



## Stream ciphers

---

## The One Time Pad

# Symmetric Ciphers: definition

Def: a **cipher** defined over  $(\mathcal{K}, \mathcal{M}, \mathcal{C})$

is a pair of “efficient” algs  $(E, D)$  where

$$E: \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C} \quad , \quad D: \mathcal{K} \times \mathcal{C} \rightarrow \mathcal{M}$$

$$\text{s.t. } \forall m \in \mathcal{M}, k \in \mathcal{K}: D(k, E(k, m)) = m$$

- $E$  is often randomized.  $D$  is always deterministic.

# The One Time Pad

(Vernam 1917)

First example of a “secure” cipher

$$\mathcal{M} = \mathcal{C} = \{0,1\}^n , \quad \mathcal{K} = \{0,1\}^n$$

key = (random bit string as long the message)

# The One Time Pad

(Vernam 1917)

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

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

msg: 0 1 1 0 1 1 1

key: 1 0 1 1 0 1 0

CT:



Indeed:

$$D(K, E(K, m)) = D(K, K \oplus m) = K \oplus (K \oplus m) = (K \oplus K) \oplus m = 0 \oplus m = m$$

You are given a message ( $m$ ) and its OTP encryption ( $c$ ).

Can you compute the OTP key from  $m$  and  $c$ ?

No, I cannot compute the key.

Yes, the key is  $k = m \oplus c$ . 

I can only compute half the bits of the key.

Yes, the key is  $k = m \oplus m$ .

# The One Time Pad

(Vernam 1917)

Very fast enc/dec !!

... but long keys (as long as plaintext)

Is the OTP secure? What is a secure cipher?

# What is a secure cipher?

Attacker's abilities: **CT only attack** (for now)

Possible security requirements:

attempt #1: **attacker cannot recover secret key**

$$E(k, m) = m \quad \text{would be secure}$$

attempt #2: **attacker cannot recover all of plaintext**

$$E(k, m_0 \parallel m_1) = m_0 \parallel k \oplus m_1 \quad \text{would be secure}$$

Shannon's idea:

**CT should reveal no “info” about PT**

# Information Theoretic Security

(Shannon 1949)

Def: A cipher  $(E, D)$  over  $(\mathcal{K}, \mathcal{M}, \mathcal{C})$  has perfect secrecy if

$$\forall m_0, m_1 \in \mathcal{M} \quad (\text{len}(m_0) = \text{len}(m_1)) \quad \text{and} \quad \forall c \in \mathcal{C}$$

$$\Pr[E(k, m_0) = c] = \Pr[E(k, m_1) = c]$$

where  $k$  is uniform in  $\mathcal{K}$  ( $k \leftarrow \mathcal{K}$ )

# Information Theoretic Security

Def: A cipher  $(E, D)$  over  $(K, M, C)$  has **perfect secrecy** if

$$\forall m_0, m_1 \in M \quad (|m_0| = |m_1|) \quad \text{and} \quad \forall c \in C$$

$$Pr[E(k, m_0) = c] = Pr[E(k, m_1) = c] \quad \text{where } k \leftarrow K$$

- 
- ⇒ Given CT can't tell if msg is  $m_0$  or  $m_1$  (for all  $m_0, m_1$ )
  - ⇒ most powerful adv. learns nothing about PT from CT
  - ⇒ no CT only attack!! (but other attacks possible)

Lemma: OTP has perfect secrecy.

Proof:

$$\forall m, c: \Pr_{k \in \mathcal{K}} [E(k, m) = c] = \frac{\#\text{keys } k \in \mathcal{K} \text{ s.t. } E(k, m) = c}{|\mathcal{K}|}$$

So: if  $\forall m, c: \#\{k \in \mathcal{K} : E(k, m) = c\} = \text{const.}$

$\Rightarrow$  cipher has perfect secrecy

Let  $m \in \mathcal{M}$  and  $c \in \mathcal{C}$ .

How many OTP keys map  $m$  to  $c$  ?

None

1 

2

Depends on  $m$

Lemma: OTP has perfect secrecy.

Proof:

For OTP:  $\forall m, c : \text{if } E(k, m) = c$

$$\Rightarrow k \oplus m = c \Rightarrow k = m \oplus c$$

$$\Rightarrow \boxed{\#\{k \in \mathbb{K} : E(k, m) = c\} = 1}$$

$\Rightarrow$  OTP has perfect Secrecy 

# The bad news ...

Thm: perfect secrecy  $\Rightarrow |K| \geq |M|$

i.e. perfect secrecy  $\Rightarrow$  key-len  $\geq$  msg-len

$\Rightarrow$  hard to use in practice !!

# End of Segment



## Stream ciphers

---

Pseudorandom  
Generators

# Review

Cipher over  $(K, M, C)$ : a pair of “efficient” algs  $(E, D)$  s.t.

$$\forall m \in M, k \in K: D(k, E(k, m)) = m$$

Weak ciphers: subs. cipher, Vigener, ...

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

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

Lemma: OTP has perfect secrecy (i.e. no CT only attacks)

Bad news: perfect-secrecy  $\Rightarrow$  key-len  $\geq$  msg-len

# Stream Ciphers: making OTP practical

idea: replace “random” key by “pseudorandom” key

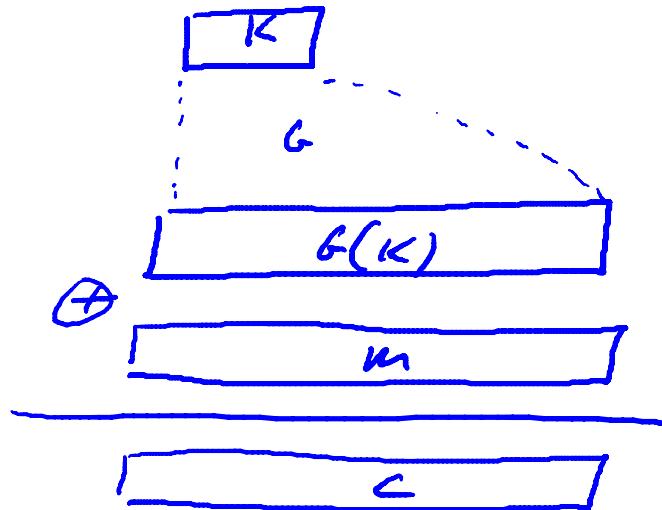
PRG is a function  $G: \underbrace{\{0,1\}^s}_{\text{seed space}} \rightarrow \{0,1\}^n$   $n \gg s$

(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 !!

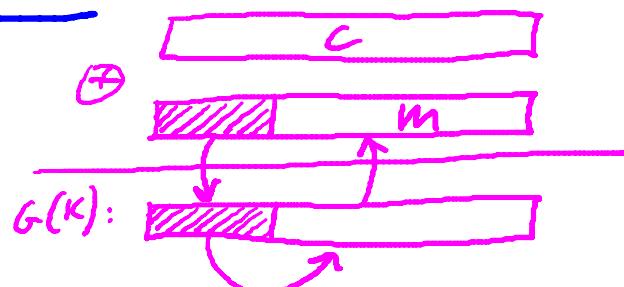
- Need a different definition of security
- Security will depend on specific PRG

# PRG must be unpredictable

Suppose PRG is predictable :

$$\exists i: \left. g(k) \right|_{1, \dots, i} \xrightarrow{\text{alg}} \left. g(k) \right|_{i+1, \dots, n}$$

Then:



even  $\left. g(k) \right|_{1, \dots, i} \rightarrow \left. g(k) \right|_{i+1}$   
is a problem!

# PRG must be unpredictable

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

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

$$\Pr_{\substack{K \leftarrow g_K}} \left[ A(G(K)) \Big|_{i, \dots, i} = G(K) \Big|_{i+1} \right] > \frac{1}{2} + \varepsilon$$

For non-negligible  $\varepsilon$  (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"  $\varepsilon$

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

Is  $G$  predictable ??

