# CIS 5560

# Cryptography
# Lecture 5

**Course website:**

pratyushmishra.com/classes/cis-5560-s25/

Slides adapted from Dan Boneh and Vinod Vaikuntanathan

# Announcements

- **HW 1 is out;** due Friday, Feb 7 at 5PM on Gradescope
  - Covers PRGs, OTPs, and semantic security
  - Get started today and make use of office hours, HW party!

# Recap of last lecture

# PRG $\Longrightarrow$ Semantically Secure Encryption

(or, How to Encrypt n+1 bits using an n-bit key)

- Gen$(1^k) \rightarrow k$:
  - Sample an $n$-bit string at random.
- Enc$(k, m) \rightarrow c$:
  - Expand $k$ to an $n+1$-bit string using PRG: $s = G(k)$
  - Output $c = s \oplus m$
- Dec$(k, c) \rightarrow m$:
  - Expand $k$ to an $n+1$-bit string using PRG: $s = G(k)$
  - Output $m = s \oplus c$

## Correctness:

$Dec(k, c)$ outputs $G(k) \oplus c = G(k) \oplus G(k) \oplus m = m$

Distinguisher $D(y)$:

1. Get two messages $m_0, m_1$, from Eve and sample a bit $b$
2. Compute $b' \leftarrow \text{Eve}(y \oplus m_b)$
3. Output $b' = b$, output “0”
4. Otherwise, output “1”

**World 0**

$$\Pr[D \text{ outputs "0"} \mid b = 0 \ (y \text{ is pseudorandom})] = \Pr[\text{Eve outputs } b' = b \mid b = 0] = \rho \geq 1/2 + 1/p(n)$$

**World 1**

$$\Pr[D \text{ outputs "1"} \mid b = 1 \ (y \text{ is random})] = \Pr[\text{Eve outputs } b' = b \mid b = 1 = \rho' = 1/2$$

Therefore,

$$\left| \Pr[D \text{ outputs "PRG"} \mid y \text{ is pseudorandom}] - \Pr[D \text{ outputs "PRG"} \mid y \text{ is random}] \right| \geq 1/p(n)$$

# PRG $\Longrightarrow$ Semantically Secure Encryption

(or, How to Encrypt n+1 bits using an n-bit key)

**$Q1$:** Do PRGs exist?

(Exercise: If P=NP, PRGs do not exist.)

**$Q2$:** How do we encrypt longer messages or many messages with a fixed key?

(**Length extension**: If there is a PRG that stretches by one bit, there is one that stretches by polynomially many bits)

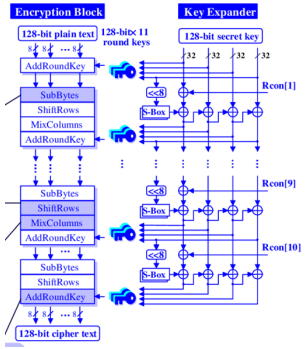
(**Pseudorandom functions**: PRGs with exponentially large stretch and "random access" to the output.)

# Constructing PRGs: Two Methodologies

## The Practical Methodology

**1. Start from a design framework**
(e.g. “appropriately chosen functions composed appropriately many times look random”)

**2. Come up with a candidate construction**

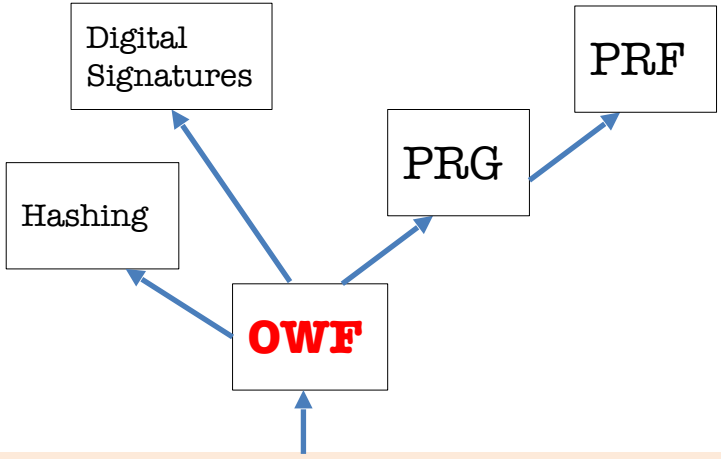

Rijndael
(now the Advanced Encryption Standard)

