Criptografía y Seguridad Computacional **2016-01**

Clase 3: 16/03/2016

Profesor: Fernando Krell *Notas: Geraldine Monsalve*

Notación

- PPT: Un algoritmo PPT es un algoritmo que corre en tiempo polinomial en el largo del input y que puede tomar decisiones aleatorias. PPT es en inglés *Probabilistic Polinomial Time*.
- PRG: Sigla para describir un generador pseudo-aleatorio. En inglés *Pseudo-Random Generators*.
- PRF: Sigla para describir una función pseudo-aleatoria. En inglés *Pseudo-Random Function*.
- U_n: Distribución uniforme sobre *strings* de n bits ($\{0,1\}^n$).
- $x \sim \mathcal{S}$: Elemento x es escogido de acuerdo a la distribución $\mathcal{S}$.
- $\mathcal{F}_n^m$: Conjunto de todas las funciones de $\{0,1\}^n$ a $\{0,1\}^m$.
- $\exists$, $\exists!$, $\nexists$ y $\forall$: Existe al menos uno, existe un único, no existe y para todo.
- $\mathsf{negl}(\cdot)$: Función negligible (asintóticamente menor al inverso de cualquier polinomio),

1. Pseudoaleatoriedad

Así como la indistinguibilidad computacional de cifrados es una relajación computacional de la seguridad perfecta de esquemas, la pseudoaleatoriedad lo es de la aleatoriedad. Informalmente podemos decir que la pseudoaleatoriedad hace referencia a una distribución sobre strings de un cierto largo, tal que es *computacionalmente* indistinguible a la distribución uniforme sobre strings del mismo largo. Formalizaremos esta noción con la definición de un generador pseudoaleatorio.

Definición 1. *Sea $\ell(\cdot)$ un polinomio y G un algoritmo de tiempo polinomial determinístico, tal que sobre cada input uniformemente aleatorio $s \in \{0,1\}^n$ entrega un string de largo $\ell(n)$. Diremos que G es un* ***generador pseudoaleatorio*** *si se cumple:*

1. *Expansión: Para cada n, se cumple $l(n) > n$.*
2. *Pseudoaleatoriedad: Para todo algoritmo PPT (probabilistic polinomial-time) distinguidor $\mathcal{D}$, existe una función negligible $\mathsf{negl}(\cdot)$, tal que:*

$$|\Pr[D(r) = 1] - \Pr[D(G(s)) = 1]| \leq \mathsf{negl}(n)$$

con $r \in \{0,1\}^{\ell(n)}$, $s \in \{0,1\}^n$ elegidos uniformemente.

Ejemplo 2. *Sea $G : \{0,1\}^n \to \{0,1\}^{n+1}$, tal que $G(x) = x||\bigoplus_{i=0}^{n}(x_i)$. ¿Es G un generador pseudoaleatorio?*
Consideremos el distinguidor $D(y)$ que toma un string de largo $n+1$:

1. *Calcula $b = \bigoplus_{i=0}^{n-1}(y_i)$*
2. *Entrega como output 1 si $b = y_n$ y 0 en el caso contrario.*

Si le entregamos un string aleatorio y a D, $P[D(r) = 1] = \frac{1}{2}$. Por otro lado, si le entregamos un string y generado por G, $P[D(G(x))] = 1$, ya que por construcción de G siempre se cumplirá la condición $\bigoplus_{i=0}^{n-2}(y_i) = y_{n-1}$. Luego:

$$|P[D(r) = 1] - P[D(G(s)) = 1]| = \tfrac{1}{2} \gg \mathsf{negl}(n)$$

Concluimos entonces que G no es un generador pseudoaleatorio.

Consideremos la cantidad de strings que pueden ser generados por G. Un generador pseudoaleatorio $G : \{0,1\}^n \to \{0,1\}^{l(n)}$, puede generar 2^n elementos diferentes, mientras que $\{0,1\}^n$ tiene $2^{l(n)}$ elementos. Luego $\Pr[x \in G()] = \frac{2^n}{2^{\ell(n)}}$.
En el caso particular $\ell(x) = 2x$, tendremos que $P[x \in G()] = \frac{1}{2^n}$.

¿Cómo construir generadores? Lamentablemente, no está demostrado que existan. Sin embargo, la comunidad científica está bastante convencida de su existencia. En particular, si se demuestra la existencia de funciones unidireccionales (funciones fáciles de computar, pero difíciles de invertir), entonces se asegura la existencia de estos generadores. La multiplicación de números primos es un candidato fuerte a ser una función unidireccional. En la práctica se utilizan algoritmos heurísticos para implementar generadores pseudoaleatorios (RC4 es uno de ellos).

