To what extent does the application of prime numbers in RSA cryptography contribute to data security, and how do the mathematical principles behind RSA encryption affect its real-world applications and vulnerabilities?

Mathematics: Analysis and Approaches Higher Level

Word count: 3562

2024

Table of Contents

1. Introduction

Lacking a security system that would allow users to explore the digital world safely is unfathomable in the digital age since the majority of information is based on the Internet. Almost all security measures developed to keep users safe are based on cryptography. Cryptography is a method of protecting information and communications through the use of codes so that only those for whom the information is intended can read and process it (Richards). Many different cryptography methods are based on mathematical properties. These mathematical-based encryption methods are further evidence that mathematics is a practical science with a wide range of real-world applications.

The Rivest-Shamir-Adleman (RSA) encryption method, an asymmetric cryptography method based on prime numbers and modular arithmetic, will be explicitly covered in this essay. The name “RSA” comes from its inventors, Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman, who created it while on the faculty at MIT. Although Clifford Cock had previously invented an equivalent system for the British GCHQ back in 1973, the “RSA” project was officially published in 1977 (Telecom Italia). RSA is one of the most widely used public key cryptography algorithms in the world, as it is used to encrypt messages and digital signatures. Its security is based on the difficulty of predicting mathematical problems such as the factorization of large numbers. In "asymmetric" encryption or “public key encryption”, there are two separate keys that are used in the process. A private key is kept secret and used for decryption, while a public key is for encryption.

Research Question: To what extent does the application of prime numbers in RSA cryptography contribute to data security, and how do the mathematical principles behind RSA encryption affect its real-world applications and vulnerabilities?

This essay aims to analyze and investigate the mathematical part underlying this encryption method. Step by step we will explore the algorithm that RSA follows, finding its strengths and weaknesses, providing a deeper understanding of the key generation progress, encryption, and decryption algorithms of RSA, as well as the implications, applications, and mathematical foundations of this encryption method. Upon perusing this essay, the readers will attain a heightened comprehension of cryptosystems in a broader context and their paramount significance in the realm of cybersecurity.

2. Background and Theory

But, how does it work? It is essential to familiarize yourself with the fundamentals of number theory since they form the basis of how RSA functions. Understanding RSA cryptography's key generation, encryption, and decryption methods requires an understanding of ideas like prime numbers, modular arithmetic, and congruence. RSA encryption generally operates on the basis that the technique is simple to calculate in one way but nearly hard to calculate in the opposite direction (also known as trap door functions). Data can be encrypted with one key using the trap door functions properties in a way that can only be unlocked by the pair's private key. The mathematical basis behind trap functions is a method called prime factorization.

2.1 Prime factorization

Basically, **prime factorization** is the process of writing any number as a product of primes ("Prime Factorization Numbers"). A **prime number** is any number that can only be divided by 1 or itself. Prime factorization stands as a formidable barrier, rendering encryption exceptionally challenging to break. To illustrate this point, consider the selection of two random integers $p = 31$, $q = 42$. This signifies that 1302 is the product of combining these numbers together. Forget the p and q now. Would you be able to determine the two numbers that makeup 1302 just by knowing the product if you were informed that it is a product of two numbers? This example underscores the significance of prime factors, emphasizing their role in the security of encryption. Consider how challenging this method would be with even more tough numbers. In reality, it is almost impossible to calculate greater figures with today's processing capacity. The size of the primes in a real RSA implementation varies, but in 2048-bit RSA, they would come together to make keys that are 617 digits long (Lake). This illustration is used as a model for the creation of public keys.

2.2 Modular Arithmetic

As mentioned already, another mathematical concept that is vital for understanding RSA cryptography is modular arithmetic. **Modular arithmetic** is a system of arithmetic for integers, which considers the remainder. In modular arithmetic, numbers "wrap around" upon reaching a given fixed quantity (this given quantity is known as the modulus) to leave a remainder (Lake). Modular arithmetic is often used in asymmetric cryptosystems, as values can reset to zero when they reach a certain modulus value, which can be helpful when it comes to decrypting a message. The most important thing is the exploration of **modulo**, a math operation that finds the remainder when one integer is divided by another. In writing, it is frequently abbreviated as mod, or

represented by the symbol % (Computer Hope). To provide an example, when we compute 11 modulo 4, we are simply determining the residual that occurs from dividing 11 by 4. The modulo procedure returns a value of 3, indicating that $11 mod 4 = 3$. Another term that would be necessary, is the inverse modulo. A **modular inverse** of an integer A mod C is the B value that makes $A \times BmodC = 1$.

