Informe Criptografía
Algoritmo RSA

Matemática Discreta

Diego Fernández Gutiérrez

Donostia, Octubre 2021

Índice general

Capítulo 1

Criptografía RSA

1.1. Introducción

Lo que definimos como criptografía, es una manera de proteger ciertos mensajes o escritos que queremos que se guarden en secreto para que solo lo pueda interpretar el receptor al que se lo estemos enviando, con el fin de que nadie pueda leer nuestro mensaje secreto y de proteger nuestra información de desconocidos.

Pero, ¿todo esto como empezó y a quién se le ocurrió?, para empezar esto tiene lugar desde hace más tiempo del que imaginamos, empezó a usarse en la antigua Roma con el código César, en honor a Julio César, en el que se introdujo la criptografía la cual utiliza el algoritmo ROT-13, con el fin de comunicarse con países aliados para que sus mensajes no sean interceptados por sus enemigos. Pero mucho más adelante, en mitad de la Segunda Guerra Mundial, cuando los alemanes empezaron a utilizar las Máquinas Enigma, una máquina con teclas para escribir cualquier mensaje encriptado, con la forma de una máquina de escribir común. Su funcionamiento consistía en escribir un mensaje normal con las teclas, pero cada vez que se pulsa una, la posición de los rotores cambiaba, así saldría un mensaje totalmente diferente al que se está escribiendo. De este modo, para conseguir el mensaje original se deberá de descifrar. Para ello, cuenta con un sistema simétrico, es decir, que el sistema de descifrado es análogo al de cifrado, luego para conseguir el mensaje original bastaba con escribir el mensaje que había impreso la máquina y ésta devolvía una a una las letras del mensaje original. Este mecanismo era utilizado en especial para transmitir mensajes secretos a las espaldas de los americanos, con el objetivo de comunicarse con países aliados con el fin de que los demás países no interceptasen el mensaje. Utilizaban mensajes cifrados y así lo interpretarían los receptores a los que les llegará el mensaje. Pero, ¿esto realmente era un truco infalible para la comunicación secreta?, la respuesta es no, ya que aún con las complejísimas máquinas de cifrado que existían en esa época y con claves tremendamente difíciles de descifrar, aun así era posible interceptar esos mensajes, todo fue obra del matemático Alan Turing, quien fue capaz de descifrar los mensajes encriptados, salvando así millones de vidas.

El concepto de la criptografía es sencillo, consiste en mandar un mensaje secreto a una

persona sin que nadie lo pueda leer, es decir, un mensaje que solo puedan entender ellos dos. Para ello, se deberá de codificar dicho mensaje, para más tarde cifrarlo, y así la otra persona se encargará de descifrarlo y decodificarlo, de esta manera, nadie más podrá leer ese mensaje.

En el mundo existen muchos tipos de criptografía diferentes con su respectivo algoritmo, el cual se encargará de hacer todo el proceso siguiendo un protocolo. Ejemplos de algunos de estos pueden ser, la Clave César, la cual consta de un algoritmo ROT-13 y es simétrica, la Criptografía RSA, la cual cuenta con una clave pública y una clave privada, convirtiéndola en una clave asimétrica. ¿Pero qué quiere decir que una clave sea simétrica o asimétrica? una clave simétrica es un método que utiliza una clave y transforma un mensaje a otro totalmente diferente y sin sentido, y solamente los usuarios que saben esta clave pueden descifrar el mensaje oculto. En cambio, una clave asimétrica consta de dos claves en vez de una, la clave pública y la privada, y su protocolo es encriptar los mensajes con una clave pública y recuperarse con una privada, de la misma forma que los mensajes encriptados con una clave privada se recuperarían con una pública. La clave privada, es una clave que no debería saber nadie salvo el usuario del mensaje y será el método de descifrar los mensajes. Al contrario, la clave pública, puede ser enviada a cualquiera para así mandar un mensaje encriptado al usuario.

Hoy en día, la criptografía se usa mucho más de lo que imaginamos, por ejemplo, es utilizada bastante en el sistema de las tarjetas convencionales, llamadas así mimas, Tarjetas Criptográficas, que constan de tener un sistema de criptografía RSA de hasta 3840 bits. Este sistema lo utilizan en especial los DNIs para la identificación instantánea y las tarjetas de crédito para la base de datos de los bancos y así tener un sistema protegido y de difícil acceso. Con este sistema, se puede poner en práctica la firma digital, la cual, consta del mismo sistema de criptografía (concretamente, un sistema asimétrico que utiliza el algoritmo RSA), lo interesante de este sistema es que es casi imposible de falsificar, ya que ésta es equivalente a la firma física de un documento, no sería posible cambiar absolutamente nada, y con esa firma digital, nadie que haya operado con ella puede negar cualquier acción que haya hecho con ella, a no ser que el sistema haya caído y alguien se haya apropiado de ella (aunque no suele ser el caso, ya que cuenta con un sistema de ciberseguridad avanzado).

