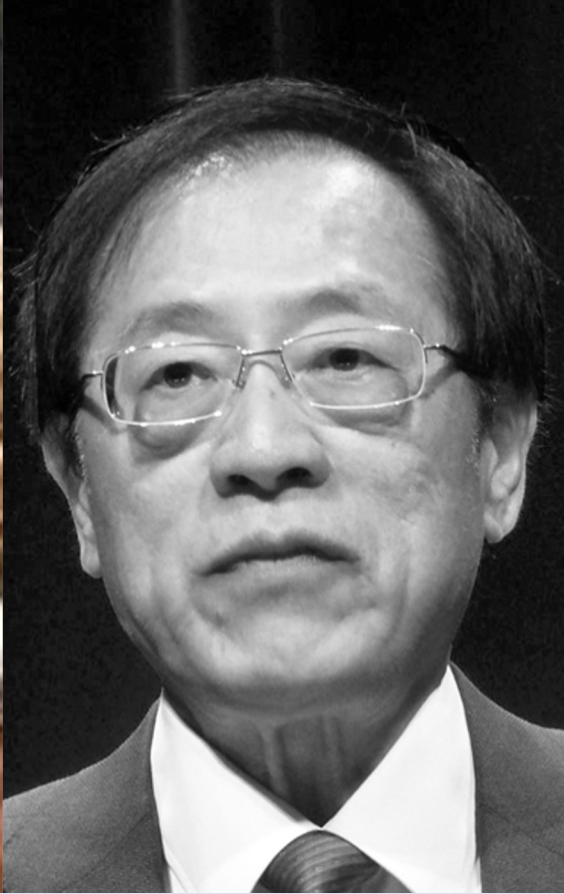


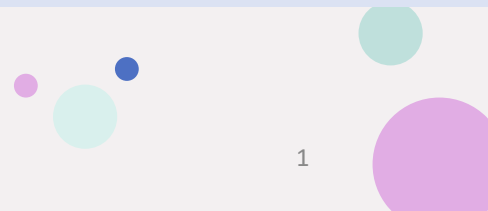


Lecture 5 (Lec 4 cont...)

Pseudorandom Generators, Stream Ciphers

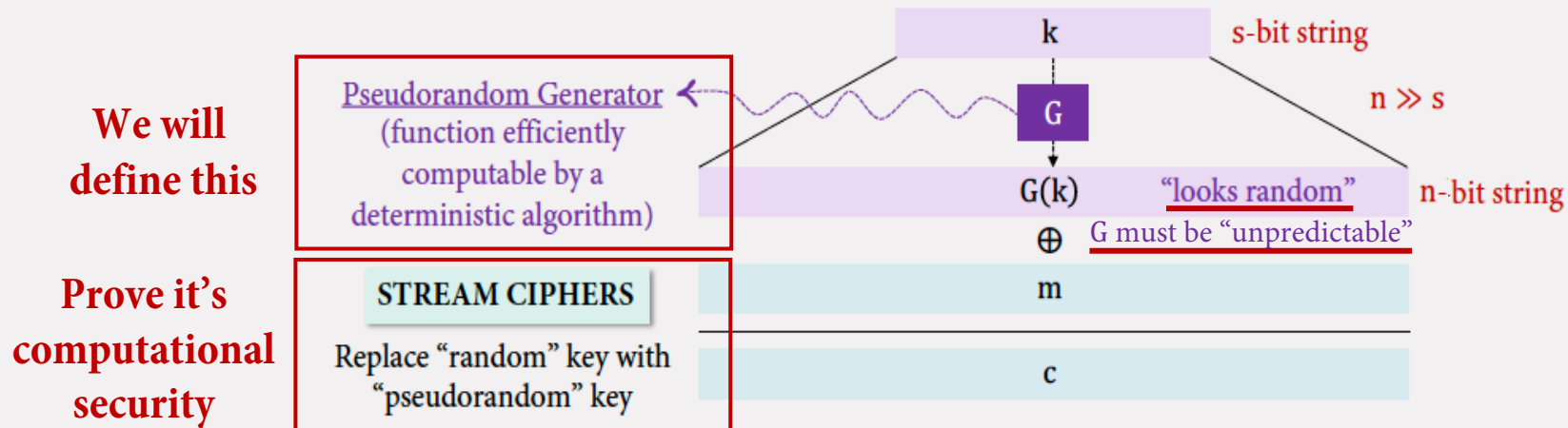
CS409/6011: Introduction to Cryptography

Sruthi Sekar



Recap

- Two Relaxations to overcome drawbacks of perfect security
 - Efficient Eve (Eve $\in$ **PPT**)
 - Eve can succeed but only with negligible probability.
- Computational Indistinguishability of Symmetric Key Encryption (SKE).
- We had a candidate construction in mind for computational SKE in [Lecture 2](#)



Today



Pseudorandom Generators
(PRGs)