2.3 Euler's totient function $\varphi(n)$ /Totative

The next mathematical property that we need to know is Euler's totient function (φ), which is based on the prime numbers theory. The **totient** $\varphi(n)$ of a positive integer n is defined to be the number of positive integers less than or equal to n that are coprime or relatively prime (two numbers that have only 1 as common factor) to n (Marius Tărnăuceanu). For example, the **totative** of the number 8 is φ(8)= 4 because there are 4 integers less than and coprime with 8 which are 1,3,5 and 7. Because this number is objectively small when it comes to the actual number used for encryption, we can calculate the totative number by just counting them. But what happens in reality? How can the totative of a huge number be calculated fast and easily?

To start finding a general formula that will make the key process easier, we have to recall some mathematical equations, such as

$$\varphi(n) = \text{ the number of elements that are relatively prime to } n$$

$$n = pq \text{ (where p and q are prime numbers.)}$$

To get (n), the number of elements that are relatively prime to n, we can begin by computing the number of elements that are not relatively prime to p or q. Then we may get (n) by subtracting this count from n. We know that the elements that are not relatively prime to p and q must have at least p or q as one of its factors. Putting every number that has p and q as a factor we get:

$0p, 1p, 2p, \ldots, kp, \ldots, p(q-1)$ (There are q elements in total)

$0q, 1q, 2q, \ldots, jq, \ldots, q(p-1)$ (There are p elements in total)

Also, we have to take into consideration the number pq cause it can obviously be divided by both numbers p and q. In order to include the number we are just going to add 1.

That gives a count of

$$n - p - q + 1$$

Knowing that $n = pq$, we have:

$$= pq - p - q + 1 = p(q-1) - (q-1) = (q-1)(p-1)$$

In fact, this method is the derivative of **Euler's function** which will be helpful for understanding the RSA cryptosystem (Parker). By using Euler's function $\varphi(N) = (p-1)(q-1)$, we are able to figure out a general and fast way to find the totative of any prime number, no matter the number's size.

3. Generation keys process

To start the process of encryption, it is necessary to generate the keys. In order to do so, we need two prime numbers, which will be selected through the primality test algorithm, which states if a number is prime or not. I'll choose small numbers, such as $p = 7$ and $q = 11$, to make it simpler and easier to grasp. After finding two prime numbers p and q user "A" follows a step-by-step algorithm called the algorithm of encryption.

1st step: Find the product of the numbers

The first step is to find N which is a number created by multiplying p and q *(*$N = p \times q$*)*. The product of these numbers $N = 77$ is also important as it will be a part of the public key.

2nd step: Calculate $\varphi(n)$ *function*

In order to do so, we are going to use the Euler's formula that we proved before.

$$\varphi(N) = (p - 1) \times (q - 1) = (7 - 1) \times (11 - 1) = 6 \times 10 = 60.$$

3rd step: Find the number "e*"*

The other component of the public key is the number "e" which must meet specific criteria and isn't chosen randomly. The first rule is that e must fall within the range of 1 and $\varphi(n)$ ($1 < e < \varphi(n)$). The second property is that e has to be coprime with $\varphi(n)$, because if e shares common factors with $\varphi(n)$ it could create vulnerabilities in the encryption. So by being coprime with, $\varphi(n)$ we ensure that the encryption process remains a one-way function. Using an online tool, I found that the number $e = 7$ meets both criteria ("Coprime Calculator"). It could be any number that meets both criteria, such as 11, 13, 17, 29, 31, etc, but I purposely selected the lowest one as it will make our example easier to understand. In general, the higher e is, the harder is to break the encryption.

