Entendiendo el sistema de encriptación más usado en el mundo y rompiendolo con teoría de números

Alexander Guadalupe Vázquez Marín

May 2023

1. Introducción

Imagínate por un momento que de la noche a la mañana desaparecieran Internet, los teléfonos y todos los dispositivos electrónicos. Si quisiéramos comunicarnos con alguien que vive lejos, tendríamos que recurrir al antiguo metodo de enviar una carta. Sin embargo, enviar mensajes a través de cartas plantea ciertos problemas de seguridad. En cualquier momento, el cartero podría leer el contenido de la carta o alguien podría robarla en algún descuido.

Ahora, considera la situación en la que necesitas comunicarte con alguien a distancia y deseas asegurarte de que nadie más lea la información que estás a punto de compartir. Ya sea que estés enamorado y estés enviando una carta de amor muy emotiva, un chef que envía la receta secreta de su plato estrella o un general en guerra dando instrucciones a sus soldados, ¿qué harías para evitar que un tercero revele el contenido de la carta en caso de robo?

Precisamente para resolver este problema, desde la antigüedad se ha recurrido al uso de la criptografía, el arte y la técnica de escribir con procedimientos o claves secretas, de tal forma que lo escrito solamente sea inteligible para quien sepa descifrarlo. Uno de los ejemplos más tempranos de criptografía se encuentra en Mesopotamia alrededor de 1500 a.C., donde se descubrió un mensaje encriptado en unas tablas de arcilla que detallaba la fabricación de vidrio de cerámica.

Si bien en la era moderna el uso de cartas ha quedado obsoleto como nuestro principal medio de comunicación, el Internet tampoco está exento de que la información que se envía pueda ser interceptada por un tercero. Todos los días compartimos información sensible a través de Internet, como mensajes a nuestra familia y amigos, fotos y documentos guardados en la nube, o nuestros datos bancarios al realizar compras en línea. No es tarea imposible para un hacker interrumpir la comunicación entre dos sitios web y revelar la información que se está compartiendo. Para prevenir este tipo de situaciones, los desarrolladores, siendo conscientes de este riesgo, recurren al uso de la criptografía.

2. Un buen método

Como ya vimos la encriptación de mensajes juega un papel fundamental en nuestra vida cotidiana, y es crucial contar con un sistema sólido para salvaguardar nuestra información de posibles ataques. Crear un sistema de encriptación desde cero no es una tarea excesivamente complicada. Una idea básica podría ser sustituir cada letra de una palabra por un número correspondiente, de manera que solo aquellos que conozcan el número correspondiente a cada letra puedan interpretar el contenido. Sin embargo, este enfoque no es muy robusto, ya que los códigos basados en sustitución pueden ser descifrados mediante métodos estadísticos. Por ejemplo, en el idioma español, las letras tienen frecuencias de aparición diferentes y se ubican en posiciones específicas dentro de una palabra. Teniendo esto en cuenta, alguien podría descifrar el mensaje de manera relativamente sencilla mediante conjeturas, sustituyendo los números más comunes por letras ampliamente utilizadas. Con este método, eventualmente se encontraría una parte del mensaje que tenga sentido y, a partir de ahí, desencriptar el resto no sería tan complicado.

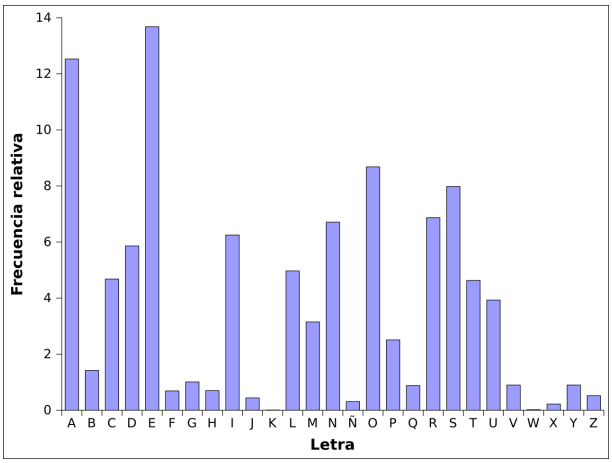

Figura 1: Distribución del uso de las letras en el idioma español.