Definitions



Computationally Secure
SKE

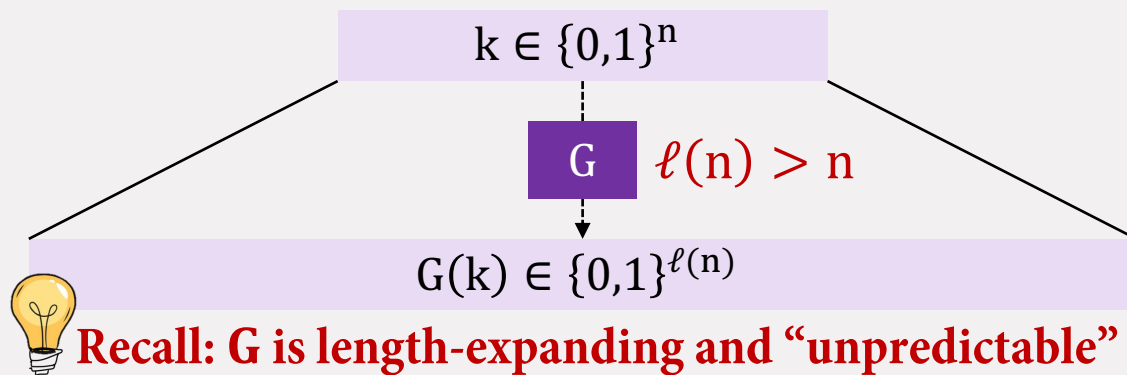
**(Security Proof
by Reduction)**



Building PRGs

Pseudorandom Generators

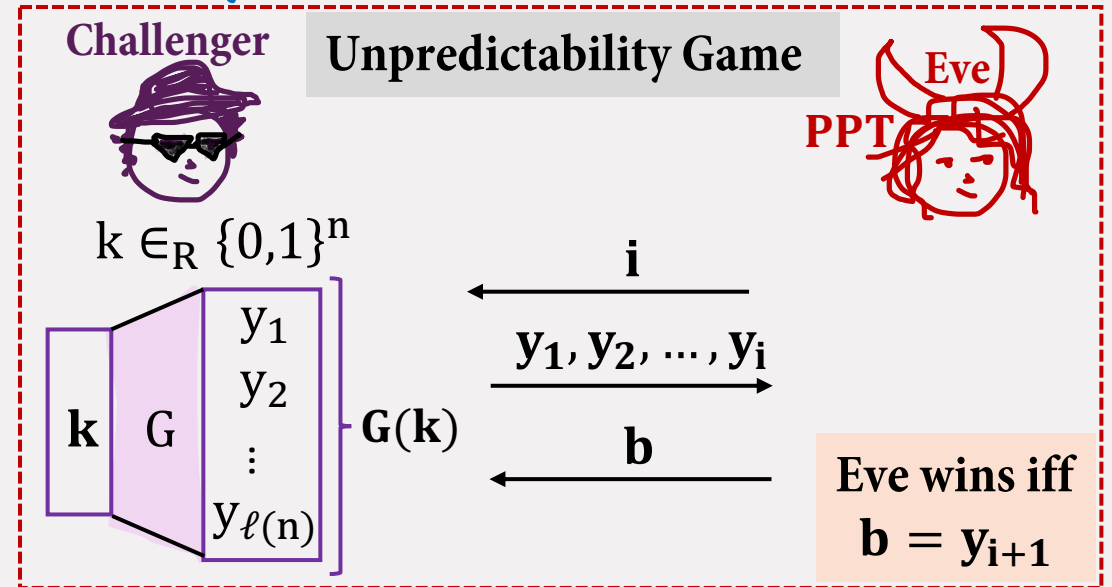
Unpredictability



Definition 1: Unpredictability

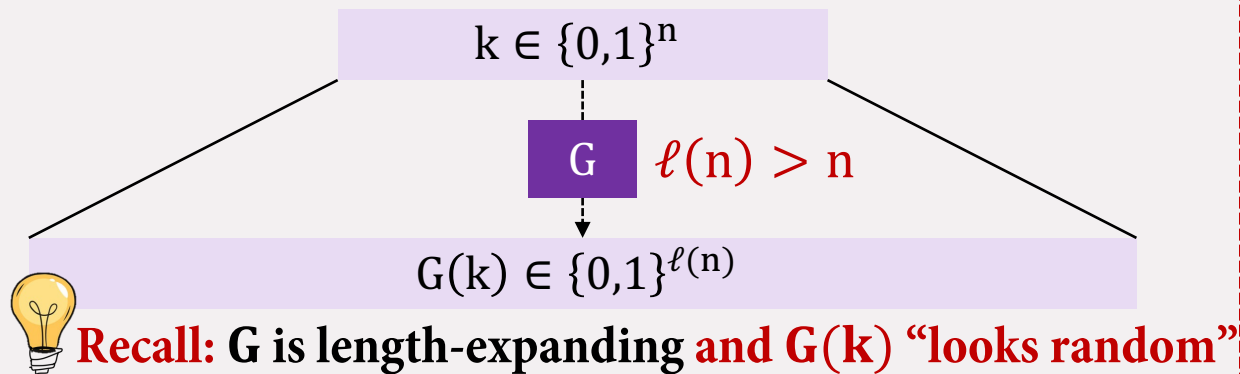
Let ℓ be a polynomial and let G be a **deterministic polynomial-time algorithm** such that for any n , and any $k \in \{0,1\}^n$, $G(k) \in \{0,1\}^{\ell(n)}$, where $\ell(n) > n$ (expanding). We say G is **next-bit unpredictable** if for every **PPT Eve**, there exists a negligible function **negl** such that, $\forall i$

$$\Pr_{k \in_R \{0,1\}^n} [\text{Eve}(\underbrace{G(k)|_{1,2,\dots,i}}_{\text{First } i \text{ bits of } G(k)}) = \underbrace{G(k)|_{i+1}}_{(i+1)\text{-th bit of } G(k)}] \leq 1/2 + \text{negl}(n)$$