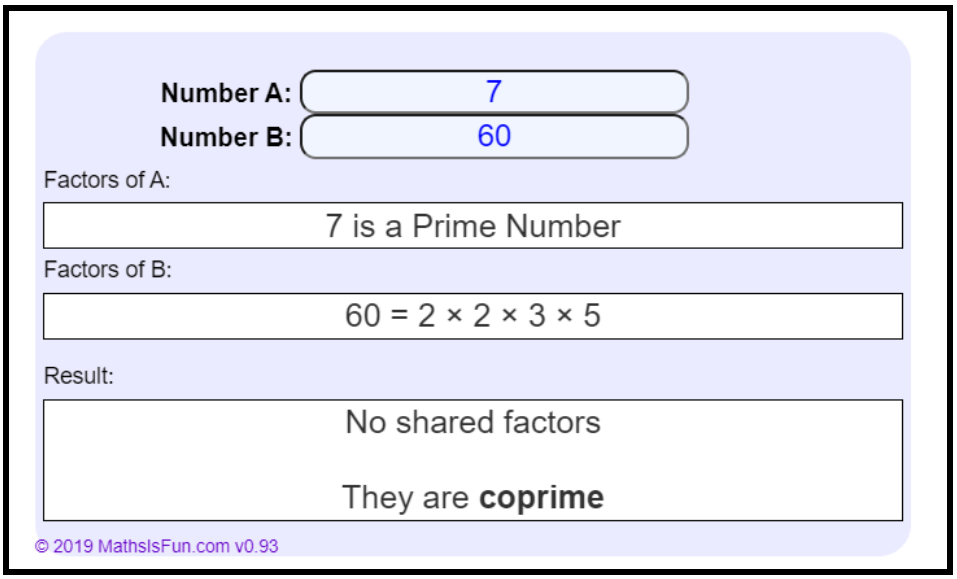

Figure 1: Screenshot taken by me

So the final public key for user A will be $(e,\ N)$, and more specifically $(7,\ 77)$. The fact that N can be divided by e doesn't give any problems or vulnerabilities to our system as the attacker cannot use this information accurately. In fact, this key will be the tool for any encryption involving user A.

As RSA is an asymmetric cryptographic method, a second requirement is the private key which will be in a form $(d, \varphi(N))$. This key is used to decipher the encrypted message and reveal its original content. There is no other way, except knowing this key, that can encrypt the message. Using Euclid's extended algorithm, we are able to compute d which will be the inverse of e modulo $\varphi(N)$. As mentioned earlier, a part of the private key is represented by the number "d" (decryption), which must adhere to the equation: $d \equiv \frac{1}{e}\ (mod\ \varphi(n))$. This equation seems to be asking you to divide 1 by 7, but this is not the actual case. In fact, its purpose is actually to compute the modular inverse of e and $\varphi(N)$. The process of calculation of modular inverse is

similar to the standard modulo operation but follows an algorithm that is called Extended Euclidean Algorithm. To make the process easier and faster, a modular inverse calculator machine that is available online will be used (Timur). In the integer section, the value of e is 7 for our case. In the modulo section, the value of $\varphi(N)$ is 60.

