Exploration in Elliptical Curve Cryptography and Rivest-Shamir-Adleman Cipher

Introduction

The topic of my math IA is number theory, specifically exploring the different methods of cryptography, specifically the Rivest, Shamir, Adleman Cipher and Elliptical Curve Cryptography. I was introduced to the general topic of cryptography in my programming class where I was asked to write algorithms for some basic cryptography methods, such as the Caesar cipher where you shift the alphabet by a certain integer to encrypt a message. Then, during my spare readings, I was introduced to quantum cryptography through some introductory books in quantum mechanics in Grade 10. In addition, one of my close friends from my local robotics club was also curious about cryptography, so we were able to collaborate and research this topic just out of curiosity. We researched how cryptography is used in terms of its usage in cybersecurity and learned how to break some low-security encryption methods. However, most of our research stayed surface level, not truly understanding the mathematical deductions behind those methods.

When it came to writing this math IA, I was reminded of my learning in cryptography and wanted to explore further about the math behind how we transmit information through encrypted numerical messages. Hence, the aim of this IA is to help you learn and understand these two methods of cryptography with a strong focus on its underlying mathematical operations. You will be working a lot with modular arithmetic in the RSA section of this exploration; thus, this IA may also broaden your mathematical understanding with topics not covered in our curriculum.

What is Cryptography?

Cryptography has been utilized by society to protect confidential information that needs to be communicated remotely. The complexity of cryptosystems has dramatically increased as we develop further technology to enhance its difficulty to decipher. The earliest known use of cryptography can be traced back to ancient civilizations such as Egypt, where hieroglyphs were sometimes encrypted to conceal sensitive information.

Throughout history, cryptography played a pivotal role in wars, diplomacy, and espionage. During World War II, for instance, the Enigma machine, a cipher device used by the Germans, was cracked by British mathematician Alan Turing and his team, significantly impacting the outcome of the war. The advent of computers and the internet in the 20th century brought about a revolution in cryptography. Modern cryptography relies heavily on complex mathematical algorithms, such as RSA (Rivest-Shamir-Adleman) and AES (Advanced Encryption Standard), which are fundamental in securing data transmission, protecting sensitive information, and enabling secure online transactions (Sidhpurwala, 2013). As technology continues to advance, cryptography remains at the forefront of cybersecurity, playing a crucial role in safeguarding data privacy and securing digital communications across various domains, including finance, healthcare, government, and everyday personal interactions.

To understand the mathematics used in cryptography, modular arithmetic is an important mathematical topic to understand as we will be using them excessively throughout. Sometimes, mathematicians are only interested in the remainder of a division. Hence, we can say that the following are all equivalent:

$$\frac{23}{7} = 3\; remainder\; 2$$

$$23\; mod\; 7 = 2$$

$$23 \equiv 2\; (mod\; 7)$$

RSA Cipher

RSA or Rivest-Shamir-Adleman Cipher is the most well-known and popular asymmetrical encryption algorithm to secure information, commonly used in banking systems. It uses an asymmetric cryptosystem using number theory. Its security is provided by the large difference in time required to perform multiplication to decrypt the key versus the long duration needed to perform prime factorization of large integers in the hope of breaking the cipher (Xin Zhou & Xiaofei Tang, 2011).

The RSA cryptosystem is an example of an asymmetrical encryption system, requiring a pair of distinct keys, one public and one private. The public key is set of two integers which are given to the message sender by the message receiver; the private key is also composed of two integers, but they will not be enclosed to anyone but the message receiver. Most of the mathematics in the RSA section is derived from the original published paper of this algorithm by Rivest, Shamir, and Adleman in 1978.

To understand the encryption and decryption process, we can start by describing the procedure with a mathematical statement. That is to represent the encryption procedure with E, and the decryption with D. Hence, to decipher numerical message M, I can use this following statement to say that I can deciphering the encrypted message M:

$$D\big(E(M)\big) = M.$$

To encipher a deciphered message M, I can change the order of E and D:

$$E(D(M)) = M.$$

General Encryption & Decryption Methods

Now that we understand the general process of encryption and decryption, we can get started on the mathematical part to understand what E and D actually mean. The public key as mentioned before is a set of two positive, non-zero integers, in this case we will call them (e, n) (but do not confuse this e with Euler's number, as the one used here refers to *encrypted*). Similarly, the private key is composed of (d, n).