Pseudorandom Generators

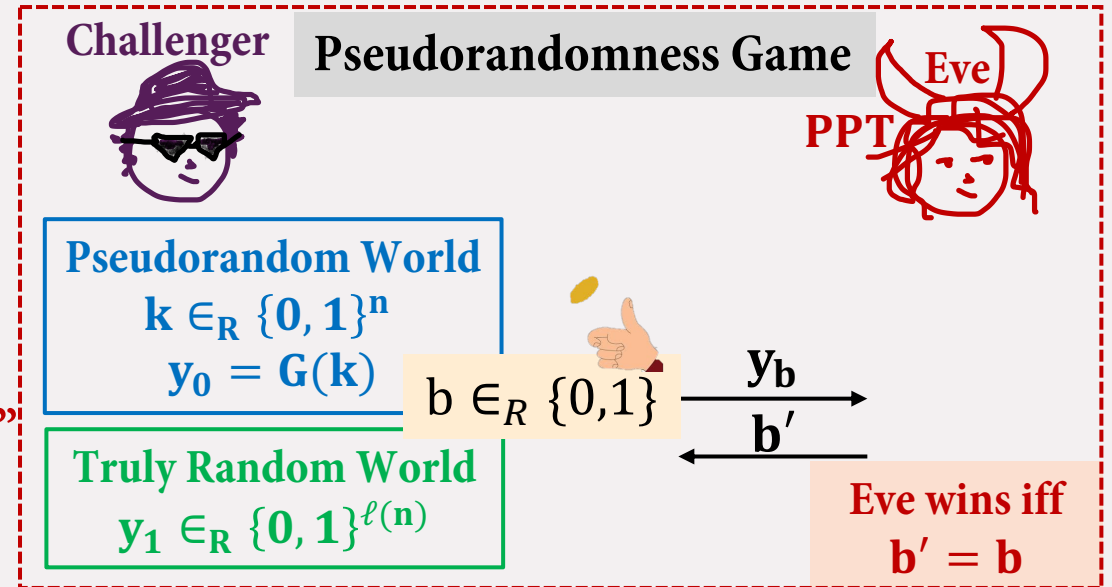
Pseudorandomness



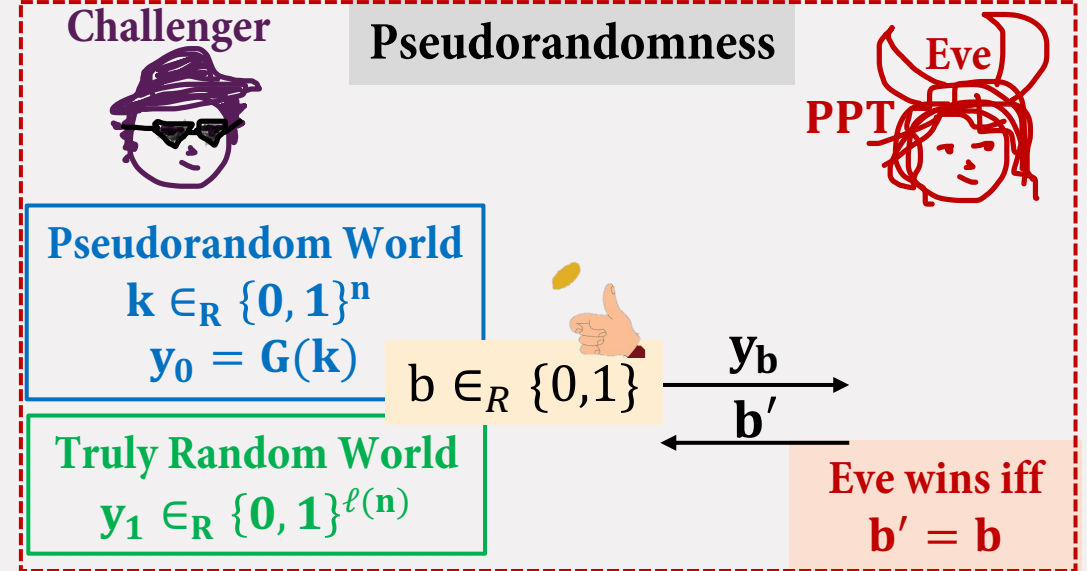
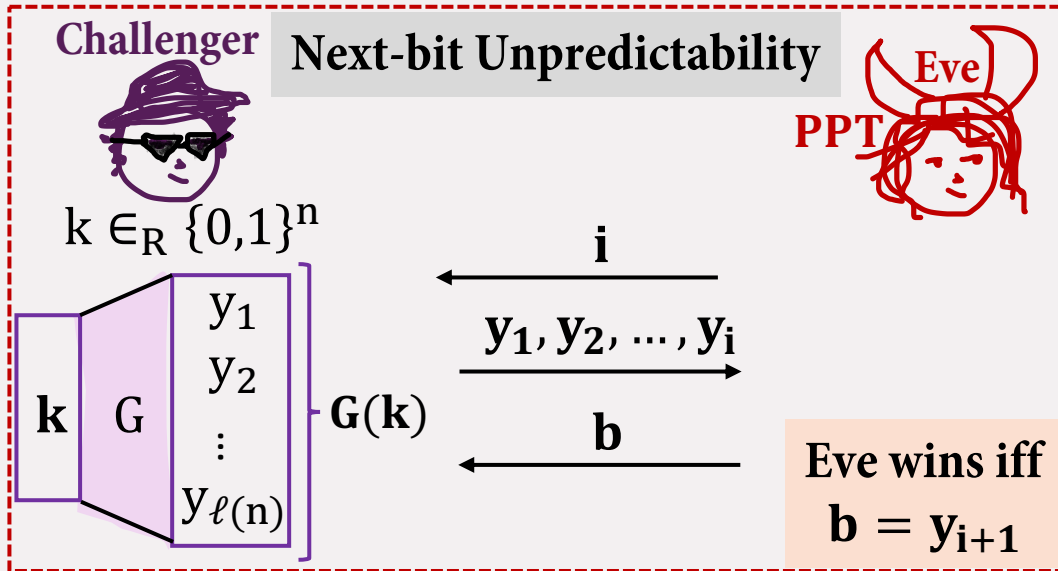
Definition 2: Pseudorandomness

Let ℓ be a polynomial and let G be a **deterministic polynomial-time algorithm** such that for any n , and any $k \in \{0,1\}^n$, $G(k) \in \{0,1\}^{\ell(n)}$, where $\ell(n) > n$ (expanding). We say G is **pseudorandom** if for every **PPT Eve**, there exists a negligible function **negl** such that,

$$\left| \Pr_{k \in_R \{0,1\}^n} [\text{Eve}(G(k)) = 1] - \Pr_{r \in_R \{0,1\}^{\ell(n)}} [\text{Eve}(r) = 1] \right| \leq \text{negl}(n)$$



