The Vulnerabilities to the RSA Algorithm and Future Alternative Algorithms to Improve Security

James Johnson
William & Mary

The Vulnerabilities to the RSA Algorithm and Future Alternative Algorithms to Improve Security

Researcher: James Johnson
Mentor: Chris Shenefiel, Adjunct Professor of William and Mary
William and Mary

Fall 2023

Abstract

The RSA encryption algorithm has secured many of the largest systems, which include bank systems, data encryption in emails, several online transactions, etc. Benefiting from the use of asymmetric cryptography and properties of number theory, RSA was widely regarded as one of most difficult algorithms to decrypt without a key, especially since by brute force, breaking the algorithm would take thousands of years. However, in recent times, research has shown that RSA is getting closer to being efficiently decrypted classically, using algebraic methods, (fully cracked through limited bits) in which elliptic-curve cryptography has been thought of as the alternative that is stronger than RSA. However, the biggest issue that faces RSA, as well as other cryptographic algorithms, such as elliptic curve, is the development of quantum computing. Mathematically, several algorithms, such as Shor's algorithm, have been proven to decrypt RSA's algorithm within a span of hours, using a quantum computer, meaning our security systems are at risk of collapsing. Research needs to address this issue, as security is compromised, and several algorithms have been created to become quantum resistant. Through classical methods (post-quantum cryptography), hash and lattice-based algorithms have used properties of group theory and number theory to create irreversible functions that even a quantum computer would not be able to decrypt. Additionally, the emergence of quantum cryptography has led research towards devising encryption algorithms based on quantum-mechanical properties that would allow for eavesdropping to be detected and for systems to remain secure.

1 Introduction

RSA's algorithm has been considered the standard for encryption algorithms that is used to protect some of the biggest systems in the nation, including VPNs, email services, web browsers, etc. The encryption is a form of asymmetric cryptography, meaning it requires a public and private key. Take two people, Alice and Bob, in which Alice has the public key that anyone can access with Bob having its associated private key that only he controls, and vice versa. Alice is able to encrypt her message/data with her public key (available to the public), but only Bob has the private key that is able to decrypt the information; hence, Alice and Bob have a secure way of being able to communicate with each other without anyone eavesdropping [1]. For RSA's algorithm, the public key decrypts a message into an incredibly large number that is the product of two large numbers and the private key is the two prime numbers that factor into the public key's large number. The idea is that it is incredibly hard to find the two prime numbers and is then secure, as it would take a computer incredibly long to find them. The two biggest questions are: how to check if the two numbers are prime and why exactly it takes so long to decrypt.

1.1 Primality Test

In order for RSA to be effective, we need a way to generate two large enough prime numbers. Luckily, we have primality tests that can determine whether a number is prime, and can do so in a fairly efficient amount of time. The most used primality test is known as the Miller-Rabin test:

Let n be a large number you are trying to check is prime and let $a, b \in \mathbb{Z}$. Then, consider

$$n - 1 = a2^b$$

where a is odd. Now we choose $c \in \mathbb{Z}/n\mathbb{Z} = \{[0]_n, [1]_n, ..., [n-1]_n\}$. Then we can say n is prime if either:

(i) $c^a \equiv 1 \pmod{n}$
(ii) $c^{a2^i} \equiv -1 \pmod{n}$ for some $i \in \{0, 1, ..., b-1\}$

The proof for the test working is simply using the contrapositive of Fermat's Little Theorem:

Theorem 1. *If p is a prime number, then for any $a \in \mathbb{Z}$, $a^{p-1} \equiv 1 \pmod{p}$*

This method of testing is not in fact deterministic, but we can say with great confidence that the number we check is prime if the proceeding conditions are satisfied. [2]

1.2 Inefficiency in Decrypting RSA

Now that we can effectively implement RSA by finding the two large primes, it is important to understand why RSA is so effective, in which we will see through the standard "best" model in decrypting the algorithm:

Let $N = pq$, where p, q are prime numbers.