Yes, given the first bit I can predict the second

No,  $G$  is unpredictable

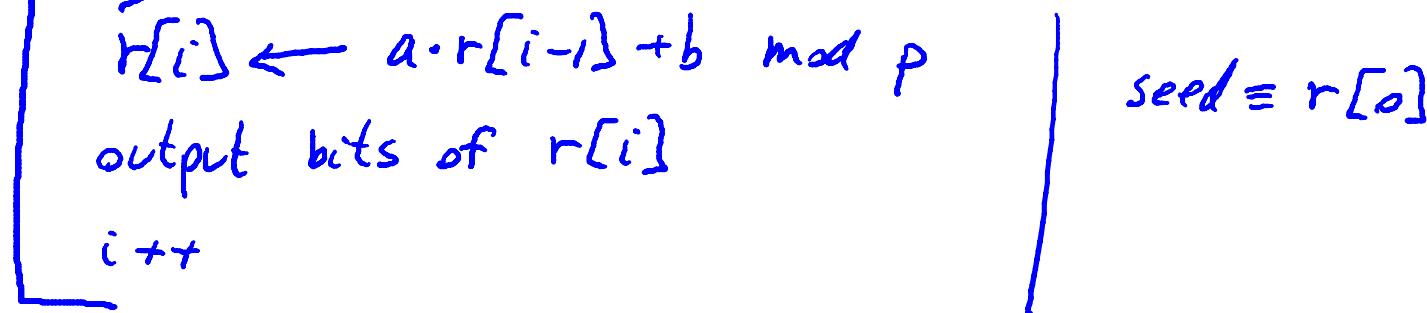
Yes, given the first  $(n-1)$  bits I can predict the  $n$ 'th bit 

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  $r[i] \gg 1$

never use random()  
for crypto !!  
(e.g. Kerberos V4)

# End of Segment



## Stream ciphers

---

Negligible vs.  
non-negligible

# Negligible and non-negligible

- In practice:  $\epsilon$  is a scalar and
  - $\epsilon$  non-neg:  $\epsilon \geq 1/2^{30}$  (likely to happen over 1GB of data)
  - $\epsilon$  negligible:  $\epsilon \leq 1/2^{80}$  (won't happen over life of key)
- In theory:  $\epsilon$  is a function  $\epsilon: \mathbb{Z}^{\geq 0} \rightarrow \mathbb{R}^{\geq 0}$  and
  - $\epsilon$  non-neg:  $\exists d: \epsilon(\lambda) \geq 1/\lambda^d$  inf. often ( $\epsilon \geq 1/\text{poly}$ , for many  $\lambda$ )
  - $\epsilon$  negligible:  $\forall d, \lambda \geq \lambda_d: \epsilon(\lambda) \leq 1/\lambda^d$  ( $\epsilon \leq 1/\text{poly}$ , for large  $\lambda$ )

# 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



# PRGs: the rigorous theory view

PRGs are “parameterized” by a security parameter  $\lambda$

- PRG becomes “more secure” as  $\lambda$  increases

Seed lengths and output lengths grow with  $\lambda$

For every  $\lambda=1,2,3,\dots$  there is a different PRG  $G_\lambda$ :

$$G_\lambda : K_\lambda \rightarrow \{0,1\}^{n(\lambda)}$$

(in the lectures we will always ignore  $\lambda$ )

# An example asymptotic definition

We say that  $\mathbf{G}_\lambda : \mathcal{K}_\lambda \rightarrow \{0,1\}^{n(\lambda)}$  is predictable at position  $i$  if:

there exists a polynomial time (in  $\lambda$ ) algorithm  $A$  s.t.

$$\Pr_{k \leftarrow \mathcal{K}_\lambda} [ A(\lambda, \mathbf{G}_\lambda(k) \Big|_{1,\dots,i}) = \mathbf{G}_\lambda(k) \Big|_{i+1} ] > 1/2 + \varepsilon(\lambda)$$

for some non-negligible function  $\varepsilon(\lambda)$

# End of Segment



## Stream ciphers

---

### Attacks on OTP and stream ciphers

# Review

**OTP:**  $E(k,m) = m \oplus k$  ,  $D(k,c) = c \oplus k$

Making OTP practical using a PRG:  $G: K \rightarrow \{0,1\}^n$

**Stream cipher:**  $E(k,m) = m \oplus G(k)$  ,  $D(k,c) = c \oplus G(k)$

Security: PRG must be unpredictable (better def in two segments)

# Attack 1: two time pad is insecure !!

Never use stream cipher key more than once !!

$$C_1 \leftarrow m_1 \oplus \text{PRG}(k)$$

$$C_2 \leftarrow m_2 \oplus \text{PRG}(k)$$

Eavesdropper does:

$$C_1 \oplus C_2 \rightarrow$$

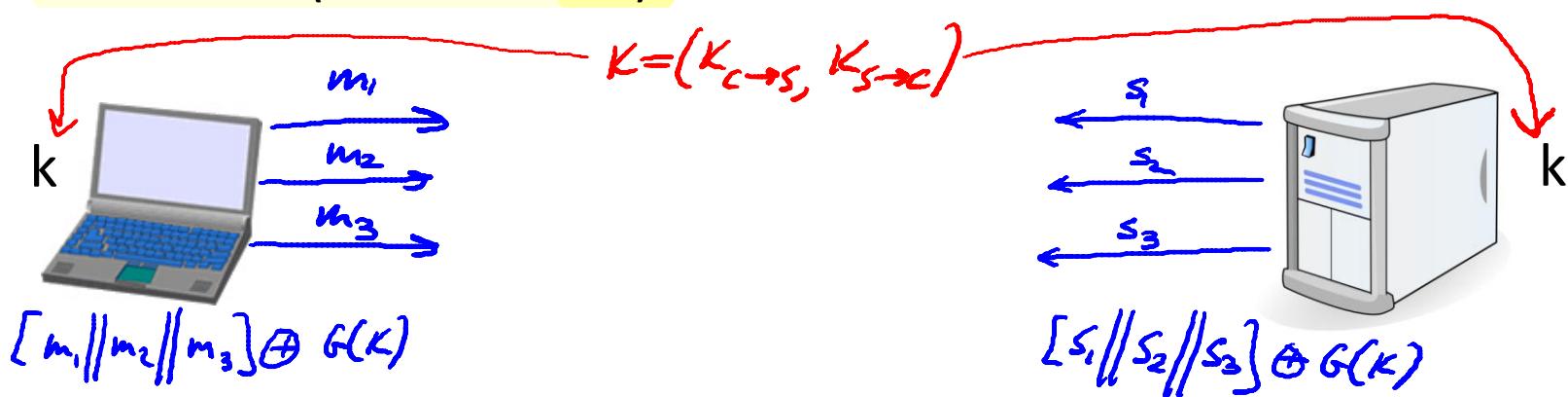


Enough redundancy in English and ASCII encoding that:

$$m_1 \oplus m_2 \rightarrow m_1, m_2$$

# Real world examples

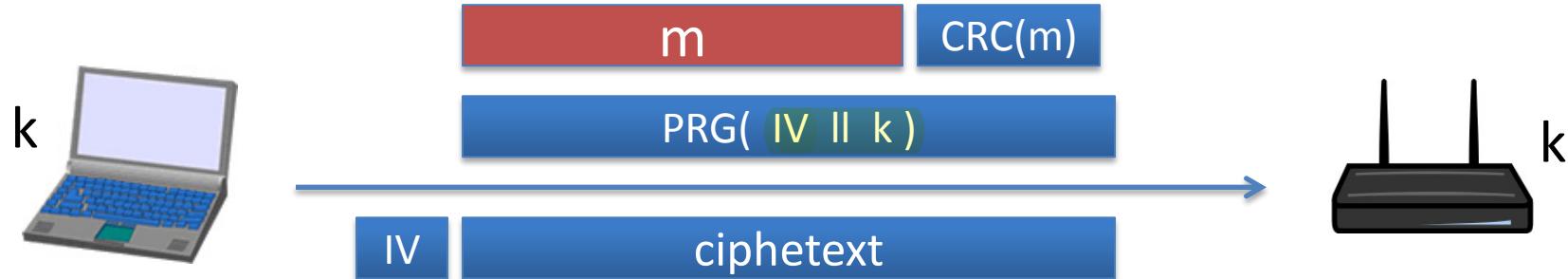
- Project Venona
- MS-PPTP (windows NT):



