¿Cómo hacer demostraciones de seguridad?

Pensamos en dos "actores", el adversario A y un simulador S (que conoce todas las claves y simula, para el adversario, una ejecución real).

Planteamos un JUEGO. Si la probabilidad de que el adversario gane es despreciable (en el parámetro de seguridad) $\Rightarrow$ el esquema queda demostrado seguro.

A este nivel...

+ Basta con:

  → que entendáis la idea detrás de este tipo de demostración

  → que sepáis ver A TRAVÉS DE LOS JUEGOS cuando un esquema es claramente INSEGURO.

  ¡→ No espero que aprendáis a hacer demostraciones completas de seguridad

[Los Juegos]

CASO CPA — $\mathcal{A}$ tiene pk y conoce la descripción de los tres algoritmos que atacan, (K, E, D)

OW

1. $\mathcal{J}$ → ejecuta K y le pasa pk al adversario $\mathcal{A}$. $\mathcal{J}$ cifra m al azar. Pasa ese cifrado c^* a $\mathcal{A}$.
2. $\mathcal{A}$ → hace cálculos, pasa un valor m^* como respuesta a $\mathcal{J}$.

$\mathcal{A}$ gana si $m = m^*$.

IND

1. $\mathcal{J}$ ejecuta K y le pasa pk al adversario $\mathcal{A}$
2. $\mathcal{A}$ elige dos mensajes, m_0 y m_1, y los envía a $\mathcal{J}$
3. $\mathcal{J}$ "tira una moneda", es decir, elige un bit $b^* \in \{0,1\}$ al azar, y cifra m_{b^*}. Llamamos c^* al resultado (reto), que $\mathcal{J}$ le pasa a $\mathcal{A}$
4. $\mathcal{A}$ tiene que decidir si c^* es un cifrado de m_0 ó de m_1. Decide que es de m_b y da b como salida

Sabemos que, siempre $P[b = b^*] = 1/2$ ($\mathcal{A}$ puede acertar por azar)

$\mathcal{A}$ gana si $\underbrace{|P[b = b^*] - 1/2|}_{\varepsilon}$ no es despreciable

↑ a esto se le llama ventaja de $\mathcal{A}$

NM

1. S ejecuta K y le pasa pk a A

2. S cifra t mensajes $m_1^*, \dots, m_t^*$, obteniendo $c_1^*, \dots, c_t^*$ que le pasa a A.

3. A intenta construir c^* y R, de modo que
   - c^* sea cifrado valido de un texto m^*
   - $R(m^*, m_1^*, \dots, m_t^*) = 1$.

es decir, A construye un cifrado válido de un texto m^*, que se relaciona con $m_1^*, \dots, m_t^*$ con una fórmula R que él conoce.

A gana si, efectivamente, c^* es un cifrado válido de un cierto m^* con $R(m^*, m_1^*, \dots, m_t^*) = 1$

CASO CCA

Los juegos son iguales solo que ahora $\mathcal{A}$, además de tener toda la información pública, tiene acceso a un oráculo de descifrado $\mathcal{O}_D$.

$\mathcal{O}_D$ recibe valores de entrada y devuelve:

→ su texto claro correspondiente, si la entrada es un cifrado correcto

→ $\perp$ si la entrada no es un cifrado correcto.

Hay dos escenarios:

CCA_1 → Sólo le dejamos $\mathcal{O}_D$ al adversario ANTES de conocer los retos, que son:

$$\begin{cases} OW \rightarrow c^* \\ IND \rightarrow c^* \\ NM \rightarrow c_1^*, \dots c_t^* \end{cases}$$

CCA_2 → $\mathcal{A}$ tiene $\mathcal{O}_D$ <u>todo el rato</u> pero no le dejamos llamar con los retos, es decir