Step 1: Take $m \in \mathbb{Z}_+$ such that $m < N$. Calculate $gcd(m, N)$:

- If $gcd(m, N) \neq 1$, then you got lucky and you are done (extremely unlikely to happen)
- Else, continue

Step 2: Let $f : \mathbb{Z}_+ \to \mathbb{Z}_+$ such that $f(a) = m^a \pmod N$ and find the period p such that $f(0) = 1 \pmod N = m^p \pmod N = f(p)$:

- If p is odd, go back to the beginning and choose a different m
- Else, continue

Step 3: Since p is even, $(m^{p/2} - 1)(m^{p/2} + 1) = m^p - 1 = 0 \pmod N$:

- If $m^{p/2} \pm 1 \neq 0 \pmod N$, go back to the beginning and choose a different m
- Else, use Euclidean Algorithm to find p, q

The runtime for the algorithm is around $O(\sqrt{N})$ and for a 2048 digit number, it would take a classical computer around 300 trillion years and as a result, RSA, in practice, has been very effective in keeping information private. However, with the emergence of various research ideas, in both classical and quantum methods, RSA is proving to be extremely vulnerable for the future and will end up being ineffective. [3]

2 Methods to Breaking RSA

Many classical and quantum approaches have been developed in order to break RSA. Many researchers have been trying to create computers that are more powerful to use the standard approach of identifying the prime numbers and have been slowly, but surely, working towards a computer that is powerful enough to break RSA. Other more successful methods have involved researchers creating more efficient algorithms to determine the prime numbers and have used advanced concepts of number theory and linear algebra to develop these algorithms, whether it is classical or quantum. Two of the biggest algorithms created involve one classical approach, general number field sieve, and a quantum approach, Shor's algorithm.

2.1 Classical Approach: General Number Field Sieve

One of the biggest successes have come through General Number Field Sieve Theory, in which a more efficient algorithm has been developed to find the two prime numbers:

Let $N = pq$, where p, q are prime numbers. Let $s, r \in \mathbb{Z}$ such that $s^2 \equiv r^2 \pmod{N}$. Then:

$$\begin{aligned} s^2 &\equiv r^2 \pmod{N} \\ \rightarrow s^2 - r^2 &\equiv 0 \pmod{N} \\ \rightarrow pq|(s^2 - r^2) \\ \rightarrow pq|(s-r)(s+r) \end{aligned}$$

Since p and q are prime, p cannot divide both $s - r$ and $s + r$, and same with q. Hence:

$$\begin{cases} p|(s-r) \vee p|(s+r) \\ q|(s-r) \vee q|(s+r) \end{cases}$$

Then, you can create a table of possibilities to determine what values of p and q are possible; then, you have a probability of $\frac{2}{3}$ that you found the prime factors.
This method uses a fair amount of number theory but it is important to understand exactly why this works:

Consider a polynomial of degree d:

$$f(x) = a_0 + a_1 x + a_2 x^2 + ... + a_d x^d$$

Let $m \in \mathbb{Z}^+$ such that:

$$f(m) = a_0 + a_1 m a_2 m^2 + ... + a_d m^d \equiv 0 \pmod{N}$$

Consider $\theta \in \mathbb{C}$ and the ring $\mathbb{Z}[\theta]$. Let $a(\theta) = A, b(\theta) = B \in \mathbb{Z}[\theta]$. Using Euclidean Algorithm:

$$\begin{aligned} a(\theta)b(\theta) &= e(\theta)f(\theta) + c(\theta) \\ \rightarrow a(\theta)b(\theta) &\equiv c(\theta) \pmod{f(\theta)} \end{aligned}$$

