HIGH SCHOOL EDITION
Journal of Student Research
Volume 12 Issue 1 (2023)

Mathematics Behind the RSA Algorithm

Rayan Das[1] and Guillermo Goldsztein[#]

[1]Archbishop Mitty High School, San Jose, CA, USA
[#]Advisor

ABSTRACT

Our world is becoming more interconnected through technology, with some notable examples being social media and online payment services. With this higher connection comes the increased risk of information, ranging from harmless text messages to credit card information, being stolen. These risks make cryptography, the field regarding the securing of messages, vital in order to minimize the chance of these data breaches. Essentially, this field is about creating, improving, and implementing various algorithms to encrypt and decrypt messages so only the sender and receiver can see the messages. This paper reviews one such algorithm, the RSA algorithm, and the mathematical concepts behind it.

Introduction

Imagine a scenario where two classmates want to send messages in a way so that only they can understand them. They devise a system in which each letter in their message is shifted over by one (a becomes b, b becomes c … z becomes a). For example, if one classmate wants to say "hello," he would send the message "ifmmp" to his friend. The process of translating "hello" to "ifmmp" is called encryption, and the translation of "iffmp" back to the original message "hello" is called decryption. However, this particular process to encrypt messages is very basic and can be very easily deciphered by any outside viewers. Therefore, there is a field called cryptography that is dedicated to the study and implementation of advanced processes to secure messages.

The idea of concealing messages is certainly not a new one. The Spartans used the earliest recorded form of cryptography, in which a piece of parchment containing the message is wrapped around a baton so that the message is scrambled when unwrapped, and the correct message can only be determined when a baton with the exact same dimensions is used. Various other civilizations also had their own methods to make sure their messages stayed a secret, but the major breakthroughs in cryptography were made during the World Wars in order to keep the enemy from understanding important messages. Specifically, rotor cipher machines were invented during this time period; these machines involved a series of rotors being hardwired to substitute letters (similar to the classmate example) so that only another machine with the same hardwiring can decrypt the messages. The last phase in the evolution of cryptography was its transition to a more digital format as computers became more mainstream. The process itself is similar, involving various rounds of substitutions and permutations, but now it is done with advanced algorithms to make the scrambled messages even harder to decrypt. Today, cryptography is used everywhere, from messaging platforms to e-commerce and more [5].

One such algorithm used in cryptography is the RSA algorithm. Essentially, this algorithm involves a public key, one that everyone can see, for encryption and a corresponding private key for decryption. In this paper, we will discuss the mathematical concepts, which are heavily based on number theory, behind this algorithm.

The paper will be organized as follows. Section 2 explains the greatest common divisor, then section 3 goes over modular arithmetic and properties pertinent to this algorithm. Sections 4 and 5 elaborate upon the Euclidian Theorem and Bézout's Lemma, respectively, and section 6 goes over Euler's Totient Function and

Euler's Theorem. All these concepts are then used in the theorem in section 7, which is finally applied to real life in section 8, the conclusion.

Greatest Common Divisor (GCD)

Definition 1 *Given two integers a and b, their greatest common divisor, denoted by* gcd(a, b), *is the largest positive integer that divides a and b.*

For example, gcd(8,20) = 4 because 8/4 = 2, 20/4 = 5, and no larger integer can evenly divide both 8 and 20. Note that gcd(a, 0) = a because any number can divide 0.

Modular Arithmetic

A key branch of number theory is modular arithmetic, which deals with remainders. In this section, we will only cover the aspects of modular arithmetic that are important for the RSA algorithm.

Definition 2 *Given two integers a and b, we say that a* mod *b returns the remainder when a is divided by b.*

For example, 20 mod 8 would be 4 because 20/8 is 2 remainder 4. Note that b cannot be 0 because any number divided by 0 is undefined. Also, we can write a mod $b = c$ as $a = b\lfloor a/b \rfloor + c$, where $\lfloor\ \rfloor$ denotes the floor function, which returns the largest integer smaller than the input.

Definition 3 *Given three integers x, y, and n, we can say* $x \equiv y$ (mod n) *if x/n and y/n have the same remainder. This is often phrased as "x and y have the same residue modulo n."*

For example, $20 \equiv 12$ (mod 8) because 20/8 and 12/8 both have a remainder of 4. In other words, 20 and 12 have the same residue modulo 8.