Por lo tanto, para que un sistema de encriptación sea efectivo, es necesario ir más allá de simplemente convertir las letras en números. Necesitamos encontrar un método que transforme los números de tal manera que resulte imposible para cualquier persona ajena al destinatario del mensaje desencriptarlos.

Con esto en mente, podemos tener una idea de los requisitos que debe cumplir un buen sistema de encriptación. En primer lugar, debe ser realizable, es decir, debemos poder encriptar el mensaje en un tiempo razonable. Sería inconveniente utilizar un método que tome demasiado tiempo para encriptar. Además, como ya mencionamos, debe ser casi imposible para terceros reconstruir el mensaje original, pero al mismo tiempo el destinatario debe poder descifrarlo en un tiempo razonable.

También debemos tener en cuenta la dificulad de que, en el entorno de Internet, normalmente no es posible reunirnos con la persona a la que queremos enviar un mensaje y acordar un código o algoritmo que ambos vayamos a emplear para encriptar y desencriptar los mensajes. Todos estos requisitos y restricciones hacen que nos percatemos de que si bien es sencillo proponer un sistema de encriptación, proponer uno que sea realmente robusto y confiable es una tarea extremadamente complicada.

3. El sistema RSA

El sistema que vamos a explorar cumple con todas las características que mencionamos anteriormente. Se trata del sistema de encriptación RSA, creado en 1978 por Ronald Rivest, Adi Shamir y Leonard Adleman, de quienes proviene su nombre. Este sistema ofrece una notable seguridad y flexibilidad, ya que no requiere que el remitente y el destinatario se encuentren en persona antes de intercambiar mensajes. Estas cualidades han convertido al sistema RSA en uno de los más utilizados, siendo empleado ampliamente por bancos, sitios web e incluso gobiernos.

Antes de proporcionar una explicación precisa del algoritmo, vamos a ofrecer una descripción intuitiva de cómo funciona este sistema de encriptación mediante la siguiente analogía:

Imaginemos a Fernando, quien le pidió a sus fammiliares que le envien por correo las recetas secretas de su familia. Naturalmente, no desea que si alguien intercepta el mensaje pueda leer las recetas. Para resolver este problema, Fernando tiene la idea de crear una caja fuerte con dos llaves: una llave maestra que puede abrir la caja fuerte y ver su contenido, y una llave pública que solo puede abrir un orificio para introducir objetos, pero no permite extraerlos.

Fernando envía por correo la caja fuerte a cada uno de sus familiares, junto con la llave pública. Él se queda con la llave maestra. Cada persona recibe la caja fuerte e introduce su mensaje usando la llave pública. Luego, envían de vuelta la caja fuerte por correo.

Finalmente, Fernando utiliza su llave maestra para abrir la caja fuerte y revisar su contenido. Este sistema aporta bastante seguridad pues en caso de que alguien haya interceptado la caja fuerte, no podrá ver su contenido, ya que solo tiene acceso a la llave pública que solo permite introducir objetos, pero no abrir la caja fuerte.

En resumen, así funciona el sistema de encriptación RSA: la persona interesada en recibir mensajes genera dos llaves, una pública y una privada. La privada es a la que anteriormente llamamos "llave maestra". Esta persona expone la llave pública para que cualquiera que quiera mandar un mensaje secreto pueda cifrarlo usando dicha llave. Luego, esta persona publica el mensaje y es allí donde nosotros usamos la llave privada para descifrar el contenido del mensaje.

Si bien es fácil comprender por qué es tan bueno este sistema de encriptación, realmente es difícil imaginar cómo es posible cifrar un mensaje sin saber cómo descifrarlo después. La manera en que el algoritmo RSA logra esto es mediante

el uso de herramientas de la teoría de números y aprovechando el hecho de que es muy difícil factorizar un número muy grande en sus factores primos así como resolver un problema conocido como el logaritmo discreto.

3.1. conocimientos preliminares

En este sistema de encriptación, como se mencionó anteriormente, se utilizan herramientas de la teoría de números. Por lo tanto, consideramos importante proporcionar algunas definiciones y resultados clave para comprender el método. Sin embargo, la revisión de las demostraciones de los siguientes teoremas se considera opcional. Como nota, para evitar repetición, cuando se haga mención de un número nos referimos a un número entero a menos que se indique lo contrario.