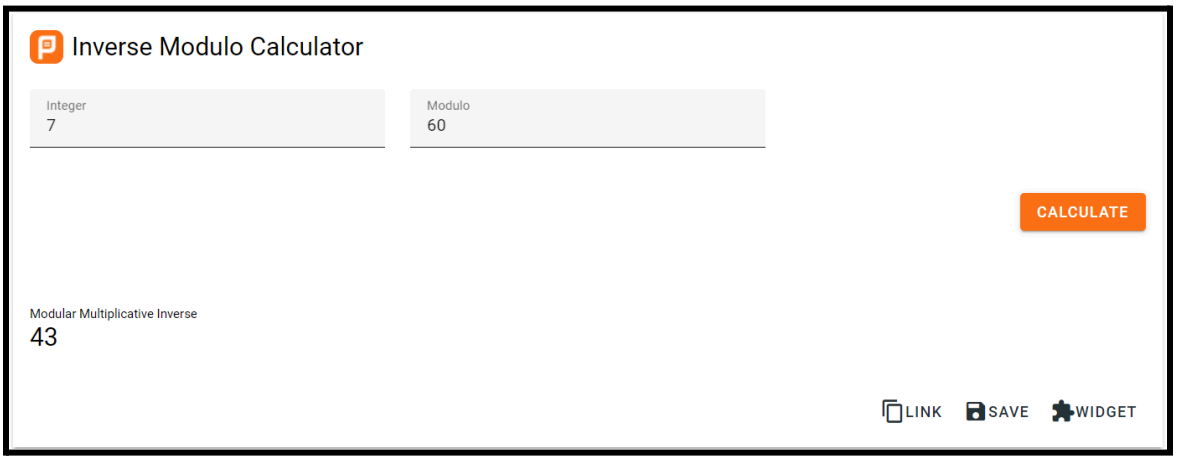

Figure 2: Screenshot taken by me

After entering the numbers into the calculator, the output reveals that d equals 43, which means that our decryption key will be $(43, 60)$.

Therefore, the key process must be understood to obtain both keys required for the cryptosystem. This enables us to illustrate how the RSA algorithm works.

4. Encryption method

Let us assume that User A desires to send a message "m" to User B. It is necessary to convert the letter "m" into a particular number since the cryptosystem can only operate with numbers and not letters. However, who determines the specific number to which a certain character is linked? In the field of computer science, a commonly agreed standard code facilitates universal and efficient processes. In ASCII (American Standard Code for Information Interchange), the

lowercase "m" is "109" in decimal (01101101 in binary) (ibid). So in reality, the message m that user A wants to send is going to be the number 109. To make the calculations more approachable and understandable, we are going to agree that $m = 9$. We are making this adoption because the cipher text is growing exponentially and the result 109^e is going to be a huge number that cannot be calculated.

After converting the letters to numbers we are able to start the encryption process. To begin with, we have to calculate the ciphertext $c = m^e \text{mod} n$, which will basically be a non-sense message that will be sent to user B. To make the modulo operation easier an online calculator will be used to calculate our ciphertext (LLC), and we are going to fill "a" with the number $m^e = 4,782,969$ and "b" with $n = 77$.

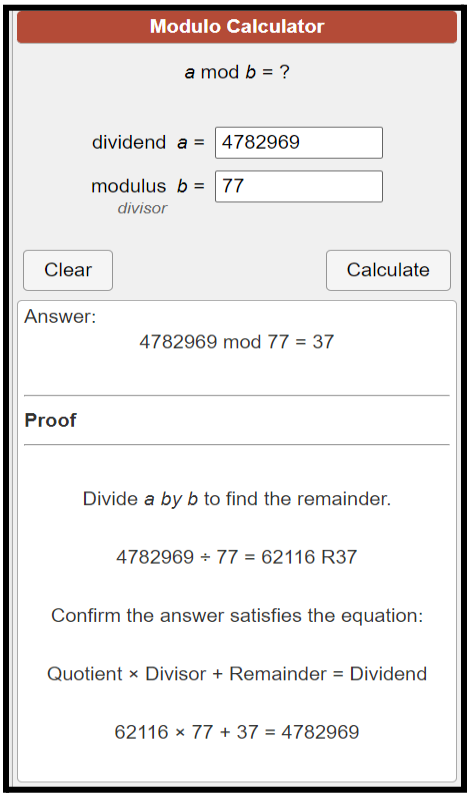

Figure 3: Screenshot taken by me

After inputting to the tool the examples given numbers, we get that $c = 37$.

With this method, we possessed a confidential message, denoted as 9 based on our agreement, and aimed to keep it secure. To achieve this, we utilized a public key for encryption, resulting in an encrypted output or ciphertext '37'. By sending a ciphertext to user B we ensure that even if the information included will get leaked somehow, the data gathered by the attack are useless as the attacker cannot decrypt it without knowing the d key. The decryption process, exclusively accessible to the key owner, will reveal the original message we intended to convey, which is '9'.

5. Decryption method

After receiving the ciphertext, User B executes a step-by-step algorithm to decrypt it, known as the "Decryption Method," which transforms the encoded message into its original form. The equation used for decryption remains identical to that employed for encryption, except for the substitution of different values. Instead of "e" the private key is subtracted, and rather than "m" the ciphertext "c" is applied. Consequently, the final decryption equation is $m = c^{d} mod n$. So applying our examples we have that $m = 37^{43} mod 77$. Because calculating huge numbers such as 37 raised to the power of 43 is almost impossible with an average calculator, we will use an online RSA decryption calculator provided in Internet (Popyack).