Multiplication Rules

Multiplication and division in modular arithmetic work exactly how they do in regular equations. For example, $20k \equiv 12k$ (mod 8) for any k because the remainders are both multiplied by k. This property can be extended to the following:

Property 1 *If* $a \equiv b$ (mod n) *and* $c \equiv d$ (mod n), *then* $ac \equiv bd$ (mod n).

Note that this idea can also be used for exponentiation, which is just repeated multiplication:

Property 2 *If* $a \equiv b$ (mod n) *, then* $a^c \equiv b^c$ (mod n).

Euclidean Algorithm

Theorem 1 *Let* $c = a$ mod b, *where* $b \neq 0$. *Then,*
gcd(a,b) = gcd(b,c).

Proof*:* If $a < b$, then after one iteration of the algorithm we get gcd(b, a) because a mod b would be a, making $c = a$ as well. Therefore, for the sake of simplicity, let's assume $a > b$.

By definition, $a = bx + c$ and thus $c = a - bx$ for some integer x. A number that divides b and c would divide $bx + c$, which means a common divisor of b and c is also a common divisor of a and b. Similarly a number that divides a and b would divide $a - bx$, which means a common divisor of a and b is also a common divisor of b and c. These two conditions mean a and b share the same common divisors with b and c, so their greatest common divisors would naturally be the same [2].

Bézout's Lemma

Theorem 2 *There always are integers u and v such that* $ua + vb = \gcd(a, b)$.

Algorithm 1
Bezout(a, b)
Input. a, b
Output. Lowest positive value of u, corresponding v

$d = \gcd(a, b)$
If $b = 0$:
    Return 1, 0
Else:
    $w, x = \text{Bezout}(b, a \bmod b)$
    $u, v = x, w - x\lfloor a/b \rfloor$
    $\bar{u} = u \bmod \frac{b}{d}$
    $k = \frac{\bar{u} - u}{b/d}$
    $\bar{v} = v - k\frac{a}{d}$
    Return $\bar{u}, \bar{v}$

Explanation*:* If b = 0, then d = a, so u = 1 and v = 0 because 1a + 0b = a.

For all other cases, we can start by using the Euclidian algorithm to realize that $wb + x(a \bmod b) = \gcd(b, a \bmod b) = \gcd(a, b) = ua + vb$ for some integers w and x. We know $a \bmod b$ can be written as $a - b\lfloor a/b \rfloor$, so after reorganizing variables, we get $xa + (w - x\lfloor a/b \rfloor)b = \gcd(a, b)$. This means $u = x$ and $v = w - x\lfloor a/b \rfloor$.

However, it turns out that there are an infinite number of (u,v) pairs that satisfy the equation. For any given u and v, all the pairs in the form $(u + k\left(\frac{b}{d}\right), v - k\left(\frac{a}{d}\right))$, are also solutions for any integer k because the $k\left(\frac{ab}{d}\right)$ terms cancel out. For this algorithm, we need to find the lowest positive value possible for $u + k\left(\frac{b}{d}\right)$, which we can call $\bar{u}$. Solving the expression for k leads to $k = \frac{\bar{u} - u}{b/d}$. Additionally, $\bar{u}$ is just the remainder when u is divided by $\frac{b}{d}$, so we know that $\bar{u} = u \bmod \frac{b}{d}$. Because we know $\bar{u}$ and k, we can now find $\bar{v}$, which is defined as the corresponding v-value for $\bar{u}$ [4].

Euler's Totient Function

Definition 4 $\varphi(n)$ is *the number of positive integers less than or equal to the integer n that are also relatively prime to n.*

Theorem 3 (Euler's Theorem) *If* $\gcd(a, n) = 1$ *for some integers a and n, then* $a^{\varphi(n)} \equiv 1 \pmod{n}$.

Proof*:* Let $E = \{r_1, r_2 \dots r_{\varphi(n)}\}$ be the $\varphi(n)$ numbers less than n that are relatively prime to n. If r_x is relatively prime to n, then ar_x is relatively prime to n as well because a is relatively prime to n. From this, we can conclude that the residues of the elements in $F = \{ar_1, ar_2 \dots ar_{\varphi(n)}\}$ modulo n are found only in set E.

To prove that set F is a permutation of set E modulo n, we must show that every element in set F is unique. Let's initially assume some two elements ar_x and ar_y in set F have the same residue modulo n. Because a and n are relatively prime, we can use Bézout's identity to write the following equation:
$ua + vn = 1$.