Theorem 2. *Let $f(x)$ be a polynomial with $\mathbb{Z}$ coefficients, $\theta \in \mathbb{C}$, $m \in \mathbb{Z}/N\mathbb{Z}$ where $f(m) \equiv 0 \pmod{N}$. Then there exists a unique mapping $\phi : \mathbb{Z}[\theta] \rightarrow \mathbb{Z}/N\mathbb{Z}$ such that:*
(i) $\phi(a + b) = \phi(a) + \phi(b)$
(ii) $\phi(ab) = \phi(a)\phi(b)$
(iii) $\phi(1) = 1$
(iv) $\phi(\theta) \equiv m \pmod{N}$

You can prove the theorem using properties of ring homomorphisms and properties of uniqueness to show $\phi(\theta) \equiv 0 \pmod{N}$. [4]

Now consider $\alpha^2 \in \mathbb{Z}[\theta]$ where α^2 is a perfect square and $y^2 \in \mathbb{Z}[\theta]$ where β^2 is a perfect square. Since $\mathbb{Z}[\theta], \mathbb{Z}/N\mathbb{Z}$ are rings, they are closed under multiplication. Hence, there exists a finite set U with (a, b) coordinates such that:

$$\Pi_{(a,b)\in U}(a + b\theta) = \alpha^2$$
$$\Pi_{(a,b)\in U}(a + bm) = y^2$$

Now let $x = \phi(\beta)$. From the properties and the previous theorem:

$$\begin{aligned} x^2 &= \phi(\beta) \cdot \phi(\beta) \\ &= \phi(\beta^2) \\ &= \phi(\Pi_{(a,b)\in U}(a + b\theta)) \\ &= \Pi_{(a,b)\in U}\phi(a + b\theta) \\ &= \Pi_{(a,b)\in U}(a + bm) \\ &= y^2 \end{aligned}$$

Hence, a factorization exists since we've shown $x^2 \equiv y^2 \pmod{N}$, meaning $x^2 - y^2 \equiv 0 \pmod{N}$.

In order to find the perfect squares, we use the ideas of smoothness and using sieving techniques, we find the pair (a, b) where $a + b\theta$ is smooth over an algebraic number base and the $a + bm$ is smooth over an a rational factor base, in which then we create vector representations and the Legendre symbol to find the perfect squares (for more information, reference [4]).

General number field sieve has been implemented and is successful for numbers around 768 bits (232 digit number), which takes approximately a couple weeks to process, significantly faster than our current best algorithm. [5] Researchers are projected to break numbers of 1024 bit length and will eventually work their way towards breaking 2048 bit numbers, highlighting the vulnerabilities of RSA. We do not have a projected date to which such classical methods will break RSA, but there have been several competitions and offers that have attracted more research towards this area and as cryptography and mathematics is a constantly changing field, the uncertainty of when RSA can be broken should raise concern [5]. However, quantum algorithms are able to quickly factor 2048 bit-length numbers and Shor's algorithm poses the biggest threat to RSA.

2.2 Quantum Approach: Shor's Algorithm

Shor's algorithm does not directly decrypt RSA; however, the algorithm is extremely effective in finding the periods of functions and recalling from Section 1.2, finding the period of function f is what makes decrypting RSA extremely inefficient. [3]

When dealing with a quantum system, we consider qubits $|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}$ and $|1\rangle = \begin{pmatrix}1\\0\end{pmatrix}$. In order to make a system "quantum", we need to entangle the qubits, meaning measuring one of the qubits will affect the measurement of the other. First, we create an orthonormal basis of vectors, say $\{v_1, v_2, ..., v_n\}$, meaning we create a set of the least amount of vectors that will span the vector space and $\forall i, j$ such that $v_i \neq v_j$, $v_i \cdot v_j = 0$. (Orthonormal basis allow for clear ways to represent different vectors in a quantum system). Another important concept is the idea of superposition, which is simply the linear combination of qubits associated with different probabilities, in which the probabilities give the likelihood of measuring a certain outcome. Our goal is to manipulate quantum systems in which we can maintain probability amplitudes, yet be able to receive certain information at a much more efficient rate (with the use of a quantum computer). Now we define a few definitions that are important in manipulating quantum systems:

Definition 1. *A matrix M is normal if $MM^\dagger = M^\dagger M$, where $M^\dagger$ is the complex conjugate of the matrix.*

Definition 2. *A matrix U is unitary if $UU^\dagger = I_n$*

Unitary matrices are important in performing unitary transformations, since the transformations rotate the axes of the qubit vectors, but maintain their normality and hence, preserve probability amplitudes. [3] Now, through these essential ideas and principles of quantum circuits and entanglement, Shor's Algorithm involves two fundamental processes.

