

Clase 12: confidencialidad e integridad, el cifrado autenticado

Fernando Virdia, versión: 0.1, mayo 2025

7 Cifrado Autenticado

Resumen: hemos conseguido

- SKES para mensajes múltiples y arbitrariamente largos
- MACs para cadenas de bits arbitrariamente largas

Que nos falta: integrar ambos mecanismos.

Vamos a protegernos de ambos escenarios limitando la posibilidad de $\mathcal{A}$ de modificar cifrados.

Definición 21 (INT-CTXT). Sea $\perp$ un símbolo que indica rechazo. Sea $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ un SKES con $\perp \in \mathcal{M}$, tal que Dec puede retornar $\perp$ si recibe un cifrado no válido.

$\text{Exp}^{\text{CTXT}}(\mathcal{A})$	$E(m)$
1: $k \xleftarrow{\$} \text{Gen}()$	1: $c \leftarrow \text{Enc}(k, m)$
2: $Q_c \leftarrow \{\}$	2: $Q_c \leftarrow Q_c \cup \{c\}$
3: $c^* \leftarrow \mathcal{A}^E()$	3: return c
4: $m^* \leftarrow \text{Dec}(k, c^*)$	
5: $b \leftarrow \llbracket m^* \neq \perp \wedge c^* \notin Q_c \rrbracket$	
6: return b	

Decimos que Π otorga (ε, t, q) -integridad de cifrado (INT-CTXT, “ciphertext integrity”), si cualquier adversario $\mathcal{A}$ que corre en tiempo $\leq t$ y hace $\leq q$ queries a $\mathcal{O}$, tiene ventaja

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

Comentario 40. Claramente, PRF-CTR no satisface INT-CTXT, dado que $(c_0, c_1 \oplus \Delta)$ descifra a $m \oplus \Delta$ para cualquier Δ .

Definición 22 (AE). Sea Π un SKES. Decimos que Π otorga cifrados $(\varepsilon_{\text{CPA}}, \varepsilon_{\text{CTXT}}, t, q)$ -autenticados (AE, “authenticated encryption”), si otorga seguridad $(\varepsilon_{\text{CPA}}, t, q)$ -IND-CPA y $(\varepsilon_{\text{CTXT}}, t, q)$ -INT-CTXT.

Comentario 41. Bajo definición de seguridad AE, $\mathcal{A}$ no puede distinguir entre cifrados de mensajes diferentes por IND-CPA, y no puede tampoco generar nuevos cifrados válidos (de la nada, o modificando cifrados que obtuvo a través de un oráculo de cifrado) por INT-CTXT.

Realizar un SKES que otorga seguridad AE es relativamente simple.

Comentario 42. Un SKES que otorga AE, también otorga una noción que no vamos a explorar de inmediato, IND-CCA. Exp^{CCA} es igual a Exp^{CPA} , pero $\mathcal{A}$ tiene acceso a oráculos de cifrado $E(m_0, m_1)$ y de descifrado $D(c)$, donde el segundo puede ser utilizado solamente para descifrar cifrados no retornados por $E(\cdot, \cdot)$. En el ámbito de la criptografía simétrica, $\text{AE} \Rightarrow \text{IND-CCA}$ dado que $\mathcal{A}$ no puede generar cifrados válidos a menos de no obtenerlos de $E(\cdot, \cdot)$. De consecuencia, un adversario IND-CCA contra un SKES AE no puede generar cifrados para pasarle al oráculo de descifrado D que no resulten en un rechazo $D(c) = \perp$.

Definición 23 (EtM). Sean $\Pi = (\text{Gen}^\Pi, \text{Enc}, \text{Dec})$ un SKES, y $\Sigma = (\text{Gen}^\Sigma, \text{Tag}, \text{Ver})$ un MAC. Definimos el SKES $\Pi' = (\text{Gen}', \text{Enc}', \text{Dec}')$ “encrypt-then-MAC” (EtM).