3.1.1. Definición de divisibilidad

Sea $d \neq 0$ decimos que d divide a n y lo denotamos de la forma $d \mid n$ si podemos a escribir a n como $n = c * d$ para algún entero c. También decimos n es un multiplo de d, que d es un divisor de n, o que d es un factor de n. Ejemplo: $7 \mid 14$ ya que $14 = 2 * 7$

3.1.2. Definición de máximo común divisor

Sea d un divisor común de a y b, es decir $d \mid a$ y $d \mid b$ entonces decimos que d es el máximo común de divisor, si d es el entero más grande que cumple que es divisor común de a y b, es costumbre denotar al máximo común de a y b como $(a, b) = d$. Como nota, también decimos que a y b son primos relativos si $(a, b) = 1$

3.1.3. Teorema 1, identidad de bezoût

Sea $(a, b) = d$, entonces d puede expresarse como una combinación lineal de a y b, es decir $ax + by = d$.
Demostración: Primero asumimos que $a \geq 0$ y $b \geq 0$. Usamos inducción en n, donde $n = a + b$. Si $n = 0$, entonces $a = b = 0$ y podemos tomar $d = 0$ con $x = y = 0$. Supongamos, entonces, que el teorema se ha demostrado para $0, 1, 2, ..., n - 1$. Por simetría, podemos suponer $a \geq b$. Si $b = 0$, tomamos $d = a$, $x = 1$, $y = 0$. Si $b \geq 1$, aplicamos el teorema a $a-b$ y b. Dado que $(a-b) + b = a = n - b \geq n - 1$, se cumple la suposición de inducción y hay un divisor común d de $a-b$ y b de la forma $d = (a-b)x + by$. Este d también divide $(a-b) + b = a$, por lo que d es un divisor común de a y b y tenemos $d = ax + (y-x)b$, una combinación lineal de a y b. Para completar la prueba, debemos demostrar que todo divisor común divide d. Pero un divisor común divide a y b y, por linealidad, también divide a d. Si $a \geq 0$ o $b \geq 0$ (o ambos), podemos aplicar el resultado recién demostrado a $|a|$ y $|b|$. Entonces, hay un

divisor común d de $|a|$ y $|b|$ de la forma

$$d = |a|x + |a|y$$

Si $a < 0$, $|a|x = -ax = a(-x)$. De manera similar, si $b < 0$, $|b|$ = b(-y). entonces d es nuevamente una combinación lineal de a y b.

3.1.4. Definición congruencia entre números

Decimos que dos números a y b son congruentes módulo n si se cumple que $n \mid a-b$, lo denotamos de la siguiente manera, $a \equiv b \pmod{n}$. Ejemplo: $22 \equiv 2 \pmod{10}$, pues $22-2 = 20 = 2 * 10$, luego $10 \mid 22-2$

3.1.5. Definición inverso multiplicativo módulo n

Decimos que d^{-1} es el inverso multiplicativo de d modulo n si se cumple que $d^{-1} * d \equiv 1 \pmod{n}$.

3.1.6. Teorema 2, existencia de inverso

$(a, m) = 1$ si y solo si existe un x tal que $a * x \equiv 1 \pmod{m}$.

Demostración: Si $(a, m) = 1$, entonces por el teorema 1, identidad de bozoût, existe x y y tal que $ax * my = 1$, reordenando $ax - 1 = (-y)m$, es decir $a * x \equiv 1 \pmod{m}$. De manera conversa, si $a * x \equiv 1 \pmod{m}$, entonces existe un y, tal que $ax + my = 1$, entonces $(a, m) = 1$.

3.1.7. Definición función $\phi(n)$ de Euler

Si n es un número entero positivo, entonces $\phi(n)$ se define como la cantidad de enteros positivos menores a n y primos relativos con n. Ejemplo: $\phi(6) = 2$ pues solo $1, 5$ son primos relativos con 6, por otra parte $\phi(7) = 6$, pues $1, 2, 3, 4, 5, 6$ son primos relativos con 7.

