MA-53G Intr. a la Criptografía	Semestre Primavera, 1996

Cátedra No. 19 : 29 de Octubre

Transcriptor: Jens Hardings

1 Firmas Digitales

Antes de abordar directamente el tema de las firmas digitales, es útil definir qué es lo que esperamos de ellas, y para ello veamos primero por qué se utilizan las firmas convencionales:

1. Autentificación, i.e. convencer al lector de un documento que éste tuvo su origen en el firmante.
2. Para evitar falsificaciones.
3. No reusables, i.e. no se puede usar la misma firma en otro documento.
4. El documento firmado no se puede alterar.
5. No Repudiación: al firmar un documento, el emisor no puede negar haberlo enviado.

Las características enumeradas arriba son las deseables para cualquier sistema ideal de firmas. En la práctica, los sistemas convencionales de firmas utilizados no satisfacen todas ellas. A pesar de esto último, dan un grado de seguridad aceptable, dada la dificultad y el riesgo que implica el falsificar una firma.

Es interesante implementar un sistema de firmas para documentos o mensajes digitales, de donde surge el concepto de *firmas digitales*. En este proceso, surgen diferencias sustanciales, dada la naturaleza del sistema. Por ejemplo, en un sistema de firmas digitales la firma debe ser una secuencia de bits, para que pueda formar parte del mensaje. Como los bits se pueden reproducir trivialmente, es imposible evitar que se saquen copias a documentos firmados. En el caso de un sistema convencional es más fácil de evitar, ya que al copiar un documento, la firma pierde validez. Por ejemplo, al fotocopiar un documento se puede apreciar la diferencia entre la firma original y la fotocopia.

Dadas las diferencias notables entre el sistema tradicional y uno digital, es necesario plantearse algunas preguntas:

- ¿ cómo evitar copias ?
- ¿ cómo "pegar" las firmas a un documento/archivo ?
- ¿ cómo hacer una firma universalmente verificable ?

En base a lo anterior, diremos que los requisitos básicos de un sistema digital que pretenda emular las firmas tradicionales son:

1. La firma debe depender del documento a ser firmado.

   Si no dependiese del documento, sería trivial que cualquier persona firme un documento a nombre de otra (recordar que una firma digital será una secuencia de bits).

2. La firma debe depender (parcialmente) de información que es privada a quien firma.

   Debe depender de información secreta para evitar copias, pero además debe permitir verificar el origen del documento (no repudiación).

3. Debe ser fácil producir y verificar una firma. Aquí existe un compromiso entre la eficiencia de la implementación y la seguridad que ofrecerá el sistema.

4. Las firmas deben ser computacionalmente difíciles de falsificar. Esto para evitar dos tipos de ataques:

   - crear un mensaje para una firma existente. En el caso general, este ataque es menos importante, dado que típicamente el mensaje generado no tendrá sentido. Pero hay casos en que esto puede ser bastante grave, si se llega a cambiar el monto de un cheque, por ejemplo.
   - crear una firma para un mensaje dado. Este caso es más crítico que el anterior, dado que al estar expuestos a un ataque de este tipo, se pierde toda la confiabilidad en el sistema.

5. Deben ser fáciles de guardar, para poder autentificarlas en el momento que se necesite.

2 Firmas basadas en Criptosistemas Simétricos

Este tipo de firmas están basadas en un criptosistema clásico o simétrico, y además se supone la existencia de un árbitro en el que todos los participantes confían, que denotaremos por T (Trusted).

Entonces, tenemos un criptosistema $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, donde e_K es la función de encripción y d_K su correspondiente función de decripción para una clave K dada. Cada participante en el sistema tendrá su propia clave secreta, la cual será conocida por T y el participante mismo.

Queremos construir un protocolo entre A y B, donde A tiene la clave K_a y B tiene la clave K_b. Además, introducimos una estampa de tiempo (Timestamp) T_m, la cual tiene por objeto dar información adicional sobre el momento de la firma del documento.

El protocolo queda como sigue:

1. A : genera el mensaje M a enviar y le entrega a M el mensaje concatenado con la firma:
$$A \to T : M \circ e_{K_a}(M).$$

2. T : verifica que la segunda parte del mensaje corresponde realmente a la firma de A, i.e. debe verificar que $d_{K_a}(M) = M$. Si la firma no corresponde al mensaje, significa que no fue A el que mando el mensaje y se termina.

3. T : transmite el mensaje recibido de A (incluyendo la firma), concatenado con T_m, todo esto encriptado usando la clave K_b:
$$T \to B : e_{K_b}(M \circ e_{K_a}(M) \circ T_m).$$

4. B : decripta lo anterior y obtiene el mensaje de A con su respectiva firma y estampa de tiempo:
$$B : M \circ e_{K_a}(M) \circ T_m.$$

Veamos ahora cómo verificar que A realmente firmó el mensaje M. Para ello, supongamos que B quiere convencer a otro participante, C, de que A le mandó el mensaje M:

1. B: le envía el mensaje que recibió anteriormente:
$$B \to T : e_{K_b}(M \circ e_{K_a}(M) \circ T_m).$$
2. T: decripta ese mensaje, obteniendo $M \circ e_{K_a}(M) \circ T_m$. Luego, verifica que $e_{K_a}(M)$ corresponda a la firma de A para el mensaje M, i.e. que el mensaje provino originalmente de A, y transmite el mensaje a C:
$$T \to C : e_{K_c}(M \circ e_{K_a}(M) \circ T_m).$$
3. C: recibe el mensaje anterior y descifrando obtiene:
$$C : M \circ e_{K_a}(M) \circ T_m.$$

Después de este proceso, C puede estar seguro de que A le mandó el mensaje M a B, y dependiendo de la información que contenga T_m, también podrá saber cuándo se envió el mensaje, o cuantas veces se ha validado, etc.