Quantum Fourier Transform:

Take $|x\rangle = |x_n x_{n-1} ... x_1 x_0\rangle$ and let $N = 2^n$. Then:

$$|x*\rangle = \frac{1}{\sqrt{N}}(|0\rangle + e^{\frac{2\pi i x_1}{2^1}}|1\rangle) \otimes (|0\rangle + e^{\frac{2\pi i x_2}{2^2}}|1\rangle) \otimes ... \otimes (|0\rangle + e^{\frac{2\pi i x_n}{2^n}}|1\rangle)$$

In order to generate the quantum circut for QFT:
(1) Hadamard Gate:

$$H|x_k\rangle = \begin{cases} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) & x_k = 0 \\ \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) & x_k = 1 \end{cases}$$
$$= \frac{1}{\sqrt{2}}(|0\rangle + e^{\frac{2\pi i x_k}{2}}|1\rangle)[6]$$

(2) Unitary Rotation:

$$UROT_k|x_j\rangle = e^{\frac{2\pi i x_j}{2^k}}|x_j\rangle$$
$$x_j = 0 \rightarrow e^{\frac{2\pi i x_j}{2^k}}|x_j\rangle = 0$$
$$x_j = 1 \rightarrow e^{\frac{2\pi i x_j}{2^k}}|x_j\rangle = e^{\frac{2\pi i}{2^k}}|1\rangle$$
$$\rightarrow UROT_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{pmatrix}$$

which applies the phase $e^{\frac{2\pi i}{2^k}}$ for the state $|1\rangle$.

Putting it all together, we form the quantum fourier transform:
Step 0: $|x\rangle = |x_1 x_2 ... x_n\rangle$
Step 1: $[|0\rangle + e^{\frac{2\pi i x_1}{2}}|1\rangle] \otimes |x_2 x_3 ... x_n\rangle$
Step 2: $[|0\rangle + e^{\frac{2\pi i x_2}{2^2}} e^{\frac{2\pi i x_1}{2}}|1\rangle] \otimes |x_2 x_3 ... x_n\rangle$
Step n: $[|0\rangle + e^{\frac{2\pi i x_n}{2^n}} ... e^{\frac{2\pi i x_2}{2^2}} e^{\frac{2\pi i x_1}{2}}|1\rangle] \otimes |x_2 x_3 ... x_n\rangle$ [6]

Quantum Phase Estimation:
Given $U|\psi\rangle = e^{i\theta_\psi}|\psi\rangle$, we can extract θ_ψ given the ability to prepare $|\psi\rangle$ and the ability to prepare $|u\rangle$:

Step 0: $|0\rangle|\psi\rangle$
Step 1: $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle|\psi\rangle + |1\rangle|\psi\rangle)$
Step 2: $\frac{1}{\sqrt{2}}(|0\rangle|\psi\rangle + |1\rangle e^{i\theta_\psi}|\psi\rangle)$
Step 3: $\frac{1}{\sqrt{2}}((\frac{|0\rangle+|1\rangle}{\sqrt{2}})|\psi\rangle + e^{i\theta_\psi}(\frac{|0\rangle-|1\rangle}{\sqrt{2}})|\psi\rangle = \frac{1}{2}[|0\rangle(1+e^{i\theta_\psi}) + |1\rangle(1-e^{i\theta_\psi})]\psi\rangle$
Prob(measuring $|0\rangle$)= $|\frac{1}{2}(1+e^{i\theta_\psi})|^2$
Prob(measuring $|1\rangle$)= $|\frac{1}{2}(1-e^{i\theta_\psi})|^2$

