EasyChair Preprint

№ 14978

Decrypting the Future: Quantum Computing and The Impact of Grover's and Shor's Algorithms on Classical Cryptography

Aryan Sahni and Ridhima Sahni

September 21, 2024

Decrypting the Future: Quantum Computing and The Impact of Grover's and Shor's Algorithms on Classical Cryptography

Anonymous

Abstract. This paper aims to provide a direct analysis of the capabilities of quantum computation algorithms, particularly the Shor and Grovers algorithm on the basis of their time complexity and brute force capability. Shor's algorithm provides us with the capability to find out the prime factors of large primes exponentially faster than with classical systems. This proves to be a threat to the classical cryptosystems of date that rely on the incapability of the classical algorithms to calculate the prime factors of large primes. Grover's algorithm provides us with a quadratic speed up in the search capabilities of the computer systems which will have a major impact in brute forcing capabilities of the cryptosystem keys and hashes. We have also analysed the implications of these algorithms to the classical cryptosystems existing today and any major areas of improvement in the security algorithms that can be done to make them more secure.

Keywords: Quadratic speedup, exponential speedup, Shor's algorithm, Grover's algorithm, search capabilities, brute force.

1 Introduction

The advent of quantum computing has brought with itself a unique set of problems related to the security of the all-important systems in the world as we know it. The algorithms leading the charge are Shor's and Grover's. The Shor algorithm relies on the predictive power of the algorithm when it comes to generating prime factors of a large prime which classical computers have a really hard time computing. Grover's algorithm relies on the search capabilities of the quantum computers and the varied states that the qubits can exist in as a basis to provide a quadratic speed up to the classical searching algorithms prevalent today. The charge of these two algorithms in the cryptographic world have made them a direct threat to all the most popular algorithms that exist today and provides a way for bad actors to crumple the entire chain of security that major services rely on so passionately. Cryptography relies on keys that secure your data from the rest of the world such that a third party does not get access to your data and only those you want can access that data the very simple baseline method that people have come up with is a simple solution being used to secure our personal space for a long time a lock and key system wherein the user can lock their data via a secret key (encrypt) and only the people with a key same as that key or a key that is

also designed to open that lock can access that information. Forming the basis for symmetric and asymmetric cryptosystems. The backbone holding these systems together is the key and encrypting that key has become a whole new field of study. There is a severe need to switch to the post-quantum alternative with the rapid progress in quantum computation technology so as to secure our future from the bad actors who may try to steal our information and use it for unethical purposes. It is emphasized that quantum algorithms like Shor's and Grover's pose significant threats to both symmetric and asymmetric cryptosystems. Shor's algorithm can efficiently break RSA, ECC, and other public-key cryptosystems based on integer factorization and discrete logarithms, while Grover's algorithm reduces the security level of symmetric cryptography (such as AES) by effectively halving the key length. (gidney C, 2019).

1.1 Shor's Algorithm

The standing logic to make these keys unique and protected is a mammoth task and the asks us to create a trapdoor logic where it is easy for one side to synthesize and difficult to impossible in their lifetime for the other side to break. Shor's algorithm targets that trapdoor system that is responsible for protecting a lot of networks and sensitive information that is multiplying two primes is easy but finding out the prim factors that were used to synthesize that number becomes extremely difficult. Shor's algorithm gives us a way of estimating a better guess 33.75 percent of the time that we start with a bad guess as to what the prime factor could be. The algorithm chooses a very unlikely prime and transforms it into a better guess that may strike the target. According to Euclidean geometry to find a factor of a number say 'N' We don't need to find a pure factor of N we can just find a number that shares a factor with N. There the process goes as follows we take our unlikely prime and raise it to a power say p such that it satisfies Shor's equation, (Hayward, 2008)

$$g^p = m * N + 1$$

then it also satisfies the expression, (Hayward, 2008)

$$g^{\frac{p}{2}} \pm 1$$

which has a very high probability to share factors with N. To begin the quantum computation we set a quantum computer in such a way that we send in a superposition of numbers and the computer raises the superposition to all possible powers it can be raised to (g^x) we keep track of all the powers and the number subsequently obtained the pass the result through another quantum computer that checks how much greater the value is to a multiple of N($m * N$). The crux of the Shors algorithm is to destructively interfere with all the non-answers and only get the answers that satisfy our conditions. Now P in this case has a repeating property such that, (Hayward, 2008)