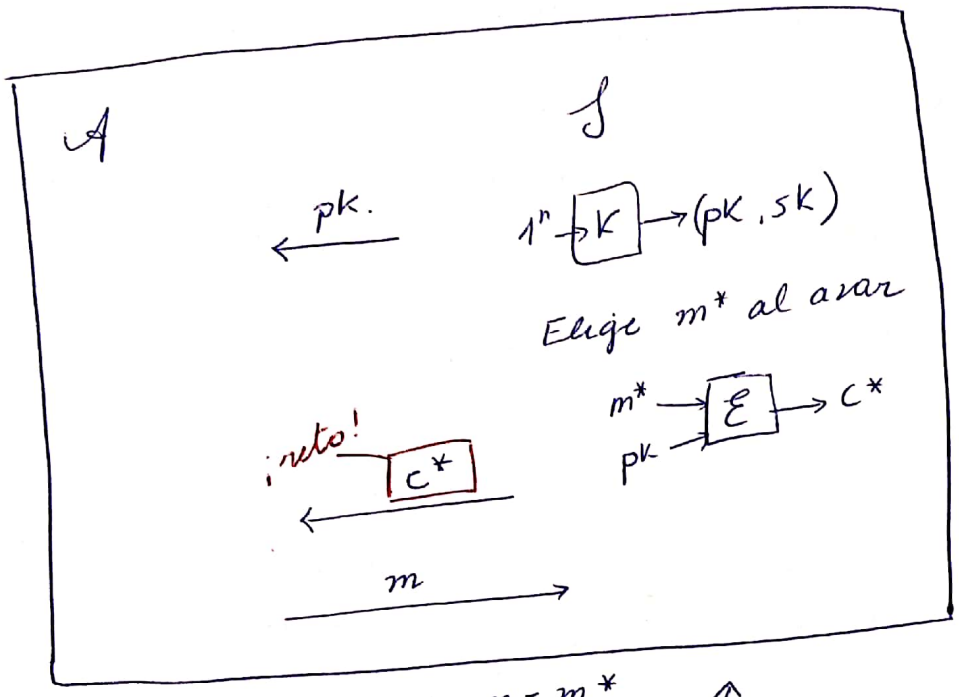
<u>IND|OW</u> $\quad c^* \rightarrow \mathcal{O}_D \rightarrow \perp$

<u>NM</u> $\quad c_i^* \rightarrow \mathcal{O}_D \rightarrow \perp$

$(i = 1, \dots, t)$.

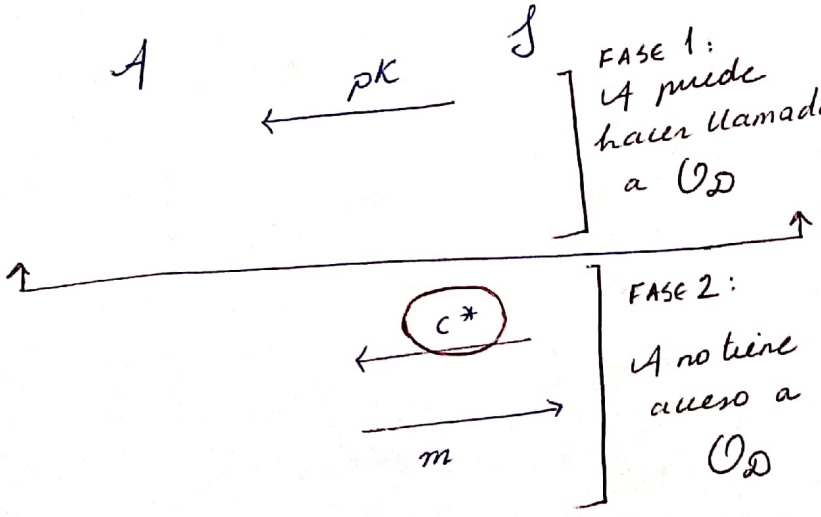
[JUEGOS DE SEGURIDAD - DIAGRAMAS ESQUEMÁTICOS]

OW-Seguridad CPA



A gana si $m = m^*$

OW - CCA1 (variante del juego anterior)

