

Clase 20: firmas “full-domain hash”

Fernando Virdia, versión: 0.1, mayo 2025

Vamos ahora a definir un DSS que utiliza TDPs y funciones de hash para firmar $m \in \{0,1\}^*$.

Definición 46 (TDP-FDH, o “hash-then-invert”). Sea $\Pi = (\text{Gen}, \text{Eval}, \text{Invert})$ una TDP, y sea $\{H_f\}_f$ una familia de funciones de hash $H_f := \{0,1\}^* \rightarrow \mathcal{D}_f$, donde $f: \mathcal{D}_f \rightarrow \mathcal{D}_f$. Definimos el DSS $\Pi' = (\text{Gen}', \text{Sign}', \text{Ver}')$ de hash de dominio completo (FDH, “full-domain hash” o “invert-then-sign”),

$\text{Gen}'()$	$\text{Sign}'(\text{sk} = (t, f), m)$	$\text{Ver}'(\text{pk} = f, \sigma)$
1: $(t, f) \leftarrow \text{Gen}()$	1: $h \leftarrow H_f(m)$	1: $h \leftarrow H_f(m)$
2: $\text{sk} \leftarrow (t, f)$	2: $\sigma \leftarrow \text{Invert}(t, f, h)$	2: $h' \leftarrow \text{Eval}(f, \sigma)$
3: $\text{pk} \leftarrow f$	3: return σ	3: return $\llbracket h' = h \rrbracket$
4: return (sk, pk)		

Comentario 67. Mas allá del tipo de primitiva usado, la mayoría de los DSS efectivamente firman hashes de un mensaje, no el mensaje directamente (“hash-then-sign”). “Deshacer” la estructura algebraica de m puede ser útil para obtener seguridad. Por ejemplo, esto es necesario si se usa la famosa TDP de Rivest, Shamir y Adleman (RSA).

Comentario 68. Usamos $m \in \{0,1\}^*$, pero uno podría definir TDP-FDH sobre cualquier dominio S tal que $H: S \rightarrow \mathcal{D}_f$ es una función de hash.

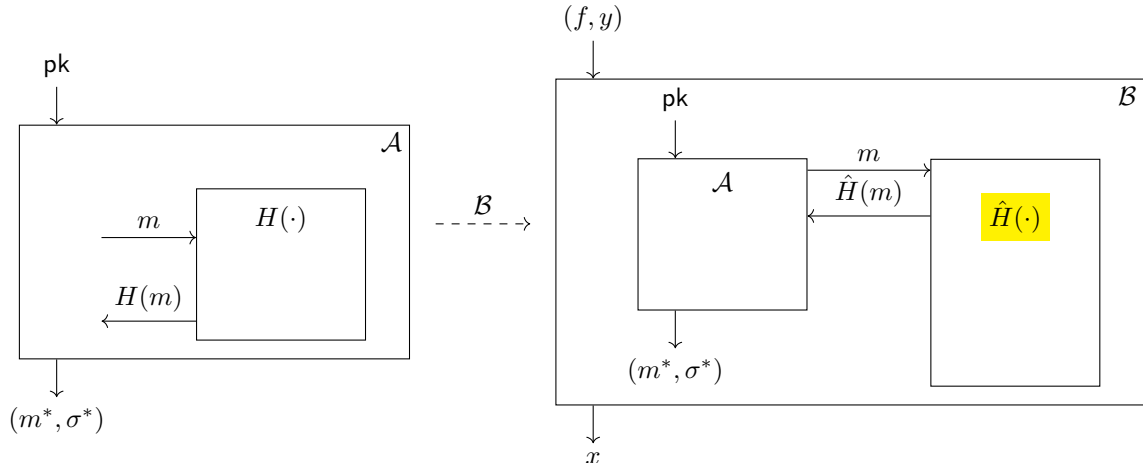
Comentario 69. Escribimos H_f , pero generalmente la función de hash es predeterminada públicamente de manera de valer para diferentes conjuntos $\mathcal{D}_f$.

Comentario 70. Como habrán notado, el esquema es muy parecido a usar una PRF para realizar un MAC, reemplazando la PRF (o PRP!) con la TDP.

Finalmente, demostramos que una TDP-FDH otorga SUF-CMA.