In order to encrypt a message, I can use the following correlation unique to the RSA cipher to produce ciphertext C. That is, the encrypted version of message M. In the equation below, we can see that to encrypt a message, we would simply raise the numerical message to the power of e provided in the public key and modulo n (find the remainder of M^e when divided by n.)

$$C \equiv E(M) \equiv M^e (mod\ n)$$

To decrypt an encrypted message that has been sent to me, I would use my own set of private keys (d, n). And use a similar process as above described in the following formula to find the original message.

$$D(C) \equiv C^d (mod\ n)$$

Up to this point, we have been introduced to the algorithm to decipher and encipher the messages, but not yet to what e, d, and n are. In the following writing, I will show you how to define these variables and find them in relation to one another.

First, we need to choose two prime numbers to use as factors for n, named p and q.

$$n = p \cdot q$$

But you may ask, how hard is it to find these prime numbers, and how large are they typically. Since the application of RSA is mostly over computers, I have discovered that the two primes are usually around 300-digits, if not even larger. While computers are efficient at performing these searches and calculations, they are not so quick compared to the ECC method which will be discussed in the latter paragraphs.

In the RSA method, the d component of the public key must be relative primes with n. Relative primes are a set of numbers that may not be considered as prime numbers individually, but do not share any common factor other than 1 (e.g. 8 and 9 are both not prime numbers, but they are considered as relative prime pairs as they only share a factor of 1). To help us find d, we can take advantage of Euler's Totient function $\varphi(n)$ which exactly finds the coprime of a certain integer; in this case, finding the coprime of d and n.

$$\varphi(n) \; = (p-1)(q-1)$$

This relationship between d, p, and q can also be described in terms of GCD (greatest common denominators). For which the GCD of d and $(p-1)(q-1)$ must be 1 for them to be co-primes.

Now, the only piece of the key missing is e, which is defined to be the multiplicative inverse of d, $mod\ (p-1)(q-1)$**.** This introduces a new field of arithmetic called modular arithmetic, just like how we use remainders when first learning how to divide numbers.

The need to calculate e brings up a new concept called modular inverses, which has a unique property as follow:

$$A^{-1}\ mod\ C \equiv 1$$

Or

$A \cdot b \equiv 1\ mod\ C$, where A, b, and C are all integers.

Hence, only when this statement is true, is A relative prime to b as their remainder remains as 1 when divided by all factors shared between A and C. To put this equation into perspective of RSA cipher, and the relationship between e and d are as follows, and we can find the e by reversing the operations.

$$d^{-1}\ mod(p-1)(q-1) \equiv 1$$

or

$$e \cdot d \equiv 1\ (mod\ (p-1)(q-1))$$

Finally, we have found a way to find all our integers needed to produce our public and private key with respect to p and q (Rivest et al., 1978).

Worked Example

To better understand the mathematical algorithm, I have found worked examples to be very effective. This example would follow the general process of key creations if I were to need to cipher an encrypted message. For the sake of simplicity and calculations, we will use two small prime numbers for this example.

Finding n:

$$let\ p = 17, q = 41$$

$$n = p \cdot q$$

$$n = 17 \cdot 41$$

$$n = 697$$

Finding $\varphi(n)$:

$$\varphi(n) = (p-1)(q-1)$$

$$= (17-1)(41-1)$$

$$= (16)(40)$$

$$= 640$$

Choosing d, the selection of d is random, as long as it is a large relative prime to $\varphi(n)$:

$$gcd\big(d, (p-1)(q-1)\big) = 1$$

$$\gcd(d, 640) = 1$$

$$d = 49$$

Finding e:

$$e \cdot d \equiv 1\ (mod\ (p-1)(q-1))$$

$$e \cdot 49 \equiv 1\ (mod\ 640)$$

Computing the result of e would be very tedious and difficult by modern arithmetic methods. However, by using Euclid’s Theorem or Fermat’s Little Theorem, which computes for the coprime of d , we can find a shortcut to find it (since it is not the focus of this IA, the two theorems will not be explained in detail). I have found a modular inverse calculator online and received the following value for e for the above equation.

$$e = 209$$

Now n, $\varphi(n)$, e, and d are all found, the ciphering can begin!