# Constructing PRGs: Two Methodologies

**The Foundational Methodology (much of this course)**

**Reduce to simpler primitives.**

**"Science wins either way" –Silvio Micali**




***well-studied***, average-case hard, problems

# One-way Functions: The Definition

A function (family) $\{F_n\}_{n\in\mathbb{N}}$ where $F(\cdot) : \{0,1\}^n \to \{0,1\}^{m(n)}$ is **one-way** if for every p.p.t. adversary $A$, the following holds:

$$\Pr\left[F_n(x') = y \;\middle|\; \begin{array}{r} x \leftarrow \{0,1\}^n \\ y := F_n(x) \\ x' \leftarrow A(1^n, y) \end{array}\right] = \mathrm{negl}(n)$$

- Can always find *an* inverse with unbounded time
- … but should be hard with probabilistic polynomial time

**One-way Permutations**:

One-to-one one-way functions with $m(n) = n$.

# OW**P** → PRG, Attempt #2

**Let $F : \{0,1\}^n \to \{0,1\}^n$ be a one-way permutation**

**Imagine there existed $B : \{0,1\}^n \to \{0,1\}$ such that the following was a PRG**

PRG($k$)

1. Output $F(k) \,||\, B(k)$

**What properties do we need of $B$?**

1. One-way: can't find $k$ from $B(k)$
2. Pseudorandom: $B(k)$ looks like a random bit
3. Unpredictable: $B(k)$ is unpredictable given $F(k)$

# Hardcore Bits

**HARDCORE PREDICATE**

For any $F: \{0,1\}^n \to \{0,1\}^m$, $B: \{0,1\}^n \to \{0,1\}$ is a **hardcore predicate** if for every efficient $A$, there is a negligible function $\mu$ s.t.

$$\Pr\left[b = B(x) \,\middle|\, \begin{matrix} x \leftarrow \{0,1\}^n \\ b \leftarrow A(F(x)) \end{matrix}\right] = 1/2 + \mu(n)$$

# Today's Lecture

- OWPs → PRGs
- PRG Indistinguishability → PRG Unpredictability

# OWP → PRG

# OW**P** → PRG, Attempt #2

**Let $F : \{0,1\}^n \to \{0,1\}^n$ be a one-way permutation**

**Imagine there existed $B : \{0,1\}^n \to \{0,1\}$ such that the following was a PRG**

PRG($k$)

1. Output $F(k) \,||\, B(k)$

**What properties do we need of $B$?**

1. One-way: can't find $k$ from $B(k)$
2. Pseudorandom: $B(k)$ looks like a random bit
3. Unpredictable: $B(k)$ is unpredictable given $F(k)$