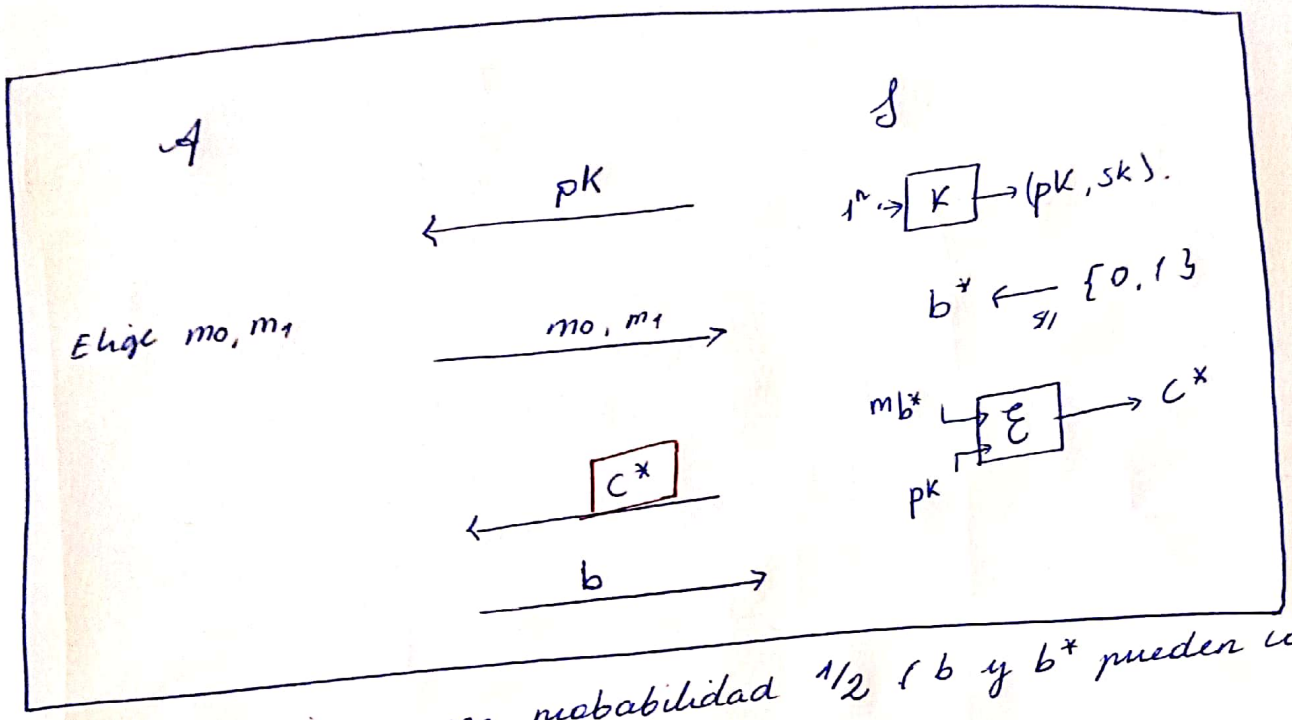


OW - CCA2

A tiene acceso a O_D durante todo el ataque, ¡pero! O_D no acepta c^* como entrada

$c^* \rightarrow O_D \rightarrow \perp$

IND - Seguridad CPA

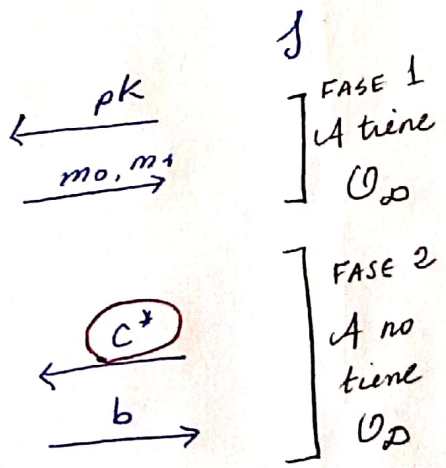


A acierta siempre con probabilidad 1/2 (b y b^* pueden coincidir por azar), pero A gana si

$$|P[b = b^*] - 1/2|$$ es NO DESPRECIABLE

Suele escribirse $\varepsilon(n) = |P[b=b^*] - 1/2|$, y llamar a ε "función ventaja del adversario". Recordad que n es el parámetro de seguridad.

