CRISTIAN RIVERA ZAPATA

ALCALA DE HENARES - MADRID

SEGURIDAD INFORMÁTICA

JOSE ANTONIO / LAS NAVES SALESIANOS/ GRADO MEDIO 2ºB

Actividad voluntaria Examen 1a Evaluacion

ROMPER EL ALGORITMO DE CIFRADO RSA

INDICE

[¿QUE ES? 2](#_Toc88156340)

[GENERACIÓN DE CLAVES 3](#_Toc88156341)

[IDEA DEL ALGORITMO 4](#_Toc88156342)

[ESQUEMAS 4](#_Toc88156343)

[SEGURIDAD 5](#_Toc88156344)

[CONSIDERACIONES PRÁCTICAS 5](#_Toc88156345)

[METODO PARA ROMPER EL ALGORTIMO 6](#_Toc88156346)

[CONSECUENCIAS DE UNA ROTURA DEL ALGORITMO RSA 7](#_Toc88156347)

[FACTORIZAR NÚMEROS GRANDES 7](#_Toc88156348)

[ARREGLAR LAS DEBILIDADES DE RSA 7](#_Toc88156349)

[IMAGINAR UN NUEVO CRIPTOSISTEMA QUE NO SEA RSA 7](#_Toc88156350)

[CONCLUSIÓN PERSONAL RSA 9](#_Toc88156351)

WEBGRAFIA

[Figure 1 2](#_Toc88156358)

[Figure 2 3](#_Toc88156359)

[Figure 3 4](#_Toc88156360)

[Figure 4 6](#_Toc88156361)

[Figure 5 8](#_Toc88156362)

¿QUE ES?

El algoritmo fue descrito en 1977 por Ron Rivest, Adi Shamir y Leonard Adleman, del Instituto Tecnológico de Massachusetts (MIT); las letras RSA son las iniciales de sus apellidos. Clifford Cocks, un matemático británico que trabajaba para la agencia de inteligencia británica GCHQ, había descrito un sistema equivalente en un documento interno en 1973. Debido al elevado coste de las computadoras necesarias para implementarlo en la época su idea no trascendió. Su descubrimiento, sin embargo, no fue revelado hasta 1997 ya que era confidencial, por lo que Rivest, Shamir y Adleman desarrollaron RSA de forma independiente.

El algoritmo fue patentado por el MIT en 1983 en Estados Unidos con el número 4.405.829. Esta patente expiró el 21 de septiembre de 2000. Como el algoritmo fue publicado antes de patentar la aplicación, esto impidió que se pudiera patentar en otros lugares del mundo. Dado que Cocks trabajó en un organismo gubernamental, una patente en Estados Unidos tampoco hubiera sido posible.

Es un algoritmo puramente asimétrico, junto con DSA. Este algoritmo como su nombre lo indica, sirve para firmar (autenticar) y para cifrar información. Una desventaja del algoritmo DSA es que requiere mucho más tiempo de cómputo que RSA.

El sistema RSA es la primera propuesta práctica del método criptográfico de clave pública propuesto en 1976 por Whitﬁeld Diﬃe y Martin Hellman. En 1991 la RSA Security, empresa al cargo de la seguridad de RSA con sede en Massachusetts, propuso varios números semiprimos para que sea inviable hallarlos mediante ordenadores convencionales. Los métodos más eficientes de factorización de números generales que se conocen son la criba en cuerpos de números y las curvas elípticas. Sin embargo, no hay absoluta certeza de que no existan métodos eficientes de factorización, ya sea mediante un nuevo método o una nueva herramienta.

El 26 de abril de 1994 un equipo de 600 voluntarios coordinados por Atkins, Graﬀ, Lenstra y Leyland consiguió romper RSA-129, ganando los 100 $ prometidos y donándolos a la Free Software Foundation. ​No era la primera vez que se rompía RSA. Es lo que se conoció por RSA Challenge. En palabras del laboratorio, «Ahora que la industria ha alcanzado una comprensión considerablemente mayor de la seguridad criptoanalítica de los algoritmos simétricos comunes y los de clave pública, estos retos dejan de estar activo



Figure 1

RSA, al ser un cifrador asimétrico, trabaja con dos claves, una pública y una privada. Todo el contenido de texto plano, o contenido sin cifrar, que se haya encriptado con la clave pública, podrá ser descifrado mediante la clave privada, y viceversa, todo contenido cifrado con la clave privada podrá ser descifrado mediante la clave pública.

