Criptografía

Jorge Urroz[1]

ETSI Caminos, Canales y Puertos,
Universidad Politécnica de Madrid.

1. Compartiendo

Ignacio: Chicos, ¿habéis pensado lo que vais a hacer el año que viene?

Daniel: Yo ya no voy a estudiar más. Me voy a poner a trabajar en el taller de mi padre.

Hugo: ¡Que suerte tienes, bro! Ya vas a poder comprarte un coche, y ¡nos iremos de fiesta!

Daniel: Jajaja. Y ¿vosotros?

Ignacio: Yo no tengo ni idea. Sí que quiero ir a la universidad, pero la verdad no sé. ¿Qué es lo que da más pasta?

Hugo: Ahora están de moda las Mates.

Ignacio: Mates, ¿de moda? ¿Y qué hacen ahora con mates?. ¿No está ya todo hecho desde hace siglos?

Hugo: Yo que sé. Mira ahí están Sofía y Eva. Chicas, Ignacio está preocupado por lo que estudiar el año que viene.

Sofía: Yo voy a hacer Mates.

Hugo: Jajaja ¡ya os decía yo!

Eva: Yo también lo he pensado. Pero no sé si Mates, Física, o alguna Ingeniería. Es que me gustaría algo más aplicado.

Daniel: Menudos frikis. Yo me piro. Ahí os dejo discutiendo del origen del universo, que hoy hay partido.

Sofía: Uy Eva, el caso es que ahora todo el mundo quiere hacer Mates, justo porque se aplican a un montón de cosas.

Ignacio: Ah ¿si?

Sofía: Todo lo que tiene que ver con los ordenadores va de mates. En plan, cuando buscas en Google, el PC tiene que mirar un montón de páginas y decidir qué te saca, y todo es gracias a las mates. Y lo mismo con la IA, ¿sabes?

Hugo: Ahora que lo dices, yo he oido que también en el Whatsapp utilizan Mates.

Eva: Y ¿eso?

[1]jorge.urroz@upm.es

Hugo: Rollo, cuando mandas un mensaje, asegurarte de que nadie lo puede leer mas que el que lo recibe.

Ignacio: ¿Y eso cómo se hace?.

Hugo: Es como que va encriptado o algo así. Mira, lo vamos a poner en Google, y así usamos Mates, jajaja. A ver "encriptado Whatsapp"

Google: Generando... Vista creada con IA.

"El cifrado de extremo a extremo de WhatsApp es un método de seguridad que protege los mensajes y las llamadas. Esto garantiza que solo tú y la persona con la que chateas puedan leer o escuchar lo que se envía".

Ignacio: ¡Ostras! "cifrado de extremo a extremo"...

Sofía: Qué interesante. Bueno, yo me marcho, que llego tarde.

Todos: Sí, ya es hora. ¡Hasta mañana!

1.1. Un poco más tarde

Sofía está creando un grupo de Whatsapp con todos de título: Mates.

Sofía: (en el grupo Mates). Chicos, he estado mirando y el cifrado de extremo a extremo que dice Google tiene que ver con Criptografía, ¡justo una disciplina que se enseña en Mates! He encontrado un artículo en la página del grado de Mates de la Politécnica que habla sobre estas cosas, dirigido a alumnos que vayan a empezar en la universidad, para que lo conozcan. ¿Os apetece que le echemos un vistazo? 👍🏻 2

Eva: Pues sí, ¡es buena idea!

Hugo: ok.

Ignacio: Anda, pues estaría muy bien. A lo mejor nos sirve para centrarnos.

Daniel: Nada, nada, miráoslo y luego me lo contáis. Jajaja. 🤣 3

Sofía: Jajaja. Vale, pues os lo mando, y nos organizamos para leerlo. Esta es la dirección

https://blogs.upm.es/retazos-matematicas/criptografia/

2. Explicando

(Al día siguiente)

Hugo: No hay manera de entender la parte que me ha tocado. De verdad, no sé ni de lo que habla.

Sofía: Lo mío también es muy difícil

Ignacio: Sí, menos mal que lo decís. Yo tampoco entiendo bien mi parte...

Sofía: ¿Y si le pedimos al autor que nos lo explique?

Hugo: Anda ya. Nos va a mandar a la m (lo digo de forma encriptada ja ja ja)

Eva: Pues a mí me parece buena idea. Si nos dice que sí, lo entenderemos todo de una vez, y nos puede ayudar.

Ignacio: Venga, va ¿quién le escribe?

Sofía: Su correo electrónico está en el mismo artículo.

Hugo: Sí, ¿pero quién es el valiente que escribe el mensaje?

Eva: ¿Y si le preguntamos al Chatgpt?

Hugo: Jajaja qué buena idea. (Escribe en el Chatgpt). Quiero escribir una carta a un profesor de universidad para que nos explique un artículo suyo. Somos alumnos de secundaria y vamos a entrar en la universidad el año que viene.