Proof of RSA's Correctness

In order to show that our algorithm truly yields the same result when the receiver decrypts the encrypted message, we can surely use examples and trial-and-error in real testing to prove our case, but would there be outliers? Therefore, I thought that a more theoretical approach would be beneficial in proving all cases mathematically.

The key formula to center our proof on would be $e \cdot d \equiv 1\ (mod\ (p-1)(q-1))$ as it entails all variables needed to be selected and used in the RSA process. From this point, we have two goals to accomplish: show that d must be co-prime to $\varphi(n)$ and show that the message M sent is equal to decrypted message D.

First, let integer t be the greatest common divisor of d and $\varphi(n)$, and let us suppose that d and $\varphi(n)$ can be re-written in the following form:

$$d = c_1 \cdot t$$

$$\varphi(n) = c_2 \cdot t$$

It's important to remember that our central equation can be rewritten as the following:

$$e \cdot d \equiv 1 \left(mod\ (p-1)(q-1)\right)$$

$$e \cdot d \equiv 1\ (mod\ \varphi(n))$$

In this case, assuming that e and d are relatively prime, then we can rewrite this in modern arithmetic format that we are more familiar with.

$$e \cdot d \equiv 1 \left(mod\ \varphi(n)\right)$$

$$e \cdot d = c_3 \cdot \varphi(n) + 1$$

Hence:

$$e \cdot c_1 \cdot t = c_2 \cdot c_3 \cdot t + 1$$

$$t(e \cdot c_1 - c_2 \cdot c_3) = 1$$

Which implies that $t = 1$, and that d and $\varphi(n)$ are truly co-primes with each other.

Our next step is to prove that our inputted message is the same as the decrypted message. Returning to our central equation, we can again, rewrite it as the following:

$$e \cdot d = t \cdot \varphi(n) + 1$$

Substituting the encrypted method into the decryption method:

$$D\big(E(M)\big) = \big(E(M)\big)^d\ mod\ n$$

$$= (M^e)^d\ mod\ n$$

$$= M^{ed}\ mod\ n$$

Substituting the decryption method into the encryption method:

$$E\big(D(M)\big) = \big(D(M)\big)^{e}\ mod\ n$$

$$= \big(M^{d}\big)^{e}\ mod\ n$$

$$= M^{ed}\ mod\ n$$

Substituting $e \cdot d = t \cdot \varphi(n) + 1$ in to the power, we yield:

$$M^{t \cdot \varphi(n)+1}\ mod\ n$$

Since $\varphi(n)$ is divisible by $p - 1$ as described in the totient function, thus

$$M^{t \cdot \varphi(n)+1}\ mod\ n \equiv M\ mod\ p$$

By symmetry, we also have

$$M^{t \cdot \varphi(n)+1}\ mod\ n \equiv M\ mod\ q$$

Therefore, if the encryption and decryption method both yield the same result, then the RSA method must be valid (Rivest et al., 1978).

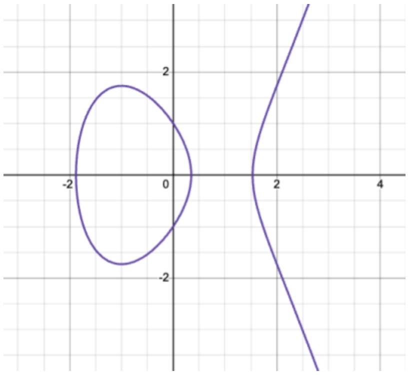
Elliptical Curve Cipher

After studying the algorithm of RSA cipher, I found the method to be more dangerous and prone to attacks as the computing power of our computer exponentially increased. Hence, I started to wonder if there is a more efficient or more secure method to encrypt and decrypt messages. I eventually came across a blog article comparing the RSA method to elliptical curve cipher (ECC). When I compared the bit-sizes between the two methods: RSA's 2048 bit achieves the same level of security as ECC's 256 bit key (Tencent Cloud, n.d.). Not only does this mean that ECC require less memory and computing power, but also that ECC is much faster at producing keys compared to RSA. Therefore, I was curious to find out what was the cause of ECC's efficiency.

