HOW QUANTUM COMPUTING BREAKS MODERN ENCRYPTION

Logan Koehn, Kalyana Gallagher, and Mattie Davis

University of Northern Colorado

Abstract

The advent of quantum computing presents a formidable challenge to the security of modern encryption techniques. This paper explores the implications of quantum computing advancements on encryption, with a focus on the potential of quantum computers, particularly through Shor's Algorithm, to decrypt data encrypted with current cryptographic standards. Beginning with a historical overview of quantum computing development and an explanation of encryption principles, we delve into the specifics of symmetric and asymmetric encryption, highlighting their vulnerabilities to quantum attacks. The core of our analysis centers on the RSA encryption algorithm, illustrating its susceptibility to quantum computing due to the efficiency of Shor's Algorithm in factoring large primes, a foundational aspect of RSA's security. We further examine the current landscape of quantum-resistant algorithms, including those emerging from the NIST Post-Quantum Cryptography Standardization Project, with special attention to lattice-based cryptography and hash-based signatures as promising avenues for securing digital communications against quantum threats. Additionally, the paper discusses quantum key distribution (QKD) as an example of quantum mechanics applied to enhance cryptographic security, albeit with practical challenges. Our conclusion underscores the urgency of transitioning to quantum-resistant encryption to safeguard against the looming threat posed by quantum computing advancements, emphasizing the need for proactive measures well before the anticipated "Q-Day" when quantum computers will be able to break current encryption algorithms. This study aims to provide a comprehensive understanding of the quantum computing threat to encryption and the ongoing efforts to develop secure post-quantum cryptographic solutions.

Introduction

The power of quantum computers has been steadily increasing since their introduction in 1998. The first quantum computer, developed by Isaac Chauang, Neil Gershenfeld, and Mark Kubinec, contained 2 qubits and could only run for a few nanoseconds (Holton, 2024). Two decades later and a quantum computer developed by Atom Computing has the largest number of qubits in any quantum computer with 1,180 qubits (Wilkins, 2023). At this rate, Q-Day could be right around the corner. Q-Day is the name given by some researchers to the day in which quantum computers become powerful enough to break modern encryption algorithms. The

consensus is that a quantum computer will need around 4,000 qubits to be powerful enough to reach this point (Herman, 2021), but researchers are unsure when this will occur. Some estimates predict that Q-Day could come as soon as 2025 (Varanasi, 2023), so early and quick preparation to defend against this threat is necessary. In this paper, we will discuss the foundations of modern encryption and quantum computers, the threat quantum poses in the form of Shor's Algorithm, and finally, what's being done to develop quantum resistant algorithms to prevent this threat that Q-Day poses.

Encryption Overview

Encryption involves converting "plaintext" into something called "ciphertext". Plaintext is any text that is specifically formatted in some way so that either a computer or a human can read and make sense of it. Ciphertext is an illegible transformation of this plaintext into a form that cannot be read by humans or machines, rendering it useless until it can be decrypted by converting it back into plaintext. Plaintext is converted into ciphertext using an encryption algorithm (also called a cipher) in conjunction with an encryption key. A key is a randomly generated string of bits, specific to the algorithm being used to encrypt the data. This or another key can then be used to decrypt some ciphertext data into plaintext (Sheldon et al., 2024).

The main strength of encryption algorithms is that if someone was to intercept sensitive data that has been encrypted, it would likely take them an unreasonable amount of time to figure out what cipher was used to encrypt the data, as well as the key used to encrypt the data. In general, the longer the key and the more extensive the encryption algorithm, the longer it will take hackers to be able to decrypt intercepted data (Sheldon et al., 2024).

Symmetric Encryption

There are two main types of encryption: symmetric and asymmetric. Symmetric encryption is much easier to understand than asymmetric encryption; In a symmetric encryption algorithm, plaintext is encrypted using a key. The data can then be decrypted by using the same key that was used to encrypt the data. This type of encryption is known as secret key ciphers, where the key is referred to as a shared secret since the key to data encrypted symmetrically must be known by only the sending and receiving parties (Sheldon et al., 2024).

