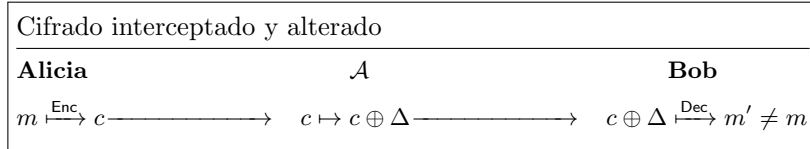


Clase 11: adversarios activos e integridad de datos

Fernando Virdia, versión: 0.1, mayo 2025

6 Integridad de mensajes

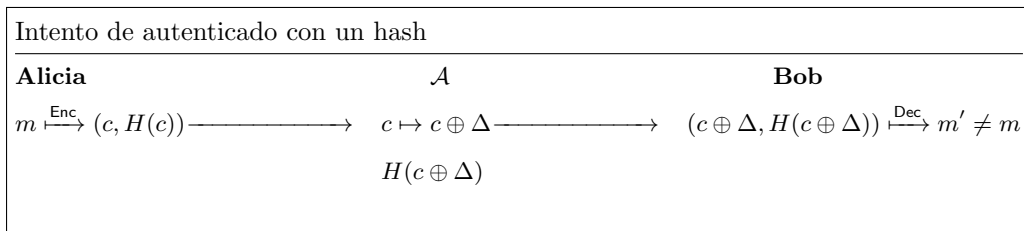
Hasta ahora pudimos garantizar la seguridad si $\mathcal{A}$ se limita a observar cifrados en transito. Ahora consideren el siguiente escenario



En este caso, un SKES no nos brinda garantías.

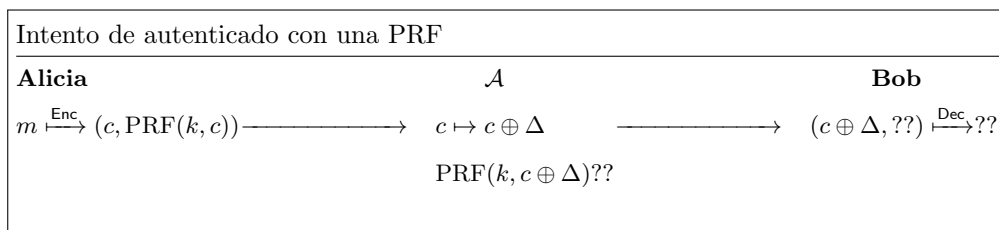
PREGUNTA: Alguien conoce algún método para chequear la integridad de datos?

Idea: Usar un checksum/hash.

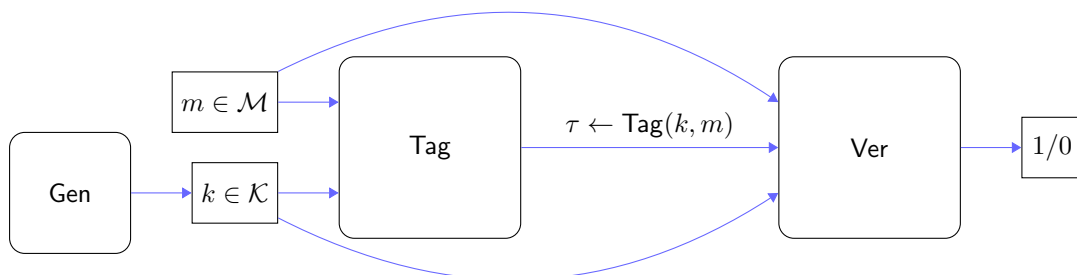


No funciona (a lo sumo, hashing ofrece “integridad sin adversarios”, aunque un código de corrección de errores es una mejor solución a ese problema). Ideas?

Idea: Y si... usamos una PRF?



Para saber si funciona, necesitamos una definición de seguridad. Que sintaxis queremos? Sirve para verificar la autenticidad de un “mensaje” (que puede ser un cifrado).