$\text{Gen}'()$	$\text{Enc}'(k, m)$	$\text{Dec}'(k, c)$
1: $k_e \leftarrow \text{Gen}^\Pi()$	1: $(k_e, k_m) \leftarrow k$	1: $(k_e, k_m) \leftarrow k$
2: $k_m \leftarrow \text{Gen}^\Sigma()$	2: $c_e \leftarrow \text{Enc}(k_e, m)$	2: $(c_e, \tau) \leftarrow c$
3: $k \leftarrow (k_e, k_m)$	3: $\tau \leftarrow \text{Tag}(k_m, c_e)$	3: $b \leftarrow \text{Ver}(k_m, c_e, \tau)$
4: return k	4: $c \leftarrow (c_e, \tau)$	4: $m \leftarrow \text{Dec}(k_e, c_e)$
	5: return c	5: if $b = 1$:
		6: return m
		7: else return $\perp$

Lemma 13. Sean Π , Σ , Π' como en Definition 23. Supongamos que Π otorga $(\varepsilon_{\text{CPA}}, t, q)$ -IND-CPA y Σ otorga $(\varepsilon_{\text{MAC}}, t, q)$ -MAC. Entonces Π' otorga $(\varepsilon'_{\text{CPA}}, \varepsilon'_{\text{CTXT}}, t', q')$ -AE con $\varepsilon'_{\text{CPA}} = \varepsilon_{\text{CPA}}$, $\varepsilon'_{\text{CTXT}} = \varepsilon_{\text{MAC}}$, $t' \approx t$, $q' = q$.

Demostración. Por definición de AE, necesitamos demostrar que

- Π' otorga $(\varepsilon'_{\text{CPA}}, t', q')$ -IND-CPA,
- Π' otorga $(\varepsilon'_{\text{CTXT}}, t', q')$ -INT-CTXT.

IND-CPA: Dado un adversario $\mathcal{A}$ contra Π' que juega Exp^{CPA} , construimos $\mathcal{B}$ contra Π que juega Exp^{CPA} ,

$\mathcal{B}^E()$	$\mathcal{O}(m_0, m_1)$
1 : $k_m \leftarrow \text{Gen}^\Sigma()$	1 : $c_e \leftarrow E(m_0, m_1)$
2 : $b' \leftarrow \mathcal{A}^\mathcal{O}()$	2 : $\tau \leftarrow \text{Tag}(k_m, c_e)$
3 : return b'	3 : $c \leftarrow (c_e, \tau)$
	4 : return c

Del punto de vista de $\mathcal{A}^\mathcal{O}$, el ambiente dentro de $\mathcal{B}$ es idéntico al de $\text{Exp}^{\text{CPA}}(\mathcal{A}, b)$. Dado que $\mathcal{B}$ retorna el output de $\mathcal{A}$,

$$\Pr[\text{Exp}^{\text{CPA}}(\mathcal{B}, b) \Rightarrow 1] = \Pr[\text{Exp}^{\text{CPA}}(\mathcal{A}, b) \Rightarrow 1], \quad \forall b \in \{0, 1\}.$$

De consecuencia, $\text{Adv}(\text{Exp}^{\text{CPA}}, \mathcal{B}) = \text{Adv}(\text{Exp}^{\text{CPA}}, \mathcal{A})$, $t' \approx t$ y $q' = q$, y Π' otorga $(\varepsilon_{\text{CPA}}, \approx t, q)$ -IND-CPA.

INT-CTXT: Ahora supongamos que un adversario $\mathcal{A}$ contra Exp^{CTXT} , y construyamos $\mathcal{B}$ contra Exp^{MAC} .

$\mathcal{B}^T()$	$\mathcal{O}(m)$
1 : $k_e \leftarrow \text{Gen}^\Pi()$	1 : $c_e \leftarrow \text{Enc}(k_e, m)$
2 : $c^* \leftarrow \mathcal{A}^\mathcal{O}()$	2 : $\tau \leftarrow T(c_e)$
3 : $(c_e^*, \tau^*) \leftarrow c^*$	3 : $c \leftarrow (c_e, \tau)$
4 : // c_e^* es el mensaje “tagueado”	4 : return c
5 : return (c_e^*, τ^*)	

Fundamentalmente, c^* consiste de un “mensaje” arbitrario c_e^* y de una etiqueta τ^* . Por inspección del algoritmo Dec' del cifrado EtM, notamos que para que este retorne un mensaje $m^* \neq \perp$, es necesario que $\text{Ver}(k_m, c_e^*, \tau^*) = 1$. Si $\mathcal{A}$ logra producir una tupla c^* nueva y valida, efectivamente ha falsificado una etiqueta del MAC.