Need different keys for  $C \rightarrow S$  and  $S \rightarrow C$

# Real world examples

## 802.11b WEP:

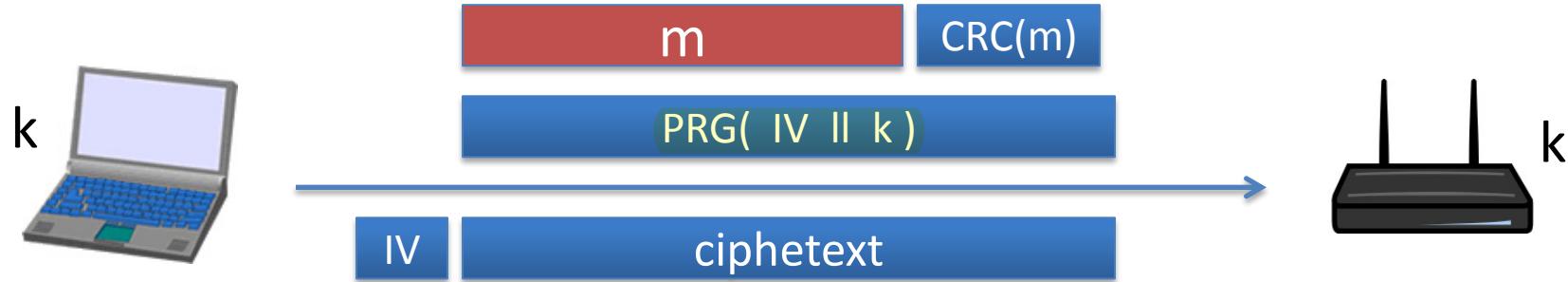


Length of IV: 24 bits

- Repeated IV after  $2^{24} \approx 16M$  frames
- On some 802.11 cards: IV resets to 0 after power cycle

# Avoid related keys

## 802.11b WEP:



key for frame #1:  $(1 \parallel k)$

key for frame #2:  $(2 \parallel k)$

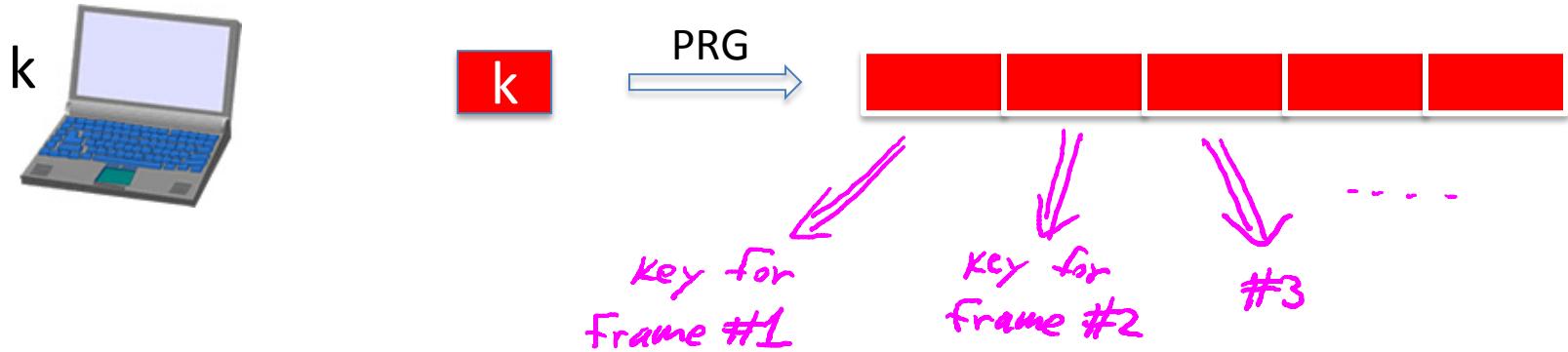
⋮       $\frac{24}{6}$  bits      ↗      ↗      1024 bits

For the RC4 PRG:

FMS2001  $\Rightarrow$  can recover 1K after  $10^6$  frames

Recent attacks  $\approx 40,000$  frames

# A better construction

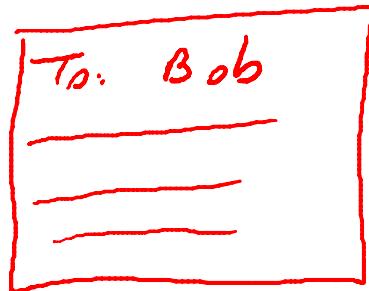


⇒ now each frame has a pseudorandom key

better solution: use stronger encryption method (as in WPA2)

# Yet another example: disk encryption

(1)



enc. disk

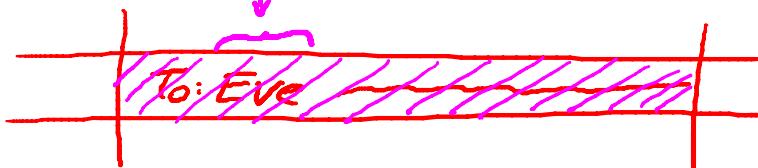


Later:

(2)



enc. disk

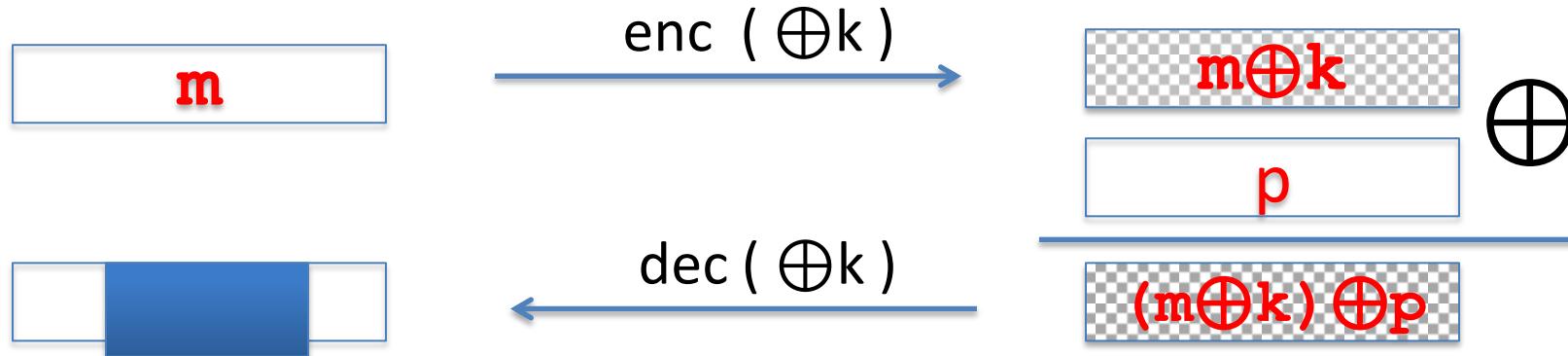


# Two time pad: summary

Never use stream cipher key more than once !!