Then, the experiment uses 1 qubit to measure θ_ψ: $\theta_\psi = 1°$: Prob(0), Prob(1)= $\{0.999, 7.6 \times 10^{-5}\}$
$\theta_\psi = 10°$: Prob(0), Prob(1)= $\{0.9924, 0.007596\}$
which is very accurate. [6]

In RSA, we take a look at step 2 of Section 1.2, which is the challenge of implementing the algorithm in a reasonable amount of time. When using, Shor's algorithm, we can use quantum fourier transform and quantum phase estimation to find the period p such that for $f(a) = m^a \pmod N$, $f(0) = f(p)$ and would then significantly decrease the time to find the period and in total, find the two prime factors of the integer. [3]

Mathematically, Shor's algorithm breaks RSA and the only thing preventing Shor's algorithm from practically being implemented is the creation of a functioning quantum computer. Many environmental factors make it difficult to transmit photons and researchers have worked towards developing methods in reducing the variability caused by the environment and being able to engineer a working quantum computer. Many question whether a quantum computer is even possible; however, as time progresses, it seems quantum computers are being more and more likely to being attainable. While a classical computer takes trillions of years to decrypt RSA, a quantum computer only takes around 3 months to break the algorithm, an incredible difference in runtime and incredibly worrisome for several security systems in our world. [3]

3 Alternative Encryption Algorithms to Replace RSA

While there are several algorithms that are working towards breaking RSA, there are several algorithms that are resistant to current methods or there are no current known methods to break these algorithms.

3.1 Elliptic Curve Cryptography

One cryptographic algorithm that is resistant to current classical methods is elliptic curve cryptography, based on a curve that is in the form: $y^2 = x^3 + Ax + B$ (known as the Weierstrauss form). The curve has certain special properties:

(1) Symmetric: If (x, y) satisfies elliptic curve E, then $(x, -y)$ also does
(2) Let y' be the tangent of y. $y' = \infty$ when $y = 0$
(3) E has either 1 or 2 components, but for simplicity, we only deal with 1 component, which has 1 distinct real root.

Elliptic curves also contain special additive laws in order to get unique results from adding points on the curve:Let $P_1, P_2 \in E$ be points on the elliptic curve. Then the line passing through P_1 and P_2 passes through another point on E.

Definition 3 (Group Law 1)**.** *Let $P_1, P_2 \in E$ and $Q = (x', y')$ be the other intersection on E of the line passing through P_1 and P_2. Then we say*

$$P_1 := -Q := (x', -y') \tag{1}$$

If $P_1 + P_2$ form a vertical line, $P_1 + P_2 = \infty$

To add a point to itself:

Definition 4 (Group Law 2)**.** *Let $P \in E$, $Q = (x', y')$ on E such that the tangent line to P intersects Q; Then we say:*

$$P + P := -Q = (x', -y') \tag{2}$$

Now we look at elliptic curves under Modulo p: Take E modulo N. Then:

$$P_1 + P_2 = -Q = (x', -y') \equiv (z_1', z_2') \pmod{N}$$

This gives a certain number of possible points and the larger the N, the more possible values.

Along with the elliptic curves themselves, we can look at the order of points:

Definition 5. *Let $kP := P + P + ... + P$ (k times). The order of p is the minimum k such that $kP = \infty$*

The idea of elliptic curve cryptography is to select a public key P_0 that is generated by a given elliptic curve and the private key is k, which is not necessarily the order, but it follows that $P_0 = kP$. The idea is that it is a trapdoor function: given a k, it is easy to find P_0, but doing the reverse is much more difficult, especially given a large enough N to be the modulo. In most instances, the graph of the curve is not given and instead is a grid representation of several clumped points which do not indicate any sort of correlation; hence, classically, it is currently impossible to devise an efficient algorithm to decrypt elliptic curve. [7]