A very simple and old example of this is the Caesar Shift Cipher, which was an encryption algorithm invented by the ancient Romans. This simple symmetric encryption algorithm works by taking a plaintext message and shifting the letters of each character by a certain number of letters in the alphabet. For example, the plaintext CAT could be shifted by 3, resulting in the ciphertext FDW. The "key" to this algorithm is 3, and this same key can be used to decrypt the algorithm by reversing it, shifting each letter back by 3 (Sheldon et al., 2024).

More recently, more advanced and secure symmetric encryption algorithms were developed, including AES, DES, Diffie-Hellman key exchange, Twofish, and many more (Sheldon et al., 2024). The only symmetric encryption algorithm that achieves perfect secrecy is an algorithm known as the one-time-pad. This algorithm works similarly to the Caesar cipher in the sense that it shifts each character in the plaintext message by a certain amount. The difference is that the one-time-pad shifts each character by a random amount based on the number of available options for each character. This is the only algorithm that achieves perfect secrecy because the ciphertext message generated does not give away any information as to what its plaintext message was. However, because of the key needing to be the same length of the plaintext message, and the fact that a new key must be generated for every encrypted message, this algorithm isn't often used in practicality (Liu & Badr).

Asymmetric Encryption

Asymmetric encryption is a little bit more difficult to understand than symmetric, as it actually involves 2 different types of keys. Asymmetric encryption is used primarily for the secure transfer of data from one person to another. In this type of encryption, each unit sending or receiving a message has 2 different keys: a public and a private key. A unit's public key is public and can be accessed by any other unit on the network. A unit's private key, however, is kept private, and only its owner should have it. Data that is encrypted with any unit's public key can then only be decrypted by that unit by using its private key (Cobb, 2021).

A common example of this involves three actors, Alice, Bob, and Eve. Alice wants to send Bob a message, but is worried that Eve is going to intercept her message and read it herself. Using asymmetric encryption, she is able to accomplish this. She starts by retrieving Bob's public key, then encrypting her message with it. She then sends it to Bob. Since the message was

encrypted using Bob's public key, the message can now only be decrypted by his private key. This means that when Bob receives it, he can read the message by decrypting it with his private key since only he should have access to his private key. If Eve intercepts the message, she won't be able to make sense of it due to her not having that private key.

RSA (Rivest-Shamir-Adleman) Encryption

One of the oldest and most widely used asymmetric encryption algorithms is called RSA, or Rivest-Shamir-Adleman. Being an asymmetric encryption algorithm, its main goal is to allow for data encrypted with a public key to only be able to be decrypted by a related private key, and to allow for public keys to be derived from private keys, but not vice versa. It accomplishes this by exploiting the fact that classical computers take an extremely long time to find the factors of very large prime numbers. It is very easy to multiply two factors, but extremely difficult to figure out what those two numbers were from the prime number calculated (Cobb, 2021). However, unlike classical computers, quantum computers are actually able to find the factors of large prime numbers with relative ease using something called Shor's algorithm. This is the key reason why quantum computers threaten modern encryption .

Overview of Quantum Computing

Quantum computers are computers that are able to take advantage of quantum mechanics, making them much faster than conventional computers under the right circumstances. Classical computers function through the use of objects that act as microscopic on/off switches. These switches can very strictly only be either on or off. This allows computers to store data and perform calculations by using a binary base 2 number system as opposed to our base 10 number system. Quantum computers, instead of using these transistors and capacitors, use quantum bits, or qubits. These are quantum particles that act as a bit in a traditional computer, but process information very differently (Caltech).

