Seguridad de Generadores Pseudoaletorios

Alan Reyes-Figueroa

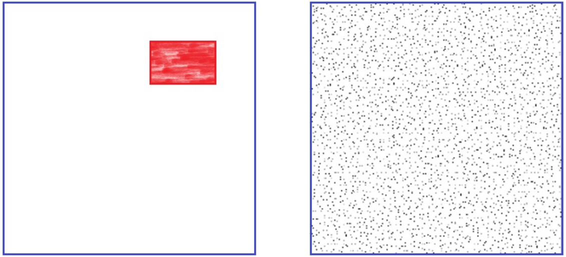
Criptografía y Cifrado de Información (Aula 07) 10.agosto.2021

Seguridad de PRGs

Queremos definir cuándo un generador pseudo-aleatorio (PRG) $G : \mathcal{K} \to \{0,1\}^n$ es seguro, en el sentido que el *output* $G(\mathbf{k})$ sea indistiguinble de algo completamente aleatorio.

Una forma de enunciar esto es

$$\text{distribución del output } G(\mathbf{k}) \overset{d}{=} \text{distribución uniforme en } \{0,1\}^n.$$

$\{0,1\}^n$  $\{0,1\}^n$

Para estudiar esa "uniformidad", usaremos tests estadísticos.

Test Estadísticos

Podemos entender un test estadístico que opera sobre cadenas de bits de longitud n, como un algoritmo o función $A : \{0,1\}^n \to \{0,1\}$, que devuelve una salida booleana, en función de algún criterio.

Ejemplo: Diferencia estadística entre 0's y 1's.

$$A(\mathbf{x}) = \begin{cases} 1, & |\#0(\mathbf{x}) - \#1(\mathbf{x})| < 10\sqrt{n}; \\ 0, & \text{en otro caso.} \end{cases}$$

Ejemplo: Uniformidad de los bigramas 00, 01, 10 y 11.

$$A(\mathbf{x}) = \begin{cases} 1, & |\#00(\mathbf{x}) - \frac{n}{4}| < 10\sqrt{n}; \\ 0, & \text{en otro caso.} \end{cases}$$

$$A(\mathbf{x}) = \begin{cases} 1, & |\#00(\mathbf{x}) - \frac{n}{4}|, |\#01(\mathbf{x}) - \frac{n}{4}|, |\#10(\mathbf{x}) - \frac{n}{4}|, |\#11(\mathbf{x}) - \frac{n}{4}| < 10\sqrt{n}; \\ 0, & \text{en otro caso.} \end{cases}$$

Ejemplo: Mayor secuencia de 0's (*longest run*).

$$A(\mathbf{x}) = 1 \iff \mathit{len}(\text{mayor cadena de ceros en } \mathbf{x}) < 10 \log_2 n.$$

Test Estadísticos

Cuidado!
Por ejemplo, para el test de la mayor cadena de ceros, la cadena $\mathbf{x} = 111\ldots111$ de unos siempre pasa el test, y no es aleatoria.

Obs! Un test estadístico no tiene por qué hacer las cosas de manera correcta.
Lo que se espera es que en la mayoría de los casos, haga un trabajo correcto: logre diferenciar la mayor parte de las cadenas que parecen aleatorias, de las que no parecen aleatorias.

Existen muchos test estadísticos:

- Contar probabilidades de bigramas, trigramas, ...
- Contar probabilidades de subcadenas en un bloques específico de tamaño k.
- Contar palabras o subcadenas faltantes de longitud k.
- Construir una matriz binaria con la cadena, y hallar la distribución de rank, $\det$, ...
- Contar espacios entre 0's o entre 1's.
- Contar máximos *runs* ascendentes o descendentes.

Seguridad de Test Estadísticos

Pregunta: ¿Cómo evaluar si un test estadístico A es bueno o no?