Chatgpt:

Asunto: Consulta sobre su artículo de divulgación
Estimado Profesor,

Mi nombre es Sofía, somos un grupo de estudiantes de último año de bachillerato y el próximo año comenzaremos los estudios universitarios, con mucho interés en el área de Criptografía.

Recientemente leímos su artículo publicado en la página del GeM (Grado en Matemáticas) y, aunque suena muy interesante, no conseguimos entenderlo bien. Nos gustaría saber si podría explicárnoslo un poco más.

Le agradezco mucho su tiempo y el trabajo de divulgación que realiza.
Atentamente,
Sofía.

2.1. La carrera entre Criptografía y criptoanálisis.

Todos tenemos un secreto: desde un amor o una emoción, hasta una intención que puede tener grandes consecuencias, por ejemplo, de carácter bélico o económico, o simplemente el secreto es una clave. Y por otro lado también tenemos la necesidad de compartir dicha información, ya sea con nuestros seres queridos, el ejército o nuestra empresa, o de recuperar en cualquier momento la clave para tener acceso a nuestros bienes. Esta dicotomía, entre esconder y comunicar, es el origen de la Criptografía. Según el diccionario de la RAE, la Criptografía es "El arte de escribir con clave secreta o de un modo enigmático."Se trata de una ciencia que permite cifrar un mensaje, asegurando que solo el destinatario escogido puede descifrar su contenido evitando que un tercero no autorizado, enemigo, competidor, etc, pueda acceder a la información.

Su naturaleza hace que la Criptografía haya existido a lo largo de toda nuestra historia y, a pesar de ello, no tenemos la solución definitiva al problema. De hecho por cada mecanismo criptográfico que surge para mantener oculta la información, o segura la comunicación, surge una nueva técnica del criptoanálisis para romper la seguridad y obtener mensajes que, en algunos casos, han dado un giro crucial a nuestra historia. Este es el caso por ejemplo de la desencriptación de la máquina Enigma utilizada por ejército Nazi en la segunda guerra mundial, y que fue determinante para la victoria aliada. El proceso se llevó mayormente en la oficina de inteligencia de Bletchley Park, Londres, encabezada por uno de los grandes de la historia de las Matemáticas, y precusor del diseño del ordenador, Alan Turing.

La consecuencia directa de esta lucha entre Criptografía y criptoanálisis es que las herramientas utilizadas en una y otra son cada vez más sofisticadas, llegando a ser necesario el conocimiento de Matemáticas de un nivel tan alto que solo está al alcance de unos pocos. Y, ¿por qué Matemáticas?. Empecemos desde el principio. ¿Cómo es que somos capaces de escribir una información que solo algunos pueden leer? Veámoslo.

Figura 1: **Dos armas. La Brutalidad contra el ingenio.** A la izquierda, soldados alemanes encriptando la información con la maquina Enigma durante la Segunda Guerra mundial.(Cortesía de Helge Fykse, Norway, [8] y cryptomuseum, [9]). A la derecha Alan Turing, (Wikipedia: autor desconocido).

Como hemos mencionado, para encriptar un mensaje necesitamos una clave secreta, y es el conocimiento de dicha clave el que permitirá al destinatario descifrar el mensaje. La forma más sencilla de llevar a cabo este procedimiento es compartir con anterioridad la clave secreta entre remitente y destinatario, y luego utilizarla para enviarse mensajes de forma segura. Uno de los primeros ejemplos de este tipo de algoritmo ya fue utilizado por Julio César de una forma muy sencilla de describir. Su método era sustituir cada letra del alfabeto por la letra situada tres posiciones hacia delante y, al llegar al remitente, este haría el proceso contrario, sustituyendo cada letra del mensaje por la que se sitúa 3 posiciones hacia atrás. Un ejemplo en nuestro alfabeto sería el siguiente: Daniel quiere mandar el mensaje "Qué envidia con lo de Criptografía.ª Sofía y no quiere que sus padres o sus otros amigos lo sepan. Acuerdan que la clave secreta es 3, es decir, trasladarán cada letra del mensaje en claro 3 posiciones hacia delante para obtener las letras del mensaje cifrado. De esta forma ambos construyen una tabla de sustitución, como la siguiente

Texto en Claro	A	B	C	D	E	F	G	H	I	J	K	L	M	N
Texto Cifrado	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P
Texto en Claro	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	
Texto Cifrado	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	

Daniel manda el mensaje cifrado

txhhpylgldfrpñrghfulswrjudild

y Sofía utiliza a la inversa la tabla para recuperar el mensaje. Si el mensaje lo intercepta una tercera persona, no entenderá lo que está escrito, y Daniel estará a salvo, ¿verdad?. En realidad, no es tan sencillo. ¿Cómo puede estar Daniel seguro de que nadie, salvo Sofía, será capaz de leer su mensaje?, en otras palabras, ¿cómo podemos garantizar la seguridad de nuestra comunicación si cae en manos de un tercero? De la respuesta no solo depende el futuro de Daniel, sino que, en realidad,

de alguna forma depende el futuro de la humanidad. En la actualidad, gobiernos, ejércitos, entidades bancarias, software, multinacionales, servicios de inteligencia, y en realidad cualquier situación que requiera comunicación entre dos partes, basa la seguridad de su privacidad en la Criptografía. Cada vez que se utiliza Whatsapp, tu tarjeta bancaria, transacciones con bitcoin, o cada vez que alguna compañía almacena tus datos, garantiza su seguridad mediante criptosistemas, es decir, algoritmos para encriptar el mensaje, y las Matemáticas que hay detrás de ellos.

