challenges associated with adopting and using post-quantum cryptographic algorithms* **2021**. DOI: 10.6028/nist.cswp.04282021.
8. Martin, K. M. *Cryptography: The Key to Digital Security, How It Works, and Why It Matters*; W. W. Norton & Company, Inc., 2021.
9. Barakat, M.; Eder, C.; Hanke, T. *An Introduction to Cryptography*. 2018. https://agag-ederc.math.rptu.de/~ederc/download/Cryptography.pdf.
10. Luciano, D.; Prichett, G. Cryptology: From Caesar Ciphers to Public-Key Cryptosystems. *The College Mathematics Journal* **1987**, *18*(1), 2–17. DOI: 10.1080/07468342.1987.11973000.
11. Abd Zaid, M.; Hassan, S. Survey on Modern Cryptography. *Journal of Kufa for Mathematics and Computer* **2021**, *7* (1), 1–8. DOI: 10.31642/jokmc/2018/070101.
12. Wolfram Research, Inc. *Euler's Totient Theorem -- from Wolfram MathWorld*. Wolfram MathWorld. https://mathworld.wolfram.com/EulersTotientTheorem.html (accessed 2023-12-23).
13. Rivest, R. L.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. 1978. DOI: 10.21236/ada606588.
14. Feynman, R. P. Simulating Physics with Computers. *International Journal of Theoretical Physics* **1982**, *21* (6–7), 467–488. DOI: 10.1007/bf02650179.
15. Hey, A. *Feynman and Computation*; CRC Press, 2018.
16. Gael, J. V. The Role of Interference and Entanglement in Quantum Computing. Master's Thesis, University of Wisconsin-Madison, Madison, WI, 2005. https://pages.cs.wisc.edu/~jvangael/pubs/msthesis.pdf (accessed 2023-12-23).
17. Charbon, E.; Sebastiano, F.; Vladimirescu, A.; Homulle, H.; Visser, S.; Song, L.; Incandela, R. M. Cryo-CMOS for Quantum Computing. *2016 IEEE International Electron Devices Meeting (IEDM)* **2016**. DOI: 10.1109/iedm.2016.7838410.
18. Wolfram Research, Inc. *Euclidean Algorithm -- from Wolfram MathWorld*. Wolfram MathWorld. https://mathworld.wolfram.com/EuclideanAlgorithm.html (accessed 2023-12-23).
19. Wolfram Research, Inc. *Multiplicative Order -- from Wolfram MathWorld*. Wolfram MathWorld. https://mathworld.wolfram.com/MultiplicativeOrder.html (accessed 2023-12-23).
20. Lanzagorta, M.; Uhlmann, J. The Quantum Fourier Transform. In *Quantum Computer Science*; Springer Cham, 2008; pp. 73-82. https://doi.org/10.1007/978-3-031-02512-9.
21. IBM Quantum Learning. *Quantum Fourier Transform*. https://github.com/Qiskit/textbook/blob/main/notebooks/ch-algorithms/quantum-fourier-transform.ipynb (accessed 2023-12-23).
22. Shor, P. W. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing* **1997**, *26* (5), 1484–1509. DOI: 10.1137/s0097539795293172.
23. IBM Quantum Learning. *Shor's algorithm*. https://github.com/Qiskit/textbook/blob/main/notebooks/ch-algorithms/shor.ipynb (accessed 2023-12-23).
24. Martín-López, E.; Laing, A.; Lawson, T.; Álvarez, R. F. P.; Zhou, X.; O'Brien, J. L. Experimental Realization of Shor's Quantum Factoring Algorithm Using Qubit Recycling. *Nature Photonics* **2012**, *6* (11), 773–776. https://doi.org/10.1038/nphoton.2012.259.
25. Skosana, U.; Tame, M. Demonstration of Shor's Factoring Algorithm for N=21 on IBM Quantum Processors. *Scientific Reports* **2021**, *11* (1). DOI: 10.1038/s41598-021-95973-w.
26. Gidney, C.; Ekerå, M. How to Factor 2048 Bit RSA Integers in 8 Hours Using 20 Million Noisy Qubits. *Quantum* **2021**, *5*, 433. DOI: 10.22331/q-2021-04-15-433.
27. Bennett, C. H.; Bernstein, E.; Brassard, G.; Vazirani, U. Strengths and Weaknesses of Quantum Computing. *SIAM Journal on Computing* **1997**, *26* (5), 1510–1523. DOI: 10.1137/s0097539796300933.
28. Grover, L. K. A Fast Quantum Mechanical Algorithm for Database Search. *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC '96* **1996**. DOI: 10.1145/237814.237866.
29. IBM Quantum Learning. *Grover's algorithm*. https://github.com/Qiskit/textbook/blob/main/notebooks/ch-algorithms/grover.ipynb (accessed 2023-12-23).
30. Kaderi, Y. E.; Honecker, A.; Andriyanova, I. Performance of Uncoded Implementation of Grover's Algorithm on Today's Quantum Processors. *2023 IEEE Information Theory Workshop (ITW)* **2023**, 209–214. DOI: 10.1109/itw55543.2023.10160239.
31. Kong, I. Transitioning towards Quantum-Safe Government. *15th International Conference on Theory and Practice of Electronic Governance* **2022**. DOI: 10.1145/3560107.3560182.
32. Trabesinger, A. Quantum Leaps, Bit by Bit. *Nature* **2017**, *543* (7646). DOI: 10.1038/543s2a.
33. Mandelbaum, R.; Steffen, M.; Cross, A. *Error correcting codes for near-term quantum computers*. IBM Research Blog. https://research.ibm.com/blog/error-correction-codes (accessed 2023-12-23).
34. Chang, H.-H. An Introduction to Error-Correcting Codes: From Classical to Quantum. *arXiv (Quantum Physics)* **2006**. DOI: 10.48550/arXiv.quant-ph/0602157.