Quantum Superposition

Instead of a single qubit being either strictly on or off, it instead can exist in a superposition between on and off. A quantum particle existing in a superposition between two states (in the case of a qubit, on or off, 0 or 1) can be measured, at which point it chooses one of

two states to exist as. This opens up new possibilities for computation, and through the exploitation of this, quantum computer scientists are able to perform extremely large amounts of calculations in parallel, rapidly speeding up some computations (Giles, 2021).

Quantum Entanglement

Something else that quantum computers take heavy advantage of is quantum entanglement. This is when two particles can become bound with one another in a way such that the particles exist in a single quantum state, and knowing something about the state of one particle will tell you something about the state of the other bound particle. This can be used in quantum computing for instantly obtaining information about another qubit from an entangled qubit (Giles, 2021).

A common misconception of quantum entanglement is that it allows for the instant traversal of information over space, breaking the light barrier. Einstein even called quantum entanglement “spooky action at a distance”. However, that is unfortunately not the case. What really happens with quantum entanglement is that the particles become correlated in a way so that knowing something that is already true about one entangled particle will tell you something that happens to be true of another entangled particle. This information is not actually traveling across space at all (Quantum Xchange, 2022).

In the realm of quantum cybersecurity, an exploration through the prism of quantum computing's influence on contemporary encryption methodologies unveils the intricate mechanics and profound implications of leveraging quantum phenomena—specifically entanglement and superposition—to decrypt algorithms that are presently deemed secure. A prime illustration of this challenge is the RSA encryption system, which fundamentally depends on the computational intricacy of factoring large prime numbers. The following section aims to illuminate the process by which quantum computing, particularly through the implementation of Shor's Algorithm, can substantially compromise the security foundations of the RSA encryption method, signaling a pivotal shift in the landscape of digital security.

The Undermining of RSA by Quantum Computing

RSA encryption is a fundamental element of contemporary digital security systems. It relies on the mathematical intricacy of factoring large semiprime numbers into their prime components. The effectiveness of RSA encryption stems from the widely held assumption that traditional computing systems are incapable of factoring significant numbers within a reasonable timeframe. However, with the advent of quantum computing and Shor's Algorithm, the computational paradigm has undergone a significant transformation. Shor's Algorithm employs quantum mechanical principles to factorize large numbers efficiently. As noted by renowned experts such as Peter Shor, and further explored in scholarly research by Steane, Williams, and Clearwater, this advancement poses a formidable challenge to security mechanisms centered around RSA.

Quantum Mechanics at Work: Superposition, Entanglement, and Quantum Fourier Transform (QFT)

The fundamental principles of quantum mechanics, namely superposition and entanglement, serve as the foundation of quantum computing's operational framework. The phenomenon of superposition allows a quantum bit, or qubit, to exist in multiple states simultaneously, providing the ability to process a vast array of possibilities in parallel. The concept of entanglement ensures that qubits are correlated in such a way that any change in one qubit, regardless of its distance from another, instantaneously affects the other, allowing for coordinated problem-solving approaches that surpass the capabilities of traditional computing.

The successful execution of Shor's Algorithm is reliant upon the Quantum Fourier Transform (QFT), which functions similarly to its classical counterpart by converting a sequence of numbers into a frequency representation. The QFT converts the state of qubits into quantum states reflective of probability amplitudes, a critical step in identifying the periodicities within the quantum system. This ability to discern periodicities is indispensable for factoring the large numbers that are commonly used in RSA encryption.

In essence, the principles of quantum mechanics, when applied to computing, provide a means of processing information in a way that surpasses the limitations of traditional computers. By harnessing the power of superposition and entanglement, quantum computing can solve

complex problems with unprecedented speed and accuracy, making it a promising technology for the future of computing.

The Steps to Decrypting RSA with a Quantum Computer

The process of decrypting RSA encryption via a quantum computer involves several steps. It begins with the acquisition of the RSA public key, which consists of a modulus 'n' that is the product of two large prime numbers 'p' and 'q', and a public exponent 'e'. Shor's algorithm, implemented via a quantum computer, is then used to factorize 'n' into 'p' and 'q', employing quantum parallelism and the Quantum Fourier Transform (QFT) to accelerate this process.