$$g^x\ ,\ g^x + p\ ,\ g^x - p$$

will all have the same remainder. we take all the answers obtained and scan them for the same remainder then pass all of those observations through a quantum Fourier transform that gives us the frequency of the series. Since all possible answers repeat with the same time period that is P we get the value of $\frac{1}{P}$. We inverse the value obtained and pass it through some checks if the number is even we continue else we run the whole process again. We check if the numbers sharing factors with N that are, (Hayward, 2008)

$$g^{\frac{p}{2}} \pm 1$$

are not multiples of N else we run the process again if it passes all checks then the number is sure to share a factor with N. We then use the Euclid method to find the other and can finally decrypt the message.

1.2 Grover's Algorithm

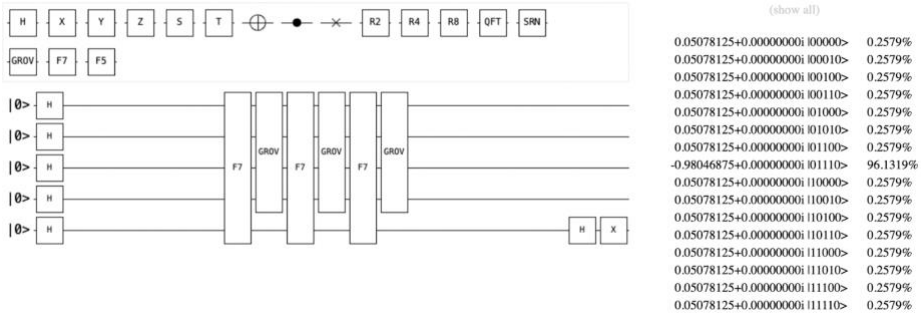

Fig. 1. The following figure gives a simulation of the capabilities of Grovers algorithm.

Grover's algorithm is a search algorithm developed by Lou Grover in the year 1996 that provides a quadratic speed up to the alternative linear search algorithms that exist in the classical space. Like the Deutsch Jozsa algorithm, it uses qubits in superposition in alternative universes to make computations the Grover algorithm does the same but for search-orientated applications. The algorithm relies on an oracle function that returns 1 for all the inputs that are desired and 0 for all the undesired inputs. The algo works step-wise as illustrated in the given image. Firstly during the initialization phase, The qubits first get passed through a Hadamard gate that creates a superposition of all the possible states of the qubits. For n qubits the number of states created is 2^n. All the states created get initialized to values with equal amplitude and the expectation of any state on a search is the same. (Jozsa, 1999)

$$|s> = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} |x>$$

f7 blocks indicate the oracle function that checks the states of the qubits and flips the phase of the input that is of the desired value. (Jozsa, 1999)

$$U_f \left| x \right> = (-1)^{f(x)} \left| x \right>$$

After the Oracle marks the solution, Grover's diffusion operator amplifies the probability amplitude of the correct state. This operator performs the inversion about the average operation, which increases the amplitude of the solution while decreasing the amplitudes of all other states. (Jozsa, 1999)

$$U_s = 2 * |s\rangle\langle s| - 1$$

$$\left|\Psi_{t+1}\right> = U_s U_f \left|\Psi_t\right>$$

The image shows three applications of the Grover diffusion operator, which is typically applied iteratively to amplify the correct state. (Jozsa, 1999)

$$\left|\Psi_t\right> = \left(U_s U_f\right)^{\mathrm{t}} \left|\Psi_{\mathrm{O}}\right>$$

The number of iterations depends on the size of the search space and is typically $O(\sqrt{N})$, where N is the number of states. The steps are repeated in a cycle to drastically amplify the strength of the required state of the qubits. The list on the right shows the resulting probability distribution of different quantum states after running the algorithm. You can see one state that is 01110 has a significantly higher probability than the others (around 96.1319%), indicating it is the marked state.

2 Comparative Analysis: Quantum Algorithms vs. Classical Cryptography

2.1 Shor's Algorithm

Time Complexity.