2. Construcción basada en generadores pseudo-aleatorios

Sea G un generador pseudoaleatorio con factor de expansión $\ell(n)$. Construiremos un esquema de cifrado de llave privada $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ para mensajes de largo $\ell(n)$ como:

- Gen : input : 1^n, output : $k \sim U_n$ (Uniforme en n bits)
- Enc : input : $k \in \{0,1\}^n, m \in \{0,1\}^{\ell(n)}$, output : $c := G(k) \oplus m$
- Dec : input : $k \in \{0,1\}^n, c \in \{0,1\}^{\ell(n)}$, output : $m := G(k) \oplus c$

Teorema 3. *Si G es un generador pseudoaleatorio, entonces la construcción anterior es un esquema de encriptación de llave privada con cifrados computacionalmente indistinguibles.*

Demostración. Suponemos que la construcción no tiene cifrados indistinguibles, esto es, existe un adversario $\mathcal{A}$, y un polinomio $p(\cdot)$ tales que

$$\Pr[\mathsf{Exp}^{\mathtt{ind}}_{\Pi,A}(n) = \text{“}\mathcal{A} \text{ gana”}] \geq \frac{1}{p(n)}$$

$\mathcal{A}$ puede distinguir entre $G(k) \oplus m_0$ y $G(k) \oplus m_1$ con una probabilidad mayor a cualquier función negligible. Consideremos el siguiente distinguidor $\mathcal{D}$:

1. $m_0, m_1 \leftarrow \mathcal{A}(1^n)$.
2. $b \xleftarrow{\$} \{0,1\}$
3. $b' \leftarrow \mathcal{A}(y \oplus m_b)$
4. output 1 si $b = b'$ y 0 en otro caso.

Analizamos ahora cómo se comporta $\mathcal{D}$ en el caso que y es $G(k)$ para $k \sim U_n$, y en el caso que $y \sim U_{\ell(n)}$

- CASO 1: $y = G(k)$. Tendremos que

$$\Pr[D(y) = 1] = \Pr[\mathsf{Exp}^{\mathtt{ind}}_{\Pi,A}(n) = 1] = 1/2 + \frac{1}{p(n)}$$

- CASO 2: $y \sim U_{\ell(n)}$. Entonces tenemos que $P[D(y) = 1] = \frac{1}{2}$. Ya que $y \oplus m_b$ es uniforme y por lo tanto no revela nada sobre m_b.

Concluimos entonces que $\mathcal{D}$ distingue con ventaja $\frac{1}{p(n)}$, lo cual es una contradicción. Por lo tanto el adversario $\mathcal{A}$ no puede existir y el esquema tiene cifrados computacionalmente indistinguibles. □

$\underline{\mathsf{Exp}^{\mathtt{ind}}_{\Pi,\mathcal{A}}(\lambda)}$**:**

1. $k \leftarrow \mathsf{Gen}(1^\lambda)$
2. $m_0, m_1 \leftarrow \mathcal{A}(1^{\lambda^{\mathsf{Enc}_k()}})$
3. $b \xleftarrow{\$} \{0,1\}$
4. $b' \leftarrow \mathcal{A}(\mathsf{Enc}_k(m_b))^{\mathsf{Enc}_k()}$

output "$\mathcal{A}$ gana" sii $b' = b$

Figura 1: Experimento $\mathsf{Exp}^{\mathtt{ind\text{-}CPA}}_{\Pi,\mathcal{A}}(\lambda)$

3. Ataques de textos plano elegidos (CPA)

Consideremos una nueva versión del experimiento de indistinguibilidad. En este nuevo experimento consideramos adversarios que tienen cierta influencia sobre cuales mensajes serán cifrados. Modelamos seguridad en este escenario dándole al adversario $\mathcal{A}$ el poder para obtener cifrados correspondientes a textos planos de su elección. Formalmente, $\mathcal{A}$ tiene acceso a Enc_k como un oráculo o caja negra, lo que espresamos como $\mathcal{A}^{\mathsf{Enc}_k(\cdot)}$.

Definición 4. *Un esquema de encriptación de llave privada* Π *tiene cifrados indistinguibles bajo ataques de texto plano escogido si para todo adversario PPT* $\mathcal{A}$ *existe una función negligible* negl*, tal que:*

$$\Pr[\mathsf{Exp}^{\mathit{ind\text{-}CPA}}_{\Pi,\mathcal{A}}(\lambda) = \text{“}\mathit{A\ gana}\text{”}\,] \leq \frac{1}{2} + \mathsf{negl}(\lambda)$$

4. Funciones aleatorias y pseudoaleatorias

Consideremos el conjunto $\mathcal{F}^m_n$ de todas las funciones $f : \{0,1\}^n \to \{0,1\}^m$. Para describir completamente cada una de ellas necesitamos $2^n m$ bits. Luego, tenemos $2^{2^n m}$ funciones distintas.
Consideremos ahora las funciones $F : \{0,1\}^\lambda \times \{0,1\}^n \to \{0,1\}^m$. Cada llave $k \in \{0,1\}^\lambda$ define una función, por lo tanto 2^n funciones F_k distintas.
Diremos que F es pseudoaleatoria si:

1. **Eficiencia**. Existe un algoritmo que dado una llave k, compute $F(k,x)$ en tiempo polinomial.

