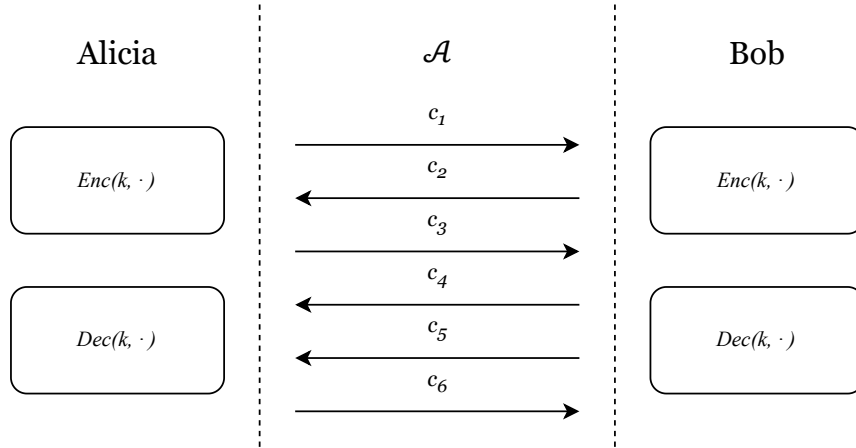


Clase 8: confidencialidad para mensajes múltiples

Fernando Virdia, versión: 0.1, mayo 2025

4 Cifrar más de un mensaje

Como anteriormente, queremos poder usar un SKES, pero generar mas de un mensaje.



PREGUNTA: Ideas sobre como definiré seguridad “multi-mensajes”?

Podríamos permitirle a $\mathcal{A}$ de ver $c_1, \dots, c_n$. Vamos a hacer algo mas: le vamos a permitir pedir el cifrado de mensajes arbitrarios! O sea: sin darle la clave k , le vamos a otorgar un oráculo que calcula $\text{Enc}(k, \cdot)$.

PREGUNTA: Esto podría parecer insólito: como podría $\mathcal{A}$ acceder a $\text{Enc}(k, \cdot)$? Ideas?

- Los mensajes podrían ser previsibles, dándole a $\mathcal{A}$ m y $\text{Enc}(k, m)$.
- $\mathcal{A}$ podría tener control del input directamente, por ejemplo si es un comerciante que introduce en el POS el valor a cobrar, y quiere usar $(m, \text{Enc}(k, m))$ para aprender la clave k .
- $\mathcal{A}$ podría saber que el mensaje cifrado es consecuente a alguna acción propia.

Nos inspiramos a la definición de (ϵ, t) -secrecy (c.f. Definition 12), donde el adversario trata de distinguir $\text{Enc}(k, m_0)$ de $\text{Enc}(k, m_1)$. En este caso imponemos $|m_0| = |m_1|$ dado que toleraremos $\mathcal{M} = \{0, 1\}^*$.

Definición 16 (IND-CPA security). Sea $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ un SKES.

$\text{Exp}^{\text{CPA}}(\mathcal{A}, b)$	$\mathcal{O}(m_0, m_1)$
1: $k \leftarrow \text{Gen}()$	1: if $ m_0 \neq m_1 $
2: $b' \leftarrow \mathcal{A}^{\mathcal{O}}()$	2: return $\perp$
3: return b'	3: $c \leftarrow \text{Enc}(k, m_b)$
	4: return c

Decimos que Π otorga (ϵ, t, q) -indistinguibilidad bajo ataques de mensaje elegido (IND-CPA, “indistinguishability under chosen-plaintext attacks”), si para todo adversario $\mathcal{A}$ que corre en tiempo $\leq t$ y hace $\leq q$ queries a $\mathcal{O}$, su ventaja es

$$\text{Adv}(\text{Exp}^{\text{CPA}}, \mathcal{A}) := \left| \Pr[\text{Exp}^{\text{CPA}}(\mathcal{A}, 0) \Rightarrow 1] - \Pr[\text{Exp}^{\text{CPA}}(\mathcal{A}, 1) \Rightarrow 1] \right| \leq \epsilon.$$

Comentario 27. Las definiciones de (ε, t) -secret y $(\varepsilon, t, 1)$ -IND-CPA son equivalentes, por lo que (ε, t) -secrety a veces se lo llama (ε, t) -OT-IND-CPA.

Comentario 28. Existen definiciones “intermedias” entre (ε, t) -secrety y (ε, t, q) -IND-CPA. Como referencia, vean [KL20] o [BS23].

Ejemplo 2.