Definición 19 (MAC). Sean $\mathcal{M}, \mathcal{T}$ conjuntos, $\mathcal{K}$ un conjunto finito. Decimos que $\Pi = (\text{Gen}, \text{Tag}, \text{Ver})$ es un código de autenticación de mensajes (MAC, “message authentication code”) con espacio de llave $\mathcal{K}$, de mensajes $\mathcal{M}$, y de etiquetas (tags) $\mathcal{T}$, si

- Gen es un algoritmo aleatorio $\text{Gen}: \emptyset \rightarrow \mathcal{K}$,
- Tag es un algoritmo (posiblemente aleatorio) $\text{Tag}: \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{T}$,
- Ver es un algoritmo (posiblemente aleatorio) $\text{Ver}: \mathcal{K} \times \mathcal{M} \times \mathcal{T} \rightarrow \{0, 1\}$,

donde

$$\forall k \in \mathcal{K}, m \in \mathcal{M}, \quad \Pr[\text{Ver}(k, m, \text{Tag}(k, m)) = 1] = 1.$$

Comentario 35. Para proteger comunicaciones, va a ser necesario transmitir un tag τ junto a los datos transmitidos. Por eso, idealmente queremos que τ sea lo mas corto posible. Posiblemente $|\tau|$ seria independiente de $|m|$.

Comentario 36. Un sistema con Tag determinista tiene la ventaja de que Ver es muy simple:

$$\text{Ver}(k, m, \tau) := \text{return } \llbracket \text{Tag}(k, m) = \tau \rrbracket$$

Dada la sintaxis de MACs, cualquier $\Pi = (\text{Gen}, \text{Tag}, \text{Ver})$ es un MAC si $\text{Ver}(k, m, \tau) := 1$. Necesitamos definir una noción de seguridad para que “ $1 \leftarrow \text{Ver}(k, m, \tau)$ ” tenga un significado.

Definición 20 (MAC security). Sea $\Pi = (\text{Gen}, \text{Tag}, \text{Ver})$ un MAC con espacios $\mathcal{K}, \mathcal{M}, \mathcal{T}$.

$\text{Exp}^{\text{MAC}}(\mathcal{A})$	$T(m)$
1: $k \leftarrow \text{Gen}()$	1: $\tau \leftarrow \text{Tag}(k, m)$
2: $Q \leftarrow \{\}$	2: $Q \leftarrow Q \cup \{(m, \tau)\}$
3: $(m^*, \tau^*) \leftarrow \mathcal{A}^T()$	3: return τ
4: $b \leftarrow \llbracket \text{Ver}(k, m^*, \tau^*) = 1 \wedge (m^*, \tau^*) \notin Q \rrbracket$	
5: return b	

Decimos que Π es (ε, t, q) -fuertemente infalsificable en un ataque de mensajes elegidos (SUF-CMA, “strongly unforgeable under chosen-message attacks”), si cualquier adversario $\mathcal{A}$ que corre en tiempo $\leq t$ y usa $\leq q$ queries a $T(\cdot)$, tiene ventaja

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

Comentario 37. Hay cosas que se pueden cuestionar en esta definición.

- $\mathcal{A}$ gana si genera un cualquier nuevo tag, inclusive sobre un mensaje que ha ya sido autenticado. Que problema hay en generar una nueva etiqueta para algo que sabemos ser autentico?
- Al $\mathcal{A}$ le permitimos generar tags. Por que también no verificarlos? ‘Podemos notar que para Π con $\text{Ver}(k, m, \tau) := \llbracket \text{Tag}(k, m) = \tau \rrbracket$, $T(\cdot)$ puede ser usado también para verificar. Usaremos esta definición y tipo de MAC para simplificar la exposición. Mas allá de la estructura de Ver , se puede demostrar que otorgarle oráculos de verificación al adversario no lo ayuda de manera significativa [BS23, § 6.2].
- Obtener integridad de comunicaciones no necesariamente excluye todo ataque por parte de adversarios activos. Por ejemplo, la noción de seguridad MAC no cubre ataques donde $\mathcal{A}$ produce “replays” de mensajes previamente autenticados. Estas no son falsificaciones del MAC, pero de cualquier manera pueden si ser peligrosas para el usuario. Piensen en el replay de una orden de compra.

Ahora si podemos demostrar que usar una PRF nos permite autenticar datos.

