

stream cipher 2 + PSEUDO RANDOM GENERATORS

→ Como fazer o OTP prático? (Problema das chaves longas!)
↳ Ideia: substituir a chave aleatória por pseudo-aleatória

PRG

* DEF: É uma função $G: \{0,1\}^s \rightarrow \{0,1\}^n$, $n \gg s$
OUTPUT

5
↓
seed

→ somente é a única coisa aleatória, sendo se montando um algoritmo determinístico.



$$C = E(K, m) := m \oplus G(K)$$
$$D(K, c) := c \oplus G(K)$$

* Mas... Esse padrão não tem um segredo perfeito. Sua chave é menor que a mensagem.

* Precisamos de uma nova definição de segurança e novas propriedades.

PRG DEVE SER IMPREVISÍVEL

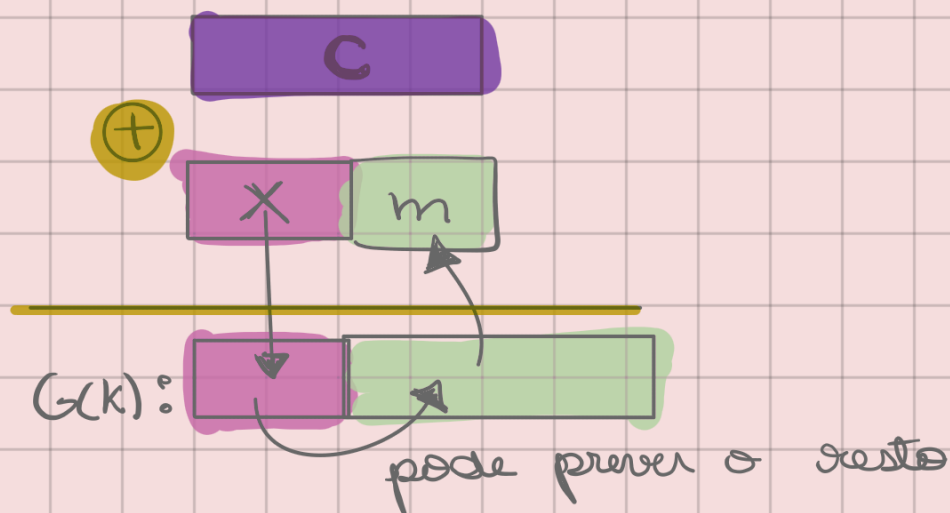
→ Ele não consegue prever.

$$\exists i: G(K) \Big|_{1, \dots, i} \xrightarrow{\text{Alg.}} G(K) \Big|_{i+1, \dots, n}$$

→ dados os primeiros termos, obtém a lógica de todos os restantes.

Então:

Exemplo SMTP: começa sempre com "X", atacante já sabe um prefixo, o que se faz:



Questão: $G: K \rightarrow \{0,1\}^n$ para todo $K: X \oplus K(G(K)) = 1$
• G é imprevisível?

R: Não, dado $n-1$ bits é possível prever o último bit.

PRG FRACO

→ não utilizado em criptografia.

* Gerador congruente linear: 3 parâmetros: 2 inteiros: A, B
1 primo: p

→ $r[0] = \text{seed}$

$r[i] \leftarrow a \cdot r[i-1] + b \pmod p$
output são alguns bits de $r[i]$
 $i++$

loop

→ fácil de prever.

→ nunca utilizar essa biblioteca para criptografia

* Global random():

$r[i] \leftarrow (r[i-3] + r[i-3]) \% 2^{32}$
output $r[i] > 1$

Insignificância e não-insignificância

neg

non-neg

* Na prática: • ε "é um escalar e

• $\varepsilon_{\text{non-neg}}$: $\varepsilon \geq 1/2^{30}$

• ε_{neg} : $\varepsilon \leq 1/2^{80}$

p/ muitos λ

* Na teoria: • ε é uma função $\varepsilon: \mathbb{Z}^{\geq 0} \rightarrow \mathbb{R}^{\geq 0}$ e

• $\varepsilon_{\text{non-neg}}$: $\exists d: \varepsilon(\lambda) \geq 1/\lambda^d$ ($\varepsilon \geq 1$ / polinômio)

• ε_{neg} : $\forall d, \lambda \geq \lambda_d: \varepsilon(\lambda) \leq 1/\lambda^d$ ($\varepsilon \leq 1$ / polinômio)

Exemplos:

* $\varepsilon(\lambda) = 1/2^\lambda$: non.

* $\varepsilon(\lambda) = 1/\lambda^{1000}$: non-neg.

* $\varepsilon(\lambda) = \begin{cases} 1/2^\lambda & \text{para } \lambda \text{ ímpar} \\ 1/\lambda^{1000} & \text{para } \lambda \text{ par} \end{cases}$ non-neg.