Definición

*Sea $G : \mathcal{K} \to \{0,1\}^n$ un PRG, y sea $A : \{0,1\}^n \to \{0,1\}$ un test estadístico. La **ventaja** de A respecto de G es*

$$\text{Adv}(A, G) = \left|\mathbb{P}_{\mathcal{K}}[A(G(\mathbf{k})) = 1] - \mathbb{P}_{\{0,1\}^n}[A(\mathbf{x}) = 1]\right|.$$

Observaciones:

- $\text{Adv}(A; G)$ debe entenderse como la diferencia entre la distribución de que las cadenas generadas por G pasen el test A, contra la distribución de que cadenas aleatorias en $\{0,1\}^n$ pasen el test.
- Adv es una diferencia de probabilidades, luego $\text{Adv} \in [0,1]$.
- Entre más cercano a 0, quiere decir que el test A es incapaz de reconocer la diferencia entre cadenas generadas por G, y cadenas puramente aleatorias.
- Entre más cercano a 1, A reconoce de forma satisfactoria la diferencia (mayoría).

Seguridad de Test Estadísticos

Ejemplo: (ejemplo muy simple).
Supongamos que $A : \{0,1\}^n \to \{0,1\}$ es un test estadístico que siempre devuelve $A(\mathbf{x}) = 0$, $\forall \mathbf{x}$.

$$\text{Adv}(A, G) = \big|\mathbb{P}_{\mathcal{K}}[A(G(\mathbf{k})) = 1] - \mathbb{P}_{\{0,1\}^n}[A(\mathbf{x}) = 1]\big| = |0 - 0| = 0.$$

Esto quiere decir que A no puede distinguir entre cadenas generadas por G y cadenas aleatorias.

Ejemplo:.
Supongamos ahora que construimos un generador PRG $G : \mathcal{K} \to \{0,1\}^n$ con la siguiente propiedad: $\mathbf{msb}_1(G(\mathbf{k})) = 1$ para $\frac{2}{3}$ de todas las claves $\mathbf{k} \in \mathcal{K}$.

Definimos $A : \{0,1\}^n \to \{0,1\}$ el test estadístico $A(\mathbf{x}) = \mathbf{msb}_1(\mathbf{x})$.

$$\text{Adv}(A, G) = \big|\mathbb{P}_{\mathcal{K}}[A(G(\mathbf{k})) = 1] - \mathbb{P}_{\{0,1\}^n}[A(\mathbf{x}) = 1]\big| = \left|\tfrac{2}{3} - \tfrac{1}{2}\right| = \tfrac{1}{6}.$$

Se suele decir que A *quiebra* el generador G con ventaje de $\frac{1}{6}$.

Definición

Decimos que un generador pseudo-aleatorio $G : \mathcal{K} \to \{0,1\}^n$ es un ***PRG seguro*** *si para todo test estadístico eficiente $A : \{0,1\}^n \to \{0,1\}$, la ventaja de A sobre G es negligible:*

$$\text{Adv}(A,G) < \varepsilon, \qquad \textit{con } \varepsilon \textit{ negligible } \ (e.g.\ \varepsilon < \tfrac{1}{2^{30}}).$$

Obs!

- La definición anterior quiere decir que básicamente, cualquier test estadístico eficiente no va a poder distinguir entre las cadenas generadas por G, y cadenas puramente aleatorias.
- Importante!, la idea es que G es PRG seguro, si ninguna batería de tests eficientes no logra ventaja suficiente. (Dicho de otra forma, G sobrevive a toda batería de test eficientes).
- No es posible demostrar matemáticamente si un generador G es PRG seguro. (Si fuera posible, esto equivaldría a demostrar $P \neq NP$).
- En la práctica, usamos heurísticas (baterías de tests).

Baterías de Tests

Mencionamos algunos conjuntos o baterías de tests más usados en la práctica, para evaluar PRGs.

- Publicación Especial **800-22 de NIST**, que es el estándar de facto en el campo. `https://csrc.nist.gov/projects/random-bit-generation/documentation-and-software`.

  `https://github.com/GINARTeam/NIST-statistical-test` (Implementación en Python).

- **Diehard tests** (1995). Desarrolladas por George Marsaglia. `https://en.wikipedia.org/wiki/Diehard_tests`.