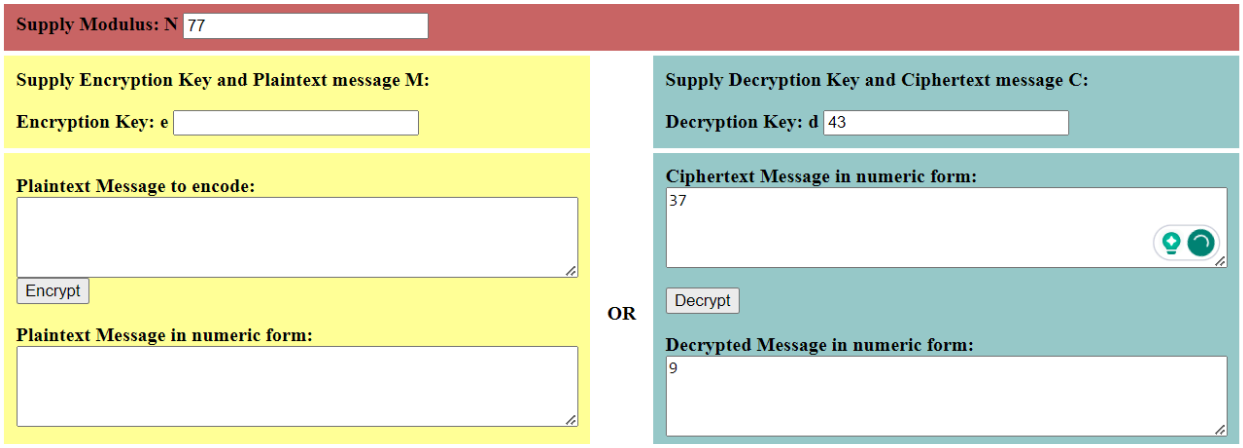

Figure 4: Screenshot taken by me

After applying all data to the site, the initial number 9 is obtained. Knowing that we agreed that 9 is equal to the letter “m”, the receiver will successfully read the initial message. The RSA scheme is then successfully completed by following a step-by-step algorithm, resulting in the secure delivery of a message from User "A" to User "B".

6. Brute force attack

Like any other cryptography system, RSA includes flaws that hackers may find and exploit using a number of different techniques. A Brute Force attack is one of these techniques where the attacker systematically attempts every possible key to decipher the ciphertext. Since the modulus N is finite, the attacker attempts to factor it. This implies that by attempting each potential number that can factor N one at a time and recording the results, the attacker will ultimately be able to decode the message. In our example, we used small numbers so we could demonstrate an example of the scheme in an easier way. This indicates that since there aren't many factors that might possibly be the modulo N, cracking the ciphertext will be simple.

In order for the attack to start we will have to use a step-by-step process called the Fermat factoring algorithm, which is based upon being able to factor the difference of 2 squares (Christensen).

$$x^2 - y^2 = (x + y)(x - y)$$

From the attacker's perspective, we only know the public key (7,77). Which means that we know that $e = 7,\ n = 77$. So how can we break into finding the $\varphi(n)$?

Step 1: Find $\sqrt{n}$

$$\sqrt{77} \approx 8.774$$

Step 2: Define the number k

k is the rounded number of $\sqrt{n}$. So in the example, $k = 9$

Step 3: Find $k^2 - n$

For the third step, we have to see if $k^2 - n$ gives a square number. If the scheme is easy to crack that would give a square number. If it doesn't give a square number, we increase k by 1 until we find a square number.

So we calculate $(k + 1)^2 - n$, $(k + 2)^2 - n$, $(k + 3)^2 - n$..., $(k + c)^2 - n$ until we find a square number. Luckily for our example, $k^2 - n$ gives a square number:

$$k^2 - n = 9^2 - 77 = 81 - 77 = 4 = 2^2$$

Step 4: Factor n by using the identity $x^2 - y^2 = (x + y)(x - y)$

$$n = 77 = 9^2 - 2^2 = (9 - 2)(9 + 2) = 7 \times 11$$

The attacker broke the code and now knows that $p = 7$, $q = 11$. With this information, the attacker easily find $\varphi(n)$ just by using Euler's totient function.

$$\varphi(n) = (p - 1)(q - 1) = 6 \times 10 = 60$$

Knowing that $\varphi(n) = 60$, the attacker can easily find the decryption key to decrypt the ciphertext just by following the private key generation process that was shown in section 3.