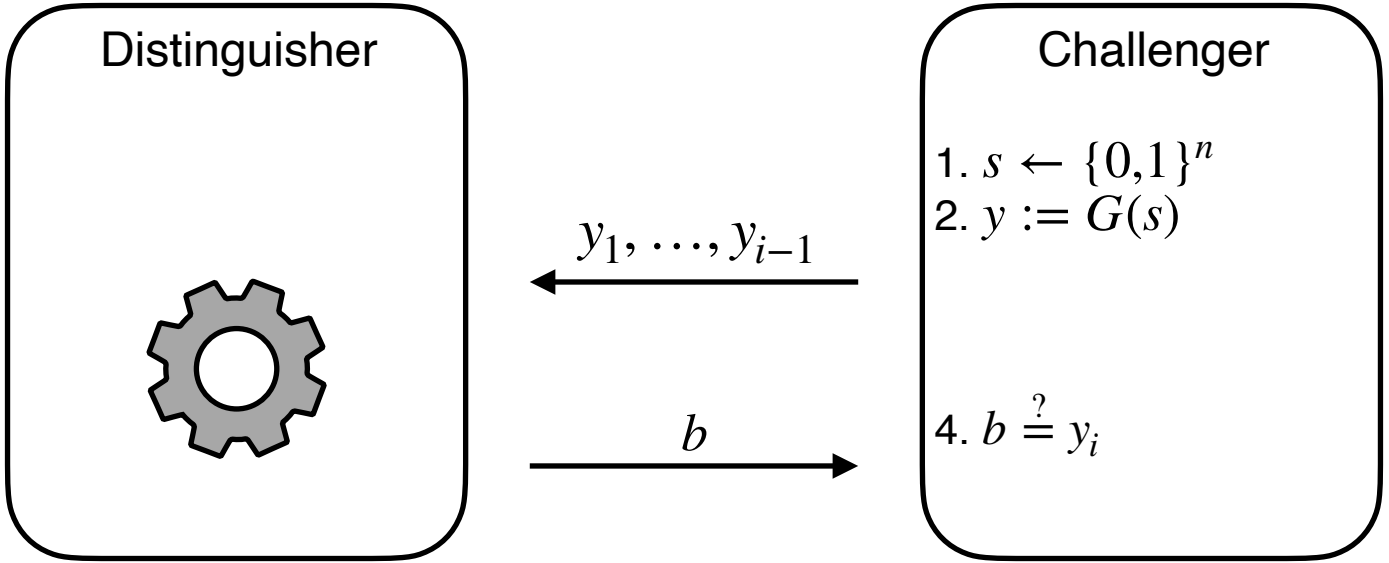
# OWP $\Rightarrow$ PRG

**Theorem**

Let $F$ be a one-way permutation, and let $B$ be a hardcore predicate for $F$.
Then, $G(x) := F(x) \,||\, B(x)$ is a PRG.

**Proof (next slide)**: Use next-bit unpredictability.

# PRG Next-Bit Unpredictability




$$\Pr\left[A(y_1, \ldots, y_{i-1}) = y_i \,\middle|\, \begin{array}{r} s \leftarrow \{0,1\}^n \\ y \leftarrow G(s) \end{array}\right] = 1/2 + \varepsilon(n)$$

# PRG Def 2: Next-bit Unpredictability

**Definition [Next-bit Unpredictability]:**

A deterministic polynomial-time computable function G: $\{0,1\}^n \rightarrow \{0,1\}^m$ is next-bit unpredictable if:

*for every PPT algorithm P (called a next-bit predictor) and every* $i \in \{1,\ldots,m\}$, *if there is a negligible function* $\mu$ *such that:*

$$\Pr\left[ y \leftarrow G(U_n) : P(y_1 y_2 \ldots y_{i-1}) = y_i \right] = \frac{1}{2} + \mu(n)$$

Notation: $y_1, y_2, \ldots y_m$ are the bits of the m-bit string $y$.

# Def 1 and Def 2 are Equivalent

**Theorem:**

A PRG G is indistinguishable if and only if it is next-bit unpredictable.

# NBU and Indistinguishability

- Next-bit Unpredictability (NBU): Seemingly much weaker requirement. Only says that next bit predictors, a particular type of distinguishers, cannot succeed.

- Yet, surprisingly, Next-bit Unpredictability (NBU) = Indistinguishability.

- NBU often much easier to use.

# OWP $\Rightarrow$ PRG

**Theorem**: $G$ is a PRG assuming $F$ is a one-way permutation.

**Proof**: Assume for contradiction that $G$ is not a PRG. Therefore, there is a next-bit predictor $P$, and index $i$, and a polynomial $p$ such that

$$\Pr\left[P(y_1,\ldots,y_{i-1}) = y_i \,\middle|\, \begin{array}{r} x \leftarrow \{0,1\}^n \\ y \leftarrow G(x) \end{array}\right] = 1/2 + 1/p(n)$$

Observation: The index $i$ has to be $n+1$. Do you see why?

Hint: $G(x) := F(x)\,||\,B(x)$ and we *know* $F(x)$ is uniformly distributed

# OWP ⇒ PRG

**Theorem**: $G$ is a PRG assuming $F$ is a one-way permutation.

**Proof**: Assume for contradiction that $G$ is not a PRG. Therefore, there is a next-bit predictor $P$, and polynomial $p$ such that

$$\Pr\left[P(y_1,\ldots,y_n) = y_{n+1} \;\middle|\; \begin{array}{r} x \leftarrow \{0,1\}^n \\ y \leftarrow G(x) \end{array}\right] = 1/2 + 1/p(n)$$

# OWP $\Rightarrow$ PRG

**Theorem**: $G$ is a PRG assuming $F$ is a one-way permutation.

**Proof**: Assume for contradiction that $G$ is not a PRG. Therefore, there is a next-bit predictor $P$, and polynomial $p$ such that

$$\Pr\left[P(F(x)) = B(x) \,\middle|\, \begin{array}{c} x \leftarrow \{0,1\}^n \\ y \leftarrow G(x) \end{array}\right] = 1/2 + 1/p(n)$$

So, $P$ can figure out $B(x)$ and break hardcore property! QED.

