Clase 19: autenticado de clave publica

Fernando Virdia, versión: 0.0.1, junio 2024

10 Firmas digitales

Finalmente Alicia y Bob pueden comunicar de manera segura, sin tener que intercambiar y proteger una clave secreta intercambiada de antemano: solo requieren la clave publica del otro.

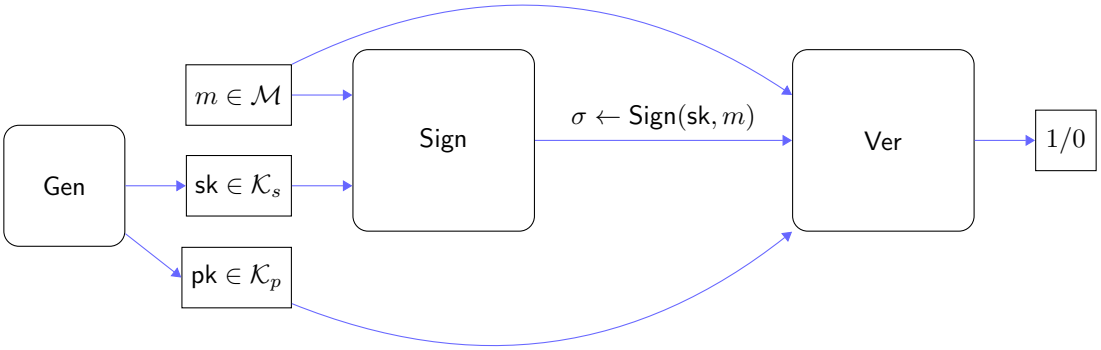
Por ejemplo, Bob podría publicar pk_B en un directorio publico, tipo la guía telefónica. Alicia bajaría pk_B, y mandaría $\mathsf{Enc}(pk_B, \text{“De Alicia: esta es mi clave: } pk_A\text{”})$ a Bob, para poder intercambiar mensajes.

PREGUNTA: Falta algo?

- Como hace Bob para saber que fue justamente Alicia a mandarle el mensaje? $\mapsto$ Alicia podría publicar su clave en la guía.
- Sobre todo, como hace Alicia para saber que pk_B le pertenece a Bob?

Para resolver este problema, necesitamos dos componentes mas: *firmas digitales*, y una *infraestructura de claves publicas*. En esta sección, vamos a construir firmas digitales.

Las firmas digitales sirven para comprobar la autenticidad de un mensaje m, sin requerir una clave secreta compartida. Pueden pensarlo como una versión publica de los MACs, y por eso sintaxis y noción de seguridad son parecidas.



Las firmas pueden solo ser generadas con sk, y pueden solo ser verificadas con pk.

Definición 35 (DSS) *Sean $\mathcal{K}_p$, $\mathcal{K}_s$, $\mathcal{M}$, Σ conjuntos. Sea $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ una tripla de algoritmos eficientes, tales que*

- Gen *es un algoritmo aleatorio,* $\mathsf{Gen}\colon \emptyset \to \mathcal{K}_s \times \mathcal{K}_p$ *que retorna una tupla (*sk, pk*);*
- Sign *es un algoritmo posiblemente aleatorio,* $\mathsf{Sign}\colon \mathcal{K}_s \times \mathcal{M} \to \Sigma$ *que retorna una firma σ;*
- Ver *es un algoritmo determinista* $\mathsf{Ver}\colon \mathcal{K}_p \times \mathcal{M} \times \Sigma \to \{0,1\}$*, que retorna 1 si la firma es “valida” y 0 si no.*

Decimos que Π es un esquema de firma digital (DSS, “digital signature scheme”) δ-completo si

$$\forall m \in \mathcal{M}, \quad \Pr_{\substack{(\mathsf{sk},\mathsf{pk})\sim\mathsf{Gen}(),\\ \$\ en\ \mathsf{Sign}}}[\mathsf{Ver}(\mathsf{pk}, m, \mathsf{Sign}(\mathsf{sk}, m; \$)) \neq 1] \leq \delta.$$

Comentario 60 *Como en el caso de PKE, el espacio de mensajes “firmables” puede depender de pk, $\mathcal{M}_{\mathsf{pk}} \subset \mathcal{M}$.*

Si recuerdan, un MAC seguro se dice “fuertemente infalsificable”, y excluye que $\mathcal{A}$ pueda retornar una tupla (m^*, σ^*) validos y “nuevos”. Vamos a usar una definición parecida para los DSS.

Definición 36 (SUF-CMA) *Sea* $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ *un DSS. Definimos un experimento* $\mathsf{Exp}^{\mathrm{SUF}}$.

$\mathsf{Exp}^{\mathrm{SUF}}(\mathcal{A})$	$S(m)$
$(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}()$	$\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m)$
$Q \leftarrow \{\}$	$Q \leftarrow Q \cup \{(m, \sigma)\}$
$(m^*, \sigma^*) \leftarrow \mathcal{A}^S(\mathsf{pk})$	**return** σ
$// \ (m^*, \sigma^*) \notin Q$	
$b \leftarrow \mathsf{Ver}(\mathsf{pk}, m^*, \sigma^*)$	
return b	