Once 'p' and 'q' are secured, the computation of the private key commences by determining Euler's totient function ($\phi(n)$) and subsequently utilizing the Extended Euclidean Algorithm to ascertain the modular multiplicative inverse of 'e' mod $\phi(n)$, culminating in the derivation of the private key 'd'. This sequence of operations showcases the computational efficiency of quantum computers in comparison to classical computers, and underscores the potential for RSA encryption to be decrypted in the quantum computing era.

Implications and the Path Forward

The theoretical and empirical strides in quantum computing portend an imminent era wherein conventional public-key cryptography systems, such as RSA, might fail to ensure secure communications. This foresight necessitates a strategic shift towards the cultivation and embracement of quantum-resistant encryption methodologies to preclude potential quantum computational exploitations.

The dialogue on quantum-resistant cryptography transcends academic curiosity, propelled by an urgent imperative to foresee and counteract the security quandaries introduced by quantum computing. With the continuous progression in quantum computing research and development, the exploration of quantum-resistant algorithms and the fortification of security protocols emerge as critical endeavors to preserve the integrity and confidentiality of digital communications in the forthcoming post-quantum epoch.

The confluence of quantum computing and cybersecurity represents a significant shift in the digital encryption sphere. The capabilities of quantum computers, as exemplified by Shor's

Algorithm, highlight the need for the cryptographic community to adapt to these emerging challenges. As we move towards the quantum revolution, the need for quantum-resistant cryptography has become a global mandate to safeguard our digital infrastructure against future vulnerabilities.

To address these challenges, there is a growing impetus to develop new cryptographic tools and protocols that can withstand the power of quantum computers. This requires a deep understanding of the underlying principles of quantum computing, as well as the development of new encryption techniques that can resist quantum attacks.

In this context, the cryptographic community is working to identify new post-quantum cryptographic algorithms that can provide high levels of security against quantum attacks. These new algorithms are being designed to be resistant to attacks from both classical and quantum computers, ensuring that our digital infrastructure is secure even in the face of rapidly advancing technologies.

In summary, the convergence of quantum computing and cybersecurity is a critical issue that demands the attention of the cryptographic community. The development of post-quantum cryptographic algorithms and protocols represents an important step towards securing our digital future, and it is essential that we continue to invest in these efforts to ensure that our digital infrastructure remains resilient in the face of emerging threats.

Quantum Key Distribution

In the world of cryptography, quantum mechanics are already being used. A notable example of this is the key distribution method known as quantum key distribution, or QKD for short. Quantum key distribution is a key distribution method which “utilizes the unique properties of quantum mechanical systems to generate and distribute cryptographic keying material using special purpose technology.” (National Security Agency, n.d.). In this system, the polarity of a given photon is measured, then sent to the recipient in that state. The key is generated by measurement of the photon’s polarity, which is then relayed to the recipient, who then has to measure the polarity with the correct filter on to build the key. In addition to this, since in a quantum system measurement causes a disruption in the system, this allows any eavesdroppers on the line to be easily detected by the sender or the receiver. This is the basis of

the first QKD protocol known as BB84. BB84 was proposed in 1984 by Bennett and Brassard, which is where the acronym for the name comes from ("Lecture 12: Quantum key distribution," n.d.). The BB84 protocol is what every other QKD protocol is based on.