IND-CCA1

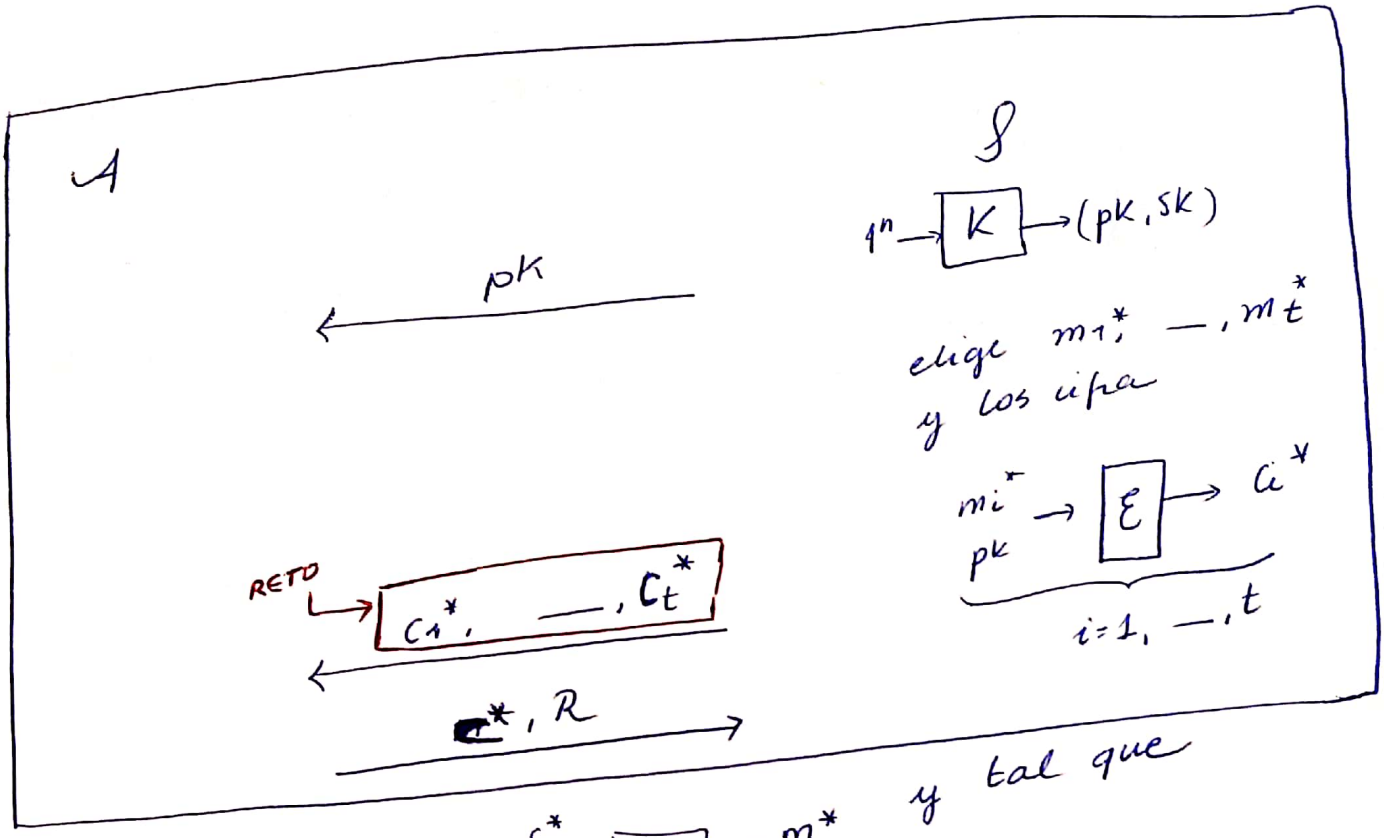


IND-CCA2

A tiene acceso a O_D durante todo el ataque pero no puede llamar a O_D con c^*

$c^* \rightarrow O_D \rightarrow \perp$

NM - Seguridad CPA

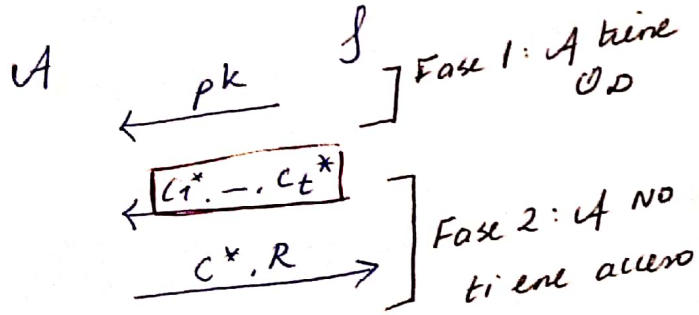


A gana si $\exists m^*$ con $c^* \to \boxed{D} \to m^*$ (sk) y tal que

$R(m^*, m_1, \dots, m_t) = 1$.