Quick Quiz

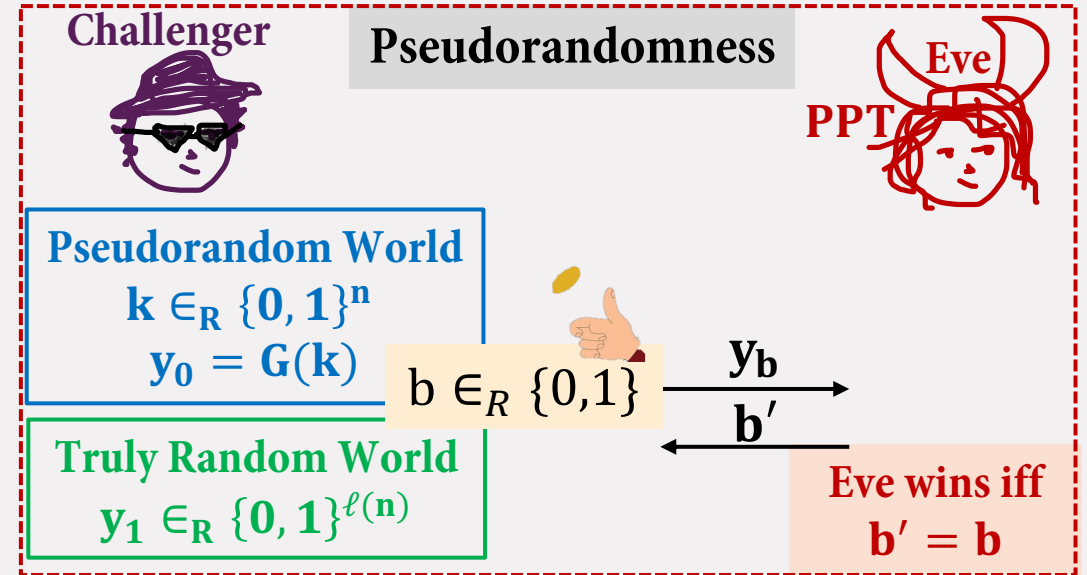
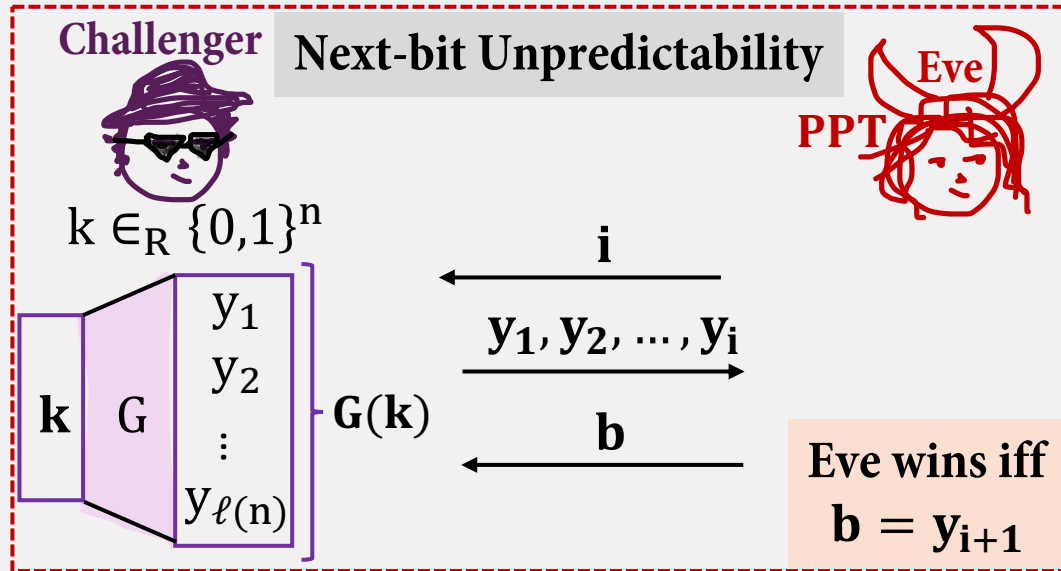


What if Eve is unbounded?

What if $P = NP$?

Neither definition will hold!

Quick Quiz



PRG or not? (G_1, G_2 are PRGs)

$$G(k) = kk', k' = k_1 \oplus k_2 \oplus \dots \oplus k_n$$

$$G(k) = G_1(k)0$$

$$G(k) = G_1(k) \oplus G_2(k)$$

$$G(k) = G_1(k)G_2(k)$$

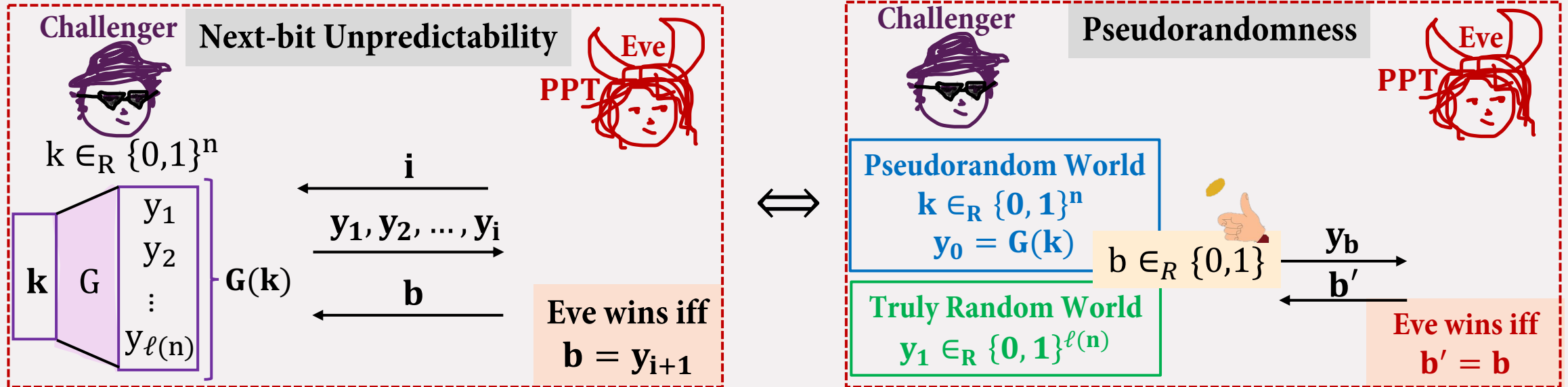
$$G(k_1 k_2) = G_1(k_1)G_2(k_2)$$

This just means concatenation.

$$G(k) = y_{\ell(n)} y_{\ell(n)-1} \dots y_1, \text{ where } G_1(k) = y_1 y_2 \dots y_{\ell(n)}$$

HW 1: Formally prove this!

PRGs: Equivalence of Defs 1 and 2



Theorem 1 [**Definition 1** $\Leftrightarrow$ **Definition 2**]

Let ℓ be a polynomial and let G be a **deterministic polynomial-time algorithm** such that for any n , and any $k \in \{0,1\}^n$, $G(k) \in \{0,1\}^{\ell(n)}$, where $\ell(n) > n$ (expanding).

G is pseudorandom if and only if G is next-bit unpredictable.

Pseudorandom Generators

Pseudorandomness $\Rightarrow$ Next-bit Unpredictability

Theorem 1A

Let ℓ be a polynomial and let G be a deterministic polynomial-time algorithm such that for any n , and any $k \in \{0,1\}^n$, $G(k) \in \{0,1\}^{\ell(n)}$, where $\ell(n) > n$ (expanding).

If G is pseudorandom, then G is next-bit unpredictable.

Proof: Assume to the contrary that G is predictable.

To show: G is not pseudorandom

$\exists \text{ Eve}^{\text{predictor}}$ such that $\exists$ some i for which:

$$\Pr[\text{Eve}^{\text{predictor}}(G(k)|_{1,2,\dots,i}) = G(k)|_{i+1}] > \frac{1}{2} + \text{non-negl}$$

What is non-negligible?

μ is non-negligible if $\exists$ polynomial p such that
 $\forall n_0 \in \mathbb{N}, \exists n \geq n_0$ such that $\mu(n) > 1/p(n)$

Pseudorandom Generators

Pseudorandomness $\Rightarrow$ Next-bit Unpredictability

Theorem 1A

Let ℓ be a polynomial and let G be a deterministic polynomial-time algorithm such that for any n , and any $k \in \{0,1\}^n$, $G(k) \in \{0,1\}^{\ell(n)}$, where $\ell(n) > n$ (expanding).