Un resultado importante de la función $\phi(n)$ de Euler es que es una función multiplicativa. Esto significa que si tomamos x, y primos relativos, entonces $\phi(x * y) = \phi(x) * \phi(y)$

3.1.8. Teorema 3, teorema de Euler

Dados dos números a y m, si se cumple que son primos relativos, entonces:

$$a^{\phi(m)} \equiv 1 \pmod{m}$$

La demostración de este teorema se sale del alcance de este artículo, pero para los lectores con mayor formación matemática que esten interesados , este teorema puede demostrarse de manera sencilla considerando al conjunto de los residuos modulo m como un grupo abeliano finito de orden $\phi(m)$ y luego aplicando el teorema de lagrange, se deja como ejercicio para el lector terminar los detalles.

3.1.9. Definición periodo de un número

El período de un número módulo n se refiere al menor entero positivo p tal que $a^p \equiv 1 \pmod{n}$, donde a es el número dado y n es el módulo. Ejemplo: el periodo de 2 módulo 7 es 3, pues 3 es el menor natural para el que se cumple que $2^3 \equiv 1 \pmod{7}$.
Una observación relevante es que los residuos que son primos relativos con el módulo forman un grupo finito bajo el producto, luego para todo a tal que $(a, n) = 1$ se tiene que a tiene periodo finito.

3.2. En que consiste el método

Ahora hablaremos sobre como es que la teoría de números hace que sea posible encriptar un mensaje sin tener conocimiento de como desencriptarlo después, la realidad es que este enunciado tomado de manera literal no es del todo cierto, ya que en la practica si sabes como encriptar el mensaje entonces necesariamente también sabes como desencriptarlo. El punto clave del sistema RSA es que, usando la llave pulica es posible encriptar el mensaje de manera rapida, más sin embargo que a la hora de intentar desencriptarlo sin tener la llave privada, aunque sea posible hacer esto, nos encontremos con que el tiempo que nos tomaría es extremadamente grande, en la práctica esto son años de cálculo usando la computadora más potente en la actualidad. A las funciones que hacen esto de ser faciles de calcular en un sentido, pero muy dificiles en el sentido opuesto se les conoce en las ciencias computacionales como funciones de una dirección. De lo que se dieron cuenta los creadores del sistema RSA es que en la teoría de números había un ejemplo perfecto de función de una dirección y está es:

$$f(x) = x^b \pmod{n} \tag{1}$$

notemos que el cálculo de está función es bastante sencillo y rápido, simplemente elevamos x a la b y revisamos a que es congruente modulo n. Sin embargo si quisieramos tratar de computar la segunda dirección, es decir la función inversa, tendríamos que encontrar un x tal que al elevarlo a la b sea congruente con un y módulo n, a este problema se le conoce como el problema de logaritmo discreto y hasta la fecha no hay ningun algoritmo que permita hacer el computo de manera rápida ya que en el mejor algorimo conocido para cálcularlo, el tiempo de computo crece exponencialmente conforme se aumenta el tamaño del número a computar.

Muy bien, ya tenemos una función de una dirección (1), que nos permite que sea rápido cifrar el mensaje, pero muy lento intentar descifrarlo, ahora el problema que surge es que nosotros también debemos ser capaces de descifrarlo y de manera rápida con ayuda de nuestra llave privada, esto lo logramos eligiendo en (1) un n y un b que cumplan unas propieades especiales, de manera que tenindo conocimiento de la llave privada, nostros podamos resolver el problema del logaritmo discreto de manera rápida aplicando un truco de la teoría de

números, a continuación primero presentaremos el método que se sigue en el sistema RSA y luego se explicará el porqué aplicar este truco funciona.

Para generar nuestras llaves de encriptación, el primer paso es seleccionar aleatoriamente dos números primos grandes, p y q, donde $p \neq q$. Calculamos su producto para obtener n, es decir, $n = p * q$. Luego, necesitamos encontrar un número x que sea primo relativo con $(p-1)(q-1)$. Finalmente, calculamos un número y de manera que sea el inverso multiplicativo de x módulo $(p-1)*(q-1)$, es decir $x * y \equiv 1 \pmod{(q-1)(p-1)}$, De esta manera, hemos generado nuestras dos llaves: la llave privada, que utilizaremos para desencriptar, será (x, n), y la llave pública que transmitiremos será (y, n).