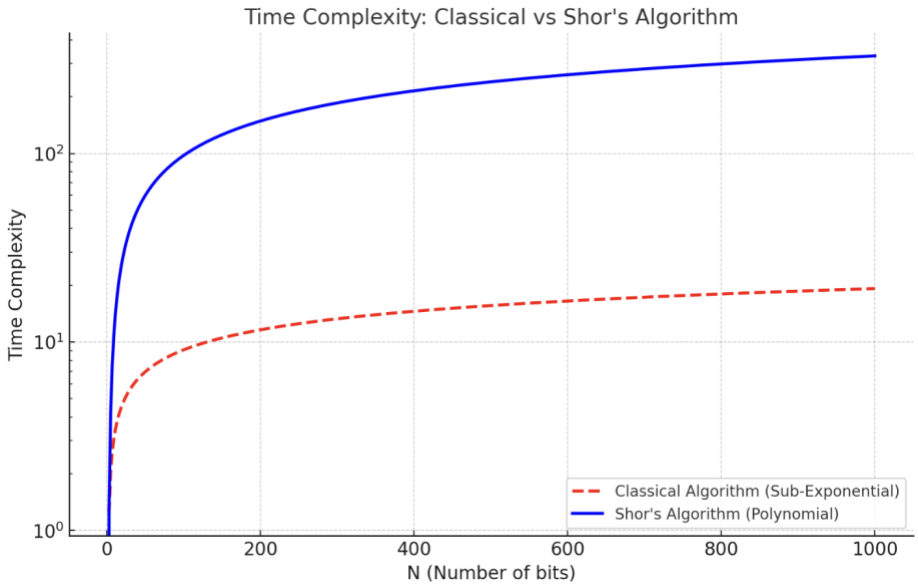

Fig. 2. Displays the difference between the time complexity of Shor's and subsequent classical algorithms.

Shor's algorithm offers a time complexity of $O((logN)^3)$ (Cao, 2009)which is a major leap providing exponential speedup over classical factoring algorithms. It leverages quantum Fourier transforms and modular exponentiation, allowing it to factor large integers efficiently on a quantum computer. The time complexity is polynomial in the number of digits of N, making it significantly faster for large N. In contrast the fastest algorithm in the classical computing side is the general number field sieve that offers a time complexity of

$$O\left(e^{(logN)^{1/3}*(\log(\log(N)))^{\frac{2}{3}}}\right)$$

(Gluher, 2020) It is a sub-exponential algorithm and represents the current practical limit for breaking cryptographic systems like RSA with large keys. The complexity is sub-exponential, meaning that as N grows, the time required to factor it increases much more rapidly than with polynomial time complexity. For large values of N, the difference in time required by these algorithms becomes dramatically large.Shor's algorithm, with its polynomial complexity, is exponentially faster than GNFS. GNFS has a sub-exponential time complexity, which grows faster than any polynomial but

slower than an exponential function. However, it is still vastly slower than Shor's algorithm for large integers. Even though GNFS is currently the best classical method, its time complexity means that factoring large integers (e.g., 2048-bit RSA keys) is computationally infeasible within a reasonable timeframe. With Shor's algorithm, a sufficiently large and error-corrected quantum computer could factor such integers in polynomial time, posing a direct threat to RSA and other cryptosystems based on the difficulty of integer factorization.

Table 1. Comparison of classical and quantum time complexities with respect to integer factorization and discrete logarithms.

Problem Type	***Classical Algorithm Complexity***	***Shor's Algorithm Complexity***
Integer Factorization (RSA)	$O(e^{(logN)^{1/3}*(\log(\log(N)))^{\frac{2}{3}}})$	$O((logN)^3)$
Discrete Logarithms (DH, ECC)	$O(e^{(logN)^{1/3}*(\log(\log(N)))^{\frac{2}{3}}})$	$O((logN)^3)$

Brute Force.