Elliptical curves are not to be confused with ellipses. So, what are elliptical curves? The elliptical curve is a smooth algebraic curve that can be described using the following equation on the Cartesian plane, where a and b are both real numbers.

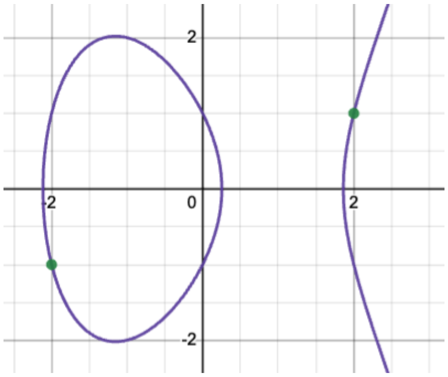
$$y^2 = x^3 + ax + b$$

The unique thing about elliptical curves is that it exhibits horizontal symmetry; any point on the curve can be reflected over the x-axis and still remain on the same curve. Furthermore, any non-vertical line can intersect the curve at three places maximum (Coates Welsh, n.d.).



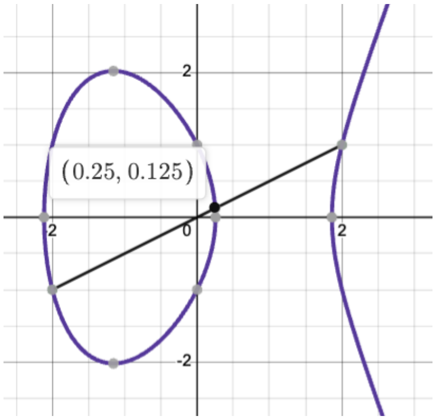
Group Theory And How They Are Used In ECC

Group theory is also essential to understanding the ECC algorithm. Groups are a set of numbers that can contain integers, rational, irrational numbers, matrices, or even shapes and functions. A group also needs to have certain properties, such as a binary operations (e.g. addition or multiplication), and satisfy four particular axiom: closure, associativity, identity, and inverse (Kurzweil & Stellmacher, 2004). In the case of ECC, we need to be able to combine the two coordinates chosen first and create a 3rd member, all of which are considered a group:



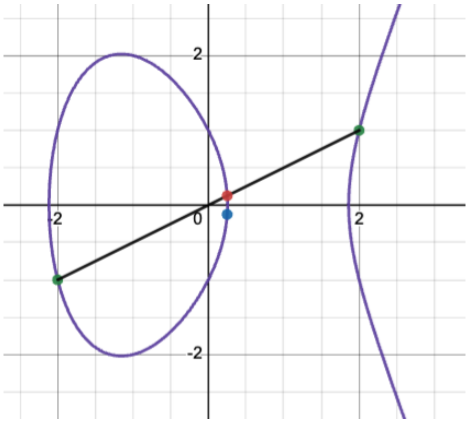
Given these properties, I discovered that ECC follows a trapdoor function, which the initial message sender can chose an initial point and reflect it, then reselect, however many times until they reach a final coordinate. This makes the algorithm hard to break as the attacker will have no idea how many times they have reflected the point (Chambers, 2021).

To start the encryption, we need to first randomly select 2 points, A and B on an elliptical curve $y^2 = x^3 - 4x + 1$. Let point $A = (2,1)$ and point $B = (-2,-1)$. According to the group axiom identity of addition, $A + B$ must also be on the elliptical curve in order for them to be in the same group. If I add them as I would normally perform vector addition, I get $(0,2)$ – but unfortunately, this is not on the curve, and by extension, not in the group. So, I eventually found that by defining the addition of $A + B$ geometric steps, I can avoid this issue. Hence, I connected A and B with a straight line. This line intersects the curve in one more place, C.