As already mentioned earlier, the numbers for our example are small numbers, which means that a program could break the encryption in milliseconds. Knowing that the actual numbers that are used in the real world are extraordinarily large numbers with many digits, this attack is practically impossible to work. In fact, hackers use computer's ability to make programs that autotest the third step, until they find a square number. Once they find a square number, they test the results by decrypting the ciphertext. Many times the decrypted text still makes no sense, as there are numerous factors that can factorize n. The fact that n factorized, doesn't necessarily mean that we found the numbers p and q. After finding multiple factors, the program saves the decrypted text in a data table. The attacker hopes that the computer after days, months, or years will finally factorize n with the actual numbers p q, which show a logical decrypted text.

7. Why is RSA widely used?

Despite being one of the first public key encryptions, RSA is currently utilized in a variety of online applications, including digital signatures, emails, bank accounts, data storage, VPNs, P2P systems, and other communications channels (Klusaitė). One of RSA's key advantages is that it has a straightforward yet safe algorithm, which keeps it relevant and in use even after many years. The inability to factor the product of two huge prime numbers underpins the security of

RSA. Even with the most powerful computers available, RSA remains extremely safe against brute force attacks as long as the key length is suitably large. Another advantage of RSA it's its versatility. It can be modified and improved when it comes to security by using a scheme that presents a new cryptography algorithm based on additive homomorphic properties called Modified RSA Encryption Algorithm (MREA). After years of use by tens of thousands of computers, RSA has profited from research and advancements. Its reputation as a dependable cryptographic solution for data security has been cemented by the fact that the cryptosystem has been tested and that its strengths and weaknesses are well recognized. Its weaknesses present opportunities for enhancements, optimizations, or even more combinations with other cryptosystem methods that would create a safer environment in this chaotic huge storage of data called "Internet".

8. How hard is it to break RSA?

RSA is generally thought to be extremely difficult to crack. Before continuing it is important to define what crack means. Cracking RSA encryption means finding the private key from the public key, which, in turn, involves factoring this product of two primes. The size of the used primes, which defines the length of the RSA key, affects how challenging this factorization is. As the complexity of the prime factorization problem grows exponentially with the key length, it is almost impossible to crack a 2048-bit key. But what happens when it comes to a key with fewer bits? The two primary needs that are essential when it comes to cracking RSA, are a high budget for a really strong computer with practically unbelievable computing abilities, and a lot of luck. There are some possible threads based on mathematical principles that can attack RSA's security, such as factoring attacks, quantum computing attacks (which will be covered later in this essay) Chosen Plaintext, and Ciphertext Attacks (OpenAI). If key lengths are insufficiently large,

factoring algorithms like the General Number Field Sieve (GNFS) may pose a danger to RSA encryption.

9. The future of RSA

As years pass by, technological abilities increase, and furthermore RSA develops rapidly. Some forms of RSA with many bits, are almost impossible to break. In the absence of potent quantum computers, RSA encryption is still secure when key lengths are sufficiently large against classical computers. But what will happen with the existence of quantum computers? Quantum computing is not a sci-fi scenario for a science fiction film; instead, it is a very likely scenario for the near future. Indeed, the US National Institute of Standards and Technology (NIST) has already said that quantum computers will, by 2029, be able to break existing public key infrastructure like 128-bit AES encryption, which is currently used to protect sensitive information sent over the Internet (McKenzie). With the help of algorithms like Shor's algorithm, quantum computers have the potential to efficiently factor large numbers, significantly weakening the security of RSA. A recent research in China reported finding a factorization method that could break a 2048-bit RSA key using a quantum system with just 372 qubits when it operated using thousands of operation steps (Yan et al.). If the discovery was accurate, it would have indicated that RSA encryption might succumb to quantum computing much sooner than most people anticipated.