This can be rewritten as $ua = 1 - vn$, and because the right side of the equation is one more than a multiple of n, we can say that $ua \equiv 1 \pmod{n}$. Because $ar_x \equiv ar_y \pmod{n}$, multiplying both by u would mean

that $r_x \equiv r_y \pmod{n}$. However, r_x and r_y are elements in set E, which has unique elements by definition. Therefore our initial assumption must always be incorrect, and thus set F has unique elements and is a permutation of set E modulo n.

Multiplying all the elements in each set would result in the following equation:

$r_1 r_2 \ldots r_{\varphi(n)} a^{\varphi(n)} \equiv r_1 r_2 \ldots r_{\varphi(n)} \pmod{n}$.

After canceling all values of r_x, we get $a^{\varphi(n)} \equiv 1 \pmod{n}$. [1, 3].

Computing kth Roots Modulo m

All the concepts discussed earlier in the paper combine to form the foundation of this theorem, which is the backbone of the RSA algorithm.

Theorem 4 *Let m, b, and k be integers so* $\gcd(b, m) = \gcd(k, \varphi(m)) = 1$. *Also let u and v be integers so that* $uk + v(\varphi(m)) = 1$. *Then, x in the equation* $x^k \equiv b \pmod{m}$ *is equal to* $b^u \bmod m$.

Proof*:* We can rewrite b as $\frac{b}{b^{\varphi(m)v}}$ because $b^{\varphi(m)} \equiv 1 \pmod{m}$ by Euler's Theorem and $\frac{b}{1^v} = b$. With exponent rules, this can further be transformed into $b^{1 - v(\varphi(m))}$, which happens to be b^{uk} due to the given condition $uk + v(\varphi(m)) = 1$. So now we have the equation $x^k \equiv b^{uk} \pmod{m}$, which means $x \equiv b^u \pmod{m}$ [4].

Special Case

Let $m = pq$, where p and q are different prime numbers. To calculate $\varphi(m)$, we can use complementary counting. There are pq total numbers less than or equal to m. There are $m/p = q$ factors of p less than or equal to m and $m/q = p$ factors of q less than or equal to m, so we subtract p and q from m. However, this process counts m twice because it's a multiple of p and q, so we must add one to get the following final expression:

$\varphi(pq) = pq - p - q + 1 = (p - 1)(q - 1)$.

Conclusion

Let's return to the scenario of the two classmates, but now they will implement the RSA algorithm. The sender wants to send a message x, and at the moment, only the sender knows what x is. Two different prime numbers, p and q, are picked such that p and q don't divide x. Additionally, because there would otherwise be an infinite number of possibilities for x, x is defined so $0 \leq x < m$, where $m = pq$. The receiver also knows what p and q are. After this, k is chosen so k and $(p - 1)(q - 1)$ are relatively prime. With these values, $x^k \bmod m$, which we can call b, is computed. Note that $\gcd(b, m) = 1$ because b is a multiple of x but neither p nor q divide x. The sender sends b as a message, so the sender, the receiver, and also any potential viewers can see b, k, and m. The message x can therefore be derived with $b^u \bmod m$, as shown in the previous section (to reiterate, u and v are defined such that $uk + v(\varphi(m)) = 1$, so u can be found with Bézout's Lemma). However, in a real-life application, m would be so big and so hard to factor that the only way x can be found is with p and q, which the receiver has but any outside viewers do not, allowing the RSA algorithm to function successfully.

Acknowledgments

I would like to thank Dr. Goldsztein from the Georgia Institute of Technology for his assistance with this paper.

References

[1] *Fermat's Little Theorem and Euler's Theorem*. (n.d.). [Handout]. Art of Problem Solving.

[2] Keef, P., & Guichard, D. (n.d.). *3.3 The Euclidean Algorithm*. Introduction to Higher Mathematics. Retrieved July 1, 2022, from https://www.whitman.edu/mathematics/higher_math_online/section03.03.html.

[3] Modular arithmetic. (2010). In A. Nayak (Ed.), *Discrete Mathematics*. U. Waterloo. https://www.math.uwaterloo.ca/~anayak/courses/ece103-s10/notes/modular-arithmetic.pdf.

[4] Silverman, J. H. (2013). A Friendly Introduction to Number Theory. India: Pearson.

[5] Simmons, G. (2022, August 2). *cryptology | Definition, Examples, History, & Facts*. Encyclopedia Britannica. https://www.britannica.com/topic/cryptology/History-of-cryptology.