Otro ejemplo de la criptografía en la sociedad actual podemos encontrarlo en las apps de comunicación más famosas, como WhatsApp, Twitter, Instagram...etc. Estas son aplicaciones de mensajería en el que se envía información de un sitio a otro, en lo que se sabe que todos los mensajes que se envían están cifrados, pero, ¿qué clase de algoritmo utiliza?, pues se conoce que por ejemplo, WhatsApp genera una pareja de claves públicas y privadas (*Identity Key Pair*), junto con un algoritmo de Criptografía de Curvas Elípticas, para después generar una segunda pareja de claves junto con la clave privada sacada anteriormente(*Signed Pre Key*), y por último genera un lote parejas que se irán renovando a medida que se agoten(*One-Time Pre Keys*). En resumen, la información que

compartes son las claves públicas del conjunto de parejas que se generan, aunque la clave privada siempre se guardará en tu dispositivo y nadie, ni siquiera WhatsApp, tiene acceso a ella. Para ver todo esto más claro, vamos a ver el proceso de enviar un mensaje a alguien por WhatsApp, para empezar el servidor nos envía una clave pública *Identity Key Pair,Signed Pre Key* y al otro usuario *One-Time Pre Keys*, y con estas claves acuerda con el otro usuario un mensaje con estas claves públicas, para ello utiliza el protocolo de claves ECDH(Elliptic-Curve Diffie–Hellman), esta clave utilizará un algoritmo HKDF (HMAC Key Derivation Function) para así crear una Root Key para cifrar los mensajes intercambiados con una clave secreta, y así sucesivamente con todos los mensajes que queramos enviar.

Cambiando un poco de perspectiva, el sistema de criptografía es bastante usado en gran de cantidad de películas y series de televisión como la de "Mr. Robot", en la que en un episodio se encuentra un mensaje cifrado y tienen que tratar de descifrarlo, en ese caso se trataba de un mensaje con una rotación de valor 13(algoritmo ROT-13), por lo que se necesitarían los números perrin para sacar ese mensaje secreto.

Por último, otro fenómeno traído de la ciencia y de la física cuántica que relaciona a la criptografía se hace llamar Criptografía Cuántica, el cual no es más que otro método de encriptar mensajes pero más eficiente y más seguro que los anteriores mencionados, para entender como funciona este nuevo protocolo lo explicaré con un ejemplo.

Para empezar pongámonos en la situación de que dos personas quieren mandarse mensajes y no quieren que nadie les descubra, ellos han entendido que los métodos de encriptación César y demás no son lo suficientemente seguros o simplemente porque tienen un coste computacional muy alto y quieren probar algo nuevo, por lo que usarán la Criptografía Cuántica. Pero para ello deberán de usar el protocolo de encriptación cuántica BB84 (inventado por Charles Bennett y Gilles Brassard en 1984) en el que genera una clave en código binario que servirá para encriptar los mensajes.

Por ejemplo, el emisor deberá de escribir una serie muy larga de bits para más tarde escribirlo en una cadena de *Qubits*, pero para ello deberá elegir una cosa, el eje del *Qubit*, es decir, deberá de elegir si codificarlo en el eje X o el eje Z del *Qubit*, si éste elige el eje X deberá de traducir los bits 0 preparándolos como Estados Derecha y los bits 1, Estados Izquierda. Pero si elige el eje Z los bits 0 y 1, se llamarán así mismos 0 y 1, formando así una cadena de *Qubits* con cuatro posibles estados. Con esta información deberá de apuntar los estados especificando el eje y su respectivo valor asignado, con esto ya es posible enviarle a la otra persona los *Qubits*, este receptor deberá de elegir aleatoriamente con qué eje medir cada *Qubit*. Pero esto tiene un riesgo, ya que el receptor no sabe con qué eje ha medido el emisor el mensaje, por lo que tiene un 50 % de probabilidades de acertar con el código, es decir, que si el receptor mide el *Qubit* con el eje Z y el emisor ha hecho lo mismo, lo han hecho bien y le ha llegado correctamente, pero si hace lo contrario

y lo mide con el eje X, le dará un valor (0 ó 1) aleatoriamente, por lo que tiene un 50 % de probabilidades de acertar.