If G is pseudorandom, then G is next-bit unpredictable.

Proof: Assume to the contrary that G is predictable.

To show: G is not pseudorandom

$\exists \text{ Eve}^{\text{predictor}}$ such that $\exists$ some i for which:

$$\Pr[\text{Eve}^{\text{predictor}}(G(k)|_{1,2,\dots,i}) = G(k)|_{i+1}] > \frac{1}{2} + \text{non-negl}$$

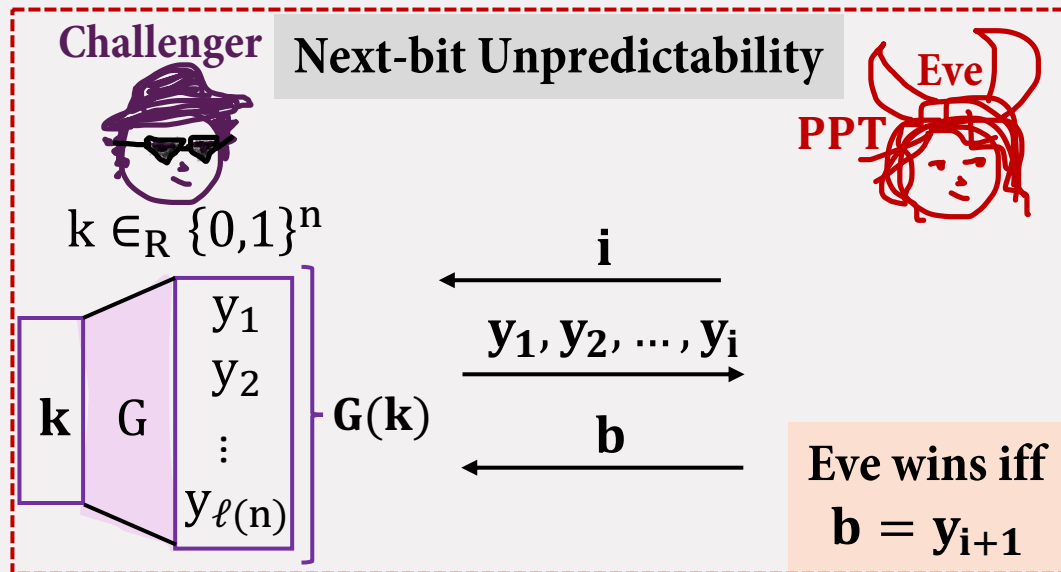
$\text{Eve}^{\text{pseudorandom}}(\mathbf{y})$:

- If $\text{Eve}^{\text{predictor}}(y_1, y_2, \dots, y_i) = y_{i+1}$, output 1
- Else output 0

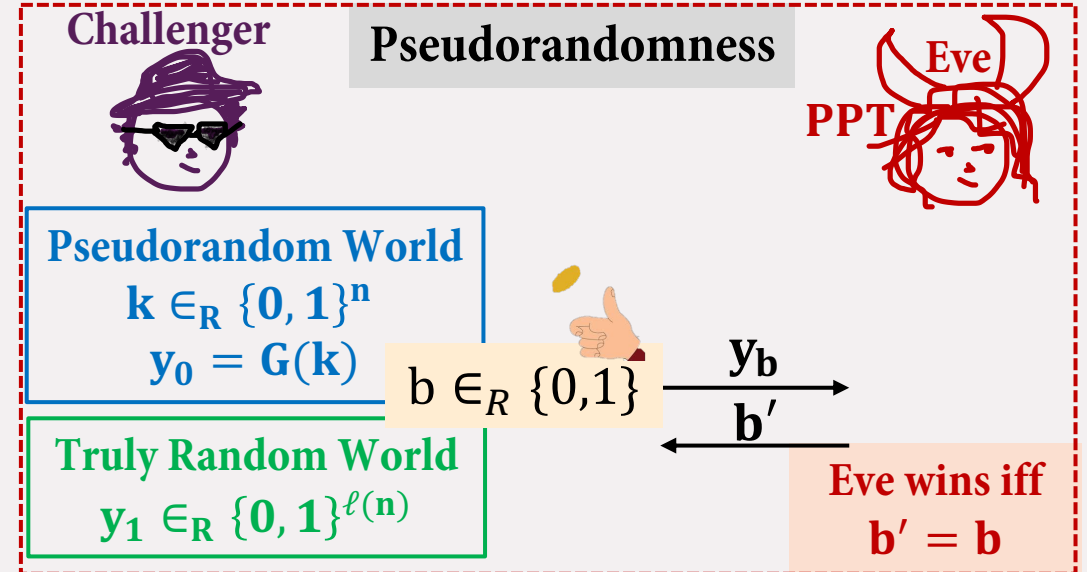
$$\begin{aligned} & \Pr_{k \in_R \{0,1\}^n} [\text{Eve}^{\text{pseudorandom}}(G(k)) = 1] - \Pr_{r \in_R \{0,1\}^{\ell(n)}} [\text{Eve}^{\text{pseudorandom}}(r) = 1] > \cancel{\frac{1}{2}} + \text{non-negl} - \cancel{\frac{1}{2}} \\ & \quad = \Pr[\text{Eve}^{\text{predictor}}(G(k)|_{1,2,\dots,i}) = G(k)|_{i+1}] > \frac{1}{2} + \text{non-negl} \\ & \quad = \Pr[\text{Eve}^{\text{predictor}}(r_1, \dots, r_i) = r_{i+1}] = \frac{1}{2} \end{aligned}$$

Pseudorandom Generators

Next-bit Unpredictability $\Rightarrow$ Pseudorandomness



$\Leftrightarrow$



Theorem 1B [Definition 1 $\Rightarrow$ Definition 2]