The other quantum key distribution protocols are E91, Device Independent Quantum Key Distribution, and Twin Fields Quantum Key Distribution. Protocol E91 was developed by Artur Ekert in 1991 and makes the notable improvement to BB84 of introducing entanglement in the key generation process, so that each party has a single identical photon at the end of a transmission (Ekert, 1991). Device Independent Quantum Key Distribution was proposed in 1998 by Mayers and Yao in their paper titled *Quantum Cryptography with Imperfect Apparatus.* This protocol allows for Quantum Key Distribution to be used by untrusted devices, so long as they pass a self-check test to verify they are secure. In addition to this, this protocol makes the transmission abort if a deviation in the system is detected, instead of sending an incorrect message. The last, most recent improvement made on BB84 is the Twin Fields Quantum Key Distribution system. Twin Fields Quantum Key Distribution was proposed in 2018 in an article by Lucamarini, Yuan, Dynes, and Shields titled, *Overcoming the rate-distance limit of quantum key distribution without quantum repeaters*. In this article, they propose their protocol as a way to overcome a large downside of quantum key distribution, which is the rate-distance limit. The rate-distance limit is the idea that without physical repeaters present in the transmission, the rate of key generation decreases exponentially with the distance at which a transfer is being sent (Takeoka et. al., 2014). Twin Fields Quantum Key Distribution claims to solve this issue by creating quantum repeaters within the system, which would theoretically remove the need for physical repeaters. However, Twin Fields Quantum Key Distribution is too new to be properly implemented, therefore the issue of rate-distance limit is still a major downside to QKD.

In addition to this issue, other issues include the problem that the authenticity of a key cannot be verified, and that this key distribution requires very expensive special purpose equipment, which according to the National Institute of Standard and Technology cost around $5,000 - $20,000 in 2004. Another part of what makes QKD so expensive is the need to lay down new fiber optic cables to have a secure line to transfer the key across. Because of these issues with quantum key distribution, the National Security Agency has recommended to not use

quantum key distribution, and instead recommends the use of post-quantum encryption algorithms.

Post-Quantum Cryptography Standardization Project

This recommendation from the NSA comes now due to the progress of the National Institute of Standards and Technology's (NIST) Post-Quantum Cryptography Standardization Project. This project was started in 2016 and is a competition which challenged the world's greatest cryptographers to make quantum-resistant algorithms. The winning algorithms from this competition will be standardized by the NIST and the NSA to be used for general encryption and for digital signatures (National Institute of Standards and Technology, 2022). In early 2017, the NIST announced that they would be accepting submissions until November 30, 2017. During the period in which they were open for submissions, they received a total of sixty-nine submissions. Over the span of five years from 2017, to 2022, the NIST conducted three rounds of testing. The purpose of these rounds was to reduce the number of submitted algorithms and declare which of the algorithms the NIST considered the best candidates for standardization. In 2022, at the end of the NIST's rigorous testing, they declared four winners. These four winners are CRYSTALS-Kyber, developed by the CRYSTALS team, CRYSTALS-Dilithium, also developed by the CRYSTALS team, FALCON, developed by Fouque et. al., and SPHINCS+, developed by Hülsing et. al. Of these algorithms, CRYSTALS-Kyber was selected to be the standard for general encryption, while the three other algorithms were selected to be used for digital signatures. According to the NIST (2022), CRYSTALS-Kyber was selected to be used for general encryption due to its small encryption keys and fast speed of operation. Besides SPHINCS+, the method used in these algorithms is called multidimensional structured lattices.

Multidimensional Structured Lattices

CRYSTALS-Kyber, CRYSTALS-Dilithium, and FALCON all use multidimensional structured lattices in their algorithms, however each of them differ in the type of problem which is used in addition to these lattices. First though, an explanation of structured lattices is necessary. A lattice is a space which contains an infinite number of points spread out across a dimension, and in a multidimensional lattice, the number of dimensions is increased from just two dimensions, to hundreds or thousands of dimensions (Alwen, 2018). Within these lattices,