- **TestU01**. Librería de software implementada en ANSI C. `http://simul.iro.umontreal.ca/testu01/tu01.html`

- **Dieharder** (2004). Elaborada por Robert Brown. `https://webhome.phy.duke.edu/~rgb/General/dieharder.php`

Baterías de Tests

Mencionamos algunos de estos tests estadísticos:

Diferencia entre 0's y 1's: Cuenta la diferencia entre el número de 0s y el número de 1s en una cadena $\mathbf{x} \in \{0,1\}^n$.

$$s = |\#0(\mathbf{x}) - \#1(\mathbf{x})|,$$

y luego calcula un p-valor dado por

$$p = 1 - erf\Big(\frac{s}{2\sqrt{n}}\Big),$$

donde $erf(\mathbf{x}) = \dfrac{2}{\sqrt{\pi}} \displaystyle\int_0^{\mathbf{x}} e^{-t^2}\,dt$, es la función de distribución normal estándar.

Al final el test A devuelve 1, si el p-valor obtenido está por encima de un p-valor crítico, por ejemplo el obtenido a partir de un nivel de significancia $\alpha \in (0,1)$.
(por ejemplo $\alpha = 0.05$, o definir un p-valor crítico de 0.01).

Baterías de Tests

Mencionamos algunos de estos tests estadísticos:

Rango de una matriz $m \times k$: Divide la cadena $\mathbf{x} \in \{0,1\}^n$ en bloques de longitud mk, y con cada bloque construye una secuencia de matrices binarias $M_i \in \mathbb{R}^{m\times k}$. Por ejemplo $m = k = 32$ es comúnmente usado.
Luego, se calcula la frecuencia de los rangos $r_i = \operatorname{rank} M_i$, para $r_i = m$, $r_i = m - 1$, y $r_i \leq m - 2$:

$$f_m = |\{M_i : \operatorname{rank} M_i = m\}|, \quad f_{m-1} = |\{M_i : \operatorname{rank} M_i = m-1\}|, \quad f_r = |\{M_i : \operatorname{rank} M_i \leq m-1\}|.$$

Luego compara estas frecuencias con una distribución χ^2, mediante el valor crítico *chisq*, y calcula el p-valor

$$p = e^{-\frac{chisq}{2}}.$$

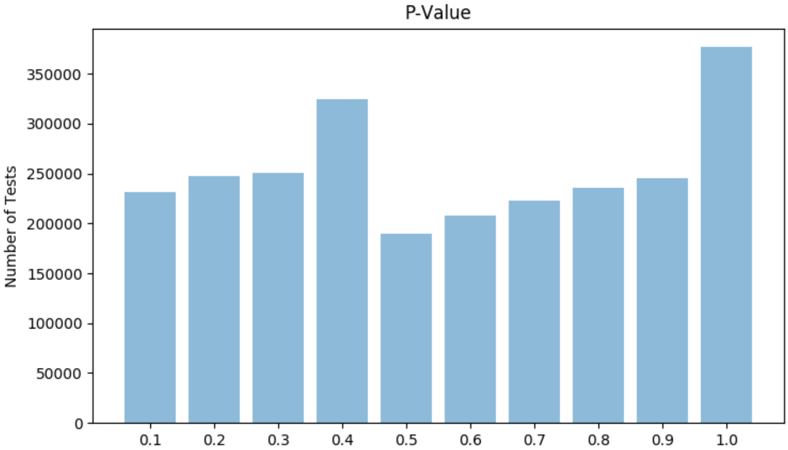
Al final el test A devuelve 1, si el p-valor obtenido está por encima de un p-valor crítico, por ejemplo el obtenido a partir de un nivel de significancia $\alpha \in (0,1)$.
(por ejemplo $\alpha = 0.05$, o definir un p-valor crítico de 0.01).