- El OTP no otorga $(\varepsilon, t, 2)$ -IND-CPA, dado que $\mathcal{A}$ puede pedir $\mathcal{O}(0^n, 0^n) \Rightarrow k$, y obtenida k , puede pedir $\mathcal{O}(0^n, 1^n) \Rightarrow b^n \oplus k$, y así recuperar b .
- Ningún SKES donde $\text{Enc}(k, \cdot)$ es determinista puede otorgar $(\varepsilon, t, 2)$ -IND-CPA! En cuanto el siguiente ataque aplica:

1. $c_1 \leftarrow \mathcal{O}(0^n, 0^n)$
2. $c_2 \leftarrow \mathcal{O}(0^n, 1^n)$
3. Si $c_1 = c_2$, entonces $b = 0$. Si $c_1 \neq c_2$, entonces $b = 1$.

Comentario 29 (SKE OT-IND-CPA $\not\Rightarrow$ SKE $(q \geq 2)$ -IND-CPA). Como vimos en Example 2, es fácil identificar algoritmos de SKE que no otorgan seguridad IND-CPA cuando mas de 1 query es otorgada al adversario. De otro lado, el OTP puede otorgar (ε, t) -secrety, o sea “OT-IND-CPA” (c.f. Remark 27), o $(\varepsilon, t, q = 1)$ -IND-CPA. Esto quiere decir que en el caso de cifrado simétrico, o SKE,

$$(\varepsilon, t, q = 1)\text{-IND-CPA} \not\Rightarrow (\varepsilon', t'q' > 1)\text{-IND-CPA con } t'/\varepsilon' \approx t/\varepsilon.$$

Definición 17 (PRF-CTR). Sea $F: \mathcal{K} \times \mathcal{D} \rightarrow \mathcal{R}$ una (ε, t, q) -PRF con $\mathcal{D} = \{0, 1\}^d$ y $\mathcal{R} = \{0, 1\}^r$. Definimos un SKES $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ con $\mathcal{M} = \{0, 1\}^{\ell \cdot r}$ y $\mathcal{C} = \{0, 1\}^{\ell \cdot r + d}$, para algún $\ell \leq 2^d/2$.

Gen()	Enc(k, m)	Dec($k, c = (c_0, c_1, \dots, c_\ell)$)
1: $k \xleftarrow{\$} \mathcal{K}$	1: $m_1 \dots m_\ell \leftarrow m$, donde $ m_i = r$	1: for $i = 1, \dots, \ell$:
2: return k	2: $c_0 \xleftarrow{\$} [0, 2^d)$, equiv. $\{0, 1\}^d$	2: $y_i \leftarrow F(k, c_0 + i - 1 \bmod 2^d)$
	3: for $i = 1, \dots, \ell$:	3: $m'_i \leftarrow y_i \oplus c_i$
	4: $y_i \leftarrow F(k, c_0 + i - 1 \bmod 2^d)$	4: $m' \leftarrow m'_1 \dots m'_\ell$
	5: $c_i \leftarrow y_i \oplus m_i$	5: return m'
	6: $c \leftarrow (c_0, c_1, \dots, c_\ell)$	
	7: return c	

Si usamos una PRP/cifrado de bloques en lugar de F , Π es el modo de cifrado “counter mode” (CTR).

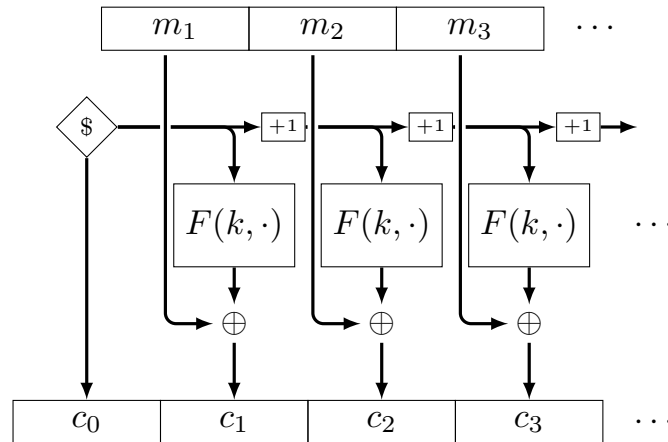


Figure 6: Diagrama de Enc en PRF-CTR. Imagen adaptada de [Ros21, Construction 8.3], usada con permiso del autor.

Comentario 30. El PRF-CTR puede ser paralelizado de manera simple, dado que los y_i pueden ser calculados de manera independiente. Por lo tanto tiene muy buena eficiencia al cifrar mensajes muy largos.

Comentario 31. *Para descifrar, no es necesario “invertir” F . Esto sugiere que si m es largo entre $(\ell - 1) \cdot r + 1$ y $\ell \cdot r$, se pueden “cortar” los últimos bits de y_n en Enc , resultando en un cifrado mas corto.*