

Clase 19: autenticado de clave publica

Fernando Virdia, versión: 0.1, mayo 2025

Finalmente Alicia y Bob pueden comunicar de manera confidencial, sin tener que intercambiar claves secretas de manera segura: 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 a Bob

- $c \leftarrow \text{Enc}(pk_B, \text{"De Alicia: esta es mi clave publica: } pk_A\text{"})$, o
- $c \leftarrow \text{Enc}(pk_B, \text{"De Alicia: usemos esta clave secreta: } k\text{"})$,

para poder intercambiar mensajes.

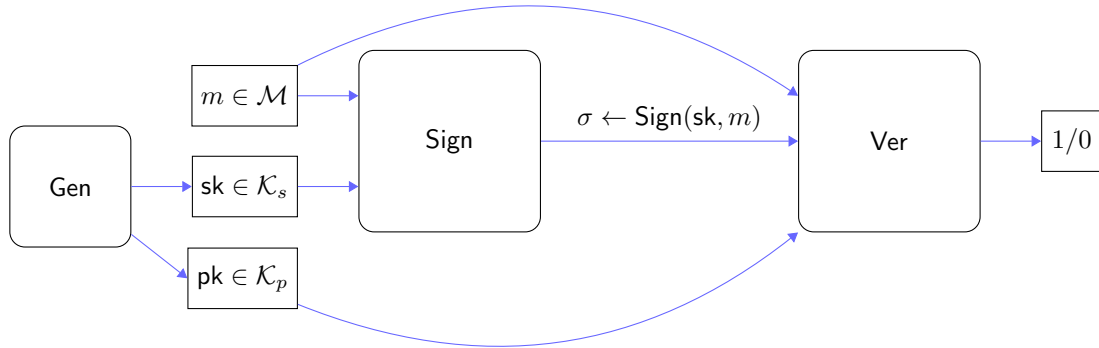
PREGUNTA: Falta algo?

- Como hace Bob para saber que fue justamente Alicia a mandarle el mensaje?
- Como hace Alicia para saber que efectivamente pk_B le pertenece a Bob?

Para resolver este problema, necesitamos dos componentes mas: *firmas digitales*, y una *infraestructura de claves publicas*.

10 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 MAC, 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 42 (DSS). Sean $\mathcal{K}_p$, $\mathcal{K}_s$, $\mathcal{M}$, Σ conjuntos. Sea $\Pi = (\text{Gen}, \text{Sign}, \text{Ver})$ una tripla de algoritmos eficientes, tales que

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

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

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

Comentario 64. Como en el caso de PKE, el espacio de mensajes “firmables” puede depender de pk , $\mathcal{M}_{pk} \subset \mathcal{M}$.

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 43 (SUF-CMA). Sea $\Pi = (\text{Gen}, \text{Sign}, \text{Ver})$ un DSS. Definimos un experimento Exp^{SUF} .

$\text{Exp}^{\text{SUF}}(\mathcal{A})$	$S(m)$
1: $(\text{sk}, \text{pk}) \leftarrow \text{Gen}()$	1: $\sigma \leftarrow \text{Sign}(\text{sk}, m)$
2: $Q \leftarrow \{\}$	2: $Q \leftarrow Q \cup \{(m, \sigma)\}$
3: $(m^*, \sigma^*) \leftarrow \mathcal{A}^S(\text{pk})$	3: return σ
4: $b \leftarrow \llbracket \text{Ver}(\text{pk}, m^*, \sigma^*) = 1 \wedge (m^*, \sigma^*) \notin Q \rrbracket$	
5: return b	

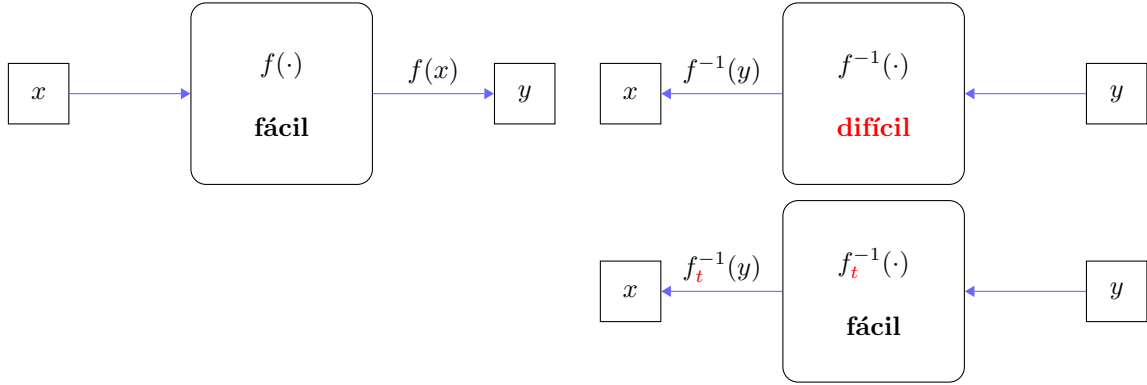
Decimos que Π es (ε, t, q) -fuertemente infalsificable bajo ataques de mensaje elegido (SUF-CMA, “strongly unforgeable under chosen-message attacks”) si cualquier $\mathcal{A}$ que corre en tiempo $\leq t$, realiza $\leq q$ queries a S , tiene ventaja

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

Comentario 65. En la definición de seguridad MAC, se podía discutir si darle a $\mathcal{A}$ acceso a un oráculo $\text{Ver}(k, \cdot)$. Esta discusión no existe con DSS, dado que $\text{Ver}(\text{pk}, \cdot)$ 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, $f: D \rightarrow D$ es una permutación (o biyección) públicamente computable, tal que $f^{-1}: D \rightarrow D$ 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 44 (TDP). Sean $\mathcal{T}, \mathcal{F}$ dos conjuntos. Sea $\Pi = (\text{Gen}, \text{Eval}, \text{Invert})$ una tripla de algoritmos, tales que

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

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

Opcionalmente, si $(\mathcal{D}_f, \times)$ es un grupo, $f: \mathcal{D}_f \rightarrow \mathcal{D}_f$ puede ser un isomorfismo de grupo, o sea dados $g, h \in \mathcal{D}_f$, $f(g \times h) = f(g) \times f(h)$.

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

Definición 45 (OW TDP). Sean $\Pi = (\text{Gen}, \text{Eval}, \text{Invert})$ una TDP. Definimos Exp^{OW} ,

$\text{Exp}^{\text{OW}}(\mathcal{A})$
1: $(t, f) \leftarrow \text{Gen}()$
2: $y \xleftarrow{\$} \mathcal{D}_f$
3: $x \leftarrow \mathcal{A}(f, y)$
4: return $\llbracket f(x) = y \rrbracket$

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

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

Comentario 66. 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. Esta omisión es necesaria si f es un isomorfismo de grupos: de otro modo, dados $f^{-1}(y_1), f^{-1}(y_2)$, $\mathcal{A}$ podría calcular la inversa $f^{-1}(y_1) \times f^{-1}(y_2) = f^{-1}(y_1 \times y_2)$ de $y_1 \times y_2$, sin pedirselo al oráculo.

Esto nos trae a nuestro ultimo postulado.

Postulado 4. Dado $\lambda > 0$, existen TDPs que otorgan λ -bits de seguridad OW y que son isomorfismos de grupo.