Stream ciphers

Pseudorandom Generators

Review

Cipher over (K,M,C): a pair of "efficient" algs (***E***, ***D***) s.t.

$$\forall\ m \in M,\ k \in K: \quad \boldsymbol{D}(k, \boldsymbol{E}(k, m)) = m$$

Weak ciphers: subs. cipher, Vigener, …

A good cipher: **OTP** $M=C=K=\{0,1\}^n$

$$E(k, m) = k \oplus m \quad , \quad D(k, c) = k \oplus c$$

<u>Lemma</u>: OTP has perfect secrecy (i.e. no CT only attacks)

Bad news: perfect-secrecy ⇒ key-len ≥ msg-len

Stream Ciphers: making OTP practical

idea: replace "random" key by "pseudorandom" key

PRG is a function $G: \{0,1\}^s \rightarrow \{0,1\}^n$ $\quad n \gg s$

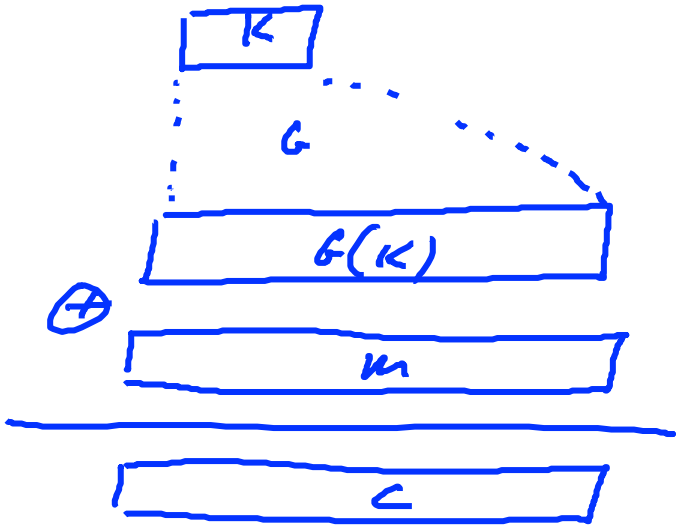
seed space

(eff. computable by a deterministic algorithm)

Stream Ciphers: making OTP practical

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

$D(K,C) = C \oplus G(K)$



Can a stream cipher have perfect secrecy?

- Yes, if the PRG is really “secure”
- No, there are no ciphers with perfect secrecy
- Yes, every cipher has perfect secrecy
- No, since the key is shorter than the message

Stream Ciphers: making OTP practical

Stream ciphers cannot have perfect secrecy !!

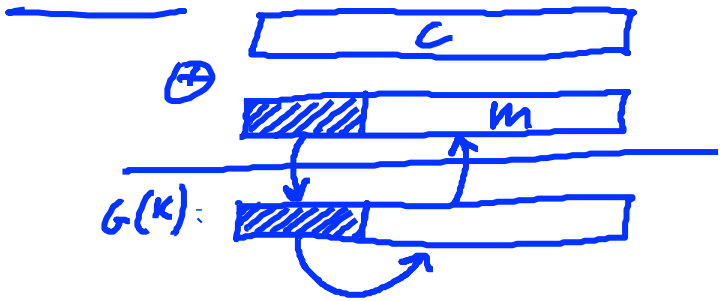
- Need a different definition of security
- Security will depend on specific PRG

PRG must be unpredictable

Suppose PRG is predictable:

$\exists i$: $G(k)|_{1,\ldots,i} \xrightarrow{\text{alg}} G(k)|_{i+1,\ldots,n}$

Then:



even $G(k)|_{1,\ldots,i} \rightarrow G(k)|_{i+1}$ is a problem!