Baterías de Tests

```
Statistical test summary:

      Test                                             P-value               Conclusion
2.01. Frequency Test:                                  0.9537486285283232    True
2.02. Block Frequency Test:                            0.21107154370164066   True
2.03. Run Test:                                        0.5619168850302545    True
2.04. Run Test (Longest Run of Ones):                  0.7189453298987654    True
2.05. Binary Matrix Rank Test:                         0.3061558375306767    True
2.06. Discrete Fourier Transform (Spectral) Test:      0.8471867050687718    True
2.07. Non-overlapping Template Matching Test:          0.07879013267666338   True
2.08. Overlappong Template Matching Test:              0.11043368541387631   True
2.09. Universal Statistical Test:                      0.282567947825744     True
2.10. Linear Complexity Test:                          0.8263347704038304    True
2.11. Serial Test:                                     0.766181646833394     True
2.12. Approximate Entropy Test:                        0.7000733881151612    True
2.13. Cumulative Sums (Forward):                       0.6698864641681423    True
2.13. Cumulative Sums (Backward):                      0.7242653099698069    True
2.14. Random Excursion Test:
         STATE           xObs                   P-Value                 Conclusion
         '-4'        3.8356982129929085     0.5733056949947805          True
         '-3'        7.318707114093956      0.19799602021827734         True
         '-2'        7.861927251636425      0.16401104937943733         True
         '-1'        15.69261744966443      0.007778723096466819        False
         '+1'        2.4308724832214765     0.7868679051783156          True
         '+2'        4.7989062888391745     0.44091173664620265         True
         '+3'        2.3570405369127525     0.7978539716877826          True
         '+4'        2.4887672641992014     0.7781857852321322          True
```

Baterías de Tests



Histograma o distribución de p-valores obtenidos en un test estadístico.

Resultados sobre PRG seguros

Teorema

Todo PRG seguro es impredecible. ($G(\mathbf{k})\big|_{1:i} \not\Rightarrow G(\mathbf{k})\big|_{i+1}$).

La idea del argumento es que si G fuera predecible, existe $1 \leq i < n$, tal que a partir de la secuencia $G(\mathbf{k})\big|_{1:i}$ se puede obtener información sobre $G(\mathbf{k})\big|_{i+1}$. Luego, hay un algoritmo A tal que

$$\mathbb{P}[A(G(\mathbf{k})\big|_{1:i}) = G(\mathbf{k})\big|_{i+1}] = \tfrac{1}{2} + \varepsilon, \qquad \text{con } \varepsilon \text{ no-negligible.}$$

Podemos entonces diseñar un test estadístico de la forma

$$B(\mathbf{x}) = 1 \iff A(\mathbf{x}\big|_{1:}) = \mathbf{x}\big|_{i+1},$$

y tendríamos $\text{Adv}(B, G) = \big|\mathbb{P}_{\mathcal{K}}[B(G(\mathbf{x})) = 1] - \mathbb{P}_{\{0,1\}^n}[B(\mathbf{x}) = 1]\big| = \big|(\tfrac{1}{2} + \varepsilon) - \tfrac{1}{2}\big| = \varepsilon$.

Teorema (Yao, 1982)

Si un PRG G es impredecible, entonces es PRG seguro. Específicamente, si para todo $i = 1, 2, \ldots, n-1$, *G es impredecible en la posición i, entonces G es PRG seguro.*

Resultados sobre PRG seguros

Consideremos p_1 y p_2 dos distribuciones de probabilidad sobre $\{0,1\}^n$.

Definición

*Decimos que p_1 y p_2 son **computacionalmente indistinguibles**, si para todo algoritmos eficiente $A : \{0,1\}^n \to \{0,1\}$, vale*

$$\left|\mathbb{P}_{p_1}[A(\mathbf{x}) = 1] - \mathbb{P}_{p_2}[A(\mathbf{x}) = 1]\right| < \varepsilon, \qquad \textit{con } \varepsilon \textit{ negligible}.$$

En ese caso, escribimos $p_1 \simeq_p p_2$.

O sea, si ningún algoritmo eficiente puede distinguir entre las distribuciones p_1 y p_2.

Ejemplo: Un generador $G : \mathcal{K} \to \{0,1\}^n$ es PRG seguro si, y sólo si, $\{G(\mathbf{k}) : \mathbf{k} \in \mathcal{K}\} \simeq_p \mathit{Unif}(\{0,1\}^n)$.