Lemma 11 (PRF $\Rightarrow$ fixed-length MAC). Sea $F: \mathcal{K} \times \mathcal{D} \rightarrow \mathcal{R}$ una (ε, t, q) -PRF. Sea $\Pi = (\text{Gen}, \text{Tag}, \text{Ver})$ un MAC con espacio de claves $\mathcal{K}$, de mensajes $\mathcal{M} = \mathcal{D}$, y de tags $\mathcal{T} = \mathcal{R}$ donde

$\text{Gen}()$	$\text{Tag}(k, m)$	$\text{Ver}(k, m, \tau)$
1: $k \xleftarrow{\$} \mathcal{K}$	1: $\tau \leftarrow F(k, m)$	1: $\tau' \leftarrow F(k, m)$
2: return k	2: return τ	2: return $\llbracket \tau' = \tau \rrbracket$

Π ofrece (ε', t', q') -seguridad MAC con $\varepsilon' = \varepsilon + \frac{1}{|\mathcal{R}|}$, $t' \approx t$, $q' = q - 1$.

Demostración.

Construimos un adversario $\mathcal{B}$ que juega Exp^{PRF} .

$\mathcal{B}^{\mathcal{O}}()$	$T(m)$
1: $(m^*, \tau^*) \leftarrow \mathcal{A}^T()$	1: $\tau \leftarrow \mathcal{O}(m)$
2: if $\mathcal{O}(m^*) = \tau^* \wedge (m^*, \tau^*) \notin Q$:	2: return τ
3: return 0 (“PRF”)	
4: else :	
5: return 1 (“random”)	

Consideremos $\text{Exp}^{\text{PRF}}(\mathcal{B}, 1)$, donde $\mathcal{O}$ es una función R realmente aleatoria.

$$\begin{aligned}
\Pr[\text{Exp}^{\text{PRF}}(\mathcal{B}, 1) \Rightarrow 0] &= \Pr[R(m^*) = \tau^* \wedge (m^*, \tau^*) \notin Q] \\
&= \Pr[R(m^*) = \tau^* \mid (m^*, \tau^*) \notin Q] \cdot \Pr[(m^*, \tau^*) \notin Q] \\
&\leq \Pr[R(m^*) = \tau^* \mid (m^*, \tau^*) \notin Q]
\end{aligned}$$

Al ser una función, R tiene una sola imagen de m^* . Dado que estamos condicionando sobre $(m^*, \tau^*) \notin Q$, estamos suponiendo que $\mathcal{A}$ nunca ha evaluado $T(m^*) = R(m^*) = \tau^*$, como para poderlo retornar. Por ende, tuvo que adivinar el valor de $\tau^* \in \mathcal{R}$. Dado que R es una función aleatoria, la probabilidad de que $\mathcal{A}$ adivine correctamente es $\Pr[R(m^*) = \tau^* \mid (m^*, \tau^*) \notin Q] = \frac{1}{|\mathcal{R}|}$.

Ahora consideramos $\text{Exp}^{\text{PRF}}(\mathcal{B}, 0)$. Aquí $\mathcal{O} = F(k, \cdot)$, una PRF. Por construcción, el entorno de $\mathcal{A}$ dentro de $\mathcal{B}$ en $\text{Exp}^{\text{PRF}}(\mathcal{B}, 0)$ es idéntico al de $\mathcal{A}$ en Exp^{MAC} , cuando el algoritmo de etiquetado evalúa $\text{Tag}(k, \cdot) = F(k, \cdot)$. Sigue que

$$\begin{aligned}
\Pr[\text{Exp}^{\text{PRF}}(\mathcal{B}, 0) \Rightarrow 0] &= \Pr[F(k, m^*) = \tau^* \wedge (m^*, \tau^*) \notin Q] \\
&= \Pr[\text{Ver}(k, m^*, \tau^*) = 1 \wedge (m^*, \tau^*) \notin Q] \\
&= \Pr[\text{Exp}^{\text{MAC}}(\mathcal{A}) \Rightarrow 1].
\end{aligned}$$