Observación 1 *implícitamente, para que el sistema funcione de la forma presentada, T_m debe satisfacer ciertas propiedades. Por ejemplo, un cliente no debe ser capaz de generar una estampa de tiempo válida para un documento dado, ya que así podría alterar la fecha, y por lo tanto la validez de un documento.*

2.1 Problemas con este sistema

El principal problema que presenta este sistema es el árbitro T, y esto por varias razones. Al tratar de implementar este sistema, uno de los primeros puntos a tratar será el cómo encontrar un T en el que todos los clientes confíen, cosa muy difícil en la práctica. T debe ser completamente confiable para todos los clientes, puesto que si da a conocer o alguien le roba información, el sistema completo queda inservible, al perder toda la confiabilidad. Ahora, si suponemos que podemos encontrar este T, que es completamente seguro y confiable, tenemos que ,como toda operación pasa a través de T, y suponemos que habrá una gran cantidad de clientes, T deberá tener una gran capacidad computacional para cumplir con todos los requerimientos que le lleguen. Si no puede procesar los requerimientos a la velocidad que llegan, se convertirá automáticamente en un “cuello de botella”, y todos los clientes tendrán que esperarlo antes de poder interactuar con otros.

3 Firmas basadas en criptosistemas de clave pública

Las exigencias más básicas que le podemos hacer a un criptosistema de firmas están dadas por la siguiente definición:

Definición 1 *Un sistema de firmas digitales públicas es una tupla* $(\mathcal{K}, \mathcal{S}, \mathcal{V})$ *donde*

1. $\mathcal{K}$ *es un conjunto de claves.*
2. *dada una clave* $k \in \mathcal{K}$*, existen los conjunto* $\mathcal{P}_k$ *y* $\mathcal{A}_k$*, y las funciones de firma* S_k *y de verificación* V_k *tal que:*
   - $S_k : \mathcal{P}_k \to \mathcal{A}_k$
   - $V_k : \mathcal{P}_k \times \mathcal{A}_k \to \{0, 1\}$
   - *dado cualquier* $x \in \mathcal{P}_k$ *se tiene que* $V_k(x, y) = 1$ *si y sólo si* $S_k(x) = y$*.*
3. V_k *es pública y* S_k *es privada.*

Para que un sistema de firmas sea implementable computacionalmente, se requiere la existencia de algoritmos eficientes para calcular S_k y V_k. Claramente, la seguridad de un sistema como el anterior debe estudiarse en el contexto de la seguridad computacional.

Además, es importante hacer notar que puede necesitarse algo más que este sistema basado en criptosistemas de clave pública. Ello queda más claro con un ejemplo: supongamos que Alicia le manda a Roberto un cheque, el cual firma electrónicamente. Entonces, Roberto lo puede cobrar en el banco y mantener una copia en su poder, la cual podría reutilizar en otro momento. Si se incluye una estampa de tiempo dentro del documento, se puede evitar este tipo de engaños.

3.1 Sistemas de firmas de una vez

Los sistemad de firmas de una vez (one-time signature schemes). Como su nombre lo dice, son sistemas donde la clave se puede utilizar una sola vez.

Sistema de firmas de Lamport

Supuesto: existe una función $f : \{0,1\}^* \to \{0,1\}^*$ de un sentido fuerte.

$\mathcal{K} = \{(y,z) \mid y \in M_{2,m}(\{0,1\}^n) \ tal\ que\ f(y_{i,j}) = z_{i,j}\}$.

$$\text{Si } (y,z) \in \mathcal{K}\ , \quad y = \begin{pmatrix} y_{0,1} & \cdots & y_{0,m} \\ y_{1,1} & \cdots & y_{1,m} \end{pmatrix} \quad z = \begin{pmatrix} z_{0,1} & \cdots & z_{0,m} \\ z_{1,1} & \cdots & z_{1,m} \end{pmatrix}.$$

Dada un clave $K = (y,z)$, la parte pública será z; y es privada. Para un mensaje $M = i_1 i_2 \ldots i_m$ con $i_k \in \{0,1\}$, la función de firma es

$$S_k(i_1 i_2 \ldots i_m) = y_{1,i_1} \ldots y_{m,i_m},$$

y la función de verificación:

$$V_k(i_1 i_2 \ldots i_m, y_1 y_2 \ldots y_m) = \begin{cases} 1 \text{ si } \forall j = 1 \ldots m\ f(y_j) = z_{i_j,j} \\ 0 \text{ si no} \end{cases}$$

Notar que la función f debe ser pública también, para poder verificar cada firma. Con este sistema, un adversario no podrá falsificar una firma a menos que pueda calcular preimágenes de f.

Una desventaja que se presenta es que se requiere una nueva clave por cada firma, dado que una vez que se firma un documento, se está entregando la mitad de las preimágenes de f para esa clave. Además, tenemos que la firma es n veces más larga que el mensaje original, y nos interesa que las firmas sean fáciles de guardar.

Este sistema fue modificado por Bose y Chaum, llegando a firmas digitales de aproximadamente la mitad del largo de las de Lamport.