Teorema 8 (OW TDP + ROM $\Rightarrow$ SUF-CMA). Sea Π una TDP que es también un isomorfismo de grupos, y sea Π' el resultante DSS obtenido como TDP-FDH. Supongamos que las funciones de hash $\{H_f\}_f$ son completamente aleatorias (o sea, trabajamos con el modelo del oráculo aleatorio). Sea $q_S \geq 1$ el número de queries al oráculo de firmas $S(\cdot)$ que le permitimos a un adversario $\mathcal{A}$ contra Π' . Si Π es (ε, t) -OW entonces Π' es (ε', t', q_S) -SUF-CMA con $t' \approx t$, $\varepsilon' \lesssim e \cdot q_S \cdot \varepsilon$, donde $e = 2,71828\dots$ es la constante de Napier.

Proof. Sea $\mathcal{A}$ un adversario contra TDP-FDH. Sea $p \in (0,1)$ una constante, el cual valor determinaremos mas tarde. Vamos a definir $\mathcal{B}$, un adversario que juega Exp^{OW} contra la TDP. Sin pérdida de generalidad, escribimos H en lugar de H_f . Dado que suponemos la función H ser completamente aleatoria, vamos a suponer que cuando $\mathcal{A}$ requiere $H(m)$, calcula este valor a través de un oráculo, que nosotros vamos a reemplazar.



Esto puede parecer raro, pero puede ser pensado de la siguiente manera. Supongamos que $\mathcal{A}$ es un programa en C, y precisa calcular un hash $H(m)$. Posiblemente, $\mathcal{A}$ obtenga H a través de una librería de enlace dinámico (una DLL) contra la cual viene compilado, dado que es improbable que la estructura de la función de hash tenga un impacto sobre la dificultad de falsificar una firma (es común suponer que los hashes son “collision resistant”).

Obtenido $\mathcal{A}$, el adversario $\mathcal{B}$ decide de re-compilarla contra una diferente DLL, que implementa otra función aleatoria $\hat{H}(\cdot)$. La intuición es que esto no tendría que reducir la probabilidad de $\mathcal{A}$ de retornar una falsificación (m^*, σ^*) respecto al DSS TDP-FDH con hash $\hat{H}$, dado que $\mathcal{A}$ es un algoritmo aleatorio, y H es una función aleatoria, por lo que la distribución de la variable aleatoria $\mathcal{A}^{\hat{H}}(f, y)$ es la misma que la de $\mathcal{A}^H(f, y)$.

Decimos que este tipo de demostración se encuentra en el *modelo del oráculo aleatorio* (ROM, “random oracle model”). Mucho ha sido debatido sobre el ROM. [KM15] cubre los aspectos mas importantes del debate. A menudo este modelo nos permite demostrar la seguridad de esquemas mas eficientes [BR93], que resisten la “prueba del tiempo” una ves desplegados. Por ejemplo, la demostración de que FO es KEM OT-IND-CCA, también se encuentra en el ROM.

$\mathcal{B}(f, y)$	$\hat{H}(m)$	$S(m)$
1 : $p \leftarrow //$ un valor particular $\in (0, 1)$	1 : if $m \in L$:	1 : $\hat{H}(m)$
2 : $Q \leftarrow \{\}$	2 : $(\dots, h) \leftarrow L[m]$	2 : $(b, s, h) \leftarrow L[m]$
3 : $L \leftarrow []$	3 : return h	3 : if $b = 1$:
4 : $\text{pk} \leftarrow f$	4 : $//$ si no, $m \notin L$:	4 : ERROR
5 : $(m^*, \sigma^*) \leftarrow \mathcal{A}^{S(\cdot), \hat{H}(\cdot)}(\text{pk})$	5 : $r \xleftarrow{\$} (0, 1)$	5 : $Q \leftarrow Q \cup \{(m, s)\}$
6 : $\hat{H}(m^*) \parallel \Rightarrow m^* \in L$	6 : $s \xleftarrow{\$} \mathcal{D}_f$	6 : return s
7 : $(b^*, s^*, h^*) \leftarrow L[m^*]$	7 : if $r < p$: $//$ con pr. p	
8 : $x \leftarrow \sigma^* \times (s^*)^{-1}$	8 : $h \leftarrow \text{Eval}(f, s)$	
9 : return x	9 : $L[m] \leftarrow (0, s, h)$	
	10 : else : $//$ con pr. $1 - p$	
	11 : $h \leftarrow y \times \text{Eval}(f, s)$	
	12 : $L[m] \leftarrow (1, s, h)$	
	13 : return h	