Para garantizar la seguridad de la comunicación, lo primero que debemos hacer es analizar las formas en que se puede romper su privacidad. Pongámonos en el papel del criptoanalista, que llamaremos R por conveniencia. R es un tercero, experto en la materia, que ataca la seguridad del criptosistema con el propósito de descifrar el contenido del mensaje. Supongamos pues que R ha interceptado el mensaje de Daniel. Aquí debemos hacer un inciso. Y es que, para poder confiar en un criptosistema, debemos ponernos en el peor de los casos. La comunidad ha adoptado un convenio universal, denominado principio de Kerckhoffs, en honor al lingüista y criptógrafo holandés del mismo nombre, que afirma que la seguridad de un criptosistema no puede depender del desconocimiento del algoritmo utilizado. Es decir, daremos por supuesto que R conoce el método, en el caso del ejemplo, el algoritmo del César, y lo que no conoce es la clave que compartieron Sofía y Daniel, es decir, el 3. La debilidad del criptosistema del César se hace evidente, y más con la potencia de los ordenadores en la actualidad. Lo único que tiene que hacer R es utilizar la fuerza bruta, es decir, cada una de los posibles números del 1 al 26, que determinan las posiciones a trasladar hacia atrás, y ver el resultado del mensaje trasladado.[2]. En nuestro caso, comenzamos con $k = 1$, luego $k = 2$, $k = 3$, etc y vemos que al restar k al mensaje cifrado aparecen los mensajes

swggoxkfkceqonqfgetkrvqitchkc
rvffñwjejbdpñmpefdsjquphsbgjb
queenvidiaconlodecriptografia

y después de tres cálculos nos muestra el mensaje en claro.

Hay varios puntos a analizar en el anterior ejemplo. Si nos damos cuenta detenidamente, observamos que ya en el algoritmo de César estamos utilizando Matemáticas, en este caso a nivel elemental. Si a cada letra le hacemos corresponder el número que indica su posición en el alfabeto, trasladar una letra tres posiciones hacia delante será lo mismo que sumarle tres, y para recuperar la letra, simplemente tendremos que restar tres, con una salvedad. Si nos aparece una "A.$^{\text{en}}$ el mensaje cifrado, ¿cómo le vamos a restar 3?. La respuesta está en el famoso concurso de "pasapalabra"de la tele. Allí se utiliza un rosco para representar el abecedario, y el concursante puede recorrerlo tantas veces como quiera de forma cíclica. Así pues, la letra que está 3 posiciones por detrás de la "A"no es más que la "X".

Un rosco no es más que un reloj, en este caso de 27 horas, y así como nosotros podemos decir que son la 1 de la tarde, o las 13, haciendo corresponder a cada hora la equivalente al sumarle 12, en nuestro rosco hacemos corresponder a cada número el equivalente al sumarle 27. De esta forma podemos sumar y restar cualquier número cayendo de nuevo en nuestro rosco. Esto es lo que se llama aritmética modular o, en este caso concreto, aritmética módulo 27, y se aprende en el primer curso del grado en Matemáticas. Una vez que hemos hecho corresponder cada letra con un número la posición que ocupa y podemos hacer operaciones aritméticas, romper el criptosistema se basará en la capacidad que tengamos de hacer operaciones.

A día de hoy la capacidad de cálculo de un ordenador hace absolutamente necesario diseñar

[2]Obviamente, no contamos trasladar 0 posiciones, porque el mensaje no está en claro

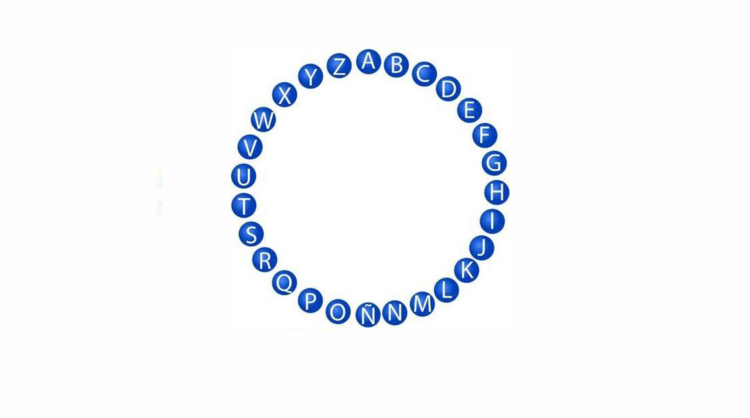

Figura 2: Rosco

criptosistemas con un conjunto de claves secretas de tamaño gigantesco, de forma que haga inviable la ruptura del mensaje por fuerza bruta. Empieza pues la carrera, y nuestro primer cometido para conseguir un criptosistema seguro es agrandar el conjunto de claves.