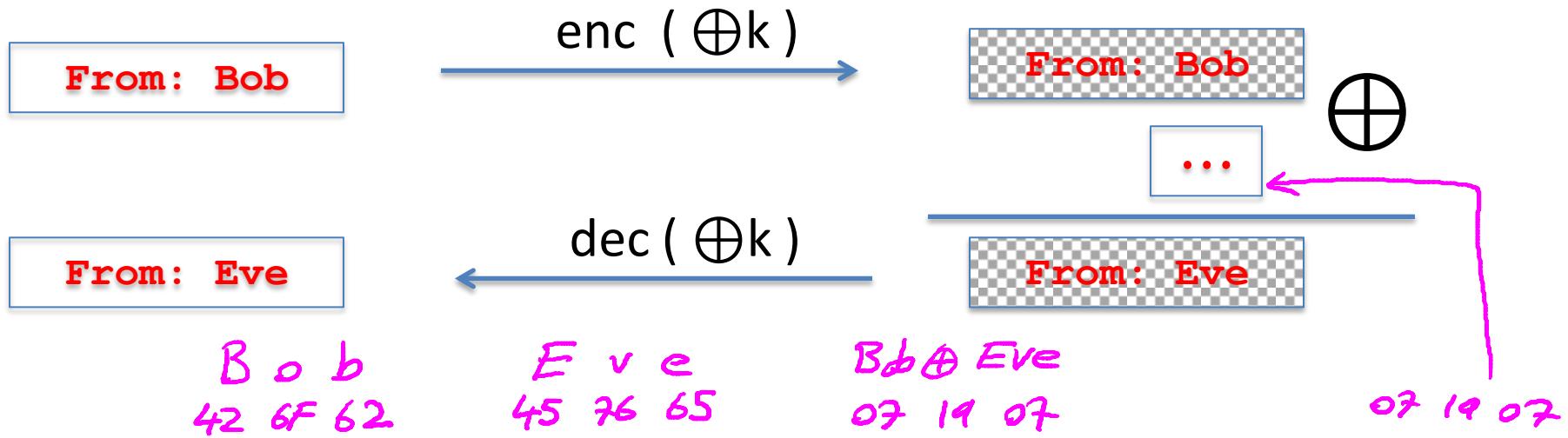
- Network traffic: negotiate new key for every session (e.g. TLS)
- Disk encryption: typically do not use a stream cipher

# Attack 2: no integrity (OTP is malleable)



Modifications to ciphertext are undetected and have **predictable** impact on plaintext

# Attack 2: no integrity (OTP is malleable)



Modifications to ciphertext are undetected and have predictable impact on plaintext

# End of Segment

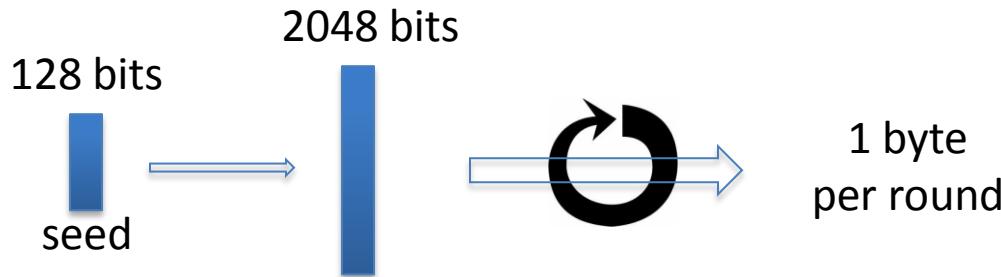


# Stream ciphers

---

Real-world Stream  
Ciphers

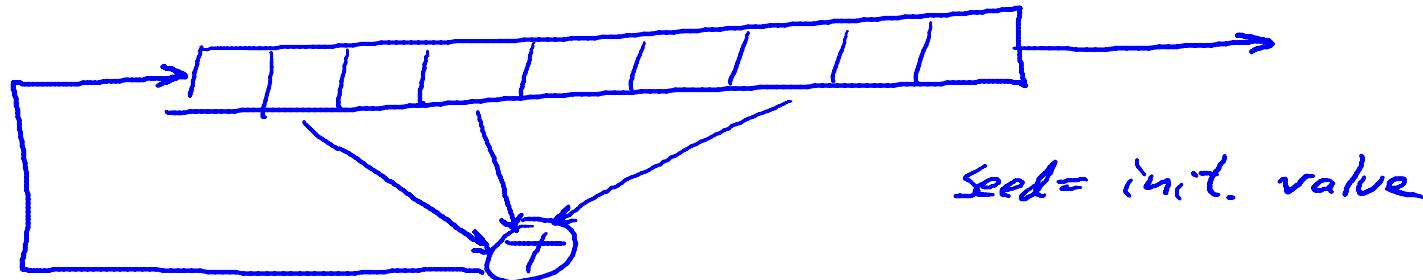
# Old example (software): RC4 (1987)



- Used in HTTPS and WEP
- Weaknesses:
  - Bias in initial output:  $\Pr[ \text{2nd byte} = 0 ] = 2/256$
  - Prob. of (0,0) is  $1/256^2 + 1/256^3$
  - Related key attacks

# Old example (hardware): CSS (badly broken)

Linear feedback shift register (LFSR):



DVD encryption (CSS): 2 LFSRs

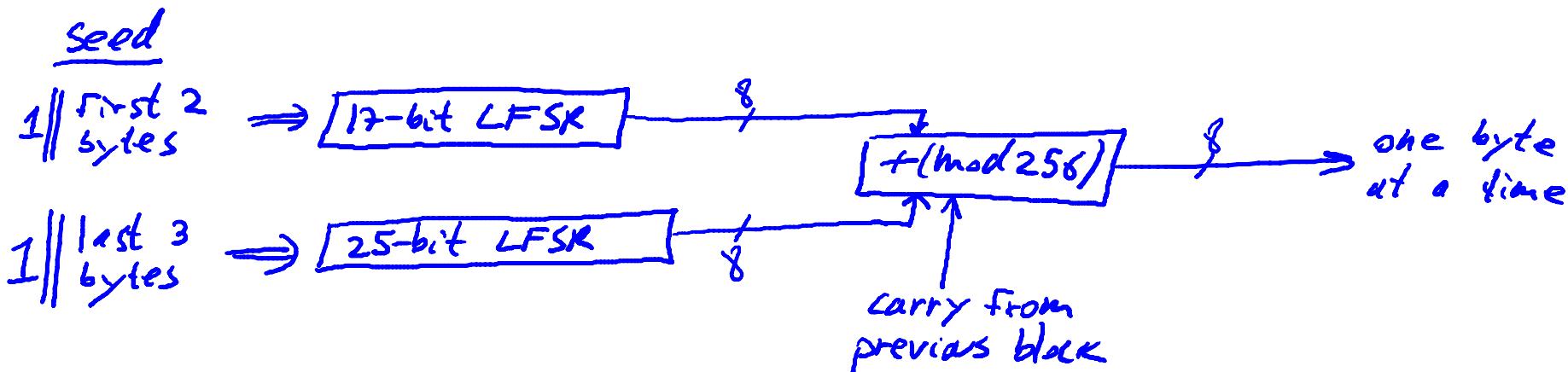
GSM encryption (A5/1,2): 3 LFSRs

Bluetooth (E0): 4 LFSRs

} all broken

# Old example (hardware): CSS (badly broken)

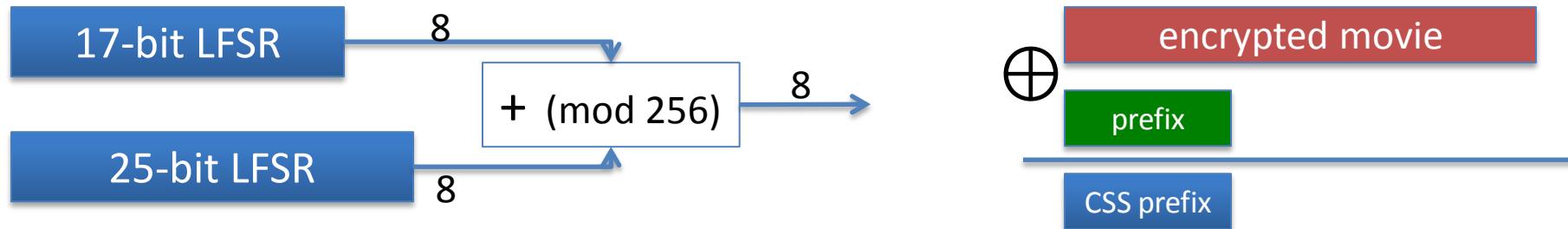
CSS: seed = 5 bytes = 40 bits



Easy to break in time  $\approx 2^{17}$

# Cryptanalysis of CSS

( $2^{17}$  time attack)



For all possible initial settings of 17-bit LFSR do:

- Run 17-bit LFSR to get 20 bytes of output
- Subtract from CSS prefix  $\Rightarrow$  candidate 20 bytes output of 25-bit LFSR
- If consistent with 25-bit LFSR, found correct initial settings of both !!