35.Shor, P. W. Scheme for Reducing Decoherence in Quantum Computer Memory. *Physical Review A* **1995**, *52* (4), R2493–R2496. DOI: 10.1103/physreva.52.r2493.
36.Bernstein, D. J.; Buchmann, J.; Dahmen, E. Post-Quantum Cryptography. *Nature* **2017**, *549* (7671), 188–194. DOI: 10.1038/nature23461.
37.European Union Agency for Cybersecurity, Bernstein, D. J.; Hülsing, A.; Lange, T. Post-Quantum Cryptography: Integration Study. *Publications Office of the European Union* **2022**. DOI: 10.2824/151162.
38.Dam, D.-T.; Tran, T.-H.; Hoang, V.; Pham, C.; Hoang, T.-T. A Survey of Post-Quantum Cryptography: Start of a New Race. *Cryptography* **2023**, *7* (3), 40. DOI: 10.3390/cryptography7030040.
39.Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, U.S. Department of Commerce. *Announcing PQC Candidates to be Standardized, Plus Fourth Round Candidates* | CSRC. https://csrc.nist.gov/news/2022/pqc-candidates-to-be-standardized-and-round-4 (accessed 2023-12-23).
40.Alagic, G.; Apon, D.; Cooper, D.; Dang, Q.; Dang, T.; Kelsey, J.; Lichtinger, J.; Liu, Y.-K.; Miller, C.; Moody, D.; Peralta, R.; Perlner, R.; Robinson, A.; Smith-Tone, D. *Status report on the third round of the NIST Post-Quantum Cryptography Standardization Process* **2022**. DOI: 10.6028/nist.ir.8413.
41.Micciancio, D.; Regev, O. Lattice-Based Cryptography. *Post-Quantum Cryptography* **2009**, 147–191. DOI: 10.1007/978-3-540-88702-7_5.
42.Nguyen, P. Cryptanalysis of the Goldreich-Goldwasser-Halevi Cryptosystem from Crypto '97. *Advances in Cryptology — CRYPTO' 99* **1999**, 288–304. DOI: 10.1007/3-540-48405-1_18.
43.Mandangan, A.; Kamarulhaili, H.; Asbullah, M. A. On the Underlying Hard Lattice Problems of GGH Encryption Scheme. *Cryptology and Information Security Conference* **2018**, 288–304.
44.Yi, X.; Paulet, R.; Bertino, E. Homomorphic Encryption. In *Homomorphic Encryption and Applications*; Springer Cham, 2014; pp 27–46. DOI: 10.1007/978-3-319-12229-8_2.
45.Kadykov, V.; Levina, A.; Voznesensky, A. S. Homomorphic Encryption within Lattice-Based Encryption System. *Procedia Computer Science* **2021**, *186*, 309–315. DOI: 10.1016/j.procs.2021.04.149.
46.Balamurugan, C.; Singh, K.; Ganesan, G.; Rajarajan, M. Post-Quantum and Code-Based Cryptography—Some Prospective Research Directions. *Cryptography* **2021**, *5* (4), 38. DOI: 10.3390/cryptography5040038.
47.Huffman, W. C.; Kim, J.-L.; Solé, P. Code-Based Cryptography. In *Concise Encyclopedia of Coding Theory*; Chapman & Hall/CRC, 2021; pp. 799-822.
48.Dey, J.; Dutta, R. Progress in Multivariate Cryptography: Systematic Review, Challenges, and Research Directions. *ACM Computing Surveys* **2023**, *55* (12), 1–34. DOI: 10.1145/3571071.
49.Peng, C.; Chen, J.; Zeadally, S.; He, D. Isogeny-Based Cryptography: A Promising Post-Quantum Technique. *IT Professional* **2019**, *21* (6), 27–32. DOI: 10.1109/mitp.2019.2943136.

Author

David Yang is an 11th-grade student studying at the University of Toronto Schools in Toronto, Canada. He is interested in mathematics and computer science, particularly on cryptography and AI systems. He wishes to leave an impact on this world.

DOI: 10.36838/v6i6.3