Formalmente, supongamos que $\mathcal{A}$ se encuentra jugando Exp^{CTXT} contra el cifrado EtM con clave de cifrado k_e . Sigue que por definición

$$\text{Adv}(\text{Exp}^{\text{CTXT}}, \mathcal{A}) = \Pr[\text{Dec}'(k_e, c^*) \neq \perp \wedge c^* \notin Q_c]$$

Sea c^* un cifrado EtM con componentes $c^* = (c_e^*, \tau^*)$. Por inspección del algoritmo Dec' del cifrado EtM, notamos que para que este retorne un mensaje $m^* \neq \perp$, es necesario que $\text{Ver}(k_m, c_e^*, \tau^*) = 1$, por lo que

$$\Pr[\text{Dec}'(k_e, c^*) \neq \perp \wedge c^* \notin Q_c] \leq \Pr[\text{Ver}(k_m, c_e^*, \tau^*) = 1 \wedge c^* \notin Q_c]$$

Ahora notemos que el entorno de $\mathcal{A}$ dentro de $\mathcal{B}^T$ cuando este ultimo juega el experimento Exp^{MAC} es exactamente idéntico al entorno de $\mathcal{A}$ dentro de Exp^{CTXT} . En particular, cada vez que $\mathcal{A}$ le pasa un mensaje m a $\mathcal{O}$, un cifrado $c_e \leftarrow \text{Enc}(k_e, m)$ viene pasado como input al oráculo de etiquetado T en Exp^{MAC} , el cual retorna una etiqueta τ sobre c_e . Esto quiere decir que $c = (c_e, \tau) \in Q_c$ dentro de $\mathcal{B} \iff (c_e, \tau) \in Q$ dentro de $\text{Exp}^{\text{MAC}}(\mathcal{B})$. Por lo tanto,

$$\begin{aligned} & \Pr[\text{Ver}(k_m, c_e^*, \tau^*) = 1 \wedge c^* \notin Q_c \text{ dentro de } \text{Dec}' \text{ dentro de } \text{Exp}^{\text{CTXT}}] \\ &= \Pr[\text{Ver}(k_m, c_e^*, \tau^*) = 1 \wedge (c_e^*, \tau^*) \notin Q \text{ dentro de } \mathcal{B} \text{ dentro de } \text{Exp}^{\text{MAC}}] \\ &= \Pr[\text{Exp}^{\text{MAC}}(\mathcal{B}) \Rightarrow 1] = \text{Adv}(\text{Exp}^{\text{MAC}}, \mathcal{B}). \end{aligned}$$

Juntando las desigualdades, obtenemos que $\text{Adv}(\text{Exp}^{\text{CTXT}}, \mathcal{A}) \leq \text{Adv}(\text{Exp}^{\text{MAC}}, \mathcal{B})$. Por lo tanto, tenemos $\varepsilon'_{\text{CTXT}} \leq \varepsilon_{\text{MAC}}$, y por construcción de $\mathcal{B}$, $t' \approx t$, $q' = q$, y Π' otorga $(\varepsilon_{\text{MAC}}, \approx t, q)$ -INT-CTXT.

Y por lo tanto, Π' otorga $(\varepsilon_{\text{CPA}}, \varepsilon_{\text{MAC}}, \approx t, q)$ -AE. $\square$

Comentario 43. Si definiéramos la seguridad MAC de manera que a $\mathcal{A}$ le fuera permitido generar nuevos tags τ' a partir de un mensaje ya etiquetado por $T(\cdot)$ (o sea, el MAC no sería “fuertemente” infalsificable), el resultante cifrado EtM no otorgaría AE. Esto es porque $\mathcal{A}$ podría ver un válido cifrado $c = (c_e, \tau)$ y generar un nuevo cifrado $c' = (c_e, \tau')$. Esto podría ser un problema en el caso específico de EtM, pero la definición de IND-CTXT es general, y no tendría que solamente funcionar para EtM.