La idea de la demostración es que para falsificar una firma, $\mathcal{A}$ tiene que invertir f sobre algun input $H(m^*)$. Dado Π' , un adversario $\mathcal{A}$ que juega Exp^{SUF} tiene acceso a un oráculo $H(\cdot)$, que vamos a remplazar con otro oráculo $\hat{H}$ de manera de “inyectar” un valor $y \in D_f$ que $\mathcal{B}$ deba invertir, en el output de $\hat{H}$.

Primero analizemos la distribución de $\hat{H}$, desde el punto de vista de $\mathcal{A}$ “dentro” de $\mathcal{B}$. Al calcular $\hat{H}(m)$, dos cosas pueden pasar:

- Si $r < p$, generamos $s \sim U(\mathcal{D}_f)$, y retornamos $h := f(s)$. Dado que f es una permutación, $h = f(s)$ también está uniformemente distribuida, $h \sim U(D_f)$.

$$\left(\forall y \in \mathcal{D}_f, \Pr_{\sigma \sim U(D)} [f(\sigma) = y] = \Pr_{\sigma \sim U(D)} [\sigma = f^{-1}(y)] = \frac{1}{|\mathcal{D}_f|} \right)$$

Por lo tanto, $\hat{H}(m) \mid r < p \sim U(D_f)$.

- Si $r \geq p$, queremos “inyectar” y , generado en Exp^{OW} , dentro de la salida $\hat{H}(m)$. Generamos $s \sim U(D_f)$, tal que $f(s) \sim U(D_f)$ también, y retornamos $y \times f(s) \sim U(D_f)$. Por lo tanto $\hat{H}(m) \mid r \geq p \sim U(D_f)$.

En ambos casos, $\hat{H}(m) \sim U(D_f)$, así como lo seria $H(m)$. En el ROM, esta técnica se llama “programación del oráculo aleatorio”.

Dentro de Exp^{SUF} , $\mathcal{A}$ tambien tiene acceso a un oráculo de firma, $S(\cdot)$. En Exp^{SUF} este es realizado usando la clave secreta, $S(m) := \text{return Sign}(\text{sk}, m)$. Al no tener sk , $\mathcal{B}$ utiliza los valores s generados internamente a $\hat{H}$ para responder $S(m)$. Recordamos que en TDP-FDH, $\text{Sign}(\text{sk} = (t, f), m) = f^{-1}(H(m))$. Al calcular $S(m)$, primero forzamos el calculo de $h \leftarrow \hat{H}(m)$. Sea $(b, s, h) \leftarrow L[m]$. Luego,

- Si h fue generado en el ramo “ $r < p$ ” ($b = 0$) de $\hat{H}$, entonces $s = f^{-1}(h)$ a $\mathcal{A}$ por construcción de $\hat{H}$, y retornamos s , que notamos ser $s \sim U(D_f) \sim f^{-1}(H(m))$.
- Si h fue generado en el ramo “ $r \geq p$ ” ($b = 1$), $f^{-1}(h) = f^{-1}(y) \times s$. Al no conocer $f^{-1}(y)$, no podemos *simular* el algoritmo de firma. $\mathcal{B}$ interrumpe su ejecución retornando un **ERROR** y fallando en el ataque. Denotamos este evento, E , y el evento que no ocurra un error, $\bar{E}$.

Finalmente, observamos que en cuanto $\mathcal{A}$ retorna (m^*, σ^*) , $\mathcal{B}$ calcula $\hat{H}(m^*)$, de manera de asegurarse que $m^* \in L$, lo que simplifica el siguiente analisis. Sean $(b^*, s^*, h^*) \leftarrow L[m^*]$. Supongamos que al calcular $\hat{H}(m^*)$, $b^* \leftarrow 1$, o sea $h^* \leftarrow y \times f(s^*)$. Si $\mathcal{A}$ logra retornar una falsificación valida, entonces $\sigma^* = f^{-1}(h^*) = f^{-1}(y \times f(s^*)) = f^{-1}(y) \times s^*$, y $\mathcal{B}$ retorna $x \leftarrow \sigma^* \times s^{*-1} = f^{-1}(y) \times s^* \times s^{*-1} = f^{-1}(y)$, ganando Exp^{OW} .