The fastest algorithm known to decrypt it is Pollard Rho's algorithm, which has a runtime of $O(\sqrt{n})$, which as seen with RSA, given large enough number, the algorithm is extremely inefficient. However, elliptic curve cryptography is only resistant to classical decryption methods; if quantum computing is possible, using Pollard Rho's Algorithm and Shor's algorithm, the encryption would be incredibly vulnerable, even though it is the current closest algorithm to replace RSA (it would likely serve as more of a "temporary" replacement, although would be incredibly effective in the long-term if quantum computing is not physically possible).

3.2 Post-Quantum Cryptography

As demonstrated, elliptic curve cryptography is an effective algorithm restricting to classical methods, but the involvement of quantum methods makes elliptic curve as vulnerable as RSA. Researchers have gotten heavily involved in discovering various algorithms that would secure systems and are resistant to both classical and quantum methods.

BB84:
One way to resist quantum methods of decrypting security systems is to create a quantum cryptographic model. Many have been mathematically developed, one most notably being BB84:

Suppose you have a sender and receiver, say Alice and Bob, and Alice wants to send information to Bob privately. BB84 states that Alice can translate the message into a series of qubits and with Alice and Bob having the same bases, which send and receive the qubits, Alice can randomly send qubits through those bases and Bob measures them randomly through his chosen bases (as allowed by quantum properties). To compare, Alice and Bob can communicate with each other publically about which qubits were sent and received through certain channels to understand the measurements of the qubits by the given bases. Suppose there was an eavesdropper, Eve, who intercepted qubits and sent qubits back to Bob. Through an idea of No Cloning Theorem, if Eve measures the intercepted qubit, then Eve cannot "clone" the qubit to send and after measurement, the qubit is affected and sent through a random channel. While Alice and Bob are exchanged the bases they used for each qubit, because of randomness, Alice and Bob can detect which measurements do not align with the chosen bases, in which they are alerted that an eavesdropper intercepted the message. [3]

These quantum algorithms certainly are "more secure" than RSA; however, these methods only alert you of an interception and does not necessarily prevent it. Moreover, it would require a quantum computer to work, in which by then, Shor's algorithm would already be established and would have already destructed RSA. Luckily, there exist a couple methods that are classical and quantum resistant.

Cryptographic Hash Algorithm:
Cryptographic hash algorithms input a message and they return hash values: a fixed length output composed of letters, numbers and symbols. Hash algorithms contain two special properties, which are incredibly useful for encryption:

(1) Collision Resistant: The values from the function are unique, meaning there are no two strings that can be inputted into the hash function to create the same string
(2) Irreversibility: It is easy to convert a given string into a hash value, but nearly impossible to convert a hash value back to its original string

Hence, hash algorithms can be considered trapdoor functions, which are needed for encryption to be quantum-secure [8]. Researchers have already developed several hash algorithms (MD5, SHA-256, BLAKE2, etc.), and while some are possible to crack with quantum methods, irreversibility and and the size of the hash generated by the algorithm have made several quantum-resistant algorithms, essentially making them effective. [8]

Lattice-Based Cryptography:
Lattice-Based Cryptography is developed from the algebraic structure of a lattice:

Definition 6. *A lattice is a space of infinite points in* $\mathbb{R}^n$ *such that points are closed under addition and subtraction and* $\exists m, M > 0$ *such that any two points have a distance greater than* m *and less than* M

The security of the encryption is based on the Shortest Vector Problem: Find the shortest nonzero lattice vector. In other words, we want to create a linear combination of vectors in the lattice vector space, such that the resulting vector is the shortest vector in the space. Without knowing the distances for all the points and the different possible coefficients for the linear combinations, it makes the problem incredibly difficult to solve and given the right lattice structure, an encryption algorithm can involve the private key being the shortest vector and makes the system secure. [9]

According to the National Institute of Standards and Technology, there have been four lattice-based cryptographic algorithms that have proven to be quantum-safe: CRYSTALS-Kyber [12], CRYSTALS-Dilithium [11], and FALCON [10]. Implementing the algorithms have proven to work and with more research towards these variations of algorithms, lattice-based algorithms could be the closest to practically being implemented and being safe from both classical and quantum methods.