GENERACIÓN DE CLAVES

El algoritmo de cifra asimétrico RSA permite tanto cifrar la información (proporcionando confidencialidad), como firmarla (proporcionando autenticidad e integridad).

Como con todos los cifrados, es importante cómo se distribuyan las claves públicas del RSA. Supongamos Eve tiene alguna forma de dar a Bob arbitrariamente claves y hacerle creer que provienen de Alicia. Eve envía a Bob su propia clave pública, como Bob cree que es de Alicia, Eve puede entonces interceptar cualquier texto cifrado enviado por Bob, descifrarlo con su propia clave secreta, guardar una copia del mensaje, cifrar el mensaje con la clave pública de Alicia, y enviar el nuevo texto cifrado a Alicia.

Las claves pública y privada se calculan a partir de un número que se obtiene como producto de dos primos grandes.

El uso más común de una llave RSA es para autenticarse por SSH en servidores web o de aplicaciones. Para esto, se debe utilizar el comando ssh especificando la llave privada a utilizar. Por su parte, el servidor al cual se desea conectar deberá tener registrada la parte pública de tu llave rsa. Esta parte pública es la que se debe subir a servicios como Digital Ocean, AWS, GitHub, Bitbucket u otros.

**SSH -I KEY\_FILE USER@SERVER.COM**

Si se especifica una passphrase al crear la llave rsa, al momento de autenticarse en un servidor se deberá digitarla nuevamente.

Cuando se crea una llave rsa por primera vez esta se intentará crear en ~/.ssh/id\_rsa tal y como se puede ver a continuación.

**07:09:46 FERMIUS@DEBIAN ~ → SSH-KEYGEN -T RSA**

**GENERATING PUBLIC/PRIVATE RSA KEY PAIR.**

**ENTER FILE IN WHICH TO SAVE THE KEY (/HOME/FERMIUS/.SSH/ID\_RSA):**

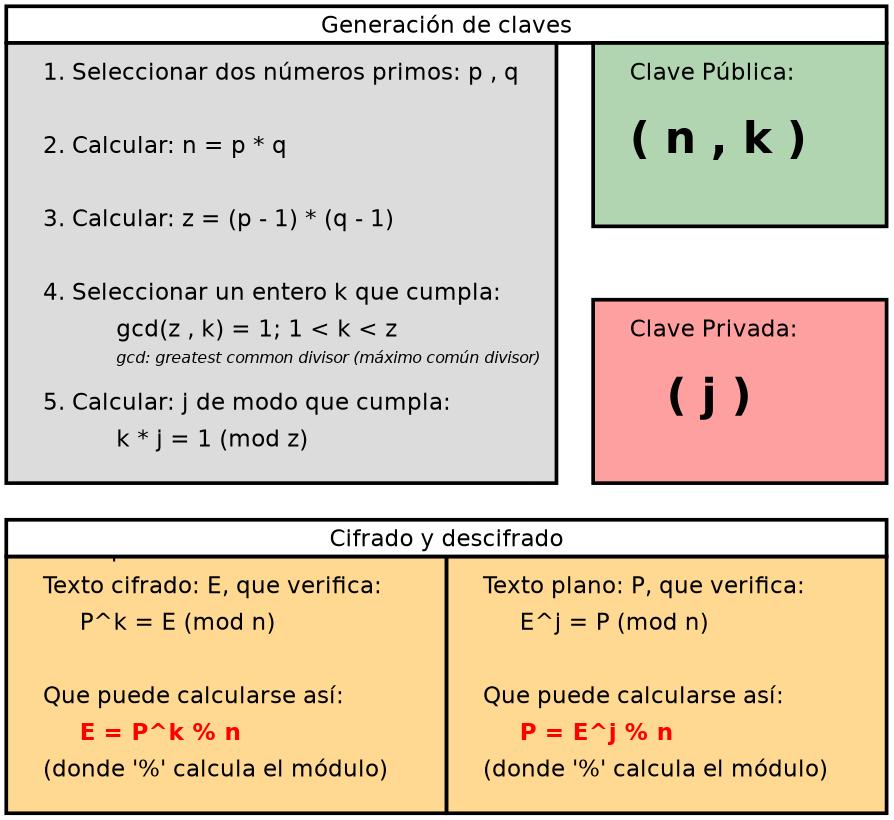
Si no se especifica ningún nombre de archivo diferente al id rsa esta llave se usará por defecto cuando no se especifique una llave. Es decir, se podría loguearse en un servido solamente con ejecutar el comando ssh así: **SSH USER@SERVER.COM**.