El Criptosistema del César en realidad consiste en escoger una permutación de las letras del alfabeto, (cada letra por la trasladada correspondiente), para impedir la lectura del mensaje, a no ser que se aplique la permutación inversa. Las permutaciones del César son simplemente sumar para cifrar y restar, su inversa, para descifrar. Pero hay muchas otras permutaciones y si pudiésemos utilizar cualquier permutación como clave secreta, habríamos agrandado el espacio de claves de manera considerable. Concretamente, tal y como se aprende en secundaria, tenemos 27! permutaciones posibles de 27 caracteres. Multiplicando los 27 primeros enteros en orden creciente o decreciente queda

$$\begin{aligned} 27! &= 1 \times 2 \times 3 \times \cdots \times 14 \times \cdots \times 25 \times 26 \times 27 = \prod_{i=1}^{27} i \\ 27! &= 27 \times 26 \times 25 \times \cdots \times 14 \times \cdots \times 3 \times 2 \times 1 = \prod_{i=1}^{27} (28-i) \end{aligned}$$

y multiplicando ambas expresiones tenemos

$$27!^2 = (1 \times 27) \times (2 \times 26) \times (3 \times 25) \times \cdots \times 14^2 \times \cdots \times (25 \times 3) \times (26 \times 2) \times (27 \times 1)\,.$$

Por simetría, basta multiplicar los primeros 14 primeros términos para obtener

$$27!^2 = 14^2 \times \left(\prod_{i=1}^{13} i(28-i) \right)^2$$

y teniendo en cuenta que $i(28-i) > 100$ para i entre 5 y 13, nos queda

$$\begin{aligned} 27!^2 &= (27 \times 52 \times 75 \times 96 \times 14)^2 \left(\prod_{i=5}^{13} (28-i)i \right)^2 \\ &> (141523200)^2 \prod_{i=5}^{13} 10^4 > 10^{52} \end{aligned}$$

con lo que $27! > 10^{26}$. Los superordenadores más potentes en la actualidad pueden calcular 1000 petaflops por segundo, es decir, unas 10^{18} operaciones por segundo, con lo que recorrer todo el conjunto de claves necesitaría 10^8 segundos, es decir unos 10 años de trabajo continuo. Definitivamente hemos agrandado el espacio de claves para evitar la fuerza bruta. Pero no es solo eso lo que necesitamos. El Criptosistema del César es super fácil de utilizar pues solo debemos sumar o restar un número menor que 27 al mensaje. La sencillez del cifrado, o descifrado si sabemos la clave, es algo que evidentemente debemos conseguir en el diseño de cualquier criptosistema. Por suerte para nosotros, implementar todo el conjunto de permutaciones es posible con un poco más de Matemáticas: a sumar y restar vamos a añadir multiplicar y dividir. En el segundo curso del GeM [4], aprenderemos que al conjunto de 27 caracteres con las operaciones de sumar, restar, multiplicar y dividir se le puede dotar de una estructura matemática llamada cuerpo, es decir, un lugar donde las operaciones elementales se pueden hacer sin salirnos del conjunto, y sin temor a equivocarnos. Además, este cuerpo es muy conveniente pues solamente tiene un número finito de elementos, 27.

En abstracto, convertir el mensaje en claro en uno cifrado, o al revés, lo podemos entender como una función sobre el cuerpo, cuyos elementos son las letras del alfabeto convertidas en números, compatible con las operaciones aritméticas elementales. Como alumnos de bachillerato quizás ya os imagináis que tales funciones no son otra cosa que los polinomios. Por ejemplo, el polinomio $x+3$ produce la traslación por 3 para encriptar el mensaje en el César y su inverso el $x-3$ sirve para desencriptar. Es importante darse cuenta de que el polinomio lo hemos construido gracias a que sabemos la clave secreta, es decir, el 3. Otros polinomios, llamados polinomios permutacionales, operan sobre otros cuerpos y permiten diseñar criptosistemas más generales. En concreto, nuestro criptosistema lo hemos diseñado sobre un cuerpo de tamaño 27, pero el César hablaba en latín y tenía un alfabeto diferente. Hoy en día podría ser en una de las aproximadamente 7000 lenguas vivas que existen en el mundo en la actualidad, con lo que necesitaremos diseñar criptosistemas sobre cuerpos de tamaño más general. Durante el grado veréis la fuerza de las Matemáticas que, en este caso, nos permitirá hacer dicha construcción de cuerpos con tamaño tan grande como queramos. Supongamos que queremos incluir en el abecedario las antiguas LL y la CH, con lo que ahora el cuerpo tiene tamaño 29, y lo denotaremos como $\mathbb{F}_{29} = \{0, 2, \ldots, 27, 28\}$ y como en el ejemplo anterior, haremos corresponder a cada letra un número de forma cíclica. Si ahora consideramos como clave secreta la permutación correspondiente al polinomio $p(x) = x^3 + 1$, el mensaje "En Mátes un Ocho", es decir, $5, 15, 14, 0, 21, 5, 20, 22, 15, 16, 13, 16$ se convertirá en el texto cifrado $10, 12, 19, 1, 11, 10, 26, 6, 12, 8, 23, 8$, que surge de elevar al cubo cada número, recordando que cada vez que nos salga un número mayor que 29, restaremos un múltiplo de 29, dando las vueltas necesarias al "rosco"de 29 casillas hasta llegar al número que corresponde entre 0 y 28.

