Miramontes Sarabia Luis Enrique

Septiembre 17, 2019

RSA

RSA es un algoritmo propuesto por Rivest, Shamir y Adleman. Es una implementación del concepto de criptosistemas de llave pública, inventado por Diffie y Hellman. Se pensó para una era donde los correos electrónicos reemplazarían a los de papel, evento que ya sucedió hace muchos años. El concepto fundamental de la criptografía de llave pública es que revelar una llave usada para cifrado no revela la correspondiente llave para descifrado. Esto abre camino en aplicaciones para establecer comunicaciones seguras sin necesitar un método seguro de transmitir llaves y también la firma digital de mensajes.

La criptografía es la metodología estándar utilizada para mantener la información en secreto, el remitente cifrará el mensaje y el receptor lo descifrará, pero en el caso que una tercera entidad no autorizada quiera leerlo sólo identificará basura. Los algoritmos utilizados en las fechas que RSA fue propuesto sufrían del problema de distribución de llaves; necesitaban de una manera para distribuir las llaves sobre un canal seguro, que no era posible en un sistema como el correo electrónico.

Un ejemplo del funcionamiento de la criptografía de llave pública serpia el siguiente: Bob debe enviar M a Alice, Bob procede a tomar E de un archivo público, entonces envía el mensaje cifrado E(M) a Alice, que a su vez lo descifra usando su llave privada D(E(M)) = M; por las propiedades de la criptografía de llave pública sólo Alice puede descifrar E(M). Cuando se cifra usando la llave privada se obtiene una propiedad adicional, autenticación. Ya que lo cifrado con la llave privada sólo puede ser descifrado con la pública, lo que asegura que el dueño de la llave privada envió ese mensaje.

El método propuesto utiliza el concepto de “trap-door one-way function”, que consiste en una función que es muy fácil de calcular en una dirección, pero muy difícil de calcularla “al revés”. Se les conoce como “trap-door” porque se puede calcular rápidamente si se conoce información especial.

El procedimiento propuesto ocupa una representación del mensaje en enteros del 0 al n-1, se debe de romper el mensaje en partes enteras dentro de este rango, el único fin es que la representación sea numérica. Para cifrar el mensaje se eleva cada número obtenido a la e modulo n.

Dando el ciphertext como resultado el residuo de M^e cuando se divide entre n. El cifrado no reduce el tamaño del mensaje, el ciphertext y el plaintext seguirán teniendo números entre 0 a n-1.

Las llaves utilizadas será un par de enteros positivos (e,n) para cifrar y otro par de enteros positivos (d,n) para descifrar. El usuario hace su llave de cifrado pública y la de descifrado privada. El cálculo de las llave se realiza encontrando n tal que n = p*q donde p y q son dos números primos, la seguridad del algoritmo radica en la complejidad de encontrar los factores p y q primos. Después se obtiene d, que será primo relativo de (p-1)*(q-1). Posteriormente e se obtiene del inverso multiplicativo de d modulo (p-1)(q-1)

La matemática usada por el algoritmo radica en teorema de Fermat-Euler, que afirma uan proposición sobre la divisibilidad de números enteros: “Si a y n son enteros primos relativos, entonces $a^{\varphi(n)} \equiv 1 \pmod{n}$. La fortaliza del algortimo radica en encontrar números primos aleatorios de gran tamaño, el recomendado por los autores es de 100 dígitos para que n tenga 200 dígitos.

En el artículo se proponen ataques posibles contra RSA, el más factible puede ser intentar factorizar n para obtener φ(n) y poder calcular d, pero afortunadamente computacionalmente factorizar números parece mucho más difícil que determinar si son primos. Usando el algoritmo propuesto por Richard Schroeppel para obtener los factores de n y usando dígitos de 200 se tardaría una cantidad de tiempo nada factible, en el orden de los miles de millones de años.

Para concluir el RSA es un algoritmo muy simple y revolucionario, ya que permitió solucionar el problema de compartir llaves sobre canales inseguros y utilizando conceptos básicos como primos, multiplicaciones y módulos ha logrado resistir como seguro hasta nuestros tiempos simplemente modificando el tamaño de las llaves.

Investigando encontré que atacaron RSA de 4096-bits utilizando una técnica llamada criptoanálisis acústico. Utilizaron un micrófono y escucharon las frecuencias que genera el regulador de voltaje de un CPU mientras se encuentra descifrando los datos. Como hemos visto en clase, estos ataques no intentan ir contra el algoritmo sino encontrar otros puntos donde los datos sean sensibles, en este caso el las frecuencias generadas por el hardware, que no tiene que ver con el algoritmo.