Stream Ciphers

Generadores pseudo-Aleatorios (PRG)

Un generador pseudo-Aleatorio es un algoritmo determinista para generar números aparentemente aleatorios. Estos números no son realmente aleatorios ya que siguen un algoritmo, pero parecen serlo.

$G:[{0,1}]^l -> [{0,1}]^L$ donde $l<=L$ siendo l la longitud de la cadena.

Intuitivamente se dice que G es seguro si es computacionalmente difícil de distinguir entre $r=G(s)$ y un número verdaderamente aleatorio. Un desafiante selecciona un número y después genera un número aleatorio de forma desconocida para el atacante. Tras eso revela un número R de forma que el atacante debe saber si el número es verdaderamente aleatorio o no, si el atacante acierta el número entonces el sistema ha sido comprometido.

PRGs Seguros

G es un PRG seguro si: $|P(b=1|b=0)-p(b=1,b=1)|<=ξ$ para un ξ extremadamente pequeño y todos los adversarios eficientes A.

G es seguro cuando no existe ninguna prueba estadística eficiente para decidir si G(s) es aleatorio con una probabilidad que no sea extremadamente pequeña. Algunos test estdísticos pueden ser:

- Frecuencia empírica de 1s y 0s
- Frencuencia empírica de substrings
- Valores Extremos: mayor secuencia de 1s

Stream Ciphers

Stream Cipher es un cifrado bsado en PRG. El Stream Ciber (E,D) definido de un PRG G es:

Encriptado: $E(s,m) = G(s)⊕m$

Desencriptado: $D(s,c)=G(s)⊕C$

Si G es un PRG seguro, entonces el Stream Cipher (E,D) construido a partir de G es semanticamente seguro.

Problemas de los stream Ciphers:

Two-time pad

Usando un mismo Stream Cipher para cifrar 2 mensajes es poco seguro. Los mensajes tanto en español como en inglés tienen suficiente redundancia como para facilitar romper parte del cifrado. La

clave generada por un Stream Cipher nunca debe ser usada más de una vez. Se puede tratar de acertar el cifrado adivinando que letra equivale a cual utilizando un listado con las letras que más se repiten en el lenguaje del mensaje. Por lo que se puede deducir estadísticamente el cifrado de un mensaje.

One-Time pad es maleable

El OPT asado en PRG es maleable, se puede utilizar un delta para cambiar el mensaje deliberadamente como forma de ataque, resultando en la destrucción del texto. c+delta = G(s)⊕m⊕delta = G(s)⊕(m+delta)

Composición de PRGs

Si tienes un PRG simple puedes combinarlo para obtener algo más sofisticado. Se recomienda realizar una construcción paralela. G'(s1,s2,.....,sn) = (G(s1), G(s2).....G(sn)) La seguridad de G' se degrada linearmente en n, por lo que si G es un PRG seguro, entonces G' también lo es.

La construcción secuencial (Construcción Blum-micali) supone G como un PRG definido en (S, RxS).

1. Se toma una semilla inicial S0
2. Para i=1....n se computa

(ri,si)

1. A

Incompleto...

Generadores lineales congruentes (LCG)

Los LCGs son suministrados en todo sistema operativo y entorno de programación como origen de números pseudo-aleatorios para simulaciones computacionales. Los LCG no son robustos para propósitos criptográficos.