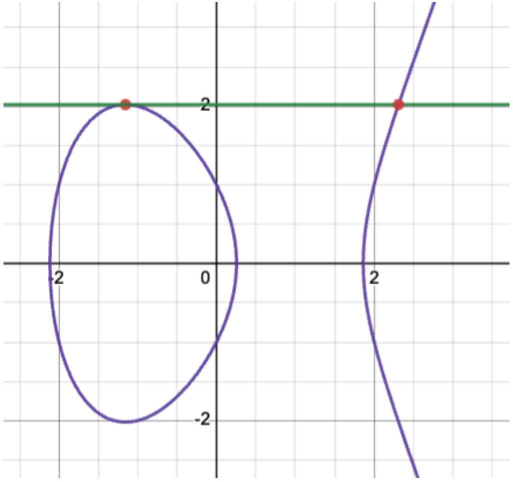


According to the inverse axiom of inverse from group theory, I should be able to reflect point C and still find it on the curve. Thus, by reflecting C around the x-axis, I then define this new point $C' = A + B$.

$$(2,1) + (-2,-1) = (\frac{1}{4}, \frac{-1}{8})$$



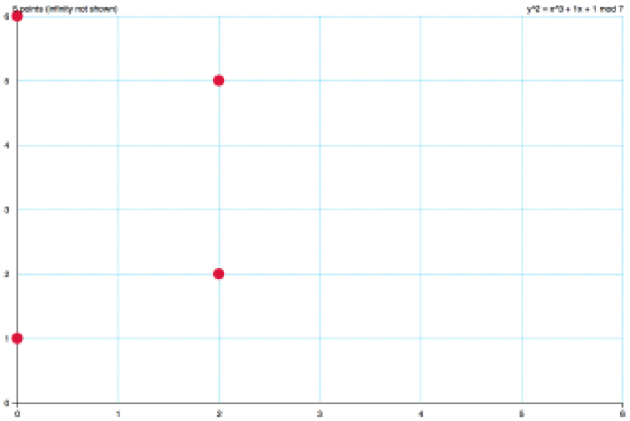
I then wondered whether there would be any limitations to this approach and discovered that since we did not limit the selection of A and B, it is still possible to have them as identical points. Here I can create the line through A which is the tangent to the curve at that point:



Using the same method as any normally selected two coordinates to create the addition group of $A + B = C'$. For On curve $y^2 = x^3 - 4x + 1$, if I start with point A $(-1.155, 2.02)$ then this transformation tells us that

$$A + A = (2.31, 2.02)$$

Since we are only interested in the positive integer groups for cryptography purposes, we will only be working with them on a finite field. Moreover, we will also be adding in a modulus function (mod prime). Using the curve $y^2 = x^3 + x + 1$, and only looking at the positive integer solutions $mod\ 7$ (the function and modulus is chosen randomly to provide a graphical explanation), we will get the following four points on the field below.



If we inspect more closely, when $x = 1$,

$$y^2 = 1^3 + 1 + 1$$

$$y^2 = 3 \ (mod\ 7)$$

Thus, when $x - 1 = 1$, there is no integer solution.

When $x = 2$,

$$y^2 = 2^3 + 2 + 1 = 11$$

$$y^2 = 11 = 4 \ (mod\ 7)$$

$$\sqrt{y^2} = \sqrt{4}$$

$$y = \pm 2 = 5 \ (mod\ 7)$$

Hence, there are two points located on the graph when $x = 2$.

