Clase 3: seguridad computacional y pseudoaleatoriedad

Fernando Virdia, versión: 0.0.1, junio 2024

Ahora vamos a ver relajaciones de la noción de *secreto perfecto*, y vamos a usar un nuevo postulado para construir algoritmos de cifrado mas eficientes del OTP.

Es común en criptografía *relajar* las definiciones de seguridad, en cambio de eficiencia. Vemos un ejemplo.

Definición 6 (One-time ε-secrecy) *Decimos que un SKES tiene* one-time ε-secrecy *(OT ε-secrecy) si dada la variable aleatoria $k \sim \mathsf{Gen}()$,*

$$\forall m_0, m_1 \in \mathcal{M}, c \in \mathcal{C}, \quad |\Pr[\mathsf{Enc}(k, m_0) = c] - \Pr[\mathsf{Enc}(k, m_1) = c]| \leq \varepsilon.$$

Comentario 8 *Un SKES con OT ε-secrecy con $\varepsilon = 0$ otorga secreto perfecto.*

EJERCICIO: [BS23, Exercise 2.5]: OT ε-secrecy $\implies |\mathcal{K}| \geq (1 - \varepsilon)|\mathcal{M}|$

Comentario 9 *Lo mas cercano ε a 0, lo mas cercano OT ε-secrecy al secreto perfecto, y lo mas cercano lo que un adversario $\mathcal{A}$ puede hacer dado un cifrado salido de un SKES con OT ε-secrecy a lo que puede hacer con un cifrado salido de un cifrado con secreto perfecto.*

En la practica, tratamos que $\varepsilon \in [2^{-256}, 2^{-128}]$. Si $\varepsilon < 2^{-\lambda}$, decimos que un SKES ofrece λ-bits de seguridad estadística.

Comentario 10 *Por que "one-time"? Supongamos que Alicia quiere comunicar con Bob, y comparte una clave de OTP k muestreada $k \sim \mathsf{Gen}()$. Alicia y Bob deciden de cifrar dos mensajes usando la misma clave:*

$$\begin{aligned} c_1 &= k \oplus m_1 \\ c_2 &= k \oplus m_2. \end{aligned}$$

Si un adversario $\mathcal{A}$ observa los cifrados c_1 y c_2 en transito, que puede aprender sobre m_1 y m_2?

$$\begin{aligned} c_1 \oplus c_2 &= (k \oplus m_1) \oplus (k \oplus m_2) \\ &= m_1 \oplus m_2. \end{aligned}$$

Esto es un problema! Por ejemplo, si m_1 es un mensaje predecible, digamos $\Pr[m_1 = \overline{m_1}] \approx 1$, $\mathcal{A}$ puede adivinar, y determinar que $m_2 = c_1 \oplus c_2 \oplus \overline{m_1}$ con probabilidad ≈ 1.

El OTP otorga seguridad si la llave ("pad") k se usa una sola vez ("one-time"). No garantiza nada si la llave se la usa mas de una vez.

Comentario 11 *OT ε-secrecy solo garantiza* confidencialidad. *No otorga garantías de integridad. Si Alicia usa un OTP para comunicarle a su banco que quiere pagar $m = \$4 = \$100_2$ a un negociante, y este intercepta el cifrado*

$$\begin{aligned} c = c_0 c_1 \dots c_9 = k_0 \ k_1 \dots k_6 \ k_7 \ k_8 \ k_9 \\ \oplus \ 0 \ \ 0 \ \dots \ 0 \ \ 1 \ \ 0 \ \ 0, \end{aligned}$$

lo intercepta y transmite $c' = c \oplus (10 \dots 0)$ al banco, este decifraría

$$m' = k \oplus c' = \$(1000000100)_2 = \$\ 2^9 + 2^2 = \$516,$$

costandole a Alicia mucho mas.

Hemos visto como se puede relajar una definición de seguridad. Ahora vamos a introducir un nuevo postulado.

3 PRGs, PRFs, PRPs

Ahora vamos a definir 3 familias de objetos criptográficos que vamos a utilizar para construir mejores cifrados.

Definición 7 (PRG) *Sea* $G\colon \{0,1\}^\ell \to \{0,1\}^m$ *una función, implementable como un algoritmo determinista eficiente. Generalmente* $\ell < m$*. Dado un adversario aleatorio* $\mathcal{A}$ *y un bit* $b \in \{0,1\}$*, definimos el siguiente experimento* $\mathsf{Exp}^{\mathrm{PRG}}(\mathcal{A}, b)$

$\mathsf{Exp}^{\mathrm{PRG}}(\mathcal{A},0)$	$\mathsf{Exp}^{\mathrm{PRG}}(\mathcal{A},1)$
1 : $s \xleftarrow{\$} \{0,1\}^\ell$	1 : $y \xleftarrow{\$} \{0,1\}^m$
2 : $b' \leftarrow \mathcal{A}(G(s))$	2 : $b' \leftarrow \mathcal{A}(y)$
3 : **return** b'	3 : **return** b'

Decimos que G *es un generador* (ε, t)*-pseudoaleatorio (PRG, "pseudorandom generator"), si para todo adversario* $\mathcal{A}$ *que corre en tiempo* $\leq t$*, la ventaja de* $\mathcal{A}$ *es*

$$\mathsf{Adv}(\mathsf{Exp}^{\mathrm{PRG}}, \mathcal{A}) := \left|\Pr[\mathsf{Exp}^{\mathrm{PRG}}(\mathcal{A},0) \Rightarrow 1] - \Pr[\mathsf{Exp}^{\mathrm{PRG}}(\mathcal{A},1) \Rightarrow 1]\right| \leq \varepsilon.$$