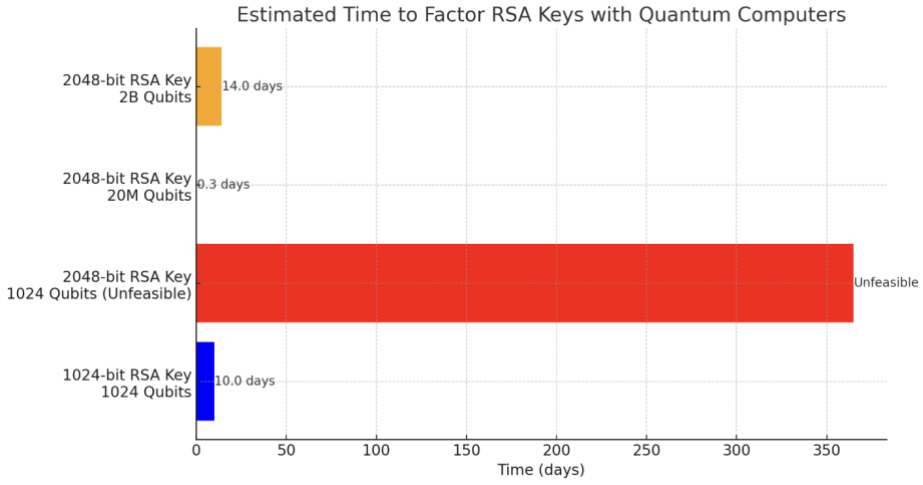

Fig. 3. Plots the estimate time to factor RSA keys with quantum computers.

1024 qubit quantum computer Factoring a 1024-bit RSA key takes approximately 10 days according to a paper by (Proos, 2003), assuming error correction and logical

qubits. Factoring a 2048-bit RSA key: requires 2048 qubits, making it currently unfeasible with current technology (Proos, 2003)20 million qubit quantum computer factoring a 2048-bit RSA key takes approximately 8 hours according to a study by (gidney C, 2019). Assuming large-scale quantum computation with error correction factoring a 2048-bit RSA key takes approximately 14 days according to a blueprint article by Jhegedus, assuming($2 * 10^9$) trapped ions.

Execution Time.

Classical Computing Based on the search results, the RSA classical decryption execution time is extremely long for large key sizes. According to one source, a classical computer would take approximately 300 trillion years to break an RSA-2048-bit encryption key. This is because factoring large integers, a crucial step in RSA decryption has not been shown to be possible in polynomial time on a classical computer. While various attacks have been discovered, such as timing attacks and side-channel attacks, they are not practical for large key sizes.

Quantum Computing. Theoretically, A quantum computer with 4099 perfectly stable qubits could break RSA-2048 encryption in 10 seconds, whereas a classical computer would take 300 trillion years (Quantum 101: Introduction to Quantum-Resilient Preparation, 2024). A study suggests that a quantum computer with 20 million qubits could break RSA-2048 encrypted messages using a revised algorithm (gidney C, 2019)In the near term, experts estimate that a quantum computer could break 2048-bit RSA encryption in approximately 8 hours (gidney C, 2019)Practically however, currently, the largest quantum computer has 72 qubits (Google Bristlecone), with an error rate of 0.6%, and the hardest problem is coherence time (Quantum 101: Introduction to Quantum-Resilient Preparation, 2024)Shor's algorithm, a quantum algorithm, can factor large integers efficiently, making it potentially capable of breaking RSA encryption. However, current quantum computers are far from having the required number of qubits and stability to implement Shor's algorithm (Wong, 2023).

2.2 Grover's Algorithm

Time Complexity.

The classical search algorithms prevalent today have in the worst case a time complexity of O(N). A classical search algorithm would need to check all N items individually to find the correct one in a list of unsorted elements. However, the Grover's method takes the same unstructured problem and performs it on a quantum computer. Grover's algorithm can find the solution in approximately O ($\sqrt{N}$) steps (Grover, 2001), offering a quadratic speedup over classical search algorithms. It performs this by iteratively applying the oracle and the Grover diffusion operator, which amplifies the probability of the correct solution state in superposition. The time complexity offered is O ($\sqrt{N}$).