Using key, generate entire CSS output

# Modern stream ciphers:

eStream

$$\text{PRG: } \underbrace{\{0,1\}^s}_{\text{seed}} \times R \rightarrow \{0,1\}^n$$


Nonce: a non-repeating value for a given key.

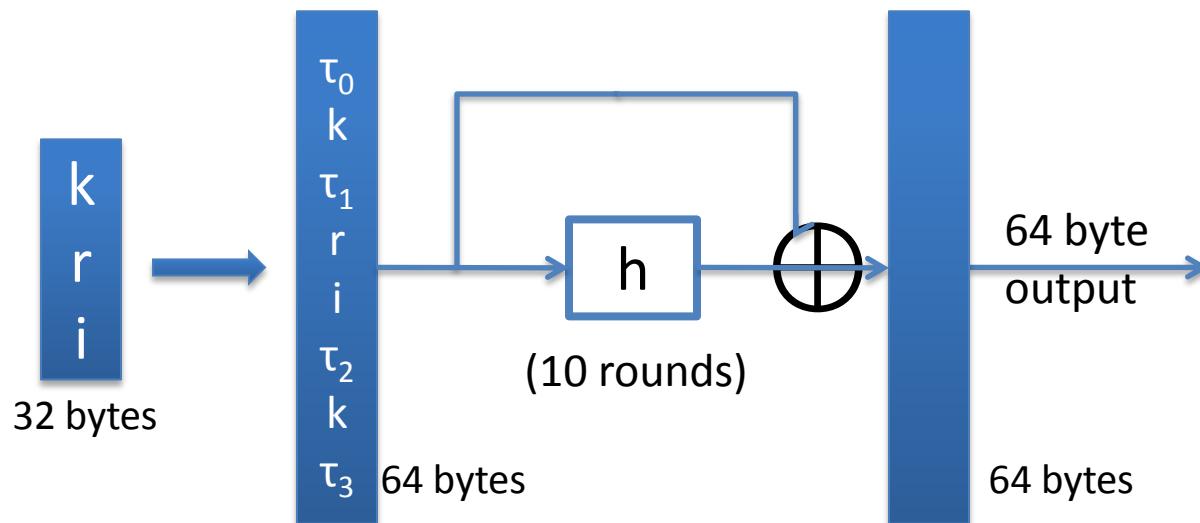
$$E(k, m ; r) = m \oplus \text{PRG}(k ; r)$$

The pair  $(k,r)$  is never used more than once.

# eStream: Salsa 20 (SW+HW)

Salsa20:  $\{0,1\}^{128 \text{ or } 256} \times \{0,1\}^{64} \xrightarrow{\text{nonce}} \{0,1\}^n$  (max n =  $2^{73}$  bits)

$\text{Salsa20}(k; r) := H(k, (r, 0)) \parallel H(k, (r, 1)) \parallel \dots$



$h$ : invertible function. designed to be fast on x86 (SSE2)

# Is Salsa20 secure (unpredictable) ?

- Unknown: no known **provably** secure PRGs
- In reality: no known attacks better than exhaustive search

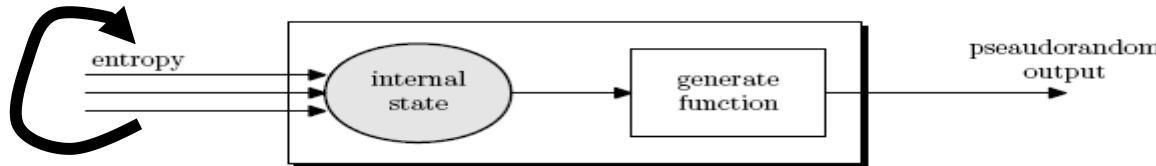
# Performance:

Crypto++ 5.6.0 [ Wei Dai ]

AMD Opteron, 2.2 GHz ( Linux)

<u>PRG</u>	<u>Speed (MB/sec)</u>
RC4	126
eStream	643
Salsa20/12	727
Sosemanuk	

# Generating Randomness (e.g. keys, IV)



Pseudo random generators in practice: (e.g. /dev/random)

- Continuously add entropy to internal state
- Entropy sources:
  - Hardware RNG: Intel **RdRand** inst. (Ivy Bridge). 3Gb/sec.
  - Timing: hardware interrupts (keyboard, mouse)

NIST SP 800-90: NIST approved generators

# End of Segment



## Stream ciphers

---

## PRG Security Defs

Let  $G: K \rightarrow \{0,1\}^n$  be a PRG

Goal: define what it means that

$[K \xleftarrow{R} \mathcal{K}, \text{ output } G(K)]$



is “indistinguishable” from

$[r \xleftarrow{R} \{0,1\}^n, \text{ output } r]$

# Statistical Tests

Statistical test on  $\{0,1\}^n$ :

an alg.  $A$  s.t.  $A(x)$  outputs “0” or “1”

*not random* ↘ *random* ↘

Examples:

$$(1) \quad A(x)=1 \quad \text{iff} \quad | \#0(x) - \#1(x) | \leq 10 \cdot \sqrt{n}$$

$$(2) \quad A(x)=1 \quad \text{iff} \quad | \#00(x) - \frac{n}{4} | \leq 10 \cdot \sqrt{n}$$

# Statistical Tests

More examples:

$$(3) A(x)=1 \text{ iff } \max\text{-run-of-}0(x) < 10 \cdot \log_2(n)$$

•  
•  
•  
•

# Advantage

Let  $G:K \rightarrow \{0,1\}^n$  be a PRG and  $A$  a stat. test on  $\{0,1\}^n$

Define:

$$\text{Adv}_{\text{PRG}}[A,G] := \left| \Pr_{K \in \mathcal{R}^K} [A(G(K))=1] - \Pr_{r \in \{0,1\}^n} [A(r)=1] \right| \in [0,1]$$

Adv close to 1  $\Rightarrow A$  can dist.  $G$  from random

Adv close to 0  $\Rightarrow A$  cannot

A silly example:  $A(x) = 0 \Rightarrow \text{Adv}_{\text{PRG}}[A,G] =$



Suppose  $G:K \rightarrow \{0,1\}^n$  satisfies  $\text{msb}(G(k)) = 1$  for 2/3 of keys in K

Define stat. test  $A(x)$  as:

if [  $\text{msb}(x)=1$  ] output “1” else output “0”

Then

$$\text{Adv}_{\text{PRG}}[A, G] = \left| \overbrace{\Pr[A(G(k))=1]}^{2/3} - \overbrace{\Pr[A(r)=1]}^{1/2} \right| =$$



# Secure PRGs: crypto definition

Def: We say that  $G:K \rightarrow \{0,1\}^n$  is a secure PRG if

$\forall$  "eff" stat. tests  $A$ :

$Adv_{PRG}[A, G]$  is "negligible"

Are there provably secure PRGs?

but we have heuristic candidates.

Easy fact: a secure PRG is unpredictable

We show: PRG predictable  $\Rightarrow$  PRG is insecure

Suppose A is an efficient algorithm s.t.

$$\Pr_{\substack{K \leftarrow \mathcal{R} \\ g_K}} [ A(g(K)|_{1, \dots, i}) = g(K)|_{i+1} ] > \frac{1}{2} + \varepsilon$$

for non-negligible  $\varepsilon$  (e.g.  $\varepsilon = 1/1000$ )

# Easy fact: a secure PRG is unpredictable

Define statistical test  $B$  as:

$$B(x) = \begin{cases} \text{if } A(x|_{1,\dots,i}) = x_{i+1} & \text{output 1} \\ \text{else} & \text{output 0} \end{cases}$$

$$r \xleftarrow{R} \{0,1\}^n : \Pr[B(r) = 1] = \frac{1}{2}$$

$$r \xleftarrow{g(k)} : \Pr[B(g(k)) = 1] > \frac{1}{2} + \varepsilon$$

$$\implies \text{Adv}_{\text{PRG}}[B, g] = \left| \Pr[B(r) = 1] - \Pr[B(g(k)) = 1] \right| > \varepsilon$$

Thm (Yao'82): an unpredictable PRG is secure

Let  $G:K \rightarrow \{0,1\}^n$  be PRG

“Thm”: if  $\forall i \in \{0, \dots, n-1\}$  PRG  $G$  is unpredictable at pos.  $i$   
then  $G$  is a secure PRG.