Seguridad Semántica

Pregunta: ¿Qué significa que un cifrado $\mathbb{E} = (E, D)$ sea seguro?
Pensemos en un atacante (*one key*) que puede recuperar sólo un texto cifrado.

- Intento 1: El atacante no puede recuperar la clave secreta $\mathbf{k}$.
  No es buen concepto de seguridad. Por ejemplo en el cifrado $E(\mathbf{k}, \mathbf{m}) = \mathbf{m}$, el atacante no recupera la clave, pero recupera toda la información original.

- Intento 2: El atacante no recuperta todo el mensage plano original $\mathbf{m}$.
  Este tampoco es buen concepto de seguridad. Por ejemplo, dados dos mensajes planos $\mathbf{m}_0$ y $\mathbf{m}_1$, en el cifrado

$$E(\mathbf{k}, \mathbf{m}_0 + \mathbf{m}_1) = \mathbf{m}_0 + E(\mathbf{k}, \mathbf{m}_1),$$

  El atacante no recupera el mensaje $\mathbf{m}_0 + \mathbf{m}_1$, en su totalidad. Sin embargo no es seguro, porque recupera parte del mensaje $\mathbf{m}_0$.

Debemos recordar la definición de Shannon de secreto perfecto: a partir del texto cifrado $\mathbf{c}$, el atacante no puede ganar información de $\mathbf{m}$. No revela información.

Seguridad Semántica

Recordemos que

Definición

*Un cifrado de Shannon $\mathbb{E} = (E, D)$ sobre el espacio $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ posee **secreto perfecto** (perfect secrecy) si para cualesquiera mensajes $\mathbf{m}_0, \mathbf{m}_1 \in \mathcal{M}$, con $len(\mathbf{m}_0) = len(\mathbf{m}_1)$, y para cualquier texto cifrado $\mathbf{c} \in \mathcal{C}$, vale*

$$\mathbb{P}(E(\mathbf{k}, \mathbf{m}_0)) = \mathbf{c}) = \mathbb{P}(E(\mathbf{k}, \mathbf{m}_1) = \mathbf{c}), \quad \forall \mathbf{k} \in \mathcal{K},$$

cuando $\mathbf{k}$ es una variable aleatoria con distribución uniforme en $\mathcal{K}$, $\mathbf{k} \sim U(\mathcal{K})$.

*Vamos a relajar un poco la definición, y vamos a decir que el cifrado $\mathbb{E} = (E, D)$ posee **secreto perfecto** si para cualesquiera mensajes $\mathbf{m}_0, \mathbf{m}_1 \in \mathcal{M}$, con $len(\mathbf{m}_0) = len(\mathbf{m}_1)$*

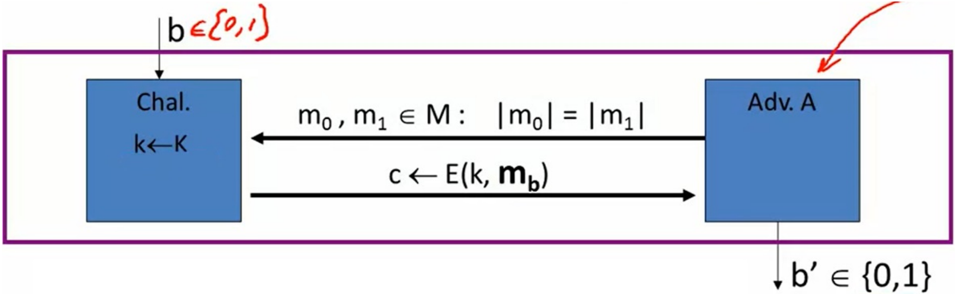
$$\{\mathbb{P}_{\mathcal{K}}(E(\mathbf{k}, \mathbf{m}_0)) = \mathbf{c})\} \simeq_p \{\mathbb{P}_{\mathcal{K}}(E(\mathbf{k}, \mathbf{m}_1) = \mathbf{c})\},$$

o sea, las distribuciones asociadas a los mensajes $\mathbf{m}_0$ y $\mathbf{m}_1$ son computacionalmente indistinguibles.