Figure 2

IDEA DEL ALGORITMO

En este ejemplo, la caja con el candado es la «clave pública» de Alicia, y la llave del candado es su «clave privada». A continuación, genera el «mensaje cifrado» es la clave pública de Alicia. Ahora Alicia descifra el mensaje en clave es la clave privada que solo Alicia conoce. La clave pública es, esto es, el módulo y el exponente de cifrado.

La clave privada es, esto es, el módulo y el exponente de descifrado, que debe mantenerse en secreto. Alicia comunica su clave pública a Bob y guarda la clave privada en secreto.

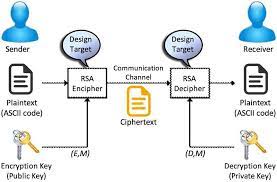


Figure 3

ESQUEMAS

Un mensaje consiste en un solo carácter ASCII NUL se codificaría como m=0, produciendo un texto cifrado de 0 sin importar qué valores de e y N son usados. Probablemente, un solo ASCII SOH produciría siempre un texto cifrado de 1. Para sistemas convencionales al usar valores pequeños de e, como 3, un solo carácter ASCII mensaje codificado usando este esquema sería inseguro, ya que el máximo valor de m sería 255, y 255³ es menor que cualquier módulo razonable. De esta manera los textos sin cifrar podrían ser recuperados simplemente tomando la raíz cúbica del texto cifrado.

Esta técnica asegura que m no caerá en el rango de textos sin cifrar inseguros, y que, dado un mensaje, una vez que este rellenado, cifrará uno de los números grandes de los posibles textos cifrados. El esquema de relleno de RSA debe ser cuidadosamente diseñado para prevenir ataques sofisticados los cuales podrían ser facilitados por la predictibilidad de la estructura del mensaje.

SEGURIDAD

El descifrado completo de un texto cifrado con RSA es computacionalmente intratable, no se ha encontrado un algoritmo eficiente todavía para ambos problemas. Proveyendo la seguridad contra el descifrado parcial podría requerir la adición de una seguridad padding scheme. Con la capacidad para recuperar factores primos, un atacante puede calcular el exponente secreto d desde una clave pública , entonces descifra c usando el procedimiento estándar. Las claves RSA son normalmente de entre 1024-2048 bits de longitud.

Por lo tanto, si n es suficientemente grande el algoritmo RSA es seguro. Un dispositivo hardware teórico llamado TWIRL descrito por Shamir y Tromer en el 2003 cuestionó a la seguridad de claves de 1024 bits. En 1993, Peter Shor publicó su algoritmo, mostrando que una computadora cuántica podría en principio mejorar la factorización en tiempo polinomial, mostrando RSA como un algoritmo obsoleto.

CONSIDERACIONES PRÁCTICAS

Buscando números primos grandes p y q por el test de aleatoriedad y realizando tests probabilísticos de primalidad los cuales eliminan virtualmente todos los no-primos. Los números p y q no deberían ser suficientemente cercanos para que la factorización de Fermat para n sea exitosa. En particular, se debe utilizar un buen generador aleatorio de números primos para el valor empleado. Es importante que la clave secreta d sea muy grande.

Aunque valores de e tan bajos como 3 se han usado en el pasado, los exponentes pequeños en RSA están actualmente en desuso, por razones que incluyen el no relleno del texto sin cifrar, vulnerabilidad listada antes. 65537 es normalmente usado como valor de e, considerado demasiado grande para evitar ataques de exponenciación pequeños, de hecho, tiene un peso de Haming suficiente para facilitar una exponenciación eficiente.

METODO PARA ROMPER EL ALGORTIMO

Un reconocido investigador alemán afirma que el algoritmo RSA fue descifrado por especialistas en análisis criptográfico, lo que puede representar un gran avance en el campo de las matemáticas o bien un riesgo de ciberseguridad. Las funciones unidireccionales que forman la base de la mayoría de los algoritmos dependen de la dificultad de resolver algunos problemas incluso con acceso a una computadora con altas capacidades de procesamiento. La seguridad de RSA, por ejemplo, se basa en la dificultad de factorizar el producto de dos números primos grandes. Como recordará, el sistema criptográfico RSA utiliza la factorización de números enteros y desde su lanzamiento se convirtió en uno de los recursos más utilizados para el cifrado digital.