If next-bit predictors cannot distinguish  $G$  from random  
then no statistical test can !!

Let  $G:K \rightarrow \{0,1\}^n$  be a PRG such that  
from the last  $n/2$  bits of  $G(k)$   
it is easy to compute the first  $n/2$  bits.

Is  $G$  predictable for some  $i \in \{0, \dots, n-1\}$  ?

- Yes 
- No

# More Generally

Let  $P_1$  and  $P_2$  be two distributions over  $\{0,1\}^n$

Def: We say that  $P_1$  and  $P_2$  are

**computationally indistinguishable** (denoted  $P_1 \approx_p P_2$ )

if  $\forall$  "eff" stat. tests  $A$

$$\left| \Pr_{x \leftarrow P_1} [A(x)=1] - \Pr_{x \leftarrow P_2} [A(x)=1] \right| < \text{negligible}$$

Example: a PRG is secure if  $\{ k \xleftarrow{R} K : G(k) \} \approx_p \text{uniform}(\{0,1\}^n)$

# End of Segment



# Stream ciphers

---

## Semantic security

Goal: secure PRG  $\Rightarrow$  “secure” stream cipher

# What is a secure cipher?

Attacker's abilities: **obtains one ciphertext** (for now)

Possible security requirements:

attempt #1: **attacker cannot recover secret key**

$$E(k, m) = m$$

attempt #2: **attacker cannot recover all of plaintext**

$$E(k, m_0 \parallel m_1) = m_0 \parallel m_1 \oplus k$$

Recall Shannon's idea:

**CT should reveal no “info” about PT**

# Recall Shannon's perfect secrecy

Let  $(E, D)$  be a cipher over  $(K, M, C)$

$(E, D)$  has perfect secrecy if  $\forall m_0, m_1 \in M \quad ( |m_0| = |m_1| )$

$$\{ E(k, m_0) \} = \{ E(k, m_1) \} \quad \text{where } k \leftarrow K$$

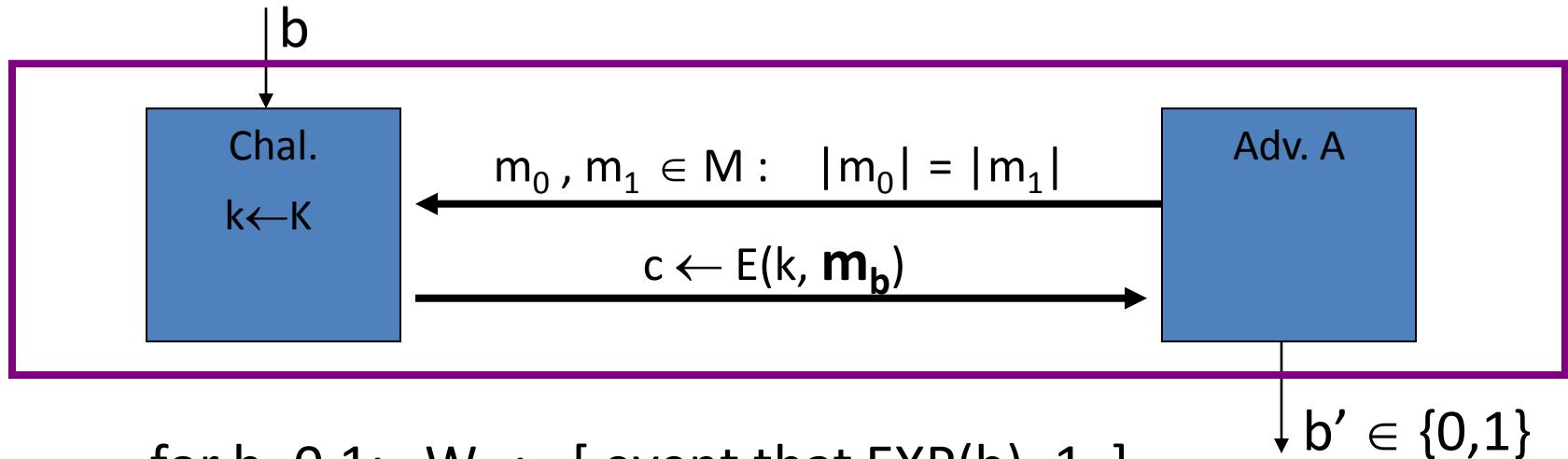
$(E, D)$  has perfect secrecy if  $\forall m_0, m_1 \in M \quad ( |m_0| = |m_1| )$

$$\{ E(k, m_0) \} \approx_p \{ E(k, m_1) \} \quad \text{where } k \leftarrow K$$

... but also need adversary to exhibit  $m_0, m_1 \in M$  explicitly

# Semantic Security (one-time key)

For  $b=0,1$  define experiments  $\text{EXP}(0)$  and  $\text{EXP}(1)$  as:



$$\text{Adv}_{\text{SS}}[A, E] := \left| \Pr[W_0] - \Pr[W_1] \right| \in [0,1]$$

# Semantic Security (one-time key)

Def:  $E$  is **semantically secure** if for all efficient  $A$

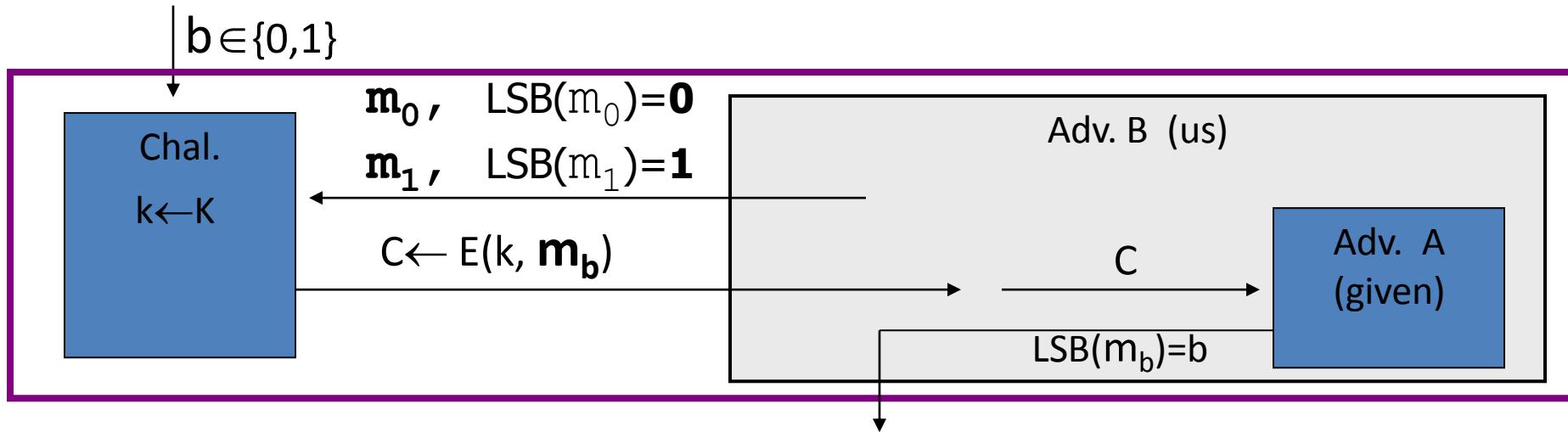
$\text{Adv}_{\text{SS}}[A, E]$  is negligible.

$\Rightarrow$  for all explicit  $m_0, m_1 \in M$ :  $\{E(k, m_0)\} \approx_p \{E(k, m_1)\}$

# Examples

Suppose efficient A can always deduce LSB of PT from CT.