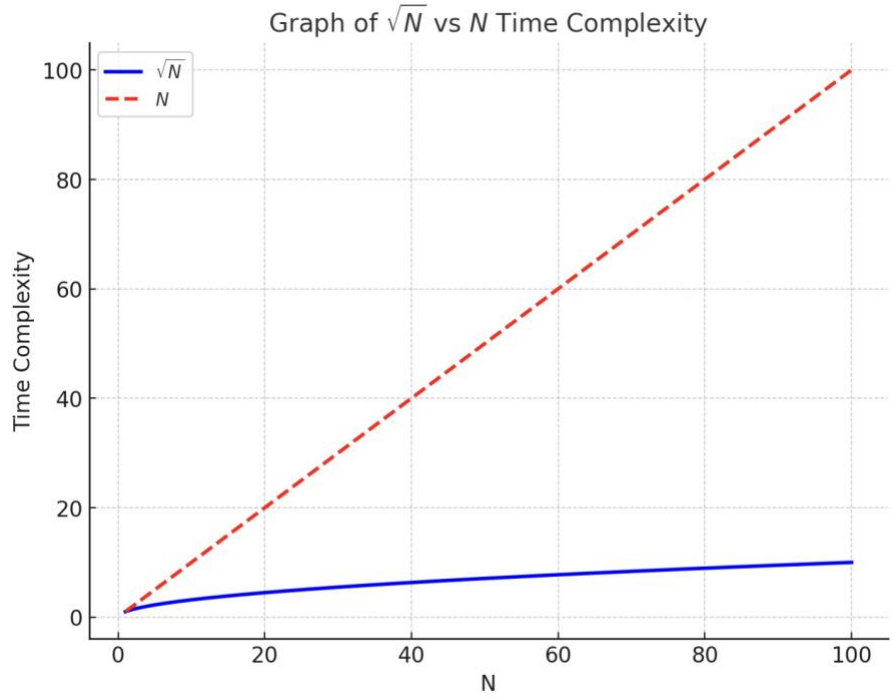

Fig. 4. The image shows the difference in the time complexity of classical search and quantum search algorithms.

Brute Force.

Symmetric-key algorithms, like AES, rely on the difficulty of brute-forcing a key to guarantee security. Grover's algorithm can speed up brute-force attacks against symmetric encryption schemes. In classical computing, finding a key through brute force requires O (2^n) time for an n-bit key. Grover's algorithm reduces this to O ($2^{\frac{n}{2}}$), which effectively halves the security level of the key length. AES-128 (128-bit key) requires Classical brute force of O (2^{128}) operations and with Grover's algorithm: O

(2^{64}) operations. Asymmetric cryptography relies on the hardness of specific mathematical problems rather than brute-force search. Grover's algorithm is less effective here because these algorithms don't typically rely on searching through a large space. Cryptographic hash functions are designed to make it hard to find collisions or preimages (inputs that produce a specific hash value). Classical brute force for finding a preimage in a hash function of size n requires O (2^n) operations. Grover's algorithm reduces this to O ($2^{\frac{n}{2}}$).

Table 2. Comparison of the Classical and quantum algorithms and their impact on the cryptographic algorithms.

Cryptographic System	***Classical Time Complexity***	***Grover's Algorithm Time Complexity***	***Impact***
Symmetric (AES-128)	O(2^{128})	O(2^{64})	Halves security (use AES-256)
RSA (Public Key)	O(2^n)	No significant impact (Shor's is more relevant)	Little impact (Shor's is a bigger threat)
SHA-256 (Hash)	O(2^{256})	O(2^{128})	Halves security (use SHA-512)

3 Discussion

3.1 RSA (Integer Factorization):

RSA encryption is based on the difficulty of factoring large numbers into primes. Classical computers struggle with this task when the numbers are sufficiently large (e.g., 2048-bit keys), but Shor's algorithm can solve the problem in polynomial time. If a sufficiently large and stable quantum computer is built, Shor's algorithm could break RSA encryption allowing attackers to decrypt messages, forge digital signatures, and compromise the security of systems that rely on RSA for authentication and data protection.

3.2 Implications for Diffie-Hellman (Discrete Logarithms):

Diffie-Hellman key exchange relies on the hardness of computing discrete logarithms, which is another problem Shor's algorithm can solve efficiently. Elliptic Curve Cryptography (ECC), which is based on the elliptic curve discrete logarithm problem, is also vulnerable. Both Diffie-Hellman and ECC are widely used for secure

communications (e.g., TLS, SSL, VPNs, etc.), and Shor's algorithm could break these cryptosystems as well.

3.3 Digital Signatures and Certificates:

Systems like digital signatures and public key infrastructures (PKI), which are essential for verifying identities and securing transactions on the internet, rely on RSA and ECC. (Rivest, 1978) Shor's algorithm could break these systems, leading to a collapse in trust across internet communications, e-commerce, and financial transactions.