10. Conclusion

Going back to the initial research question, RSA cryptography is based on the factorization of prime numbers. Historically, RSA is one of the most widely used cryptographical schemes playing a huge role in data security. Using mathematical principles such as modular arithmetic,

and Euler's totient function, RSA protects important data and creates a safe environment for users to communicate using ciphertexts. As with any other cryptographical scheme, RSA has also variabilities that can be used by attackers, to decrypt messages, using various methods such as the brute force attack. Despite our example, in reality, real-world RSA keys involve extraordinarily large numbers, rendering a brute force attack practically improbable. The future of RSA, and cryptography in general, hides many surprises with the appearance of quantum computers being imminent. The incredible calculating ability of quantum computers will change the way cryptography works forever.

11. Works Cited

Christensen, Chris. *Mathematical Attack on RSA* . 2006.

Computer Hope. "What Is Modulo?" *Computerhope.com*, 10 July 2019, www.computerhope.com/jargon/m/modulo.htm.

"Coprime Calculator." *Www.mathsisfun.com*, www.mathsisfun.com/numbers/coprime-calculator.html.

Klusaitė, Laura. "What Is RSA Encryption, and Is It Safe to Use? | NordVPN." *Nordvpn.com*, 31 Dec. 2022, www.nordvpn.com/blog/rsa-encryption/.

Lake, Josh. "What Is RSA Encryption and How Does It Work? | Comparitech." *Comparitech*, 11 Dec. 2018, www.comparitech.com/blog/information-security/rsa-encryption/.

Li, Mei, et al. "Modular Arithmetic | Brilliant Math & Science Wiki." *Brilliant.org*, www.brilliant.org/wiki/modular-arithmetic/.

LLC, CalculatorSoup. "Modulo Calculator." *CalculatorSoup*, www.calculatorsoup.com/calculators/math/modulo-calculator.php.

Marius Tărnăuceanu. "A Generalization of the Euler's Totient Function." *Asian-European Journal of Mathematics*, vol. 08, no. 04, 17 Nov. 2015, pp. 1550087–1550087, https://doi.org/10.1142/s1793557115500874 .

McKenzie, James. "When Will Quantum Computers Finally Break into the Market?" *Physics World*, 3 Apr. 2023, www.physicsworld.com/a/when-will-quantum-computers-finally-break-into-the-market/.

OpenAI. "ChatGPT." *Chat.openai.com*, OpenAI, 2023, chat.openai.com/.

Parker, Charlie. "Why Is Euler's Totient Function Equal to $(P-1)(Q-1)$ When $N=Pq$ and P and Q Are Prime in a Clean Intuitive Way?" *Mathematics Stack Exchange*, 1 Jan. 2014, www.math.stackexchange.com/questions/629949/why-is-eulers-totient-function-equal-to-p-1q-1-when-n-pq-and-p-and-q. Accessed 22 Jan. 2024.

Popyack, JL. "RSA Calculator." *Www.cs.drexel.edu* , Dec. 2002, www.cs.drexel.edu/~popyack/Courses/CSP/Fa17/notes/10.1_Cryptography/RSA_Express_EncryptDecrypt_v2.html. Accessed 23 Oct. 2023.

"Prime Factorization Numbers." *University of North Georgia*, www.ung.edu/learning-support/video-transcripts/prime-factorization-numbers.php

Richards, Kathleen. "What Is Cryptography? Definition from SearchSecurity." *Security*, www.google.com/url?q=www.techtarget.com/searchsecurity/definition/cryptography&sa=D&source=docs&ust=1694626651010003&usg=AOvVaw3jNaBr0MTysJFI4uKl6Nez. Accessed 13 Sept. 2023.

Telecom Italia. "RSA Cryptography: History and Uses." *Telsy*, 26 May 2021, www.telsy.com/rsa-encryption-cryptography-history-and-uses/.

Timur. "Online Calculator: Modular Multiplicative Inverse." *Planetcalc.com*, 2021, www.planetcalc.com/3311/.

Whitman. “3.8 the Euler Phi Function.” *Whitman.edu*, 2019, www.whitman.edu/mathematics/higher_math_online/section03.08.html.

Yan, Bao, et al. *Factoring Integers with Sublinear Resources on a Superconducting Quantum Processor*. 2022.