Una vez que tenemos las llaves, el siguiente paso es utilizar la llave pública para encriptar nuestro mensaje. Para ello, como mencionamos en secciones atrás, es una buena idea convertir las letras del texto en números, ya que es más sencillo manipular números que letras. Podemos lograr esto asignando a cada caracter su número correspondiente según la tabla ASCII. Por ejemplo, A = 65 y w = 119. Además, se acostumbra sumarle 100 al valor resultante para asegurarnos de que cada caracter tenga un número de 3 dígitos, des esta manera A= 165 y w=219. Una vez completado este procedimiento, nuestro mensaje estará representado como una secuencia de enteros, $m_1, m_2, ..., m_k$.

Luego, encriptamos cada elemento de la secuencia aplicando la llave pública de la siguiente manera. Sea e_i el elemento encriptado correspondiente a m_i. Calculamos e_i como $e_i \equiv m_i^y \pmod{n}$

Finalmente, la persona que intenta enviar el mensaje publica la secuencia $e_1, e_2, ..., e_k$. Por otro lado, la persona destinataria del mensaje utiliza su llave privada para desencriptar el mensaje aplicando la regla $m_i \equiv e_i^x \pmod{n}$ Luego, solo tiene que volver a convertir el mensaje original restando 100 a cada m_i y utilizando la tabla ASCII. A continuación, se muestra de manera resumida los pasos a seguir:

3.3. Resumen del algoritmo de encriptación RSA

1. Encuentra dos números primos grandes, p y q, $p \neq q$ y calcula su producto $n = p * q$.
2. Encuentra un número entero a que sea primo relativo con $(p-1)(q-1)$.
3. Calcula el invero multiplicativo de a modulo $(p-1)(q-1)$, y denotemos a este inverso por $y = x^{-1}$
4. Transmite la clave pública, es decir, el par de números (y, n).
5. Representa el mensaje a transmitir como una secuencia de enteros $m_1, m_2, ..., m_k$.
6. Encripta cada m_i usando la clave pública aplicando la regla

$$e_i \equiv m_i^y \pmod{n} \tag{2}$$

7. El receptor descifra el mensaje usando la regla

$$m_i \equiv e_i^x \pmod{n} \tag{3}$$

8. Vuelve a convertir la secuencia de enteros al mensaje original

3.4. ¿Por qué funciona?

En el paso 7 mencionamos que podemos descifrar el mensaje usando la regla que allí se menciona, ahora demostremos que esto de verdad se cumple, es decir, que podemos cálcular m_i como $m_i \equiv e_i^x \pmod{n}$.

Demostración:

$$\begin{aligned} x * y &\equiv 1 \pmod{(p-1)(q-1)} \quad \text{porque y es el inverso de x} \\ x * y &= k * (p-1) * (q-1) + 1 \quad \text{por definición de congruencia} \end{aligned}$$

notemos que $\phi(n) = \phi(p * q) = (p-1)(q-1)$ para $p \neq q$ esto porque como p y q son primos diferentes y $\phi(n)$ es una función múltiplicativa entonces $\phi(p * q) = \phi(p) * \phi(q)$. También notese que para cualquier número primo t se cumple que $\phi(t) = t-1$ pues por defínción de número primo, t es primo relativo con todos lo núemeros menores a él. De esto concluimos que $\phi(n) = (p-1)*(q-1)$

$$x * y = k * \phi(n) + 1 \quad \text{por el resultado anterior} \tag{4}$$

finalmente consideramos:

$$\begin{aligned} e_i &\equiv m_i^y \pmod{n} && \text{por como definimos a } e_i \\ e_i^x &\equiv (m_i^y)^x \pmod{n} && \text{elevando a la x} \\ e_i^x &\equiv m_i^{y*x} \pmod{n} && \text{por propiedades de potencia} \\ e_i^x &\equiv m_i^{k*\phi(n)+1} \pmod{n} && \text{sustituyendo (4)} \\ e_i^x &\equiv m_i^{k*\phi(n)} * m_i \pmod{n} && \text{por propiedades de potencia} \\ e_i^x &\equiv 1 * m_i \pmod{n} && \text{por el teorema de euler} \\ e_i^x &\equiv 1 * m_i \pmod{n} && \blacksquare \end{aligned}$$