Como verás el texto cifrado, a simple vista, parece completamente aleatorio y sin relación con el texto en claro, a no ser que sepas la clave secreta, es decir, x^3. ¿Sabrías encontrar el polinomio que serviría para desencriptar, es decir, que representa la permutación inversa? Como muy tarde lo sabrás cuando tomes el curso de Estructuras Algebraicas del GeM.

De nuevo las operaciones que debemos hacer no son más que sumas, restas, multiplicaciones y divisiones modulares, de muy fácil ejecución, con la ventaja de que esta vez tenemos un conjunto enorme de claves, o permutaciones, a escoger para compartir como clave secreta, y de flexibilidad tal que lo podremos diseñar sobre cualquier alfabeto.

Parece que tenemos bastante avanzado el problema de enviar mensajes cifrados: podemos diseñar un criptosistema con un tamaño de claves que impida el criptoanálisis por fuerza bruta, y que solo los participantes con la clave secreta pueden leer. Pero esto no es más que el principio. Para

que solo los participantes con la clave secreta puedan leer el mensaje, lo primero que deben hacer es compartir la clave, lo cual requiere enviar un mensaje de forma segura, por un canal público, y por tanto, inseguro. La primera idea que surge es enviar la clave por un canal seguro, al que no tienen acceso terceras personas. Por ejemplo, quedando emisor y receptor en un lugar privado y compartirla. Se dice fácil, pero según el contexto este sistema puede ser peligroso, muy costoso, o incluso inviable. Durante la segunda guerra mundial se debían enviar miles de libros de códigos entre los mandos y cada uno de los puestos al frente, naves o submarinos en plena acción de combate. Es interesante señalar que parte del trabajo efectuado por Bletchley Park para decodificar la máquina Enigma de la armada alemana, se hizo posible por la captura de libros de código del submarino U-559 el 30 de Octubre de 1942 en Port Said [2]. Este ejemplo manifiesta la dificultad que puede tener la distribución de la clave, y la enorme importancia de mantenerla en secreto. Pero no es más que un ejemplo. En la actualidad cada vez que escribes en tu navegador "https", se pone en marcha un protocolo SSL/TLS que permite enviar información encriptada entre un servidor y un cliente. Según la página worldometers [https://www.worldometers.info/] hoy, 14/9/2025, se han hecho más de 9529718723 consultas a google, es decir, más de 9 mil millones en un sólo día, entre los más de 6 mil millones de usuarios de internet, que se han enviado más de 2 billones de correos electrónicos... Sí, solo hoy. ¿Te imaginas quedar entre cada pareja de usuarios para intercambiarse las claves? Esto no es posible, salvo que lo hagamos por el mismo medio, es decir, por internet, un canal al que tiene acceso cualquier usuario, y por tanto inseguro. Para mandar la información por este canal, deberemos encriptar la información con protocolos como PGP, "pretty good privacy", para garantizar la privacidad del mensaje.

2.2. Criptografía de clave pública

¿Y cómo funciona PGP? La respuesta se encuentra en la Criptografía de clave pública. Veremos que, matemáticamente hablando, es más o menos sencilla de explicar, pero para la humanidad ha supuesto uno de los mayores avances de toda su historia. De hecho, como hemos dicho, la Criptografía es tan antigua como cualquier medio de comunicación de la humanidad, en concreto se tiene constancia de mensajes encriptados que tienen más de 4000 años de edad, y solo hace 50 años que hemos podido desarrollar la Criptografía de clave pública.

La forma de encriptar descrita hasta el momento mediante una clave simétrica, es decir, la misma para cifrar y descifrar, es compartir dicha clave secreta, y parece de sentido común, ¿no?: si un usuario cifra un mensaje con una clave secreta, el receptor tendrá que usar esa clave para descifrarlo. ¿Cómo puede ser de otra manera? La Criptografía de clave pública propone utilizar claves distintas: una para encriptar, pública para todo el mundo, con lo que evitamos compartirla, y una privada para desencriptar, que el receptor mantiene en absoluto secreto. Por ejemplo, mi clave pública va a ser mi dirección de correo electrónico, con lo que, como vosotros habéis hecho, cualquiera podrá enviarme un mensaje. Si lo hacéis mediante PGP, el mensaje se cifra automáticamente y sale de vuestro ordenador de forma ilegible. Al llegar a mi correo, el propio programa utiliza la clave secreta, solamente en mi poder, y descifra el mensaje.