2. **Pseudo-aleatoriedad** Para todo PPT $\mathcal{D}$, existe una función negligible $negl$, tal que:
$$|\Pr[\mathcal{D}^{F_k(\cdot)}(1^\lambda) = 1] - P[\mathcal{D}^{f(\cdot)}(1^\lambda) = 1]| \leq \mathsf{negl}(\lambda)$$
Donde f es elegida uniformemente en $\mathcal{F}_n^m$ y $k \sim U_\lambda$. El algoritmo distinguidor $\mathcal{D}$ tiene acceso a un oráculo que puede ser f o F_k.

Ejemplo 5. *Sea F una función con llave tal que $F_k(x) := k \oplus x$. ¿Es F una función pseudo-aleatoria?. No. Un algoritmo distinguidor con oráculo $O(\cdot)$ puede consultar $y_1 = O(x_1), y_2 = O(x_2)$, para $x_1 \neq x_2$. Si $y_1 \oplus y_2 = x_1 \oplus x_2$ output 1, en otro caso output 0. Si $O(\cdot)$ es $F_k(\cdot)$, entonces $y_1 \oplus y_2 = x_1 \oplus x_2$ siempre. En cambio si $O(\cdot)$ es una función aleatoria, entonces y_1 e y_2 son independientes de los inputs x_1 y x_2. Por lo tanto, $y_1 \oplus y_2 = x_1 \oplus x_2$ con probabilidad negligible.*

Ejemplo 6. *Sea $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ una función pseudo-aleatoria. ¿Es necesariamente $F' : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$, definida como $F'_k(x) = F_x(k)$, una función pseudo-aleatoria? No. Puede que exista una llave especifica (por ejemplo $k = 0^n$) para la cual la función F_k no luzca para nada aleatoria. Por ejemplo $F_0(x) = 0$ para todo input x. Sin embargo, la función F sigue siendo pseudo-aleatoria, ya que la llave 0^n es elegida con probabilidad negligible 2^{-n}. Por lo tanto, un algoritmo distinguidor puede consultar su oráculo con input 0^n. Si el oráculo es F'_k, el resultado sería $F_{0^n}(k) = 0^n$.*

Es importante notar que la existencia de funciones pseudoaleatorias, implica la existencia de generadores pseudoaleatorios. Se deja como ejercicio propuesto la demostración. *Hint* : Considerar el generador $G(k) = F_k(1)||F_k(2)||...||F_k(n)$.

5. Construcción esquema CPA via funciones pseudo-aleatorias

Sea F una función pseudoaleatoria de $\{0,1\}^\lambda$ en $\{0,1\}^n$. Definimos un esquema de encriptación de llave privada para mensajes de largo n, como:

- Gen. input : 1^λ, output : $k \sim U_\lambda$ (Uniforme en n bits)
- Enc. input : $k \in \{0,1\}^\lambda, m \in \{0,1\}^n$, elige $r \sim U_\lambda$, output : $\langle r, F_k(r) \oplus m \rangle$
- Dec. input : $k \in \{0,1\}^n, c = \langle r, s \rangle \in \{0,1\}^n$, output : $m := F_k(r) \oplus s$

Intuitivamente, el esquema es seguro debido a que si F es pseudo -aleatoria el valor de $s = m \oplus F_k(r)$ es indistinguible de unifome, aún sabiendo r. El único problema ocurre cuando el mismo r se usa para 2 mensajes m_1, m_2, en donde el adversario puede obtener $m_1 \oplus m_2$. Sin embargo, r es elegido uniformemente en 2^λ por lo que la probabilidad de que r se repita es negligible en λ.

Teorema 7. *Si F es una función pseudoaleatoria, entonces la construcción anterior es un esquema de llave privada y tamaño fijo es CPA seguro.*

Recordemos que un esquema es CPA seguro, si tiene cifrados indistinguibles bajo ataques de texto planos elegidos.