Para poder asegurarnos que $\mathcal{A}$ logre falsificar una firma, debemos asegurarnos que su entorno dentro de $\mathcal{B}$ sea indistinguible de el de Exp^{SUF} . Hemos visto que la distribucion de $\hat{H}(\cdot)$ es la correcta, y asi es la de $S(\cdot)$, siempre que no ocurra un **ERROR**. Por lo tanto, el entorno dentro de $\mathcal{B}$ será correcto si no ocurre un error, o sea $\bar{E}$. Antes de evaluar $S(m)$, viendo $\hat{H}(m)$, $\mathcal{A}$ no tiene ninguna información sobre si m va a causar un **ERROR** de manera más o menos probable respecto a algun otro $m' \neq m$. Por lo tanto del punto de vista de $\mathcal{A}$, $\Pr[S(m) \Rightarrow \text{ERROR}] = 1 - p$. Suponiendo que $\mathcal{A}$ usa sus q_S queries a S sobre al máximo q_S inputs diferentes, y siendo el valor de $S(m)$ fijo, $\Pr[\bar{E}] \geq p^{q_S}$.

Podemos entonces calcular la ventaja de $\mathcal{B}$ en base a la de $\mathcal{A}$ y a $\Pr[\bar{E}]$. Sean $(b^*, s^*, h^*) \leftarrow L[m^*]$.

$$\begin{aligned}
\text{Adv}(\text{Exp}^{\text{OW}}, \mathcal{B}) &= \Pr[\text{Exp}^{\text{OW}}(\mathcal{B}) \Rightarrow 1] = \Pr[f(\sigma^* \times s^{*-1}) = y] = \Pr[f(\sigma^*) = y \times f(s^*)] \\
&= \Pr[f(\sigma^*) = \hat{H}(m^*)] \\
&\geq \Pr[f(\sigma^*) = \hat{H}(m^*) \wedge (m^*, \sigma^*) \notin Q \wedge b^* = 1 \wedge \bar{E}] \\
&= \Pr[f(\sigma^*) = \hat{H}(m^*) \wedge (m^*, \sigma^*) \notin Q \wedge b^* = 1 \mid \bar{E}] \cdot \Pr[\bar{E}] \\
&= \Pr[\text{Exp}^{\text{SUF}}(\mathcal{A}^{\hat{H}(\cdot), \text{Sign}(\text{sk}, \cdot)}) \Rightarrow 1 \wedge b^* = 1] \cdot \Pr[\bar{E}] \\
&= \Pr[\text{Exp}^{\text{SUF}}(\mathcal{A}^{H(\cdot), \text{Sign}(\text{sk}, \cdot)}) \Rightarrow 1] \cdot \Pr[b^* = 1] \cdot \Pr[\bar{E}] \\
&\geq \text{Adv}(\text{Exp}^{\text{SUF}}, \mathcal{A}) \cdot (1 - p) \cdot p^{q_S} \\
\iff \text{Adv}(\text{Exp}^{\text{SUF}}, \mathcal{A}) &\leq \frac{\text{Adv}(\text{Exp}^{\text{OW}}, \mathcal{B})}{p^{q_S} \cdot (1 - p)}.
\end{aligned}$$

Al ser Π una (ε, t) -OW TDP, obtenemos $\varepsilon' \leq \varepsilon / (p^{q_S} \cdot (1 - p))$.

Ahora recordamos que $p \in (0, 1)$ es una constante de valor arbitrario, y notamos que $(p^{q_S} \cdot (1 - p)) \in (0, 1)$. Para obtener la mejor cota superior de $\text{Adv}(\text{Exp}^{\text{SUF}}, \mathcal{A})$ (o sea, la cota superior mas baja posible), queremos elegir el valor $p = p_0$ que maximiza el denominador, $g(p) := p^{q_S} \cdot (1 - p)$, con $p \in (0, 1)$.