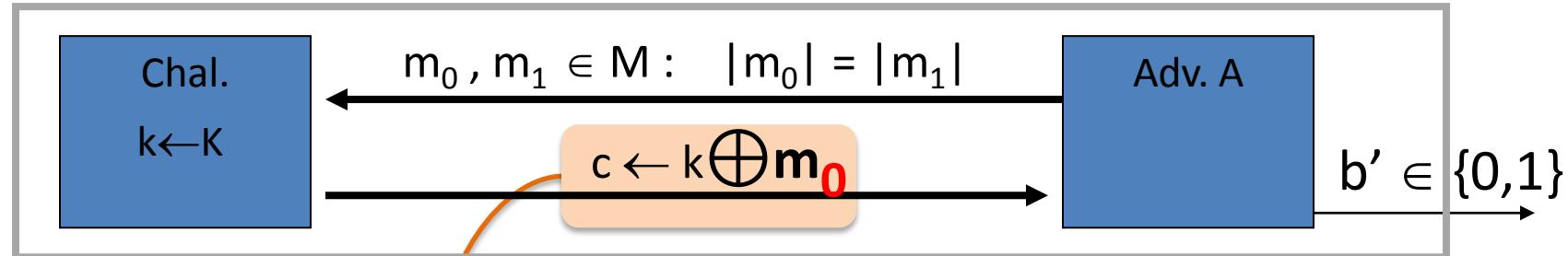
⇒  $\mathbb{E} = (E, D)$  is not semantically secure.



Then  $\text{Adv}_{\text{SS}}[B, \mathbb{E}] = \left| \Pr[\text{EXP}(0)=1] - \Pr[\text{EXP}(1)=1] \right| =$

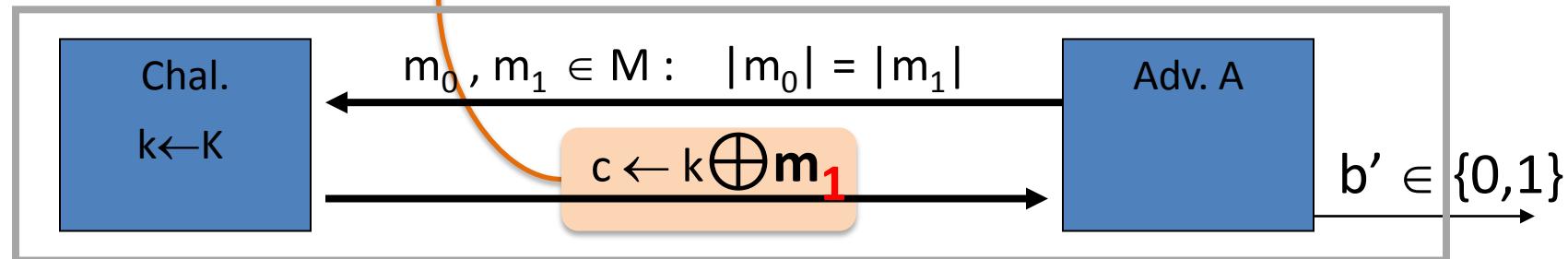
# OTP is semantically secure

$\text{EXP}(0)$ :



identical distributions

$\text{EXP}(1)$ :



For all A:  $\text{Adv}_{\text{SS}}[A, \text{OTP}] = \left| \Pr[A(k \oplus m_0) = 1] - \Pr[A(k \oplus m_1) = 1] \right|$



# End of Segment



## Stream ciphers

---

Stream ciphers are semantically secure

Goal: secure PRG  $\Rightarrow$  semantically secure stream cipher

# Stream ciphers are semantically secure

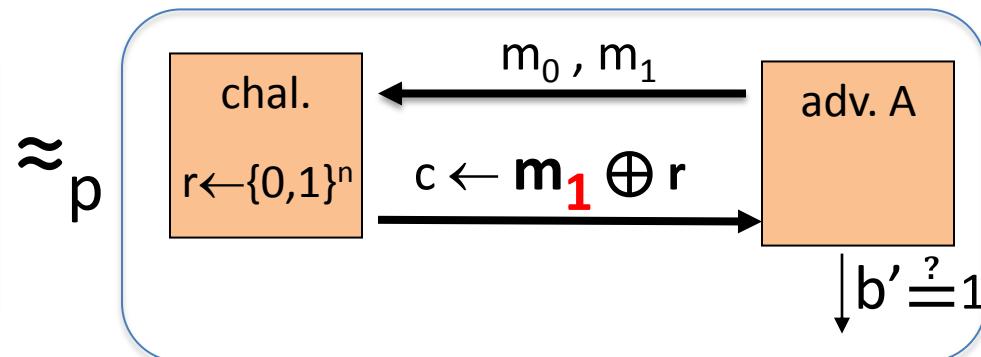
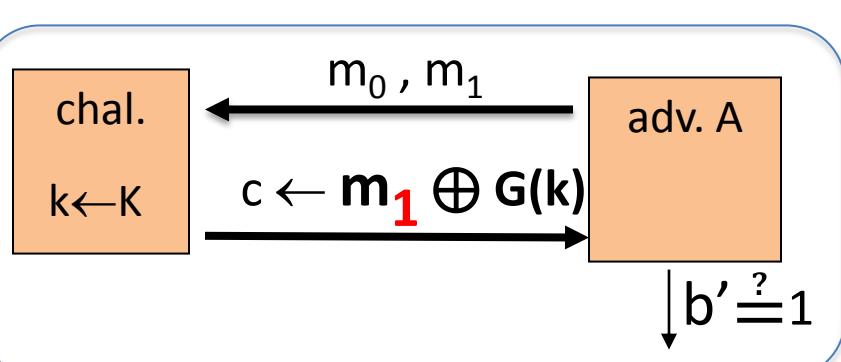
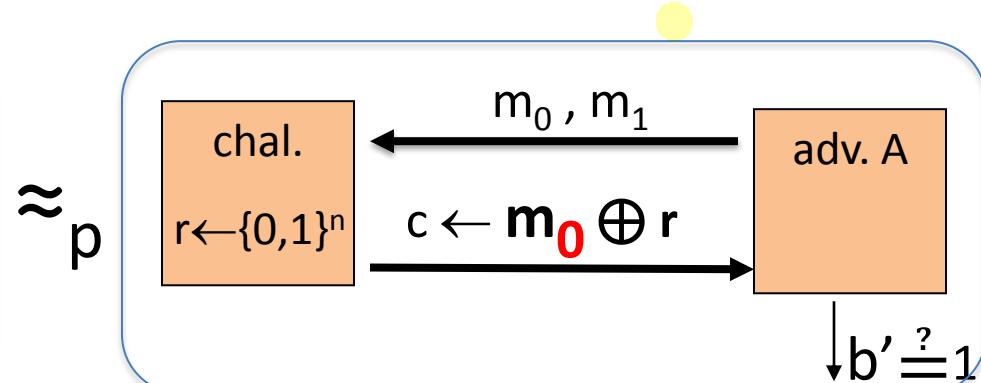
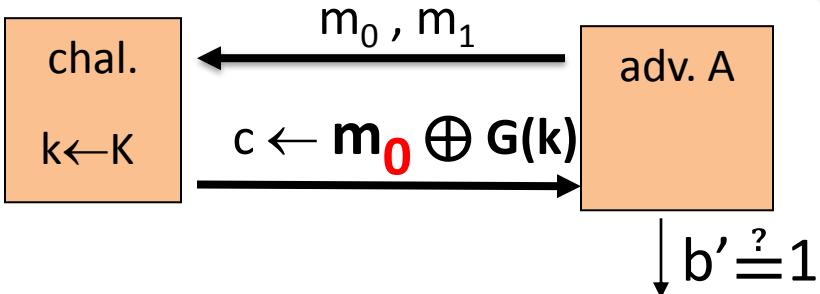
Thm:  $G:K \rightarrow \{0,1\}^n$  is a secure PRG  $\Rightarrow$

stream cipher E derived from G is sem. sec.