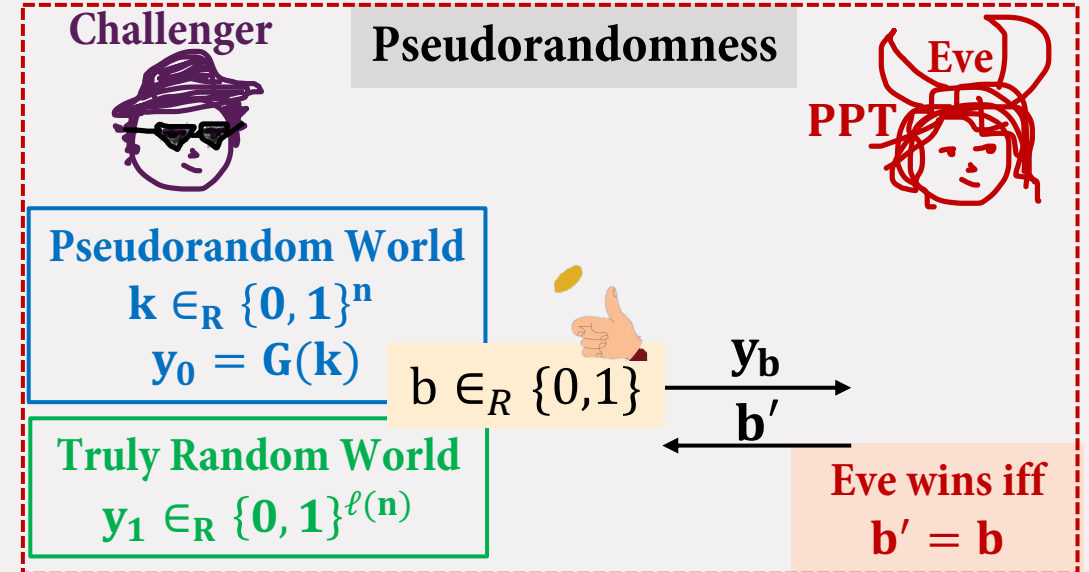
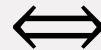
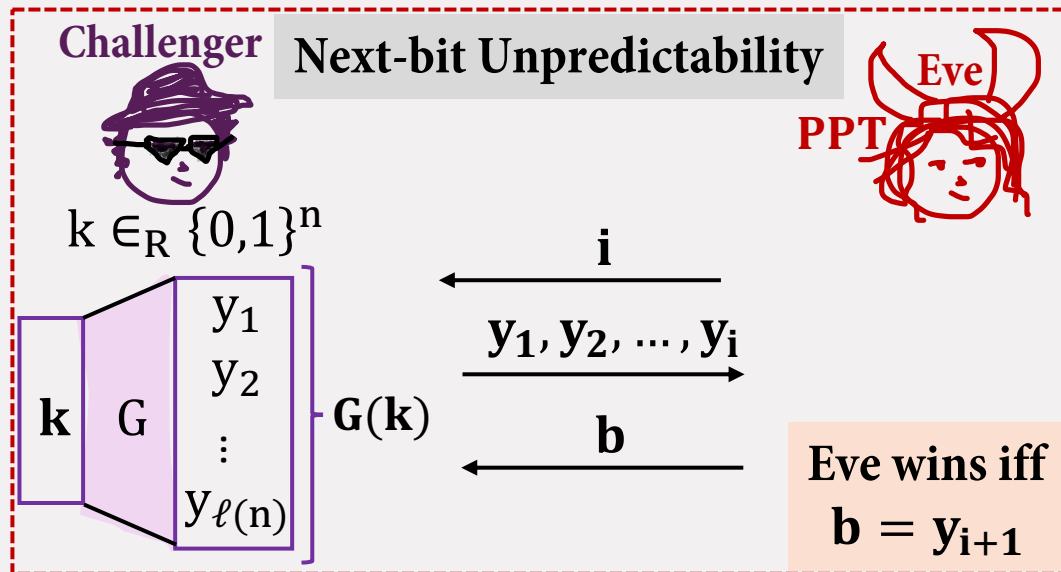
Let ℓ be a polynomial and let G be a deterministic polynomial-time algorithm such that for any n , and any $k \in \{0,1\}^n$, $G(k) \in \{0,1\}^{\ell(n)}$, where $\ell(n) > n$ (expanding).

If G is next-bit unpredictable, then G is pseudorandom.

This direction is a little harder to prove. Can Skip! If interested, see [Goldreich Vol I, Ch3]

Pseudorandom Generators

Next-bit Unpredictability $\Rightarrow$ Pseudorandomness



Theorem 1B [Definition 1 $\Rightarrow$ Definition 2]

Let ℓ be a polynomial and let G be a deterministic polynomial-time algorithm such that for any n , and any $k \in \{0,1\}^n$, $G(k) \in \{0,1\}^{\ell(n)}$, where $\ell(n) > n$ (expanding).

If G is next-bit unpredictable, then G is pseudorandom.

HW 2: Do other notions of unpredictability, say “first-bit unpredictability” imply pseudorandomness?

Today



Pseudorandom Generators
(PRGs)
(Definitions)



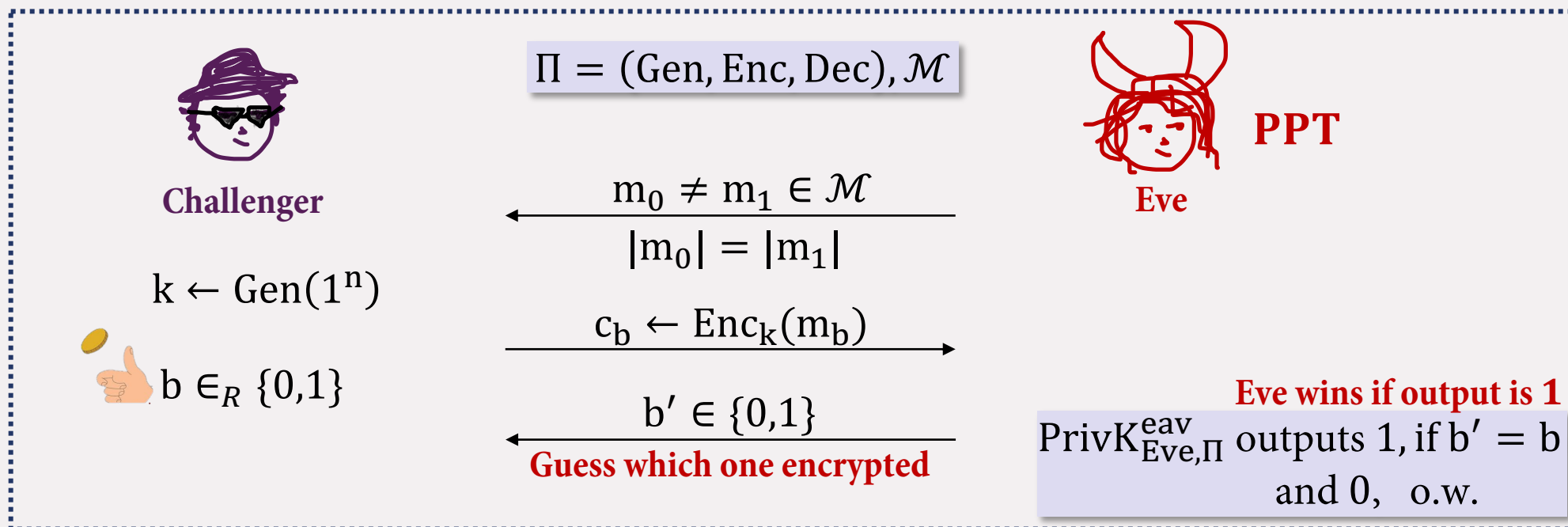
Computationally Secure
SKE
(Security Proof by
Reduction)