the encrypted data is placed somewhere inside, located on a lattice point. Another lattice point is selected as the starting point, and to decrypt the data, the computer must find the other lattice point which stores the data. Along the way, the computer must find the correct vectors to reach the data point in addition to also competing with problems which aim to impede. Given enough dimensions, this would be theoretically impossible for a quantum computer to complete in a reasonable amount of time, which is why three of the four winning algorithms utilize this method. In order for a recipient to be able to decrypt the data, the key is a list of correct vectors needed to quickly and efficiently reach the given lattice point containing the data (Veritasium, 2023). As discussed earlier, the advantage in breaking encryption algorithms that quantum computers have is that they are able to take advantage of parallelism and compute many calculations at the same time. Lattice-based cryptography is based around this advantage, and forces the computer to complete one task at a time, nullifying quantum computers' main strength. This is because in a structured lattice, the computer needs to find one vector before it can find the next vector, which may or may not be the correct vector. If the computer begins to go down an incorrect path, it may have to backtrack or restart the process due to the spatial complexity of introducing thousands of dimensions.

As stated before, the computer also must compete with other problems along the way, which is how the three lattice-based algorithms differ from each other. CRYSTALS-Kyber makes use of the learning with errors problem, which was introduced by Oded Regev in 2005 (CRYSTALS, 2020). In this problem, noise is introduced to the given equation by way of introducing mathematical errors which the computer must work with when finding where the data point is located within the lattice (Regev 2009). CRYSTALS-Dilithium and FALCON both use NTRU lattices, which is a lattice system developed by Hoffstein et. al. in 1996 as a lattice-based cryptosystem to be used for digital signatures (Silverman, 2015). The main difference between these two comes from that FALCON uses the hardness of solving the short integer solution problem (Fouque et. al., n.d.), while CRYSTALS-Dilithium makes use of the Fiat-Shamir with Aborts technique (CRYSTALS, 2021). The final winner of the Post-Quantum Standardization Project however does not use lattice-based cryptography, and instead utilizes hash-functions.

Stateless Hash-Based Signatures

SPHINCS+, the fourth winner, is unique when compared to the other winners. For starters, it doesn't utilize lattices. Instead, it uses binary hash trees full of one-time signature hash-functions (Bernstein et. al., 2015). Another unique characteristic of SPHINCS+, is that it is completely stateless (Hülsing et. al., 2023). Being stateless means that the application does not retain any information about the current run of the program and treats each run as completely independent from any others (Red Hat, 2023). According to Bernstein et. al. (2015), to achieve complete statelessness, the hash tree is rebuilt every time it is ran using a pseudorandom seed. This seed is responsible for building the shape of the tree, creating the hash-functions within each node of the tree, and for determining the location of the private key hash-function. In these trees, the public key is always located at the root node, while the private key is hidden somewhere within the tree pseudo-randomly. This method of encryption does bring a large downside with it however, and this downside is caused by the same reason that it is so strong. The fact that it is stateless leads to the program being slow and bulky, especially when compared to the much smaller and more efficient lattice-based algorithms. Even though SPHINCS+ isn't as small or efficient as the lattice-based algorithms, it is still a worthy competitor to the other algorithms. However, the main reason why it was selected is because it is based on a different mathematical approach than the other winners (National Institute of Standards and Technology, 2022). The NIST believes that it would function well as a backup to the other algorithms in case they don't function as expected once they are put to practical test against a powerful enough quantum computer.

Conclusion

In conclusion, asymmetric encryption works by converting plaintext into ciphertext using a public key and then back into plaintext using their public key. Classical computers will not be able to derive a person's private key from their public key in a reasonable amount of time. However, because of quantum's ability to perform large amounts of calculations all at once, they can do this by using Shor's algorithm. This puts people's data at risk, as the most widely used asymmetric encryption algorithm is RSA, something that is very vulnerable to this kind of attack. People have created several quantum-safe algorithms to help protect against these attacks, but innovative thinking is required to maintain data security.