Ejemplo 3. Combinar esquemas AE puede no otorgar trivialmente AE. Considérese el esquema natural obtenido concatenando cifrados: $\text{Enc}'(k, m_1 || m_2) := \text{Enc}(k, m_1) || \text{Enc}(k, m_2)$, donde Enc es parte de un SKES que otorga AE, y cada componente del cifrado $c = (c_1, c_2)$ es descifrado individualmente. ‘Este sistema no otorga AE, por que?’

Dado un cifrado (c_1, c_2) válido, el cifrado (c_2, c_1) es también válido. Esto quiere decir que $\mathcal{A}$ puede generar nuevos cifrados válidos, invalidando INT-CTXT.

Comentario 44. AE nos permite de detectar si un cifrado es autentico. Pero no impide que $\mathcal{A}$ reordene, elimine o repita cifrados en transito desde Alicia a Bob. Sin embargo, podemos intentar detectar este tipo de ataque en el nivel aplicativo, definiendo números de secuencia. Esto generalmente resulta en consideraciones dependientes de la aplicación, y que pueden encontrarse al menos parcialmente fuera del dominio de la criptografía.

En algunas aplicaciones a nivel de red, puede ser necesario transmitir datos cifrados junto a meta-datos (e.g., “packet headers”) públicos. Por ejemplo, los meta-datos podrían ser útiles para “routeear” los paquetes de datos. En esta situación, una primitiva conocida como “authenticated encryption with associated data” (AEAD) puede ser útil. Esta garantiza confidencialidad de datos, e integridad del cifrado y de los meta-datos. Los dos AEAD más famosos son AES-GCM y Chacha20-Poly1305.

Definición 24 (Un AEAD de ejemplo). Sean $\Pi = (\text{Gen}^\Pi, \text{Enc}, \text{Dec})$ un SKES, $\Sigma = (\text{Gen}^\Sigma, \text{Tag}, \text{Ver})$ un arbitrary-length MAC. Definimos un AEAD

$\text{Gen}'()$	$\text{Enc}'(k, d, m)$	$\text{Dec}'(k, d, c)$
1: $k_e \leftarrow \text{Gen}^\Pi()$	1: $(k_e, k_m) \leftarrow k$	1: $(k_e, k_m) \leftarrow k$
2: $k_m \leftarrow \text{Gen}^\Sigma()$	2: $c_e \leftarrow \text{Enc}(k_e, m)$	2: $(c_e, \tau) \leftarrow c$
3: $k \leftarrow (k_e, k_m)$	3: $\ell \leftarrow d $	3: $\ell \leftarrow d $
4: return k	4: $\tau \leftarrow \text{Tag}(k_m, \ell d c_e)$	4: $b \leftarrow \text{Ver}(k_m, \ell d c_e, \tau)$
	5: return (c_e, τ)	5: $m \leftarrow \text{Dec}(k_e, c_e)$
		6: if $b = 1$:
		7: return m
		8: else return $\perp$

Comentario 45. No proponemos una definición de seguridad AEAD. Invitamos el lector a consultar [BS23, § 9.5].

Con esto terminamos la componente simétrica. Hemos aprendido a cifrar y transmitir cifrados de manera segura entre dos partes Alicia y Bob que comparten una misma clave secreta k . Pero como pueden llegar Alicia y Bob a compartir tal clave? Claro, podrían compartirla de antemano de persona, de manera de estar seguras que ningún adversario $\mathcal{A}$ pueda verla, pero esto no es conveniente. Todos los días navegamos a paginas web, posiblemente por la primera vez, y hacemos pagos online con diferentes entidades. Pre-compartir claves con todas sería prácticamente imposible.

En la segunda parte del curso vamos a ver como transmitir mensajes de manera segura a entidades con las cuales no nos hemos nunca comunicado.