Dejando esto de lado, el receptor deberá medir todos los *Qubits* y apuntar el resultado que ha obtenido en cada uno de ellos independientemente de como lo había preparado el emisor. Una vez terminado, el receptor le envía al emisor los resultados que ha obtenido, pero solo el eje con los que lo ha medido. Inmediatamente el emisor hace lo mismo, ahora con todo esto, los dos proceden a comparar los ejes y tachan los que no han sido iguales y se quedan con los comunes. Una vez terminado este proceso y habiéndose desecho de los *Qubits* diferentes, el receptor se dispone a enviar el valor de los primeros *Qubits* para compararlos con el emisor, en el mejor de los casos, si todos los valores que le han salido al receptor son iguales, significaría que el código se ha enviado correctamente, pero ¿y si hay algún valor diferente?, podría significar que el receptor no ha medido con los mismos ejes, o simplemente que hay un intruso tratando de robar el mensaje.

Pero de esta forma, el intruso no lo tendría fácil para descifrarlo, puesto que con este protocolo es muy fácil ser atrapado, ya que el intruso interceptará los *Qubits* para medirlos él antes y devolvérselo al receptor, de alguna forma, el valor se vería comprometido, es decir, que el emisor envía un *Qubit* al receptor, pero el intruso lo atrapa antes, éste deberá de hacer lo mismo que el receptor: escoger con qué eje medir el *Qubit*. Si acierta con la medida original, el valor no se vería afectado y pasaría al receptor sin levantar sospechas, pero ¿y si se equivoca de eje y le da un valor aleatorio (0 ó 1) y no es el correcto?, al receptor le llegará un valor diferente al que el emisor había preparado, lo que daría que pensar y cabría la posibilidad de que le pillen. En verdad, con este protocolo es bastante fácil ver cuando hay alguien intentando interceptar los mensajes, porque con pocos *Qubits* puede que no tanto, pero para un mensaje largo en el que habría que enviar y medir cientos de *Qubits*, el intruso deberá de tratar de medir todos y cada uno de ellos, y si al final se equivoca con uno o dos, el emisor se daría cuenta y cancelaría la conexión.

Pasando otra vez al proceso inicial, una vez conseguida una sucesión de bits correspondientes al mensaje original, estos cogerán el resto de valores que no han hecho público y lo entenderán como una clave, luego pasarán esos números de binario a decimal, con estos números podrán identificar el mensaje con una *Libreta de un solo uso* que solo ellos dos tienen, asimismo el emisor solo deberá desplazar cada letra de su mensaje como dice la página y el receptor al recibir el mensaje deberá hacer lo mismo pero a la inversa.

1.2. Algoritmo RSA

El algoritmo RSA, inventado por Rivest, Shamir y Adleman en 1978, es una forma de cifrar y descifrar mensajes mediante la factorización de números enteros, esto utiliza una clave pública y una privada, las cuales se utilizan de la siguiente manera, es necesario conseguir la clave privada para descifrar el mensaje obtenido mediante la clave pública, es

decir, tú recibes una clave pública cualquiera, y mediante una clave privada que deberías de tener podrías descifrar el mensaje oculto, eso si, también existe la posibilidad de que puedan conseguir la clave privada mediante trucos externos. Todo esto tienen sus propias funciones que más adelante se verán implementadas, ya sean, las de cifrar y descifrar o codificar y decodificar para recibir el mensaje.

1.3. Código en R de las funciones

Para hacer uso de la Criptografía RSA en nuestros ordenadores, haremos uso de ciertas funciones escritas en el lenguaje R.
Para empezar debemos empezar haciendo uso de la función *codifica*, para codificar el mensaje que queremos enviar.

```
codifica <- function(txt) {
codetxt<-strtoi(charToRaw(txt),16L)
return(codetxt)
}
```

De esta manera, dado un mensaje en string, devolverá un vector de números ASCII.
Por ejemplo, si queremos codificar la palabra "matematicas"deberemos implementar:

```
texto1<-"matematicas"
codevector1<-codifica(texto1)
codevector1
```