Any networks that are secured through traditional encryption methods are extremely vulnerable to compromise by quantum computers' upcoming ability to break modern encryption after Q-Day. Experts say that Q-Day is anywhere between 5 and 20 years away, however, this timeframe is just a prediction and is very uncertain. Because of this, people should begin to migrate over to theoretically quantum-safe algorithms now, in the hopes that they aren't the victims of Q-day (Lohrmann, 2023).

References

Alwen, J. (2018, June 15). *What is Lattice-Based Cryptography & Why You Should Care.* Medium. https://medium.com/cryptoblog/what-is-lattice-based-cryptography-why-should-you-care-dbf9957ab717

Bernstein, D. J., Hopwood, D., Hülsing, A., Lange, T., Niederhagen, R., Papachristodoulou, L., Schneider, M., Schwabe, P., Wilcox-O'Hearn, Z. (2015). *SPHINCS: practical stateless hash-based signatures.* Department of Computer Science, University of Illinois at Chicago.

Caltech. (n.d.). What is quantum computing?. Caltech Science Exchange. https://scienceexchange.caltech.edu/topics/quantum-science-explained/quantum-computing-computers

Cobb, M. (2021, November 4). What is the RSA algorithm? definition from searchsecurity. TechTarget. https://www.techtarget.com/searchsecurity/definition/RSA

CRYSTALS. (2020, December 23). *Kyber.* CRYSTALS: Cryptographic Suite for Algebraic Lattices. https://pq-crystals.org/kyber/index.shtml

CRYSTALS. (2021, February 16). *Dilithium.* CRYSTALS: Cryptographic Suite for Algebraic Lattices. https://pq-crystals.org/dilithium/index.shtml

Ekert, A. K. (5 August 1991). Quantum cryptography based on Bell's theorem. *Physical Review Letters*, *67*(6), 661–663. https://doi.org/10.1103/PhysRevLett.67.661

Ekert, A., & Josza, R. (1996). "Quantum computation and Shor's factoring algorithm." Rev. Mod. Phys., 68, 733–753.

Fouque, P. A., Hoffstein, J., Kirchner, P., Lyubashevsky, T. P., Prest, T., Ricosset, T., Seiler, G., Whyte, W., & Zhang, Z. (n.d.). *FALCON.* Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU. https://falcon-sign.info/

Giles, M. (2021, October 20). Explainer: What is a quantum computer?. MIT Technology Review. https://www.technologyreview.com/2019/01/29/66141/what-is-quantum-computing/

Herman, A. (2021, June 07). Q-Day Is Coming Sooner Than We Think. *Forbes*. https://www.forbes.com/sites/arthurherman/2021/06/07/q-day-is-coming-sooner-than-we-think/

Holton, W. C. (2024, February 13). *quantum computer*. *Encyclopedia Britannica*. https://www.britannica.com/technology/quantum-computer

Hülsing, A., Aumasson, J. P., Bernstein, D. J., Beullens, W., Dobraunig, C., Eichlseder, M., Fluhrer, S., Gazdag, S. L., Kampanakis, P., Kölbl, S., Kudinov, M., Lange, T., Lauridsen, M. M., Mendel, F., Niederhagen, R., Rechberger, C., Rijneveld, J., Schwabe, P., & Westerbaan, B. (2023, August 02). *Home*. SPHINCS+: Stateless hash-based signatures. https://sphincs.org/index.html

Lecture 12: Quantum key distribution. (n.d.). [Lecture notes] Max Planck Institute. https://mpl.mpg.de/fileadmin/user_upload/Chekhova_Research_Group/Lecture_4_12.pdf

Liu, D., & Badr, M. (n.d.). 8.2 the one-time pad and perfect secrecy. Computer Science Teaching Labs. https://www.teach.cs.toronto.edu/~csc110y/fall/notes/08-cryptography/02-one-time-pad.html

Lohrmann, D. (2023, February 12). Quantum computers: What is Q-Day? and what's the solution? GovTech. https://www.govtech.com/blogs/lohrmann-on-cybersecurity/quantum-computers-what-is-q-day-and-whats-the-solution