4. Rompiendo RSA con computación cuántica

Una posible vulnerabilidad del sistema que podemos observar es que si un tercero, logra descomponer a n en sus factores primos entonces le resultaría sencillo, cácular la llave privada. ya que solo necesitaría encontrar el inverso de y módulo $(p-1)(q-1)$ afortunadamente, como ya hemos mencionado, el

poder factorizar a un número n en sus factores primos no es una tarea sencilla de computar. Sin embargo, recientemente, en los laboratorios Bell, se ha descubierto un algoritmo conocido como el algoritmo de Shor, que permite factorizar un número n en sus factores primos en un tiempo razonable. Este algoritmo se basa en la teoría de números y su funcionamiento es el siguiente:

4.1. Algoritmo de Shor's

1. Escoje de manera aleatoría un número a que cumpla ser primo relativo con n (esto puede verificarse con el algoritmo de Euclides.)
2. Cálcula el periodo r de a módulo n
3. Verifica que r sea divisible entre 2 y que $a^{r/2} + 1 \not\equiv 0 \pmod{n}$. Si no se cumple alguna de estas dos condiciones regresa al paso uno y eligé otro número.
4. Por definición de periodo tendremos que $a^r \equiv 1 \pmod{n}$
5. Factorizando y aplicando definición de congruencia $(a^{r/2} - 1)(a^{r/2} + 1) = k * n$
6. Computamos el máximo común divisor $(a^{r/2} - 1, n) = p$ y $(a^{r/2} + 1, n) = q$

Antes de proceder demostraremos la aseveración que mencionamos en el paso 6.
Sabemos que p es un número primo, y que $p \mid (a^{r/2}-1)(a^{r/2}+1)$ luego $p \mid a^{r/2}-1$ o $p \mid a^{r/2} + 1$. Sin perdida de generalidad asumamos $p \mid a^{r/2} - 1$ y $q \mid a^{r/2} + 1$. Para ello demostremos que p y q no pueden divir al mismo fator. Que ambos dividan al mismo factor significa que n divide a alguno de los factores, esto es una contradicción ya que $n \nmid a^{r/2} + 1$ por hipotesis (revise paso 3), y $n \nmid a^{r/2} - 1$ pues implica que $a^{r/2} \equiv 1 \pmod{n}$ lo cual es una contradicción con que r es el periodo de n ■.

Si eres observador, notarás que todos los pasos, excepto el número 2, son fáciles de calcular. Sin embargo, calcular el período de un número utilizando computación convencional es una tarea que crece exponencialmente en función del tamaño del número. Afortunadamente, el algoritmo de Shor's, utilizando computación cuántica, permite encontrar el período de un número de manera relativamente rápida. No entraremos en detalles precisos sobre cómo lo logra, ya que implica el uso de herramientas más avanzadas que están fuera del alcance de este artículo, pero debes saber que es posible realizar esta tarea.

Hasta el día de hoy, las computadoras cuánticas aún no tienen suficiente capacidad de cómputo para calcular el período de un número lo suficientemente rápido como para que el sistema de encriptación RSA deje de utilizarse. Sin embargo, se espera que en unos pocos años los equipos de cómputo adquieran la capacidad necesaria para hacerlo. Cuando eso ocurra, muchos de los algoritmos de encriptación actuales se volverán obsoletos, lo que plantea un desafío para

la seguridad informática tal como la conocemos. Por esta razón, se ha considerado prioritario desarrollar sistemas de encriptación que sean resistentes a la computación cuántica.

Referencias

[1] Colin P. Williams *Exploration in quantum computing*. Springer, 2011.

[2] Doyglas R. Stinson, Maura B. Paterson. *Cryptography Theory and Practice*. Taylor and Fracis Group, 2019.

[3] I.N. Hernstein. *Algebra abstracta*. Grupo Editoral Iberoamericana, 1988.

[4] Ivan Niven, Herbert S. Zuckerman, Hugh L. Montgomery. *An Introduction to the Theory of Numbers*. Courier Companies, Inc., Año 1991.

[5] Aaronson, Scott. *Shor i'll do it*. https://scottaaronson.blog/?p=208 , año 2010, consultado 22 de mayo de 2023

[6] Tom M. Apostol, *Analytic Number Theory*, Springer, 1976.