Building PRGs

RECALL

Computational Indistinguishability



Definition 3

An encryption scheme $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ with message space $\mathcal{M}$ is **computationally indistinguishable against an eavesdropper** if, for all **PPT** adversary **Eve**,

there exists a negligible function **negl** such that

$$\Pr[\text{PrivK}_{\text{Eve}, \Pi}^{\text{eav}}(n) = 1] \leq 1/2 + \text{negl}(n)$$

Computational One Time Pad

$$\mathcal{K} = \{0,1\}^n, \mathcal{M} = \{0,1\}^{\ell(n)}, \mathcal{C} = \{0,1\}^{\ell(n)}$$

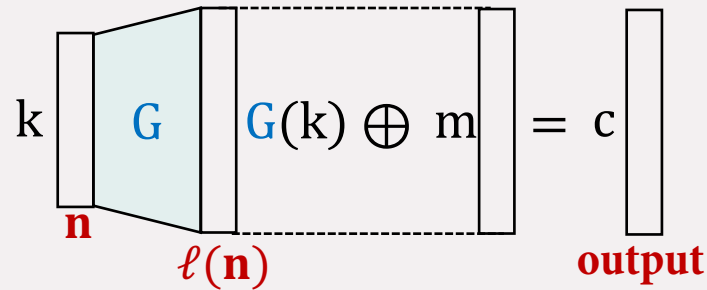
Building Block

Let $G: \{0,1\}^n \rightarrow \{0,1\}^{\ell(n)}$ be a PRG.
PRG Expansion $\Rightarrow \ell(n) > n$

Gen(1^n):

- Pick $k \in_R \{0,1\}^n$
- Output k

Enc(k, m):



Dec(k, c):

- Output $G(k) \oplus c$

Construction 1

Correctness

$$\Pr_{k \leftarrow \text{Gen}(1^n)} [\text{Dec}(k, \text{Enc}(k, m)) = G(k) \oplus (G(k) \oplus m) = m] = 1, \quad \forall m \in \{0,1\}^{\ell(n)}$$

Proof of Computational Security

Theorem 2

If G is a PRG, then Construction 1 is computationally secure.

Proof by Reduction:

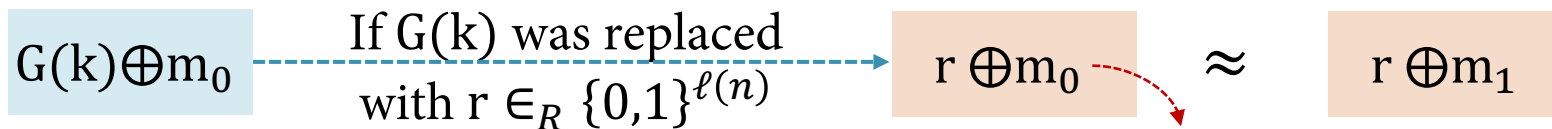
$\exists$ Eve breaking Construction 1 $\Rightarrow \exists$ Eve^{PRG} against G

$$\Pr[\text{PrivK}_{\text{Eve}, \Pi}^{\text{eav}}(n) = 1] > \frac{1}{2} + \text{non-negl} \quad \text{--- [Eq 1]}$$



INTUITION: Logical conclusions

REDUCTION:
LET'S FORMALIZE
THIS!



If [Eq 1] is true, then Eve must implicitly be distinguishing $G(k)$ from $r \in_R \{0,1\}^{\ell(n)}$

One-time pad
We know that Eve can only distinguish this with probability $1/2$

Proof of Computational Security

Theorem 2

If G is a PRG, then Construction 1 is computationally secure.

Proof by Reduction:

$\exists$ Eve breaking Construction 1 $\Rightarrow \exists$ Eve^{PRG} against G (break pseudorandomness)

$$\Pr[\text{PrivK}_{\text{Eve}, \Pi}^{\text{eav}}(n) = 1] > \frac{1}{2} + \text{non-negl} \quad \text{--- [Eq 1]}$$

Eve^{PRG}

Input: $y \in \{0,1\}^{\ell(n)}$

Goal: output 1 in pseudorandom world with a non-negligible advantage over the random world.

Output: 1, if $b' = b$
0, otherwise

Act as challenger
and run Eve

Eve^{PRG}/Challenger

Eve

$\leftarrow m_0, m_1$

$b \in_R \{0,1\}$ $\xrightarrow{c_b := y \oplus m_b}$

$\leftarrow b'$

LET'S ANALYZE THE PROBABILITIES...

Proof of Computational Security

Theorem 2

If G is a PRG, then Construction 1 is computationally secure.

Proof by Reduction: $\exists$ Eve breaking Construction 1 $\Rightarrow \exists$ Eve^{PRG} against G (break pseudorandomness)

$$\Pr[\text{PrivK}_{\text{Eve}, \Pi}^{\text{eav}}(n) = 1] > \frac{1}{2} + \text{non} - \text{negl} \quad \text{— [Eq 1]}$$

$$\Pr_{k \in_R \{0,1\}^n} [\text{Eve}^{\text{PRG}}(\underbrace{G(k)}_{\text{actual encrypted ciphertext under Enc}}) = 1] = \Pr[\text{PrivK}_{\text{Eve}, \Pi}^{\text{eav}}(n) = 1] > \frac{1}{2} + \text{non} - \text{negl} \quad (\text{by [Eq 1]})$$

$\Rightarrow$ Eve gets actual encrypted ciphertext under Enc

$$\Pr_{r \in_R \{0,1\}^{\ell(n)}} [\text{Eve}^{\text{PRG}}(\underbrace{r}_{\text{one-time pad ciphertext}}) = 1] = \Pr[\text{Eve}(r \oplus m_b) = b] = \frac{1}{2}$$

$\Rightarrow$ Eve gets one-time pad ciphertext

$$\therefore \Pr_{k \in_R \{0,1\}^n} [\text{Eve}^{\text{PRG}}(G(k)) = 1] - \Pr_{r \in_R \{0,1\}^{\ell(n)}} [\text{Eve}^{\text{PRG}}(r) = 1] > \cancel{\frac{1}{2}} + \text{non} - \text{negl} - \cancel{\frac{1}{2}}$$

Today



Pseudorandom Generators
(PRGs)
(Definitions)



Computationally Secure
SKE
(Security Proof by
Reduction)



Building PRGs

Ways to build PRGs

Two Approaches

Theoretical Constructions

Rely on well-studied hard problems!