El truco está en relacionar, de alguna forma secreta, ambas claves. Esta relación, y la seguridad de que permanece secreta la encontraremos de nuevo en las Matemáticas. La idea original para encriptar mensajes con claves de cifrado y descifrado distintas, o cifrado asimétrico como se conoce, la tienen de forma independiente Clifford Cocks en 1973 y Ron Rivest, Adi Shamir y Leonard Adleman en 1997. Curiosamente el criptosistema utilizado en la actualidad de forma masiva en todo el mundo, toma como nombre las iniciales de los segundos, a pesar de llegar a la idea más tarde. Esta injusticia histórica se basa en que Cook trabajaba para la inteligencia británica, y sus descubrimientos permanecieron como información confidencial hasta que se destaparon en 1997,

fecha en la que se reconoció la contribución original de Cook.


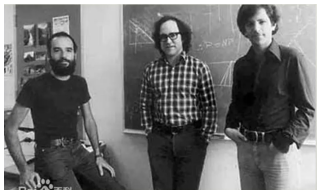

Figura 3: A la izquierda Clifford Cocks, (chalk dust magazine [6]), A la derecha Ron Rivest, Adi Shamir y Leonard Adleman, (NSF) [7].

El algoritmo RSA funciona de la siguiente forma: lo primero que tenemos que hacer es generar las claves, una pública y una privada. Para ello vamos a considerar dos primos p, q "grandes". Y vamos a decir que $n = pq$, y $\varphi = (p-1)(q-1)$. El siguiente paso es escoger dos enteros e, d coprimos con φ, y con la propiedad de que φ divide a $e \times d - 1$. En este caso diremos que $e \times d$ es 1 módulo φ. Ya tenemos las claves. La pública será la pareja (n, e), mientras que la privada será la tripleta (p, q, d). Para codificar el mensaje m, Daniel solamente tendrá que calcular $c = m^e$ módulo n, es decir, dando vueltas a un rosco de longitud n. Para descodificar, Sofía calculará c^d módulo n.

Por ejemplo, $p = 233$, $q = 257$, con lo que $n = pq = 59881$, y $\varphi = (p-1)(q-1) = 59392$. Escogemos $e = 3$ y $d = 39595$. Ahora vamos a ver cómo se manda un mensaje, por ejemplo, "ESO".

Un pequeño programa en Python como el siguiente

Cuadro 1: Código Python.

```
alfabeto=' ABCDEFGHIJKLMNOPQRSTUVWXYZ'
def codifica_letra(letra):
    return alfabeto.index(letra)
def codifica_cadena(texto):
    return [codifica_letra(c) for c in texto]
def decodifica_cadena(lista):
    return ''.join([decodifica_letra(j) for j in lista])
```

nos permite transformar las letras en números, y basta escribir en Python el texto

codifica_cadena('ESO')

para obtener la cadena de números equivalente

[5,19,15]

El mensaje que queremos mandar será el número $m = 51915$. Calculamos

$$c = m^e \pmod{n} = 51915^3 \pmod{59881} = 8482,$$

es decir, "HDHB", que resulta incomprensible. Sin embargo, si calculamos

$$c^d \pmod{n} = 8482^{39595} \pmod{59881} = 51915,$$

nos devuelve inmediatamente el mensaje original. [3]

[3] Si estas pensando en cómo recuperar el texto de los números, fíjate en que 51 no es ninguna letra, con lo que el mensaje empieza por "E". No puede ser "EAI...", luego la segunda letra es "S". Y no puede ser "ESAE", luego el mensaje es "ESO".

Así expuesto, parece un milagro que haya funcionado esta vez, y deberíamos probar que, efectivamente, el Criptosistema tiene sentido, es decir, al descifrar el mensaje encriptado, recuperamos el mensaje en claro, siempre. La razón está de nuevo en la Aritmética modular. Un curso introductorio en Teoría elemental de Números, o Aritmética, lo tendréis en el primer curso del grado y nos bastará para probar que el Criptosistema RSA funciona. Una de las claves para que funcione es la relación que le pedimos a los exponentes e, d que al multiplicar dé 1, es decir, que sean inversos el uno del otro. Para que esto sea posible necesitamos que nuestros números pertenezcan a una estructura algebraica llamada grupo.

Es posible que también parezca sospechoso que digamos que n es público, pero p y q son secretos. Este punto es crucial, y determinará el paradigma de la Criptografía de Clave pública. En nuestro ejemplo, si n es público, cualquier atacante puede simplemente factorizarlo y no tardará ni un milisegundo en conseguirlo con cualquier lenguaje, como Python, por ejemplo, o tardarás un poco más a mano, con lo que los factores p y q también serán publicos. Por otro lado, si

$$\begin{aligned} n = \quad & 13506641086599522334960321627880596993888147560566702752448514 \\ & 38515265106048595338339402871505719094417982072821644715513736 \\ & 80419703964191743046496589274256239341020864383202110372958725 \\ & 76235850964311056407350150818751067659462920556368552947521350 \\ & 08528794163773285339061097505443349998111500569772368909275 63, \end{aligned}$$

¿serías capaz de factorizarlo en menos de un milisegundo, aún sabiendo que es el producto de dos primos? En realidad, se estima que para factorizar el anterior número necesitaríamos más de 1 año de trabajo continuo de $1,5$ millones de ordenadores de 2,1Ghz de velocidad. Y si en realidad quieres tardar "solo.$^{\text{es}}$e tiempo, deberás utilizar el algoritmo clásico más rápido existente, denominado criba general de cuerpos de números, o GNFS, y tendrás que esperar hasta hacer el Máster en Matemáticas avanzadas, MUMAv, para tener el conocimiento necesario.