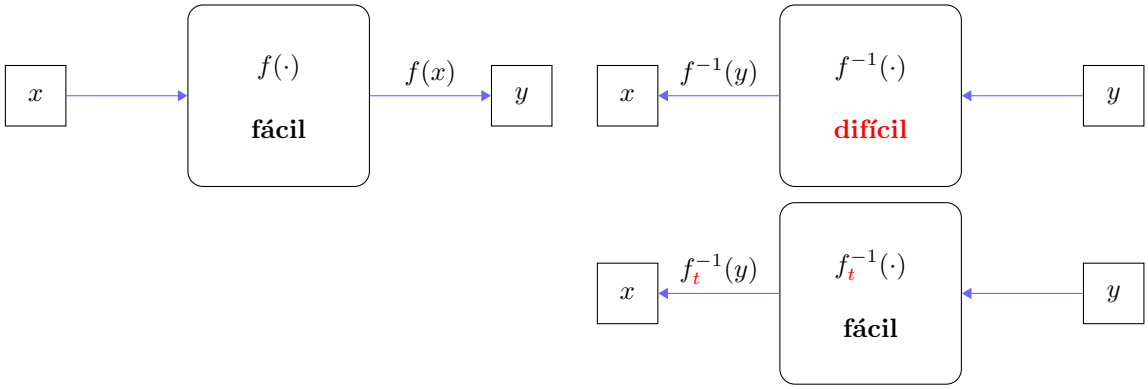
Decimos que Π *es* (ε, t, q)*-fuertemente infalsificable bajo ataques de mensaje elegido (SUF-CMA, "strongly unforgeable under chosen-message attacks") si cualquier* $\mathcal{A}$ *que corre en tiempo* $\leq t$*, realiza* $\leq q$ *queries a* S*, y retorna* $(m^*, \sigma^*) \notin Q$*, tiene ventaja*

$$\mathsf{Adv}(\mathsf{Exp}^{\mathrm{SUF}}, \mathcal{A}) := \Pr[\mathsf{Exp}^{\mathrm{SUF}}(\mathcal{A}) \Rightarrow 1] \leq \varepsilon.$$

Comentario 61 *En la definición de seguridad MAC, se podía discutir si darle a* $\mathcal{A}$ *acceso a un oráculo* $\mathsf{Ver}(k, \cdot)$*. Esta discusión no existe con DSS, dado que* $\mathsf{Ver}(\mathsf{pk}, \cdot)$ *puede ser evaluada públicamente.*

Existen principalmente cuatro maneras de realizar DSS: con protocolos de identificación, con funciones de puerta trasera, con funciones de hash y con "MPC-in-the-head". Nosotros vamos a cubrir el segundo método, en su forma clásica a través *permutaciones de puerta trasera*, que es comúnmente utilizado en practica.

La idea es la siguiente: dado un conjunto D finito, $f\colon D \to D$ es una permutación (o biyección) públicamente computable, tal que $f^{-1}\colon D \to D$ es difícil de computar, a menos de no conocer una "puerta trasera".



Para firmar un mensaje, podríamos usar f como clave publica, t como clave secreta, y generar firmas σ de mensajes *invirtiendo* f, y calculando $f(\sigma)$ como verificación.

Definición 37 (TDP) *Sean* $\mathcal{T}$*,* $\mathcal{F}$ *dos conjuntos. Sea* $\Pi = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Invert})$ *una tripla de algoritmos, tales que*

- Gen *es un algoritmo aleatorio,* $\mathsf{Gen}\colon \emptyset \to \mathcal{T} \times \mathcal{F}$ *que retorna una tupla* (t, f)*, con* $f\colon \mathcal{D}_f \to \mathcal{D}_f$ *una permutación sobre un conjunto finito* $\mathcal{D}_f$*, que puede depender de* f*.*
- Eval *es un algoritmo determinista,* $\mathsf{Eval}\colon \mathcal{F} \times \mathcal{D}_f \to \mathcal{D}_f$ *que dado* (f, x) *retorna* $f(x)$*;*
- Invert *es un algoritmo determinista* $\mathsf{Invert}\colon \mathcal{T} \times \mathcal{F} \times \mathcal{D}_f \to \mathcal{D}_f$*, que dados* (t, f, y) *retorna* $f^{-1}(y)$*.*

Decimos que Π *es una permutación de puerta trasera (TDP, "trapdoor permutation").*

La dificultad de invertir una TDP se la puede "capturar" con una definición de seguridad.

Definición 38 (OW TDP) *Sean* $\Pi = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Invert})$ *una TDP. Definimos* $\mathsf{Exp}^{\mathrm{OW}}$*,*

$\mathsf{Exp}^{\mathrm{OW}}(\mathcal{A})$

1: $(t, f) \leftarrow \mathsf{Gen}()$
2: $y \xleftarrow{\$} \mathcal{D}_f$
3: $x \leftarrow \mathcal{A}(f, y)$
4: **return** $[\![f(x) = y]\!]$

Decimos que Π *es* (ε, t)*-unidireccional (OW, "one-way"), si cualquier* $\mathcal{A}$ *que corre en tiempo* $\leq t$ *tiene ventaja*

$$\mathsf{Adv}(\mathsf{Exp}^{\mathrm{OW}}, \mathcal{A}) \coloneqq \Pr[\mathsf{Exp}^{\mathrm{OW}}(\mathcal{A}) \Rightarrow 1] \leq \varepsilon.$$

Comentario 62 *Notamos que el experimento requiere* $y \sim U(\mathcal{D}_f)$*, y requiere que a* $\mathcal{A}$ *no le sean dados oráculos de inversión! Esto puede volver las demostraciones mas difíciles.*

Esto nos trae a nuestro ultimo postulado.

Postulado 4 *Dado* (ε, t)*, existen TDPs que son* (ε, t)*-OW seguras.*