4 Conclusion

It is clear RSA has clear vulnerabilities that have been exposed by several algorithms, both classical and quantum. Researchers continue to work towards creating new algorithms to break RSA that can be practically implement, as well as work towards creating classical and quantum computers strong and accurate enough to implement these decryption methods. However, there have been alternative algorithms proven to be resistant to the emergent methods of decryption, each having their advantages and disadvantages. In terms of overall effectiveness and practicality, the algorithms that use quantum computing, such as BB84, should be lower priority, as the algorithm simply alerts of message interception and is only implemented if quantum computers exist. Elliptic curve cryptography is likely the closest to actually being implemented and it would be nearly impossible to find an effective classical decryption method, Shor's algorithm would make it extremely vulnerable, making it risky to prioritize and establish as a long-term solution (but possible for the short-term). Lattice-based cryptography and cryptographic hash algorithms seem to be the two best options as a improvement for RSA, as they are both resistant to classical and quantum methods. Although researchers continue to work towards obtaining a solution towards the "shortest vector problem" and hash algorithms being incredibly complex to implement in code, the two methods would be the best step to take in order to ensure a more secure future. Even with such known algorithms given, researchers still need to continue advancing further in our understanding of mathematics and computer science, in order to find more applications towards creating more new secure algorithms, as researchers are likely to find methods of decrypting for even what we consider to be "full-proof" in the present-day.

References

[1] pyca/cryptography. "Asymmetric Algorithms." Asymmetric Algorithms - Cryptography 42.0.0.Dev1 Documentation, Individual Contributors, 2013, cryptography.io/en/latest/hazmat/primitives/asymmetric/.

[2] Lynn, Ben. "Primality Tests." Number Theory - Primality Tests, Stanford University, crypto.stanford.edu/pbc/notes/numbertheory/millerrabin.html. Accessed 1 Dec. 2023.

[3] Li, Chi-Kwong, and Mikio Nakahara. An Invitation to Quantum Information Science and Quantum Computing. 2023. This is a rough draft of the textbook, but all the information is accurate.

[4] Case, Michael. University of Maryland, College Park, MD, A Beginner's Guide to General Number Field Sieve.

[5] Kleinjung, Thorsten, et al. International Association for Cryptologic Research, Factorization of 768-Bit RSA Modulus.

[6] Qiskit. Shor's Algorithm I: Understanding Quantum Fourier Transform, Quantum Phase Estimation. Youtube, 1 Sept. 2020, https://www.youtube.com/watch?v=mAHC1dWKNYE. Accessed 1 Dec. 2023.

[7] Dummit, Evan. Northeastern University, Boston, MA, 2021, Elliptic Curves.

[8] CSRC Content. "Cryptographic Hash Function - Glossary: CSRC." CSRC Content Editor, National Institute of Standards and Technology, csrc.nist.gov/glossary/term/cryptographic-hash-function. Accessed 1 Dec. 2023.

[9] Wickr. "What Is Lattice-Based Cryptography andamp; Why You Should Care." Medium, Wickr Crypto + Privacy Blog, 15 Aug. 2020, medium.com/cryptoblog/what-is-lattice-based-cryptography-why-should-you-care-dbf9957ab717.

[10] Pornin, Thomas. NNC Group, New Efficient, Constant Time Implementations for Falcon.

[11] Schwabe, Peter. "Crystals." Dilithium, 26 Feb. 2021, pq-crystals.org/dilithium/index.shtml.

[12] Schwabe, Peter. "Crystals." Kyber, 23 Dec. 2020, pq-crystals.org/kyber/index.shtml.