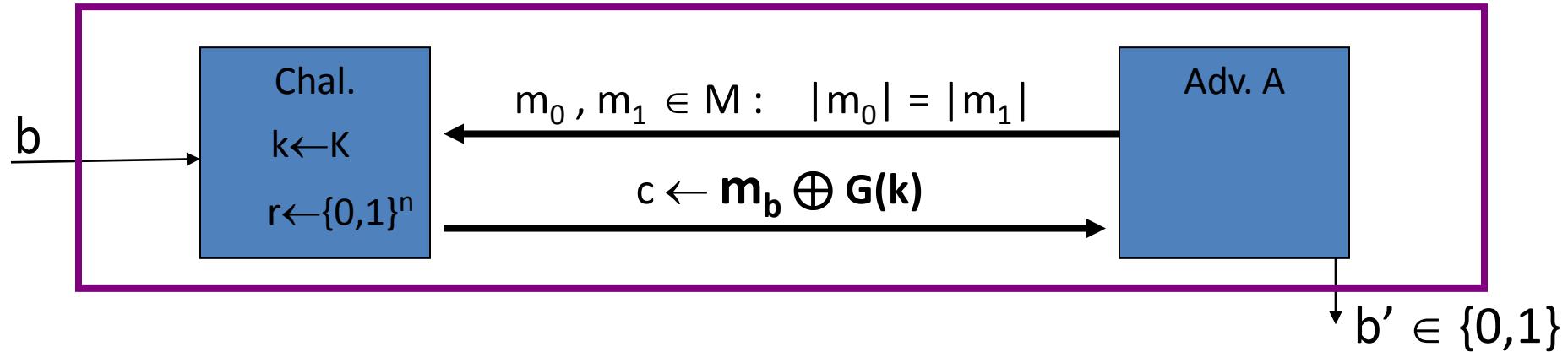
$\forall$  sem. sec. adversary A ,  $\exists$  a PRG adversary B s.t.

$$\text{Adv}_{\text{SS}}[A, E] \leq 2 \cdot \text{Adv}_{\text{PRG}}[B, G]$$

# Proof: intuition



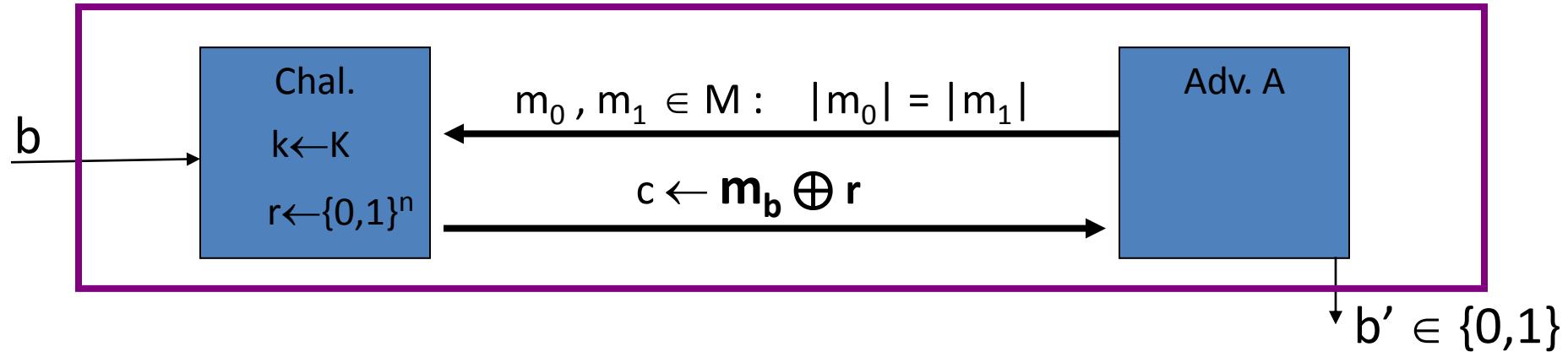
Proof: Let A be a sem. sec. adversary.



For  $b=0,1$ :  $W_b := [ \text{event that } b'=1 ]$ .

$$\text{Adv}_{SS}[A, E] = | \Pr[W_0] - \Pr[W_1] |$$

Proof: Let A be a sem. sec. adversary.



For  $b=0,1$ :  $W_b := [ \text{event that } b'=1 ]$ .

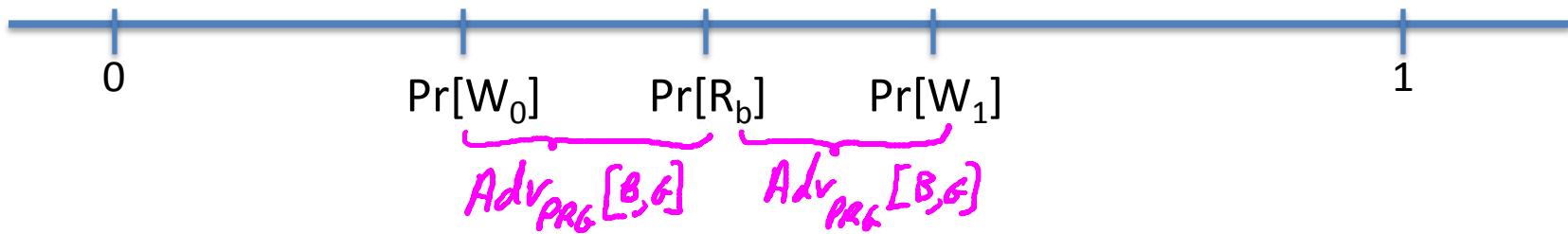
$$\text{Adv}_{SS}[A, E] = | \Pr[W_0] - \Pr[W_1] |$$

For  $b=0,1$ :  $R_b := [ \text{event that } b'=1 ]$

Proof: Let A be a sem. sec. adversary.

Claim 1:  $|\Pr[R_0] - \Pr[R_1]| = \text{Adv}_{\text{SS}}[A, \text{OTP}] = 0$

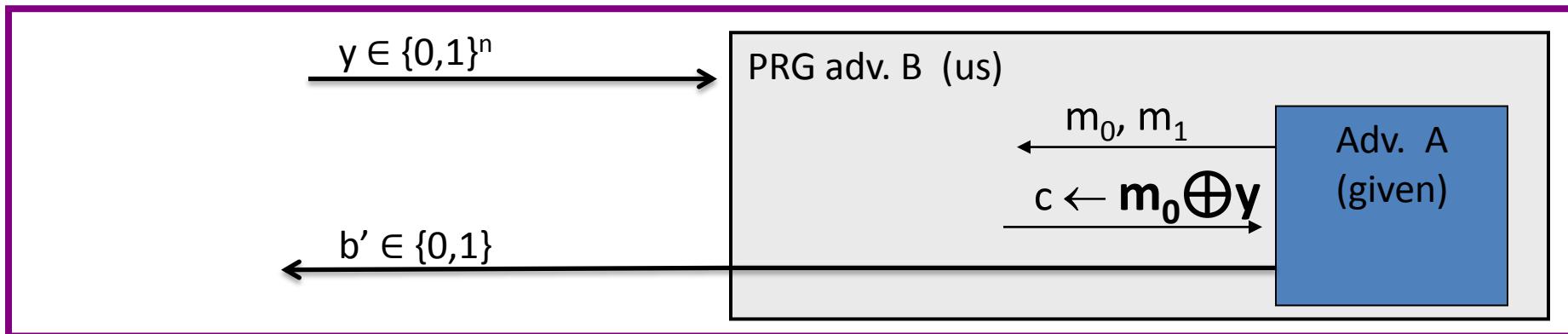
Claim 2:  $\exists B: |\Pr[W_b] - \Pr[R_b]| = \text{Adv}_{\text{PRG}}[B, G]$  For  $b=0, 1$



$$\Rightarrow \text{Adv}_{\text{SS}}[A, E] = |\Pr[W_0] - \Pr[W_1]| \leq 2 \cdot \text{Adv}_{\text{PRG}}[B, G]$$

Proof of claim 2:  $\exists B: |\Pr[W_0] - \Pr[R_0]| = \text{Adv}_{\text{PRG}}[B, G]$

Algorithm B:



$$\text{Adv}_{\text{PRG}}[B, G] = \left| \Pr_{r \leftarrow \{0,1\}^n} [B(r) = 1] - \Pr_{k \leftarrow \mathcal{R}} [B(g(k)) = 1] \right| = \left| \Pr[R_0] - \Pr[W_0] \right|$$

# End of Segment