(¡ojo! m^* tiene que existir, A tal vez ni lo conozca)

NM-CCA1



NM-CCA2

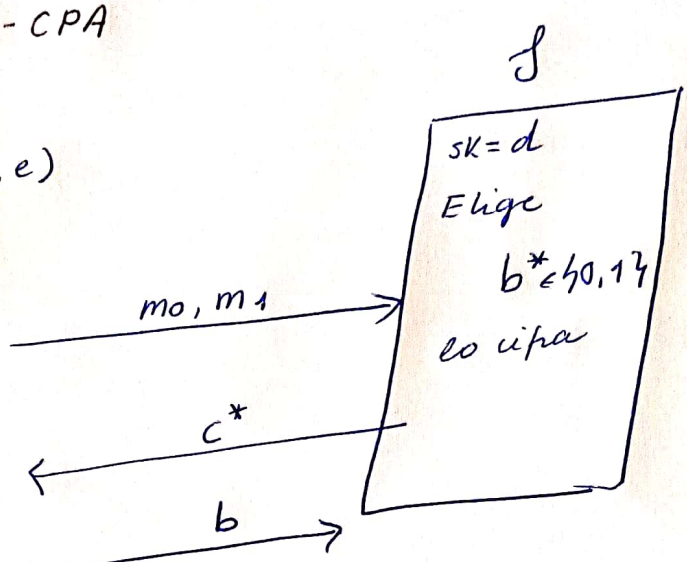
A tiene acceso a O_D durante todo el ataque

¡pero! no acepta c_i^* $(i = 1, \dots, t)$

$c_i^* \to O_D \to \perp$

[Ejemplos]

1. RSA no es IND-CPA



- A cifra m_0 y m_1, compara el resultado
  - $c^* \stackrel{?}{=} m_0^e$
  - $c^* \stackrel{?}{=} m_1^e$

Si, por ejemplo,

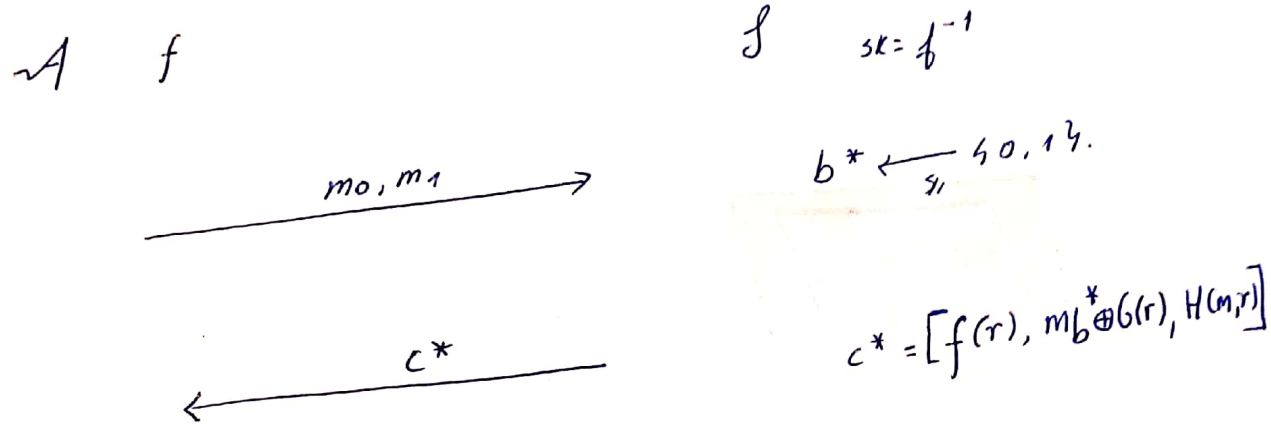
$c^* = m_1^e \bmod n$

da por sentado que $\mathcal{J}$ cifró m_1

A gana SIEMPRE porque RSA de libro de texto es DETERMINISTA, m_0 y m_1 dan siempre el mismo valor (c_0 y c_1) al cifrarse.

OBS: Un esquema de cifrado determinista NUNCA es IND

2. Bellare - Rogaway es IND-CPA.



A no tiene ni idea de si c^* viene de m_0 o de m_1.

si cifra él mismo m_0 obtiene

$$(f(\bar{r}),\ m_0 \oplus G(\bar{r}),\ H(m_0, \bar{r}))$$

si cifra luego m_1 le sale

$$(f(\hat{r}),\ m_1 \oplus G(\hat{r}),\ H(m_1, \hat{r}))$$

Cada vez que se hace una llamada al algoritmo $\mathcal{E}$ de cifrado sale un valor "r" distinto, luego las salidas son todas independientes

Así, $P[b^* = b] = 1/2$, pues A no puede hacer nada mejor que tirar una moneda y apostar al azar ...

↕ salida de A