When $x = 3$,

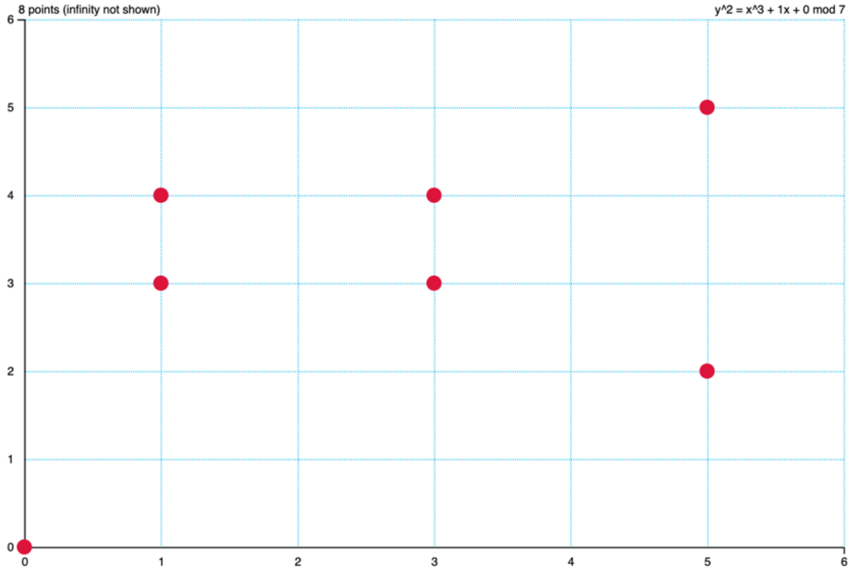
$$y^2 = 3^3 + 3 + 1 = 31$$

$$y^2 = 3 \ (mod\ 7)$$

Thus, when $x = 3$, there are no solutions that satisfy our parameter.

Therefore, only the following coordinates satisfy the equation $(mod\ 7)$: (2,2), (0,1), (0,6), (2,5).

Of course, the points formed changes as I manipulate the constants in the elliptical equation. For instance, for $y^2 = x^3 + x$, the points that satisfy the equation $(mod\ 7)$ are as follows.



The coordinate points calculated for any elliptical curve equation $(mod\ prime)$ can also be summarized using a table:

+	∞	(0,0)	(1,3)	(1,4)	(3,3)	(3,4)	(5,2)	(5,5)
∞	∞	(0,0)	(1,3)	(1,4)	(3,3)	(3,4)	(5,2)	(5,5)
(0,0)	(0,0)	∞	(1,4)	(1,3)	(5,2)	(5,5)	(3,3)	(3,4)
(1,3)	(1,3)	(1,4)	(0,0)	∞	(3,4)	(5,2)	(5,5)	(3,3)
(1,4)	(1,4)	(1,3)	∞	(0,0)	(5,5)	(3,3)	(3,4)	(5,2)
(3,3)	(3,3)	(5,2)	(3,4)	(5,5)	(1,4)	∞	(1,3)	(0,0)
(3,4)	(3,4)	(5,5)	(5,2)	(3,3)	∞	(1,3)	(0,0)	(1,4)
(5,2)	(5,2)	(3,3)	(5,5)	(3,4)	(1,3)	(0,0)	(1,4)	∞
(5,5)	(5,5)	(3,4)	(3,3)	(5,2)	(0,0)	(1,4)	∞	(1,3)

Although tables like the one above can tell us that (0,1) + (0,1) = (2,5). But, we can also approach this algebraically: I could draw a tangent line to the curve at point A (0,1), then find where it intersects the graph again at point C, and then reflect this point on the x-axis, finding C

First, I can find the slope of the elliptical curve at (0,1) using implicit differentiation:

$$y^2 = x^3 + x + 1$$

$$2y\frac{dy}{dx} = 3x2 + 1$$

$$\frac{dy}{dx} = 3x2 + 12y$$

$$\frac{dy}{dx} = 3(0)2 + 12(1)$$

$$\frac{dy}{dx} = 12$$

Next, using the gradient of the elliptical curve at (0,1) modulo 7

$$\frac{dy}{dx} \equiv 4\ (mod\ 7)$$

Then, I can find the full linear equation of the tangent by substituting in (0,1):

$$\frac{y - y_1}{x - x_1} = m$$

$$y - y_1 = m(x - x_1)$$

$$y - 1 = 4(x - 0)$$

$$y = 4x + 1$$

Next, to find C, we can try to find where the tangent line intersects the elliptical curve again by equating the two equations.

$$y^2\ =\ x^3\ + x\ +\ 1$$

$$(4x + 1)^2 = x^3\ + x\ +\ 1$$

$$16x^2 - 8x + 1 = x^3\ + x\ +\ 1$$

$$x \equiv 0, 2 (mod\ 7)$$

Lastly, by substitute $x = 2$ into the original elliptical curve, we can find the y-coordinates:

$$y^2 = 23 + 22 + 1$$

$$y \equiv 2, 5\ (mod\ 7)$$

(2,2) is the point where the tangent would touch the curve and (2,5) is the equivalent of the reflection transformation. Therefore our answer is (2,5). i.e (0,1) + (0,1) = (2,5) as required.