Como veis factorizar un número conocido aún siendo simplemente el producto de dos primos es, a día de hoy, un problema imposible de resolver en la práctica, y es por eso que, en la mayoría de los casos, la transmisión de la información está basando su seguridad en ese problema. Dicho de otro modo, si no puedes factorizar, no puedes leer el mensaje.

Es muy importante observar que, en realidad, el problema no consiste en que no sepamos factorizar, sino que más bien es un problema de tiempo. Ya en el colegio aprendemos a dividir sucesivamente por los primos anteriores al número para encontrar sus factores, pero si intentamos factorizar de esta forma el número n anterior jamás acabaremos. En este sentido, los mejores matemáticos del área, y desde hace 400 años, han dedicado innumerables esfuerzos en encontrar otra forma, otro algoritmo, para factorizar de manera más rápida. Y estos esfuerzos, como hemos mencionado, han tenido sus frutos hasta llegar a la GNFS que reduce el tiempo hasta menos de una milésima parte del algoritmo original... y aún siendo tan trascendental la mejora, no consigue resolver el problema, pues de nuevo gracias a las Matemáticas sabemos que hay infinitos números primos con lo que siempre podremos diseñar el protocolo con primos todavía más grandes, y asegurar la comunicación. Si te sorprende que los números primos tengan que ver con la seguridad de la transmisión de información, espera al tercer curso del GeM para aprender cómo la variable compleja interviene en la distribución de los primos gracias a la función zeta de Riemann. Nos queda sin embargo la difícil tarea de, entre los infinitos números primos, escoger dos que sean seguros. Este es parte de un proceso denominado generación de claves, que no es trivial en absoluto. De

hecho, una mala selección de primos puede desencadenar que la factorización se pueda completar, y así romper el criptosistema. Este fue el caso, por ejemplo, del gobierno de Estonia tras una mala selección de primos en las claves para el diseño de sus tarjetas de identidad. En 2017 se hizo público un ataque consiguiendo que más de la mitad de la población quedara expuesta tanto en su autenticidad, como en su capacidad de firma digital.

Así que la seguridad de nuestras comunicaciones, a cualquier nivel en el mundo, depende del tiempo que tarda un algoritmo en resolver un problema. No es de extrañar entonces que otra de las áreas de investigación más activas sea el diseño de nuevos algoritmos para factorizar, así como dar una estimación del tiempo de ejecución que necesitan, para lo que se necesita ser un experto en complejidad y algoritmia, parte de la cual veréis en Matemática Discreta en el segundo curso del GeM.

De entre los nuevos algoritmos que se han ido estudiando, el más sorprendente, y además potencialmente efectivo, es utilizar computación cuántica. En teoría, según un trabajo reciente, se podría factorizar un entero producto de dos primos de 2048 bits, es decir el doble de largo que nuestro n anterior, en tan solo una semana de trabajo con un ordenador cuántico de 1000 de qbits, la unidad de trabajo de la computación cuántica. El artículo todavía se está revisando para asegurarse de que su contenido es correcto, pero hace que la computación cuántica sea una forma de calcular exponencialmente más rápida, y ya muy cercana a nuestra realidad. Y es que se cree que para 2030 podremos tener ordenadores cuánticos de esas características, aunque también es cierto que no estarán al alcance de cualquiera por su elevado coste, por encima de los millones de euros. Si os interesa este tema, debéis saber que este tipo de algoritmos cuánticos los aprenderéis en la asignatura optativa de "Introducción a la codificación de la información.[en] el cuarto curso del GeM, y en cursos del Máster.

En vista del rapidísimo progreso de la computación cuántica, apoyada por las mayores empresas de comunicación, basar la seguridad de la comunicación en la factorización, pronto puede dejar de ser una buena idea, Y muy pronto deberemos sustituir dicho problema, por otro que sea resistente a esta técnica.