3.4 Symmetric-Key Cryptography:

The quantum search algorithm is a technique for searching N possibilities in only $\sqrt{N}$ (Grover, 2001) steps. the advanced search capabilities of the Grovers algorithm will reduce the security of symmetric key cryptography. AES-128 is no longer secure against quantum attacks since 2^{64} operations are feasible on a quantum computer. AES-256, which originally had 2^{256} possible keys, would be reduced to an effective key strength of 128 bits with Grover's algorithm. While this is still secure by current standards, it highlights the need to increase key lengths for post-quantum security. To maintain the security of AES-128, we may need to move to AES-256 (or higher) to protect against Grover's algorithm.

3.5 Asymmetric (Public-Key) Cryptography:

Asymmetric cryptography relies on the hardness of specific mathematical problems like factorization for RSA, discrete logarithms for Diffie-Hellman rather than brute-force search. Grover's algorithm is less effective here because these algorithms don't typically rely on searching through a large space. However, another quantum algorithm, Shor's algorithm, directly threatens these cryptosystems by solving the underlying mathematical problems efficiently.

3.6 Hash Functions:

Affects the security of hash functions, which are used in Digital signatures Password hashing Integrity verification (e.g., file checksums, blockchain). For a hash function with an output size of 256 bits (like SHA-256), Grover's algorithm would reduce the effective security to 2^{128}. This means that SHA-256 (which is widely used in blockchain technologies, SSL certificates, etc.) would offer only 128 bits of security against quantum computers. Similarly, for SHA-512, Grover's algorithm would reduce the effective security to 2^{256}.

3.7 Password Security

Grover's algorithm can also be used to brute-force passwords, which are often hashed before being stored. Passwords are typically hashed using algorithms like bcrypt, SHA-256, or PBKDF2. (Percival, 2009) If Grover's algorithm is applied, the effective strength of a hashed password decreases significantly. A password of length n, hashed using SHA-256, would take O $\left(2^{\frac{n}{2}}\right)$ operations to brute-force. This means weaker passwords become even more vulnerable to attacks. Longer and more complex passwords will be necessary to compensate for the quantum threat posed by Grover's algorithm.

4 Post-Quantum Cryptography: The Future of Cryptographic Security

The advent of quantum computers has caused a big threat to the current existing cryptographic algorithms therefore a transition to post quantum algorithms

4.1 Transition to Post-Quantum Algorithms:

lattice-based, hash-based, and code-based cryptography systems need to be adopted by organizations. NIST Post-Quantum Cryptography Standardization The National Institute of Standards and Technology (NIST) is leading efforts to standardize quantum-resistant cryptographic algorithms. Several candidate algorithms are currently being evaluated which should be adopted and integrated with the currently existing security measures.

4.2 Hybrid Cryptography:

A hybrid approach can be used where both classical like the RSA and quantum-resistant that is the lattice based cryptographic algorithms are combined.

4.3 Increase Key Size for Existing Algorithms:

Symmetric cryptography can be protected against Grover's algorithm by using longer key sizes. While this is not a complete solution, increasing key sizes for algorithms like RSA or ECC can provide temporary protection against quantum attacks by making the problems harder for current quantum computers.

4.4 Upgrade Key Exchange Mechanisms:

Replace current key exchange algorithms like Diffie-Hellman and Elliptic Curve Diffie-Hellman (ECDH) with quantum-resistant alternatives, such as Lattice-based key exchange protocols (e.g Kyber, NewHope) and Isogeny-based key exchanges (SIDH).

Asymmetric cryptography is more vulnerable to Shor's algorithm than to Grover's, and many traditional public-key cryptosystems (RSA, ECC) will need to be replaced by post-quantum cryptographic algorithms.

4.5 Plan for Cryptographic Agility:

Cryptographic agility means that systems should be designed to quickly switch between cryptographic algorithms if one becomes insecure . This will allow for a seamless transition to quantum resistant algorithms.