Claus Schorr, matemático y criptógrafo, menciona que la factorización prima se puede reducir a un problema del vector más corto, que podría resultar más fácil de descifrar. La premisa de su investigación, titulada “Factorización rápida de enteros en algoritmos SVP”, es que este proceso podría destruir el cifrado RSA. Cabe recordar que incrementar la longitud de la clave de cifrado es una respuesta común para asegurarse de que los algoritmos se mantengan vigentes en el complejo mundo de la informática. Para conocer más sobre riesgos de seguridad informática, malware, vulnerabilidades y tecnologías de la información, no dude en ingresar al sitio web del Instituto Internacional de Seguridad Cibernética



Figure 4

CONSECUENCIAS DE UNA ROTURA DEL ALGORITMO RSA

«Si la afirmación resulta correcta, es posible que una buena parte de los datos cifrados en reposo o en movimiento no estén seguros. En el momento de escribir este artículo, los matemáticos todavía están deliberando sobre la primera pregunta, pero otros están abordando la segunda pregunta y comenzando a esbozar planes sobre qué hacer si aparece una debilidad catastrófica de la nada. Están esforzándose en encontrar una base más sólida, construida a partir de múltiples algoritmos implementados con protocolos que simplifican el cambio. Algunos criptógrafos buscan reemplazos de RSA porque el algoritmo es solo un algoritmo de cifrado que puede ser vulnerable a las nuevas máquinas que explotan los efectos cuánticos en la electrónica.

FACTORIZAR NÚMEROS GRANDES

«Depende de la complejidad de factorizar números grandes. El enfoque recientemente publicado de Schnorr, que ha evolucionado en una serie de artículos publicados en los últimos años, reformula el problema de factorizar números grandes como un problema para lo que los matemáticos a veces llaman la búsqueda del vector correcto en una red multidimensional definida por números mucho más pequeños. Varias personas han circulado desafíos que son grandes números construidos como claves RSA.

ARREGLAR LAS DEBILIDADES DE RSA

Una empresa, también llamada RSA, comparte bastante historia con el algoritmo, pero las patentes han expirado, y ahora la mayoría de las implementaciones de RSA utilizadas en Internet no provienen de ellos. Simplemente cambiar a un nuevo algoritmo puede no ser tan difícil porque muchos paquetes de cifrado soportan opciones para usar diferentes algoritmos con diferentes longitudes de clave. Un desafío más profundo puede ser actualizar la infraestructura de autenticación que mantiene la jerarquía de certificados utilizada para validar las claves públicas. La versión actual de los principales navegadores viene con certificados raíz de las diferentes autoridades de certificación y muchos dependen de RSA.

En el momento de redactar este documento, los certificados de algunas de las principales autoridades de certificación como Verisign, Amazon, GoDaddy y Entrust se basan en el algoritmo RSA.

IMAGINAR UN NUEVO CRIPTOSISTEMA QUE NO SEA RSA

Los que el anuncio del NIST llama los «algoritmos de propósito general más prometedores” usan entramados y dependen de la dificultad de encontrar un elemento o vector en esta red. Hay tres esquemas de cifrado diferentes y dos algoritmos de firma diferentes. Estas construcciones matemáticas se utilizan normalmente en el almacenamiento y la transmisión de datos para recuperarse de errores, pero McEliece encontró una manera de cambiar su construcción, por lo que recuperarse del error fue difícil para todos, excepto para las personas que tenían la clave secreta correcta. Muchos dependen de claves que son mucho más grandes y gran parte de la investigación activa se dedica a buscar variantes que utilicen claves más pequeñas.

No es raro oír hablar de claves que pueden tener 500 kilobytes o más, sustancialmente más grandes que muchas de las claves actuales que pueden tener solo varios cientos de bytes. Whitfield Diffie, un criptógrafo que ayudó a crear la criptografía de clave pública señala que las nuevas propuestas podrían requerir más computación para su configuración.» «Si los sistemas postcuánticos son más costosos computacionalmente que los actuales, es posible que tenga que dedicar un minuto a la computación para negociar una clave con eBay o Amazon, digamos, el 2 de enero, conservarla todo el año y realizar la autenticación de manera clásica”. Los protocolos actuales suelen negociar una nueva clave para cada interacción.»