PRG must be unpredictable

We say that $G: K \longrightarrow \{0,1\}^n$ is **predictable** if:

$\exists$ "eff" alg. A and $\exists\, 0 \le i \le n-1$ s.t.

$$\Pr_{k \xleftarrow{R} K}\left[A(G(k))\big|_{1,\ldots,i} = G(k)\big|_{i+1} \right] > \frac{1}{2} + \varepsilon$$

For non-negligible ε (e.g. $\varepsilon = 1/2^{30}$)

Def: PRG is **unpredictable** if it is not predictable

$\Rightarrow$ $\forall$ i: no "eff" adv. can predict bit (i+1) for "non-neg" ε

Suppose $G:K \longrightarrow \{0,1\}^n$ is such that for all k: **XOR(G(k)) = 1**

Is G predictable ??

Yes, given the first bit I can predict the second

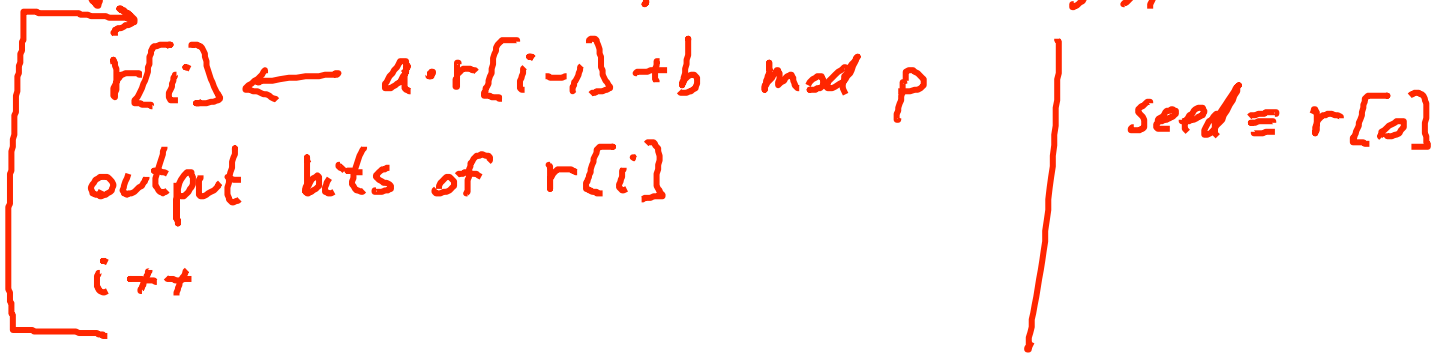
No, G is unpredictable

Yes, given the first (n-1) bits I can predict the n'th bit ⟸

It depends

Weak PRGs (do not use for crypto)

Lin. Cong. generator with parameters a, b, p:



glibc random():

$r[i] \leftarrow (r[i-3] + r[i-31]) \ \% \ 2^{32}$

output $r[i] >> 1$

never use random() for crypto !!

(e.g. Kerberos V4)

Negligible and non-negligible

- In practice: ε is a scalar and
  - ε non-neg: $\varepsilon \geq 1/2^{30}$ (likely to happen over 1GB of data)
  - ε negligible: $\varepsilon \leq 1/2^{80}$ (won't happen over life of key)

- In theory: ε is a function $\varepsilon: Z^{\geq 0} \longrightarrow R^{\geq 0}$ and
  - ε non-neg: $\exists\, d:\ \varepsilon(\lambda) \geq 1/\lambda^d$ inf. often ($\varepsilon \geq$ 1/poly, for many λ)
  - ε negligible: $\forall\, d,\ \lambda \geq \lambda_d:\ \varepsilon(\lambda) \leq 1/\lambda^d$ ($\varepsilon \leq$ 1/poly, for large λ)

Few Examples

$\varepsilon(\lambda) = 1/2^{\lambda}$: negligible

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

$$\varepsilon(\lambda) = \begin{cases} 1/2^{\lambda} & \text{for odd } \lambda \\ 1/\lambda^{1000} & \text{for even } \lambda \end{cases}$$

Negligible

Non-negligible 

End of Segment