Demostración. Probaremos el teorema por contradicción. Supondremos que el esquema construido **no** es CPA seguro y como consecuencia la función F no es pseudoaleatoria, lo que representa una contradicción. Si el esquema no es CPA seguro, entonces existe un adversario $\mathcal{A}$ que ganada el experimento $\mathsf{Exp}^{\texttt{ind-cpa}}$ con probabilidad $1/2 + 1/\text{poly}(\lambda)$. Recordemos que el adversario en este experimento tiene acceso a un oráculo de encriptación $\mathsf{Enc}_k(\cdot)$. Demostraremos entonces que existe un algoritmo distinguidor $\mathcal{D}$ con acceso a un oráculo $O(\cdot)$ que puede distinguir si $O()$ corresponde a una función 100 % aleatoria, o si corresponde a la función F con llave aleatoria k.

La idea es que el distinguidor $\mathcal{D}^{O()}$ use al adversario $\mathcal{A}$ en una simulación del experimento $\mathsf{Exp}^{\texttt{ind-cpa}}$. Para ello, $\mathcal{D}$ tiene que proveer a $\mathcal{A}$ de un oráculo de encriptación. El oráculo es simulado por $\mathcal{D}$ de la siguiente manera.

$\mathsf{Enc}'(m)$: con input m, elegimos $r \sim U_\lambda$ y devolvemos $\langle r, m \oplus O(r)\rangle$.

Si $O()$ es en realidad F_k entonces Enc' es igual a Enc_k. En cambio si $O()$ es una función aleatoria entonces Enc' devuelve *strings* aleatorios (asumiendo que r no se repite). El algoritmo distinguidor ejecuta lo siguiente:

1. $m_0, m_1 \leftarrow \mathcal{A}(1^n)^{\mathsf{Enc}'()}$
2. $b \xleftarrow{\$} \{0,1\}$
3. $b' \leftarrow \mathcal{A}(\langle r^*, O(r^*) \oplus m_b\rangle)^{\mathsf{Enc}'()}$
4. $\mathsf{output} = 1$ si $b = b$‘, O en otro caso.

Si $O()$ es $F_k()$, entonces $\mathcal{D}$ simula exactamente $\mathsf{Exp}^{\texttt{ind-CPA}}_{\Pi,\mathcal{A}}(n)$.

$$\Pr[D^{F_K()}(1^\lambda) = 1] = \Pr[\mathsf{Exp}^{\texttt{ind-CPA}}_{\Pi,\mathcal{A}}(\lambda) = 1]$$

El otro caso es que $O()$ es una función aleatoria $f()$. Sea R el evento en el que el valor r^*, usado para cifrar m_b, haya sido usado por Enc' en alguna de sus invocaciones.

$$\begin{aligned}\Pr[\mathcal{D}^{f(\cdot)}(1^\lambda)=1] &= \Pr[\mathcal{D}^{f(\cdot)}(1^\lambda)=1 \wedge R] + \Pr[\mathcal{D}^{f(\cdot)}(1^\lambda)=1 \wedge \neg R] \\ &\leq \Pr[R] + \Pr[\mathcal{D}^{f(\cdot)}(1^\lambda)=1|\neg R]\Pr[\neg R] \\ &\leq \frac{q(\lambda)}{2^\lambda} + \Pr[\mathcal{D}^{f(\cdot)}(1^\lambda)=1|\neg R] \\ &\leq \mathsf{negl}(\lambda) + \frac{1}{2}\end{aligned}$$

En donde $q(\lambda)$ es la cantidad de llamadas al oráculo $\mathsf{Enc}'()$ realizadas por $\mathcal{A}$. Debido a que $\mathcal{A}$ corre en tiempo polinomial, $q(\lambda) \leq \mathsf{TIME}(\mathcal{A}) \leq \text{poly}(\lambda)$.

Consideremos finalmente la expresión

$$|\Pr[\mathcal{D}^{F_k()}(1^\lambda)=1] - \Pr[\mathcal{D}^{f()}(1^\lambda)=1]| \leq \mathsf{negl}(\lambda)$$

Donde $k \sim U$, f uniforme en $\mathcal{F}$: conjunto de todas las funciones posibles.

Si Π no es seguro $\Pr[\mathcal{D}^{F_k()}(1^\lambda)=1] \geq \frac{1}{2} + \text{poly}(\lambda)$, $\mathcal{D}$ distingue con ventaja y luego

$$\Pr[\mathcal{D}^{F_k()}(1^\lambda)=1] - \Pr[\mathcal{D}^{f()}(1^\lambda)=1]| = |(\frac{1}{2} + \text{poly}(\lambda)) - (\frac{1}{2} + \frac{q(\lambda)}{2^\lambda})| = \text{poly}(\lambda)$$

Y lo anterior implica que F no es una función pseudoaleatoria, lo que es una contradicción. □