Ejecutar una generación de claves computacionalmente más costosa con menos frecuencia puede resultar en el mismo costo total de computación, afirma Diffie, pero a costa de «disminuir la función de forward secrecy y disminuir el valor probatorio de las firmas”.» En la década de los años setenta, Martin Hellman, otro de los matemáticos originales que comenzó a explorar la criptografía de clave pública, sugiere que el mundo puede querer desarrollar nuevos protocolos que combinen varios algoritmos distintos. En lugar de depender simplemente de un algoritmo para crear la clave aleatoria destinada a cifrar algunos datos, el protocolo debe ejecutar varios algoritmos y combinar las claves de todos ellos en una clave. Hellman ha estado pensando en cómo sobrevivir al colapso de un algoritmo durante más de una década.

Pero incluso si las rupturas catastróficas no se producen, señala que incluso los incrementos más pequeños, construidos a partir de la evolución de los algoritmos para factorización, pueden sumarse.

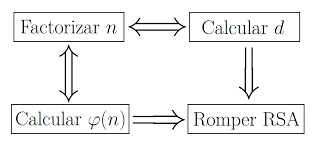


Figure 5

CONCLUSIÓN PERSONAL RSA

El algoritmo RSA es muy utilizado en todo el mundo, su manera de romperse se ve posible gracias a la inteligencia artificial, como los ordenadores cuánticos y que sus bases se reducen a “Problemas Fáciles de descifrar”, pero, aunque sean teorías que no están demostradas, significan un error o unos errores muy importantes en la seguridad de una gran mayoría de empresa a nivel mundial, por lo cual se debería encontrar unas cuantas alternativas.

**WEBGRAFÍA**

<https://es.wikipedia.org/wiki/RSA#Historia>

<https://juncotic.com/rsa-como-funciona-este-algoritmo/>

<https://noticiasseguridad.com/tecnologia/revelan-posible-metodo-para-romper-el-algoritmo-de-cifrado-rsa/>

<https://es.wikipedia.org/wiki/Problema_RSA>

<https://marvin-soto.medium.com/es-posible-romper-la-criptograf%C3%ADa-de-clave-p%C3%BAblica-786d2d787abf>

<https://blog.pleets.org/article/c%C3%B3mo-crear-una-llave-rsa-y-utilizarla>

<https://www.ocr2edit.com/es/convertir-a-txt>

<https://es.wikipedia.org/wiki/Cifrado_(criptograf%C3%ADa)>

<https://ciberseguridad.com/servicios/algoritmos-cifrado/>

<https://internetpasoapaso.com/tipos-cifrado-datos/>

<https://es.wikipedia.org/wiki/RSA>

<http://www.criptored.upm.es/crypt4you/temas/RSA/leccion1/leccion01.html>

<https://hackingenvivo.blogspot.com/2017/11/cifrado-rsa.html>

<https://www.youtube.com/watch?v=pBKQtPJE13U>

<https://juncotic.com/rsa-como-funciona-este-algoritmo/>

<http://www.dma.fi.upm.es/recursos/aplicaciones/matematica_discreta/web/aritmetica_modular/rsa.html>

<https://juncotic.com/rsa-1024bit-logran-romper-gnupg/>

<https://slidetodoc.com/algoritmo-de-cifrado-rsa-seguridad-informtica-equipo-id/>

<https://en.wikipedia.org/wiki/Key_size>

<https://en.wikipedia.org/wiki/Shor%27s_algorithm>

<https://noticiasseguridad.com/tecnologia/revelan-posible-metodo-para-romper-el-algoritmo-de-cifrado-rsa/>

<https://www.muycomputerpro.com/2012/02/16/algoritmo-rsa-no-seguro-totalmente>

<https://www.technologyreview.es/s/11209/un-ordenador-cuantico-rompera-el-cifrado-rsa-de-2048-bits-en-ocho-horas>

<https://www.muyseguridad.net/2012/02/16/investigadores-descubren-fallos-en-la-generacion-de-claves-publicas-del-algoritmo-rsa/>

<https://diarioinforme.com/la-rsa-acaba-de-ser-destruida-por-un-matematico-aleman-retirado/>