The same method is applied when adding different points; however, instead of calculating the gradient of the tangent, we must find the gradient of the line that joins the two points. However, when I tried to merge points like (2,5) and (2,2), the line that connects them does not cross the graph again; hence, we can append points an infinity, as in (2,5) + (2,2) (Sullivan, 2013).

Encryption and Decryption Methods

For more understanding and practice, let us suppose that Biff and Eunice wanted to create an encryption and decryption key using ECC. They both agreed upon an elliptical curve function $y^2 = x^3 + x + 1$ and a modulo number, $mod\ 7$.

These two parameters then create the following addition group:

+	∞	(0,2)	(0,5)	(1,1)	(1,6)	(2,2)	(2,5)	(5,2)	(5,5)	(6,0)
∞	∞	(0,2)	(0,5)	(1,1)	(1,6)	(2,2)	(2,5)	(5,2)	(5,5)	(6,0)
(0,2)	(0,2)	(1,6)	∞	(0,5)	(1,1)	(5,5)	(5,2)	(2,5)	(6,0)	(5,2)
(0,5)	(0,5)	∞	(1,1)	(1,6)	(0,2)	(2,5)	(2,5)	(6,0)	(2,2)	(5,5)
(1,1)	(1,1)	(0,5)	(1,6)	(0,2)	∞	(5,2)	(6,0)	(5,5)	(2,5)	(2,2)
(1,6)	(1,6)	(1,1)	(0,2)	∞	(0,5)	(6,0)	(5,5)	(2,2)	(5,2)	(2,5)
(2,2)	(2,2)	(5,5)	(2,5)	(5,2)	(6,0)	(0,2)	∞	(0,5)	(1,6)	(1,1)
(2,5)	(2,5)	(2,2)	(5,2)	(6,0)	(5,5)	∞	(0,5)	(1,1)	(0,2)	(1,6)
(5,2)	(5,2)	(2,5)	(6,0)	(5,5)	(2,2)	(0,5)	(1,1)	(1,6)	∞	(0,2)
(5,5)	(5,5)	(6,0)	(2,2)	(2,5)	(5,2)	(1,6)	(0,2)	∞	(1,1)	(0,5)
(6,0)	(6,0)	(5,2)	(5,5)	(2,2)	(2,5)	(1,1)	(1,6)	(0,2)	(0,5)	∞

Afterward, they would select a point on the curve. Suppose they opt for (1,1).

Biff picks a secret number, denoted as n, and then transmits nP through a public network. For instance, if chooses n = 2, then 2(1,1) = (1,1) + (1,1) = (0,2). This can be interpreted as "multiply the chosen point n times to give unique output." Subsequently, Biff transmits (0,2).

Eunice selects a secret number, denoted as m, and openly transmits mP. If, for example, Eunice chooses m = 3, then

$$3 \cdot (1,1) = (1,1) + (1,1) + (1,1) = (0,2) + (1,1) = (0,5)$$

Eunice transmits (0,5).

Both Biff and Eunice can easily compute mnP, which constitutes the secret key.

Biff receives (0,5), and then computes the private key:

$$2(0,5) = (0,5) + (0,5) = (1,1)$$

Eunice receives (0,2) and calculates for the following, which is identical to Biff's key:

$$3(0,2) = (0,2) + (0,2) + (0,2) = (1,1)$$

However, from an individual observing mP and nP and trying to attack the encryption, there's no efficient method to swiftly determine mnP. A brute force approach in this case would be exceedingly time-consuming. Thus, this method can effectively encrypt data.

Conclusion

Comparisons in Application

As ingenious as RSA is ever since it has been created, the emergence of quantum computers is well on their way to crack the ciphering of RSA, making all its calculation complexities trivial given the high computing power of quantum computers. While the quantum computers have yet to make a break on the encryption, it is projecting to be broken as of April 13, 2030. However, this date will likely approach faster than we expected given the rapid development in the quantum field. Hence, the National Institute of Standards and Technology have started a competition as early as 2016 to find a quantum-resistant cryptography method. CRYSTALS-Kyber was one amongst other two winners (Houston-Edwards, 2024). However, this does not alleviate us from the threat of the cipher breaks as our stored information may be stolen by quantum computers retroactively.