Usando la desigualdad triangular,

$$\begin{aligned}
\mathcal{A}(\text{Exp}^{\text{MAC}}, \mathcal{A}) &= \Pr[\text{Exp}^{\text{MAC}}(\mathcal{A}) \Rightarrow 1] = \Pr[\text{Exp}^{\text{PRF}}(\mathcal{B}, 0) \Rightarrow 0] \\
&= \left| \Pr[\text{Exp}^{\text{PRF}}(\mathcal{B}, 0) \Rightarrow 0] - \Pr[\text{Exp}^{\text{PRF}}(\mathcal{B}, 1) \Rightarrow 0] + \Pr[\text{Exp}^{\text{PRF}}(\mathcal{B}, 1) \Rightarrow 0] \right| \\
&\leq \left| \left(\lambda - \Pr[\text{Exp}^{\text{PRF}}(\mathcal{B}, 0) \Rightarrow 1] \right) - \left(\lambda - \Pr[\text{Exp}^{\text{PRF}}(\mathcal{B}, 1) \Rightarrow 1] \right) \right| + \left| \Pr[\text{Exp}^{\text{PRF}}(\mathcal{B}, 1) \Rightarrow 0] \right| \\
&= \text{Adv}(\text{Exp}^{\text{PRF}}, \mathcal{B}) + \left| \Pr[\text{Exp}^{\text{PRF}}(\mathcal{B}, 1) \Rightarrow 0] \right| \\
&\leq \text{Adv}(\text{Exp}^{\text{PRF}}, \mathcal{B}) + \frac{1}{|\mathcal{R}|}.
\end{aligned}$$

Dado que $\mathcal{A}$ y $\mathcal{B}$ corren en tiempos parecidos, y que $\mathcal{B}$ hace una query mas a $\mathcal{O}$ que $\mathcal{A}$ a T , Π es (ε', t', q') -MAC con $\varepsilon' = \varepsilon + \frac{1}{|\mathcal{R}|}$, $t' \approx t$, $q' + 1 = q$. $\square$

Comentario 38. Un MAC con espacios de mensajes $\{0, 1\}^d$ con $d < \infty$ (“fixed-length”), puede ser limitante al tratar de proteger la integridad de mensajes arbitrariamente largos. Hay varias maneras de extender la amplitud de los mensajes. Vamos a ver una técnica que utiliza funciones de hash.

Lemma 12 (fixed-length MAC + CR hash $\Rightarrow$ arbitrary-length MAC, hash-then-MAC). Sea $\Pi = (\text{Gen}, \text{Tag}, \text{Ver})$ un fixed-length (ε, t, q) -MAC con $\mathcal{M} = \{0, 1\}^\ell$, y $H: \{0, 1\}^* \rightarrow \{0, 1\}^\ell$ un CR-hash. Definimos un MAC $\Pi' = (\text{Gen}', \text{Tag}', \text{Ver}')$,

$\text{Gen}'()$	$\text{Tag}'(k, m)$	$\text{Ver}'(k, m, \tau)$
1: $k \xleftarrow{\$} \text{Gen}$	1: $h \leftarrow H(m)$	1: $h \leftarrow H(m)$
2: return k	2: $\tau \leftarrow \text{Tag}(k, h)$	2: $b \leftarrow \text{Ver}(k, h, \tau)$
	3: return τ	3: return b

Π' ofrece (ε', t', q') -seguridad MAC con $t' \approx t$, $q' = q$, $\varepsilon' \leq \varepsilon + \Pr[\text{coll}]$, donde coll es la probabilidad que un adversario contra el MAC Π produzca una colisión en H como parte de su funcionamiento.

Demostración. Usamos un adversario $\mathcal{A}$ contra Π' para definir un adversario $\mathcal{B}$ contra Π .

$\mathcal{B}^T$	$\mathcal{O}(m)$
1: $Q \leftarrow \{\}, S \leftarrow \{\}$	1: $h \leftarrow H(m)$
2: $(m^*, \tau^*) \leftarrow \mathcal{A}^\mathcal{O}()$	2: $\tau \leftarrow T(h)$
3: $h^* \leftarrow H(m^*)$	3: $Q \leftarrow Q \cup \{(h, \tau)\}$
4: if $(h^*, \tau^*) \in Q$:	4: $S \leftarrow S \cup \{(m, \tau)\}$
5: return $(\perp, \perp)$	5: return τ
6: else return (h^*, τ^*)	

NB: El conjunto S en $\mathcal{O}(m)$ simplifica el análisis, pero no es requerido por la implementación de $\mathcal{B}$.