Para poder realizar las operaciones de cifrado y descifrado en el RSA: multiplicar, o elevar a un exponente, dividir, encontrar un inverso, etc. debemos asegurarnos que estamos dentro de una estructura algebraica que lo permita. Así pues, los esfuerzos por sustituir el problema de la factorización por otro más resistente, pasan por encontrar otras estructuras matemáticas, a mayor nivel, que nos den con el entorno adecuado. En el año 2016 el NIST, (National Institute of Standards and Technology), lanzó un concurso entre los expertos de más alto nivel para conseguir diferentes opciones para sustituir el RSA en un escenario en el que la computación cuántica fuese real. De entre todas las propuestas se están estudiando dos, como posibles opciones sólidas, basadas en estructuras matemáticas diferentes. Por un lado, los retículos, y por otro las curvas elípticas. La primera es más o menos simple de entender y tiene su soporte en el Álgebra lineal, de primero del GeM. Es quizás esta sencillez la que ayuda a poder diseñar criptosistemas con más comodidad. De alguna forma, la idea es considerar los mensajes como una colección de números, y no solo uno como en el RSA, y operar en estas colecciones de números, y la estructura que poseen.

Detrás de las operaciones en estas colecciones de números, hay problemas difíciles de resolver, análogos al problema de la factorización. Un ejemplo sencillo de este tipo de problemas es el conocido como "problema de la mochila": supongamos que tenemos 10 pesas de diferentes pesos $1, 2, 3, 5, 7, 11, 13, 17, 19, 23$, (tu colección de números), y alguien al azar mete algunas de ellas en una

mochila. Pesamos la mochila y nos pesa 46. ¿Puedes decirme cuáles han sido las pesas escogidas?

Mientras que es fácil deducir el peso de la mochila, sabiendo las pesas utilizadas, deducir las pesas utilizadas sabiendo el peso de la mochila puede ser un problema extremadamente difícil. En realidad se sabe que con un número suficientemente grande de pesas, con pesos bien escogidos, este problema será imposible de resolver en un tiempo razonable. Dentro de la asignatura de Matemática Discreta también aprenderéis a definir la dificultad relativa de un problema en referencia al tiempo necesario para su resolución, al hablar de las clases P y NP. En este contexto está el famoso problema P versus NP, tan difícil, que forma parte de los problemas del milenio [5], y el Instituto de Matemáticas Clay te dará 1 millón de dólares si lo resuelves.

La otra estructura en estudio para el diseño de Criptosistemas seguros contra la computación cuántica la encontramos en las curvas elípticas. Así como el rosco, o círculo, ha sido definitivo en el diseño de Criptosistemas en la mayor parte de nuestra existencia, la curva elíptica es otra curva que nos proporciona mayor seguridad, a coste de un conocimiento matemático extraordinariamente superior. La gracia es conseguir la estructura de grupo, con la que podemos realizar las operaciones para encriptar y desencriptar, pero en una colección de curvas que, a pesar de ser de una complejidad mucho mayor que el círculo, nos dan una familia muy extensa, con lo que tendremos mayor flexibilidad a la hora de esconder nuestro secreto. Entender el comportamiento de las curvas elípticas requerirá cursos del GeM de Geometría, Estructuras Algebraicas, e incluso Análisis, a nivel de grado y postgrado, para entender la aritmética dentro de estas curvas y desarrollar criptosistemas seguros. Su estudio es una de las principales áreas de investigación en Teoría de Números, y de nuevo proporcionan uno de los problemas del Milenio [4], valorado en 1 millón de dólares por el Instituto Clay.

3. Decidiendo

Por fín acabo la secundaria y al terminar el curso Daniel manda un mensaje en el grupo Mates

Daniel: ¡Hola! ¿Quedamos el finde? ❤️ 2

Eva: 👏👏👏👏

Hugo: Hombre, ¡que sorpresa tu hablando en el grupo! A mi me va bien el Sábado 👏👍 3

Daniel: ¡¡¡Siiii!!! 👏🎉💃🕺🤪

Y llega el sábado...

Daniel: Chicos, se lo he dicho a mis padres. ¡¡¡Y me dejan ir a la uni a empezar Mates!!!

Sofía: ¡Qué alegría Daniel! ¡¡Yo ya te veo como el próximo Alan Turing!!

Agradecimientos: Le agradezco a Maria Jesús Vazquez la lectura de una versión previa, mejorando la presentación final, a Alicia Cantón por su sugerencia de incluir emoticones en la conversación, y a Victoria Gómez por su diseño.

Bibliografía

[1] Chatgpt, https://chatgpt.com/

[4] ¿Sabrías cuál de ellos es?

[2] Manuel J. Prieto, Historia de la Criptografía, La esfera de los libros, ISBN: 978-84-9164-737-9.

[3] Joan Gómez Urgellés, Matemáticas, espías y piratas informáticos. Codificación y criptografía National Geographic, Editorial RBA, Barcelona, 2014.

[4] Grado en Matemáticas de la UPM, https://matematicas.epes.upm.es/el-grado/plan-de-estudios/

[5] The millenium problems, https://www.claymath.org/millennium-problems/

[6] Foto de Clifford Cocks, https://chalkdustmagazine.com/interviews/clifford-cocks/

[7] Foto de Rivest, Shamir y Adleman de la NSF, https://mathshistory.st-andrews.ac.uk/Biographies/Adleman/pictdisplay/

[8] Foto de Helge Fykse, https://la6nca.net/bilder/enigma/

[9] Fotos del cryptomuseum, https://www.cryptomuseum.com/crypto/enigma/photo.htm