Elliptic curve cryptography offers advantages over RSA cryptography, needing fewer digits to pose an equally challenging problem. This efficiency translates to faster data encoding compared to RSA encryption. Presently, Bitcoin relies on elliptic curve cryptography, likely signalling its broader adoption as more data transitions into digital formats. Yet, it's important to note that the difficulty in cracking elliptic curves hasn't been definitively proven—there might exist a novel approach to solve this challenge more expediently. Numerous mathematicians and computer scientists are actively engaged in this domain.

Government intelligence agencies such as the NSA and GCHQ express keen interest in these encryption techniques. The rapid solution to this problem could grant overnight access to vast amounts of encrypted data. For instance, the security of the Bitcoin currency exchange could be compromised. Recent revelations suggest that the NSA inserted "backdoor" entries into certain elliptic curve cryptography algorithms, allowing them access to supposedly secure data transmissions. This ongoing evolution stands as a mathematical foundation in the emerging digital arms race .

Possible Exploration

As previously mentioned, the computing power required to find the initial two prime numbers in RSA is huge considering the prime’s length. This brings one to think whether there is a more efficient way of determining two unique numbers that can be used in place of the two large primes. The proposed approach combines the RSA public and private keys with an Elliptic Curve's secret and public keys through an exclusive OR operation, creating a fresh key pair. This new key pair is then utilized for encrypting and decrypting data via the RSA encryption and decryption methods. The primary focus of this proposal revolves around the creation of a robust key pair to optimize for the security of encryption and decryption processes.

References

Chambers, A. (2021, August 10). *Elliptical Curve Cryptography*. IB Maths Resources from Intermathematics. https://ibmathsresources.com/2021/08/10/elliptical-curve-cryptography-2/

Coates Welsh, M. (n.d.). *ELLIPTIC CURVE CRYPTOGRAPHY*. Retrieved February 3, 2024, from https://math.uchicago.edu/~may/REU2017/REUPapers/CoatesWelsh.pdf

Houston-Edwards, K. (2024, February 1). *Tomorrow's Quantum Computers Threaten Today's Secrets. Here's How to Protect Them*. Scientific American. https://www.scientificamerican.com/article/tomorrows-quantum-computers-threaten-todays-secrets-heres-how-to-protect-them/

Kurzweil, H., & Stellmacher, B. (2004). Basic Concepts. Universitext, 1–42. https://doi.org/10.1007/0-387-21768-1_1

Musa, U., Adebiyi, M. O., Adigun, O., Adebiyi, A. A., & Aremu, C. O. (2023). Hybrid Cloud Storage Techniques Using Rsa And Ecc. *Institute of Electrical and Electronics Engineers*. https://doi.org/10.1109/seb-sdg57117.2023.10124559

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. https://doi.org/10.1145/359340.359342

Sidhpurwala, H. (2013, August 14). *A Brief History of Cryptography*. Www.redhat.com. https://www.redhat.com/en/blog/brief-history-cryptography

Sullivan, N. (2013, October 24). *A (Relatively Easy To Understand) Primer on Elliptic Curve Cryptography*. The Cloudflare Blog; The Cloudflare Blog. https://blog.cloudflare.com/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography/

Tencent Cloud. (n.d.). Www.tencentcloud.com. https://www.tencentcloud.com/document/product/1007/39989#:~:text=It%20is%20normally%20256%20bits

Vidhya, E., Sivabalan, S., & Rathipriya, R. (2019). Hybrid Key Generation for RSA and ECC. *2019 International Conference on Communication and Electronics Systems (ICCES)*. https://doi.org/10.1109/icces45898.2019.9002197

Xin Zhou, & Xiaofei Tang. (2011, August 1). *Research and implementation of RSA algorithm for encryption and decryption*. IEEE Xplore. https://doi.org/10.1109/IFOST.2011.6021216