5 Conclusion

In conclusion, the emerging quantum technology although has opened new pathways for advancements in security has also been a major threat to the pre-existing security measures the world had in place the advanced computation capabilities render the classical cryptosystems we currently utilize useless. Though in its infancy we can already see the possible impacts of the technology on cryptography as the emergence of Shor's and Grover's algorithms have put a massive question on whether the classical cryptosystems are as fool proof as we thought they were. The need to transfer our view and approach to security in a post-quantum world has become the need of the hour we need to make sure we utilize the already available knowledge and combine it with the new emerging quantum technology to provide truly full-proof systems according to the times. There is a need for a fundamental shift in encryption practices to protect sensitive information, especially since today's most secure algorithms would be vulnerable to quantum attacks in the future. (Althobaiti, 2020). The consequences of a fall in the security of the world could be dire even as Cyber vulnerabilities in the control systems of a smart city or an automated industry may lead to catastrophic consequences (Oliva delMoral, 2024).Quantum cryptography could well be the first application of quantum mechanics at the single quantum level (Gisin, 2002) With the current growth in tech, the day is not far off when quantum computing is to become available to the masses and before that day comes it is upon us to secure our systems and adopt the new wave of tech to ensure a smooth running of the world of the internet. Cryptography as a whole will need to transition towards quantum-resistant algorithms to maintain security in a post-quantum world.

References

1. Gidney, C., & Ekerå, M. (2019). How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. *Quantum Physics*. https://scirate.com/arxiv/1905.09749v3
2. Proos, J.& Zalka, C.(2003). Shor's discrete logarithm quantum algorithm for elliptic curves. *QIC 3, (No. 4)*(2003), pp.317-344. https://arxiv.org/abs/quant-ph/0301141
3. *Quantum 101: Introduction to Quantum-Resilient Preparation | QuintessenceLabs*. (n.d.). Quintessence Labs. Retrieved August 20, 2024, from https://www.quintessencelabs.com/quantum-101
4. Wong, H. Y. (2023). Shor's Algorithm. In Introduction to Quantum Computing: From a Layperson to a Programmer in 30 Steps (pp. 289-298). Cham: Springer International Publishing.
5. Gluher, A. L., Spaenlehauer, P. J., & Thomé, E. (2020). Refined Analysis of the Asymptotic Complexity of the Number Field Sieve. arXiv preprint arXiv:2007.02730.
6. Tom, J. J., Anebo, N. P., Onyekwelu, B. A., Wilfred, A., & Eyo, R. E. (2023). Quantum computers and algorithms: a threat to classical cryptographic systems. Int. J. Eng. Adv. Technol, 12(5), 25-38.
7. Althobaiti, O. S., & Dohler, M. (2020). Cybersecurity challenges associated with the internet of things in a post-quantum world. Ieee Access, 8, 157356-157381.
8. Nahed, M., & Alawneh, S. (2020). Cybersecurity in a post-quantum world: How quantum computing will forever change the world of cybersecurity. American Journal of Electrical and Computer Engineering, 12(1), 81-93.
9. Mermin, N. D. (2003). From Cbits to Qbits: Teaching computer scientists quantum mechanics. American Journal of Physics, 71(1), 23-30.
10. Grover, L. K. (2001). From Schrödinger's equation to the quantum search algorithm. American Journal of Physics, 69(7), 769-777.
11. Ekert, A., & Jozsa, R. (1996). Quantum computation and Shor's factoring algorithm. *Reviews of Modern Physics*, *68*(3), 733.
12. Gisin, N., Ribordy, G., Tittel, W., & Zbinden, H. (2002). Quantum cryptography. *Reviews of modern physics*, *74*(1), 145.
13. Oliva delMoral, J., deMarti iOlius, A., Vidal, G., Crespo, P. M., & Martinez, J. E. (2024). Cybersecurity in Critical Infrastructures: A Post-Quantum Cryptography Perspective. IEEE Internet of Things Journal.
14. Cao, Z., & Liu, L. (2009, December). On the complexity of Shor's algorithm for factorization. In 2009 Second International Symposium on Information Science and Engineering (pp. 164-168). IEEE.
15. Jozsa, R. (1999). Searching in Grover's algorithm. arXiv preprint quant-ph/9901021.
16. Hayward, M. (2008). Quantum computing and shor's algorithm. *Sydney: Macquarie University Mathematics Department*, *1*.
17. Percival, C. (2009). Stronger key derivation via sequential memory-hard functions.
18. Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2), 120-126.