Resumen #7 - (r7)

Esteban Ignacio Durán Vargas - 2020388144

IC - 7602 - 2023 I Semestre

8.4 FIRMAS DIGITALES

Par enviar un mensaje firmado a otra parte de modo que: el receptor verifique el transmisor, el transmisor no pueda negar después el mensaje y que el receptor no pudiera elaborar el mensaje por sí mismo.

8.4.1 Firmas de clave simétrica

Se tiene una autoridad central en la que todos confíen y solo esta sabe las claves de cada uno. Cuando un usuario quiera enviar un texto firmado, genera K(B,R,t,P) con B siendo la identidad receptora, R el número aleatorio escogido, t una marca de tiempo para asegurar que sea reciente y P siendo el mensaje. La autoridad central desencripta y envía el mensaje al receptor con el texto y un mensaje firmado por la autoridad central K(A,T,P).

Si el emisor niega haberlo enviado, el receptor indica que la autoridad central no hubiera aceptado el mensaje si no hubiera sido encriptado; y lo respalda el mensaje firmado de este y este puede descifrar el mensaje original.

Se utilizan marcas de tiempo para revisar que el número aleatorio no se haya repetido y si lo fue lo descarta (con base en la antigüedad de la marca).

8.4.2 Firmas de clave pública

En la criptografía de clave pública no todos tienen que confiar en una entidad. En estos los algoritmos de encripción y desencripción generan el emismo resultado. Cada uno tiene una clave privada y existe una clave pública para todos para elaborar el mensaje. Cuando se recibe un mensaje se transforma con la privada y se almacena el texto y finalmente se descifra con la función de encripción del emisor. Si el emisor niega enviar el mensaje se puede comprobar si envió el mensaje con la función del emisor E. Como el receptor no conoce la clave privada del emisor, la única manera en la que el mensaje se pudo conseguir es si el mensaje fuera enviado por el receptor.

Esto solo sirve si la clave privada del emisor es secreta, sino cualquiera pudo enviarlo. También si el emisor cambia la clave, no se puede demostrar si aplica la nueva clave al mensaje anterior.

El estándar es RSA, pero NIST propuso DSS, que calcula difíciles logaritmos discretos. ste era demasiado secreto, lento, nuevo e inseguro (hasta que se permitieron claves de 1024 bits).

8.4.3 Compendios de mensaje

Se basa en una función de hash (MD) unidireccional que toma una parte arbitrariamente grande de texto y calcula una cadena de bits de longitud fija. Sus propiedades: Dado P, es fácil calcular MD(P), dado MD(P), es imposible encontrar P, dado P nadie puede encontrar P’ de tal manera que MD(P’) = MD(P), un cambio a la entrada de incluso 1 bit produce una salida muy diferente.

Para la tercera el hash debe ser de mayor de 128 bits, para el cuarto la función debe truncar como los de clave simétrica. Es mucho más rápido que con clave pública. El mensaje sería K(A,t, MD(P)). Si hay un reclamo, al desencriptarse se tiene el hash de P y P y es imposible que encuentra otro mensaje con el resultado.

Estos también funcionan en criptosistemas de clave pública donde el emisor calcula el compendio de un texto, firma el compendio y envía ambos y si alguien lo cambia en el camino el receptor lo verá al calcular el Hash de P.

- **MD5**

Diseñado por Ronald Rivest. Trunca los bits de manera que cada bit de salida es afectado por cada uno de entrada. Rellena el mensaje en 448 bits y después la longitud del mensaje original se agrega como entero de 64 bits para que tenga una longitud de un múltiplo de 512 bits. Finalmente, inicializa el bufer de 128 buts a un valor fijo.

Se toma un bloque de 512 bits de entrada y lo mezcla con el búfer. Con una tabla de función seno; evita sospechas de una puerta trasera que el diseñador construyó. Se hacen 4 rondas por cada bloque y continúa por todos. El búfer de 128 bits forma el compendio del mensaje.

- **SHA-1**

Igual procesa bloques de 512 bits pero genera un compendio de 160 biuts. El mensaje se procesa por el algoritmo y se obtiene un hash de 160 bits, el emisor firma el hash con su clave privada y envía al receptor el texto con el hash. El recepto calcula el hash y aplica la clave pública del emisor para esto. Si los dos concuerdan el mensaje es válido. Si es modificado en tránsito, el receptor calcula el hash y se dará cuenta. Garantiza que cualquier modificación al texto llano pueda detectarse con alta probabilidad.

Primero rellena el mensaje con bit 1 al final, seguido por tantos 0 para que sea múltiplo de 512. Al número de 65 bits que contiene la longitud del mesaje se le aplica un OR en los 64 bits de menor orden (big endian); siempre rellena al final del mensaje. SHA-1 mantiene variables de 32 bits donde acumula el hash y se inicializan a constantes. Se procesa cada bloque; las 15 palabras se copial al inicio de un arreglo de 80 palabras y se utiliza una formula para rellenarlo.

$$W_i = S^1 (W_{i-3} \text{ XOR } W_{i-8} \text{ XOR } W_{i-14} \text{ XOR } W_{i-16}) \quad (16 \le i \le 79)$$

donde S^b es la rotación circular de la palabra de 32 bits por b bits. Se inicializan 5 variables A-E según H0-H5. Cuando las 80 iteraciones del ciclo están completas, A a E se agregan a estas. Una vez procesados los primeros 512 bits se inicia el siguiente. El arreglo se reinicia desde el nuevo bloque y el hash se deja igual. Y así con los demás. Cuando termina el último, las cinco palabras de 32 bits en H se envían como la salida del hash de 160 bits.

8.4.4 El ataque de cumpleaños

Se basa en la técnica de probabilidad: ¿Cuántos estudiantes se necesitan en un grupo antes de que la probabilidad de tener dos personas con el mismo cumpleaños exceda 1/2? Apenas 23 (23*22/2=253 con una probabilidad de 1/365). Si hay una correspondencia de las entradas y salidas con n entradas y k salidas hay n(n-1)/2 entradas. Si k es menor, hay buenas probabilidades de una coincidencia. Un compendio de mensajes de 64 bits puede violarse con 2^32 mensajes y buscar dos con el mismo compendio de mensaje. Con MD5 es

difícil porque aun a una velocidad de mil millones de compendios por segundo se requieren 500 años para calcular los 2^64 compendios y no garantiza equivalencia. Sha-1 es mejor porque es más largo.