Jose Daniel Acuña Solano 2018145020

Firmas Digitales

Surge de la necesidad de avanzar de documentos en papel a digitales, para validar la autenticidad de un emisor, con el fin de proteger contra el fraude, en especial cuando hay entidades financieras involucradas.

Firmas de clave simétrica

Este tipo está enfocado en una entidad centralizada que llaman *Big Brother* en quien todos confían tiene la información correcta, el usuario da la clave secreta a BB para marcarla como la verdadera. Este modelo utiliza a BB como intermediario para validar autenticidad de los mensajes, recibe mensaje encriptado con la llave de A, y lo desencripta para luego encriptarlo con la llave de B, para finalmente enviárselo a B.

Firmas de clave publica

Surgen por el evidente problema de que el modelo anterior tiene como condición que todos confine en la autoridad central que puede ser comprometida. Este modelo consiste en dos llaves una privada que solo el propietario posee y utiliza para desencriptar, y una llave publica que envía a quien sea que se quiere comunicar con él para que encripte los mensajes con esta *public key.*

El estándar que utiliza la industria es RSA, después el Instituto Nacional de Estándares de Tecnología intento crear un algoritmo que llamaron *Digital Signature Standard* (DSS), pero este recibió criticas por se muy secreto (el protocolo) muy lento, nuevo e inseguro.

Compendios de mensaje

Nace de la idea de que las firmas creadas hasta el momento buscan autenticación y confidencialidad, pero no en todos los casos es requerida esta última. Por lo que se buscó crear una función hash que cumpliera con los siguientes cuatro criterios.

- Si ingresamos P, es fácil calcular MD(P)
- Si damos MD(P), es imposible encontrar P
- Dando P, nadie puede encontrar otro mensaje P' tal que MD(P') = MD(P)
- El más mínimo cambio en la entrada produce una salida sumamente diferente

Para cumplir con el criterio 3 se necesitan al menos 128 bits y para el ultimo se necesita truncar el mensaje. Realizar este calculo es mucho mas eficiente que encriptar el mensaje entero usando una llave pública.

El ataque de cumpleaños

La idea de este es encontrar un valor de entrada para la función hash que produzca el mismo resultado que alguna otra entrada, el nombre viene de la analogía de que la probabilidad de encontrar en un grupo dos personas con cumpleaños que coincidan no es tan baja. Por lo que es potencialmente posible *crackear* estos métodos *hash*; para protegerse de estos ataques se puede usar *inputs* extensos con salidas de tamaño menor y fijo, debido a que se da un principio de que hay más posibles entradas en lugar de salidas.