La idea de esta demostración es que si $\mathcal{A}$ falsifica correctamente $((m^*, \tau^*) \notin S)$ una etiqueta y $(h^*, \tau^*) \notin Q$, entonces $\mathcal{B}$ también falsifica correctamente. Por otro lado, si $\mathcal{A}$ falsifica correctamente $((m^*, \tau^*) \notin S)$ pero $(h^*, \tau^*) \in Q$, entonces existe algún mensaje $\bar{m} \neq m^*$ con $(\bar{m}, \tau^*) \in S$ tal que $h^* = H(m^*) = H(\bar{m})$, por lo que $\mathcal{A}$ produjo una colisión $(\bar{m}, m^*)$ de H .

Formalmente, empezamos expandiendo la ventaja de $\mathcal{A}$ contra Π' en términos de Π .

$$\begin{aligned}
\text{Adv}(\text{Exp}^{\text{MAC}}, \mathcal{A}) &= \Pr[\text{Exp}^{\text{MAC}}(\mathcal{A}) \Rightarrow 1] = \Pr[\text{Ver}'(k, m^*, \tau^*) = 1 \wedge (m^*, \tau^*) \notin S] \\
&= \Pr[\text{Ver}(k, H(m^*), \tau^*) = 1 \wedge (m^*, \tau^*) \notin S] \\
&= \Pr[\text{Ver}(k, h^*, \tau^*) = 1 \wedge (m^*, \tau^*) \notin S] \\
&= \Pr[\text{Ver}(k, h^*, \tau^*) = 1 \wedge (m^*, \tau^*) \notin S \wedge (h^*, \tau^*) \notin Q] \\
&\quad + \Pr[\text{Ver}(k, h^*, \tau^*) = 1 \wedge (m^*, \tau^*) \notin S \wedge (h^*, \tau^*) \in Q] \\
&\leq \Pr[\text{Ver}(k, h^*, \tau^*) = 1 \wedge (h^*, \tau^*) \notin Q] + \Pr[(m^*, \tau^*) \notin S \wedge (h^*, \tau^*) \in Q] \\
&= \Pr[\text{Exp}^{\text{MAC}}(\mathcal{B}) \Rightarrow 1] + \Pr[(m^*, \tau^*) \notin S \wedge (h^*, \tau^*) \in Q] \\
&= \text{Adv}(\text{Exp}^{\text{MAC}}, \mathcal{B}) + \Pr[(m^*, \tau^*) \notin S \wedge (h^*, \tau^*) \in Q]
\end{aligned}$$

Por suposición, los conjuntos S y Q están formados por mensajes m_i y hashes $h_i = H(m_i)$, donde los m_i fueron generados por $\mathcal{A}$, así como lo fue m^* . Sabemos que $h^* = H(m^*)$. Si $(m^*, \tau^*) \notin S$ pero $(h^*, \tau^*) \in Q = \{(H(m_i), \tau_i)\}_i$, entonces existe un $(m_i, \tau_i) \in S$ tal que $\tau_i = \tau^*$ y $H(m_i) = h^* = H(m^*)$ donde $m^* \neq m_i$.

Lo que quiere decir que $\mathcal{A}$ genera una colisión (m_i, m^*) de H , lo que implica que

$$\Pr[(m^*, \tau^*) \notin S \wedge (h^*, \tau^*) \in Q] \leq \Pr[\text{coll}].$$

Juntando las tres desigualdades,

$$\begin{aligned}
\text{Adv}(\text{Exp}^{\text{MAC}}, \mathcal{A}) &\leq \text{Adv}(\text{Exp}^{\text{MAC}}, \mathcal{B}) + \Pr[(m^*, \tau^*) \notin S \wedge (h^*, \tau^*) \in Q] \\
&\leq \text{Adv}(\text{Exp}^{\text{MAC}}, \mathcal{B}) + \Pr[\text{coll}].
\end{aligned}$$

Por inspección de $\mathcal{A}$ y $\mathcal{B}$, $t' \approx t$, $q' = q$ y $\varepsilon' \leq \varepsilon + \Pr[\text{coll}]$. □

Comentario 39. *Requerir un MAC y un hash contruidos de manera independiente puede ser costoso en hardware. Construcciones tipo HMAC se especializan en construir hash-then-MAC compactos a partir de funciones de hash de tipo Merkle-Damgard.*