Lucamarini, M.; Yuan, Z. L.; Dynes, J. F.; Shields, A. J. (May 2018). Overcoming the rate–distance limit of quantum key distribution without quantum repeaters. *Nature*, *557*(7705), 400–403. https://doi.org/10.1038/s41586-018-0066-6

Mayers, Dominic; Yao, Andrew (14 September 1998). Quantum Cryptography with Imperfect Apparatus. https://doi.org/10.48550/arXiv.quant-ph/9809039

National Security Agency. (n.d.). *Quantum Key Distribution (QKD) and Quantum Cryptography (QC)*. NSA/CSS. https://www.nsa.gov/Cybersecurity/Quantum-Key-Distribution-QKD-and-Quantum-Cryptography-QC/

National Institute of Standards and Technology. (2017, January 03). *Post-Quantum Cryptography Standardization.* NIST. https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization

National Institute of Standards and Technology. (2022, July 05). *NIST Announces First Four Quantum-Resistant Cryptographic Algorithms.* NIST. https://www.nist.gov/news-events/news/2022/07/nist-announces-first-four-quantum-resistant-cryptographic-algorithms

Nielsen, M. A., & Chuang, I. L. (2000). Quantum Computation and Quantum Information. Cambridge University Press.

Quantum Xchange. (2022, December 15). Quantum Entanglement Communication: Quantum Xchange. QuantumXC. https://quantumxc.com/blog/is-quantum-communication-faster-than-the-speed-of-light/

Red Hat. (2023, December 21). *Stateful vs stateless*. Red Hat. https://www.redhat.com/en/topics/cloud-native-apps/stateful-vs-stateless

Regev, Oded (2009). On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, *56*(6), 1–40. https://doi.org/10.1145/1568318.1568324

Rivest, R., Shamir, A., & Adleman, L. (1978). "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems." Communications of the ACM, 21(2), 120-126.

Sheldon, R., Loshin, P., & Cobb, M. (2024, February 7). What is encryption and how does it work?: Definition from TechTarget. Security. https://www.techtarget.com/searchsecurity/definition/encryption

Shor, P. (1994). "Algorithms for quantum computation: Discrete logarithms and factoring." Proceedings of the 35th Annual Symposium on the Foundations of Computer Science.

Silverman, J. H. (2015, January 12-16). *NTRU and Lattice-Based Crypto: Past, Present, and Future.* [Lecture notes] Rutgers University. http://archive.dimacs.rutgers.edu/Workshops/Post-Quantum/Slides/Silverman.pdf

Steane, A. (1997). "Quantum Computing." Cornell University Press.

Stein, B. P. (2004, May). *Cost Effective QKD System Developed By NIST*. NIST. https://www.nist.gov/itl/cost-effective-qkd-system-developed-nist

Takeoka, Masahiro; Guha, Saikat; Wilde, Mark M. (24 October 2014). Fundamental rate-loss tradeoff for optical quantum key distribution. *Nature Communications*, *5*(1), 5235. https://doi.org/10.1038/ncomms6235

Vandersypen, L., et al. (2001). "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance." Nature, 414, 883–887.

Varanasi, L. (2023, December 17). Brace yourself for 'Q-Day,' a global cybersecurity event that could expose our most important secrets. *Business Insider*. https://www.businessinsider.com/q-day-2025-cybersecurity-quantum-computing-data-security-privacy-china-2023-12

Veritasium. (2023, March 20). *How Quantum Computers Break The Internet… Starting Now* [Video]. YouTube. https://www.youtube.com/watch?v=-UrdExQW0cs&t=802s

Wilkins, A. (2023, October 24). Record-breaking quantum computer has more than 1000 qubits. *New Scientist.* https://www.newscientist.com/article/2399246-record-breaking-quantum-computer-has-more-than-1000-qubits/