E.g.:

PRG from average-case hardness of Subset-sum

$$G(a_1, \dots, a_n, x_1, \dots, x_n) = (a_1, \dots, a_n, \sum_i x_i a_i \bmod 2^{n+1})$$

a_i 's are $(n + 1)$ -bit numbers, x_i 's are bits.

If Subset-sum problem is hard, then

G is a PRG (expands by 1-bit).

Later in our course: One-way functions (OWF)

OWF $\Rightarrow$ PRGs

Practical Constructions

Start from a design framework
("Appropriately chosen" functions, composed
"appropriately many times" look random)

Build a candidate construction

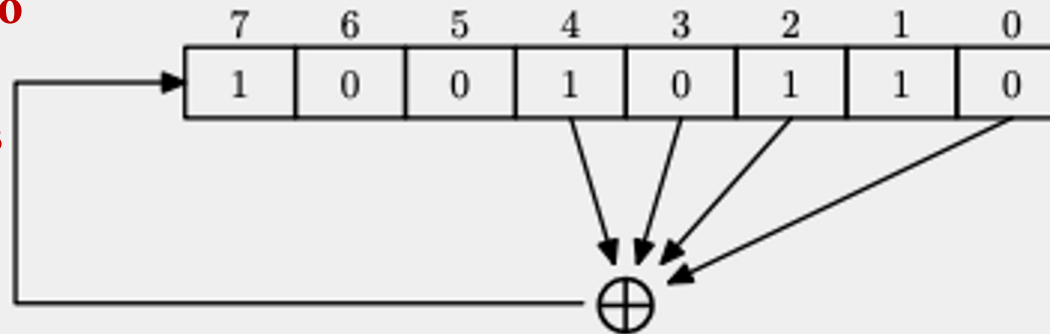
Do extensive cryptanalysis

E.g.: Salsa, ChaCha, LCG, CSS, RC4, etc.

[Ch3, Boneh and Shoup]

PRGs in the Wild

Shift all bits to
right and
insert this as
left most bit



0110100100010100...

The output bitstream
(keep outputting the bit every time)

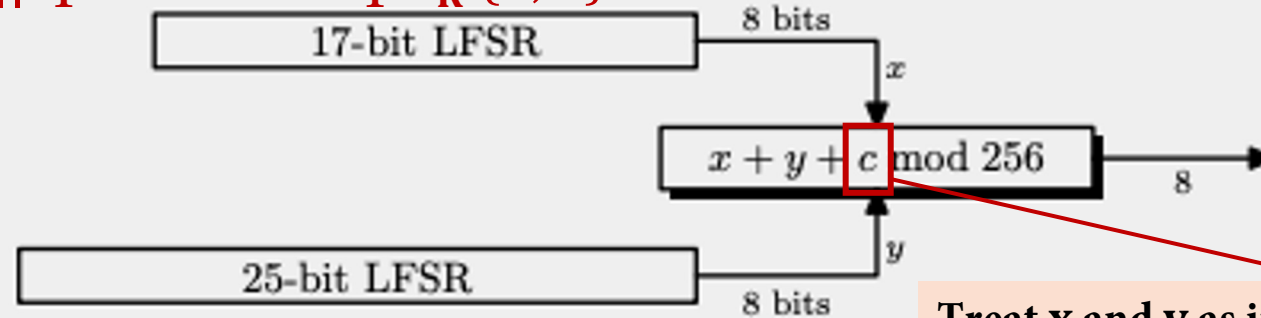
then XOR these locations

8-bit Linear Feedback Shift Register (LFSR) {4, 3, 2, 0}
(how they generate a “pseudorandom bit stream”)

(Predictable $\Rightarrow$ not Pseudorandom, but were used in Satellite TV ciphers in 1990s and early 2000s)

PRGs in the Wild

Starts with seed $1||s_1$ for random $s_1 \in_R \{0, 1\}^{16}$



Starts with seed $1||s_2$ for random $s_2 \in_R \{0, 1\}^{24}$

Treat x and y as integers in $0, \dots, 255$.
 c denotes the carry bit: initiate with $c := 0$
and if $x + y > 255$, then set $c = 1$ for
next round, else $c = 0$.

Content Scrambling System (CSS), 1980s: used in DVD encryption
(how they generate a “pseudorandom bit stream”)

HW 3: The attack on LFSR no longer works here. Is this a secure PRG then? Cannot prove security!

And there is a better attack than brute-force over all seeds:

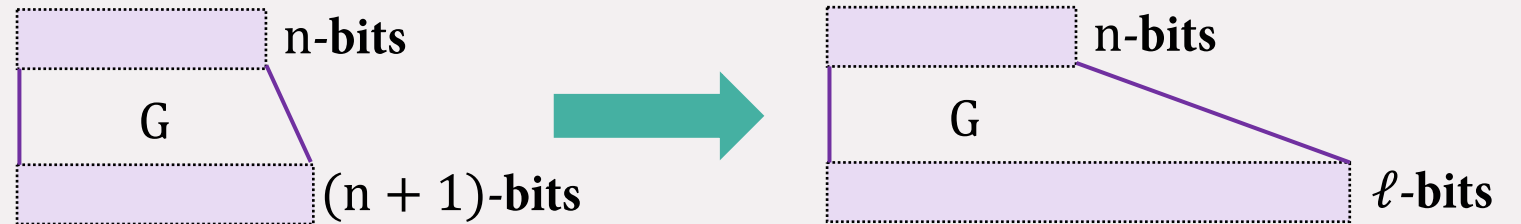
takes 2^{16} guesses to find the entire seed $s = s_1 || s_2$ –try before reading [\[3.8, Boneh Shoup\]](#)

Next Class

- Today , you saw **proof via reduction** (to the security of the underlying assumption). Next class, you will see another proof technique called **“hybrid argument”**

- More on building PRGs

- Length-Extension of PRGs





THANK YOU