Seguridad Semántica

Consideremos dos experimentos $EXP(0)$ y $EXP(1)$ de la siguiente forma.
$EXP(b)$, $b = 0, 1$, recibe un mensaje $\mathbf{m}_b \in \{0, 1\}^n$. Elige una clave aleatorial $\mathbf{k} \in \mathcal{K}$, y devuelve el texto cifrado $\mathbf{c}_b = E(\mathbf{k}, \mathbf{m}_b)$.
Esto es $EXP(0)$ devuelve $\mathbf{c}_0 = E(\mathbf{k}, \mathbf{m}_0)$, y $EXP(1)$ devuelve $\mathbf{c}_1 = E(\mathbf{k}, \mathbf{m}_1)$.

Por otro lado, tenemos un adversario A (un algoritmo o test) que recibe alguno de los cifrados $\mathbf{c}_b$ y trata de adivinar si el cifrado fue generado por $EXP(0)$ ó $EXP(1)$: produce b'.



Seguridad Semántica

Consideramos los eventos $W_b = \{EXP(b) = 1\}$, donde $b = 0, 1$.

Definimos la **ventaja semántica** de A con respecto de los experimentos E, como

$$\text{Adv}_{ss}(A, E) = \Big|\mathbb{P}_{\mathcal{K}}(W_0 = 1) - \mathbb{P}_{\mathcal{K}}(W_1 = 1)\Big|.$$

Al igual que antes, cuando $\text{Adv}_{ss}(A, E)$ es cercana a 0, esto significa que el adversario A no es capaz de distinguir si el mensaje cifrado fue generado por $EXP(0)$ ó $EXP(1)$. Cuando es cercana a 1, A es un adversario que satisfactoriamente distingue los mensajes de $EXP(0)$ y los de $EXP(1)$

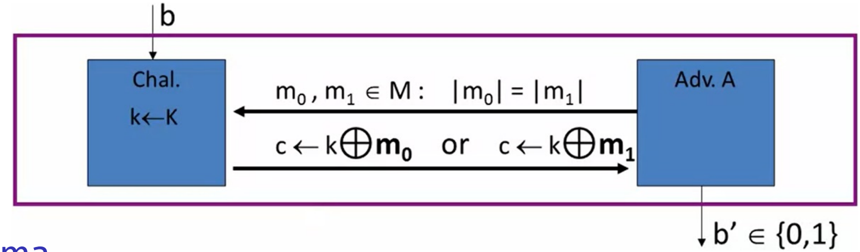
Definición

*Decimos que un cifrado $\mathbb{E} = (E, D)$ es **semánticamente seguro**, si para todo adversario o algoritmos eficiente $A : \{0, 1\}^C \to \{0, 1\}$, se tiene que*

$$\text{Adv}_{ss}(A, \mathbb{E}) < \varepsilon, \qquad \textit{para } \varepsilon \textit{ negligible}.$$

Esto es, (E, D) es semánticamente seguro, ssi, $\{E(\mathbf{k}, \mathbf{m}_0)\} \simeq_p \{E(\mathbf{k}, \mathbf{m}_1)\}$, $\forall \ |\mathbf{m}_0| = |\mathbf{m}_1|$.

Seguridad Semántica



Teorema

El cifrado OTP es semánticamente seguro.

$$\text{Adv}_{ss}(A, OTP) = \Big|\mathbb{P}_{\mathcal{K}}[A(\underbrace{\mathbf{k} \oplus \mathbf{m}_0}_{\sim \, Unif}) = 1] - \mathbb{P}_{\mathcal{K}}[A(\underbrace{\mathbf{k} \oplus \mathbf{m}_1}_{\sim \, Unif}) = 1]\Big| = \Big|\tfrac{1}{2} - \tfrac{1}{2}\Big| = 0.$$

Teorema

Si $G : \mathcal{K} \to , 1\}^n$ es PRG seguro, entonces el cifrado de flujo $E(\mathbf{k}, \mathbf{m}) = \mathbf{m} \oplus G(\mathbf{k})$ es semánticamente seguro. ***Las vulnerabilidades no son del XOR , si no que vienen del PRG.***