1. Calculamos la derivada:

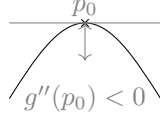
$$\begin{aligned}
g'(p) &= \frac{d}{dp} [p^{q_S}] (1 - p) + p^{q_S} \frac{d}{dp} [1 - p] \\
&= q_S \cdot p^{q_S-1} (1 - p) + p^{q_S} \cdot (-1) \\
&= p^{q_S-1} (q_S - q_S \cdot p - p) \\
g'(p) = 0 &\iff q_S - q_S \cdot p - p = q_S - p(q_S + 1) = 0 \iff p = \frac{q_S + 1 - 1}{q_S + 1} = 1 - \frac{1}{q_S + 1} =: p_0.
\end{aligned}$$

2. Verificamos que es un máximo:

$$\begin{aligned}
g''(p) &= \frac{d}{dp} [p^{q_S-1}] (q_S - q_S \cdot p - p) + p^{q_S-1} \frac{d}{dp} [q_S - q_S \cdot p - p] \\
&= (q_S - 1) \cdot p^{q_S-2} (q_S - q_S \cdot p - p) + p^{q_S-1} (-q_S - 1) \\
&= p^{q_S-2} ((q_S - 1) \cdot (q_S - q_S \cdot p - p) - p \cdot (q_S + 1)) \\
&= p^{q_S-2} (q_S \cdot (q_S - q_S \cdot p - p) - (q_S - q_S \cdot p - p) - p \cdot (q_S + 1)) \\
&= p^{q_S-2} (q_S^2 - q_S^2 \cdot p - q_S \cdot p - q_S + q_S \cdot p + p - p \cdot q_S - p) \\
&= p^{q_S-2} (q_S^2 - q_S^2 \cdot p - q_S \cdot p - q_S + \cancel{q_S \cdot p} + \cancel{p} - \cancel{p \cdot q_S} - \cancel{p})
\end{aligned}$$

$$\begin{aligned}
&= p^{q_S-2} \cdot q_S \cdot (q_S - q_S \cdot p - p - 1) \\
\implies g''(p_0) &= p_0^{q_S-2} \cdot q_S \cdot (\cancel{q_S - q_S \cdot p - p} - 1) = -q_S \cdot p_0^{q_S-2} < 0.
\end{aligned}$$

Dado que $g''(p_0)$ es negativo, $p_0 = 1 - \frac{1}{q_S+1}$ es un máximo de g (el único, dado $g'(p)$).



Ahora calculamos el denominador en p_0 ,

$$\begin{aligned}
g(p_0) &= \left(1 - \frac{1}{q_S+1}\right)^{q_S} \cdot \left(1 - \left(1 - \frac{1}{q_S+1}\right)\right) \\
&= \frac{\left(1 - \frac{1}{q_S+1}\right)^{q_S+1}}{\left(1 - \frac{1}{q_S+1}\right)} \cdot \frac{1}{q_S+1} = \frac{\left(1 - \frac{1}{q_S+1}\right)^{q_S+1}}{\left(\frac{q_S+1}{q_S+1} - \frac{1}{q_S+1}\right)} \cdot \frac{1}{q_S+1} \\
&= \left(1 - \frac{1}{q_S+1}\right)^{q_S+1} \cdot \frac{1}{q_S}
\end{aligned}$$

Dado que $\lim_{n \rightarrow \infty} \left(1 - \frac{1}{n}\right)^n = e^{-1}$ y que $q_S \gg 1$,

$$g(p_0) \approx \frac{1}{e} \cdot \frac{1}{q_S}.$$

Por lo tanto, si dentro de $\mathcal{B}$ asignamos $p \leftarrow p_0 = 1 - \frac{1}{q_S+1}$,

$$\text{Adv}(\text{Exp}^{\text{SUF}}, \mathcal{A}) \leq \frac{\text{Adv}(\text{Exp}^{\text{OW}}, \mathcal{B})}{g(p_0)} \lesssim e \cdot q_S \cdot \text{Adv}(\text{Exp}^{\text{OW}}, \mathcal{B}),$$

y $t' \approx t$, $\varepsilon' \lesssim e \cdot q_S \cdot \varepsilon$. □

Comentario 71. Notamos que las propiedades de H como función de hash (collision-resistance, etc) no juegan un rol directo en la demostración. Esto es porque ignoramos ser H un hash, y la consideramos una oráculo de función aleatoria. De manera sutil, decidimos suponer que los ataques no van a ser causados por un adversario $\mathcal{A}$ que invierta o ataque la estructura interna de H , y como esta posiblemente interactúe con la estructura de la TDP.