Y nos devolverá:

```
[1] 109  97 116 101 109  97 116 105  99  97 115
```

Con esto tendríamos el mensaje codificado, pero ahora deberemos cifrarlo para que nadie pueda acceder a él, para ello debemos sacar la Clave Pública y la Clave Privada. Las funciones para sacar estas claves son:

```
primo_relativo_pequeno <- function(m) {
zbki<-2;
while(GCD(m,zbki)!=1)
{
zbki<-zbki+1;
};
return(zbki);
}
```

```
claves_RSA <- function(p,q) {
n<-p*q;
m=(p-1)*(q-1);
r<- primo_relativo_pequeno(m)
s<-modinv(r,m)
cat("Clave publica. n=", n, "r=", r,"Clave privada, s=", s, "\n\n")
return(c(n,r,s))
}
```

Habiendo sacado nuestras claves, ahora faltaría cifrar el mensaje con la siguiente función, de esta manera nadie podrá acceder a nuestro mensaje sin nuestra Clave Privada,

```
cifrar <- function(codevector,r,n) {
vcifrado<-codevector
for (i in 1:length(codevector)) {
vcifrado[i]<-modpower(codevector[i],r,n)
}
return(vcifrado)
}
```

Con esto ya podríamos empezar a mandar mensajes secretos entre nosotros sin que nadie pueda identificarlos, pero ¿como sería si quisiéramos descifrar mensajes?
Para esto existen unas funciones muy parecidas que hacen lo mismo pero a la inversa,

```
descifra <- function(vectorcifrado,s,n) {
vcifrado<-vectorcifrado
for (i in 1:length(vectorcifrado)) {
vcifrado[i]<-modpower(vectorcifrado[i],s,n)
}
return(vcifrado)
}
```

Primero desciframos el mensaje, y después pasamos a decodificarlo con la siguiente función:

```
decodifica <- function(codetxt) {
code<-rawToChar(as.raw(codetxt))
return(code)
}
```

Con esto nos aparecerá el mensaje que había sido encriptado.

1.4. Opinión Personal

Este trabajo escrito dedicado a la criptografía, un campo que siempre me había entrado especial curiosidad e interés. Me ha parecido una buena experiencia y también una buena introducción a la carrera de Ingeniería Informática. Ya que empezando con los laboratorios prácticos, el ejercicio de salir a la pizarra a encriptar nuestros mensajes, apuntarlos en la pizarra, y descifrar los mensajes de nuestros compañeros me ha parecido una muy buena idea como introducción a todo este campo. Y admito que después de entender como funciona todo esto, me ha entrado mucho más interés en la asignatura.
Esta es la segunda vez que la curso, y tengo que decir que este segundo año me ha parecido mucho más interesante, ya que cuando vuelves a repetir lo mismo del año pasado y encima entendiendo todo lo que se explica, me hace querer aprender más acerca de la asignatura. Por ello, la búsqueda de información y la utilización de la criptografía en la sociedad actual me ha parecido una tarea muy interesante en la que he aprendido mucho y me ha hecho tener más interés sobre este tema.
Respecto al trabajo en si, la utilización del lenguaje de texto LaTeXno me ha resultado un problema, ya que estaba muy familiarizado con este, lo único que no había utilizado hasta ahora era la función "verbatim", ya que nunca lo había implementado para escribir lineas de código de otro lenguaje. Y también, nunca había escrito una bibliografía, de hecho he tenido algunos problemas a la hora de implementarla, ya que no me compilaba el proyecto por culpa de esta, por lo que he tenido que escribirla manualmente.
En conclusión, me ha gustado mucho hacer este trabajo escrito y espero que me sirva como una buena introducción a la carrera que estoy cursando y que me acuerde de lo interesante que puede llegar a ser.

Bibliografía

[1] *N. Yanofsky, M. Mannucci : Quantum Computing for Computer Scientists, Volume 9(2008)*

[2] *P. Angulo : Apuntes de Matematica Discreta, (2005)*

[3] *M. Mitani, S.Sato, I.Hinoki, V.Corp : The Manga Guide To Cryptography, (2018)*

[4] *https : //www.cert.fnmt.es/catalogo − de − servicios/tarjetas − criptograficas*

[5] *https : //www.cert.fnmt.es/content/pages*$_s$*td/html/tutoriales/tuto7.htm*

[6] *https : //empresas.blogthinkbig.com/si − whatsapp − cifra−*
comunicaciones − donde − esta − la − clave/

[7] *https : //tech − es.netlify.app/articles/es531324/index.html*

[8] *https : //www.youtube.com/watch?v = Q8K311s7EiM*

[9] *Mr.Robot, 2 Temporada, Episodio 11*

[10] *W. Dr, S. Heusler : What Can Learn about Quantum Physics from a Single Qubit*

[11] *https : //www.analyticslane.com/2018/09/07/diferencias − entre − cifrado − simetrico*
−y − asimetrico/