# Aside: Indistinguishability => Unpredictability

# 1. Indistinguishability $\Longrightarrow$ NBU

**Proof: by contradiction.**

Suppose for contradiction that there is a p.p.t. predictor $P$, a polynomial function $p$ and an $i \in \{1,\ldots,m\}$ s.t.

$$\Pr\left[ y \leftarrow G(U_n) : P(y_1 y_2 \ldots y_{i-1}) = y_i \right] \geq \frac{1}{2} + 1/p(n)$$

Then, I claim that $P$ essentially gives us a distinguisher D!

Consider $D$ which gets an m-bit string $y$ and does the following:

1. Run $P$ on the $(i-1)$-bit prefix $y_1 y_2 \ldots y_{i-1}$.
2. If $P$ returns the $i$-th bit $y_i$, then output 1 ("PRG") else output 0 ("Random").

**If $P$ is p.p.t. so is $D$.**

# 1. Indistinguishability $\Longrightarrow$ NBU

Consider $D$ which gets an m-bit string $y$ and does the following:

1. Run $P$ on the $(i-1)$-bit prefix $y_1 y_2 \ldots y_{i-1}$.
2. If $P$ returns the $i$-th bit $y_i$, then output 1 (= "PRG") else output 0 (= "Random").

We want to show: there is a polynomial $p'$ s.t.

$$\left| \Pr[y \leftarrow G(U_n) : D(y) = 1] - \Pr[y \leftarrow Um : D(y) = 1] \right| \geq 1/p'(n)$$

# 1. Indistinguishability $\Longrightarrow$ NBU

Consider $D$ which gets an m-bit string $y$ and does the following:

1. Run $P$ on the $(i-1)$-bit prefix $y_1 y_2 \ldots y_{i-1}$.
2. If $P$ returns the $i$-th bit $y_i$, then output 1 (= "PRG") else output 0 (= "Random").

$$\begin{aligned} & \Pr[y \leftarrow G(U_n) : D(y) = 1] \\ = \; & \Pr[y \leftarrow G(U_n) : P(y_1 y_2 \ldots y_{i-1}) = y_i] && \text{(by construction of D)} \\ \geq \; & \frac{1}{2} + 1/p(n) && \text{(by assumption on P)} \end{aligned}$$

# 1. Indistinguishability $\Longrightarrow$ NBU

Consider $D$ which gets an m-bit string $y$ and does the following:

1. Run $P$ on the $(i-1)$-bit prefix $y_1 y_2 \ldots y_{i-1}$.
2. If $P$ returns the $i$-th bit $y_i$, then output 1 (= "PRG") else output 0 (= "Random").

$$\Pr[y \leftarrow G(U_n) : D(y) = 1] \geq \frac{1}{2} + 1/p(n)$$

$$\begin{aligned} & \Pr\left[y \leftarrow U_m : D(y) = 1\right] \\ = \; & \Pr[y \leftarrow U_m : P(y_1 y_2 \ldots y_{i-1}) = y_i] && \text{(by construction of D)} \\ = \; & \frac{1}{2} && \text{(since y is random)} \end{aligned}$$

# 1. Indistinguishability $\Longrightarrow$ NBU

Consider $D$ which gets an m-bit string $y$ and does the following:

1. Run $P$ on the $(i-1)$-bit prefix $y_1 y_2 \ldots y_{i-1}$.
2. If $P$ returns the $i$-th bit $y_i$, then output 1 (= "PRG") else output 0 (= "Random").

$$\Pr[y \leftarrow G(U_n) : D(y) = 1] \geq \frac{1}{2} + 1/p(n)$$

$$\Pr\left[y \leftarrow U_m : D(y) = 1\right] = \frac{1}{2}$$

So, $|\Pr[y \leftarrow G(U_n) : D(y) = 1] - \Pr[y \leftarrow Um : D(y) = 1]| \geq 1/p(n)$

**Q1**: Do PRGs exist?

A: Yes, assuming OWFs

**Q2**: How do we encrypt longer messages or many messages with a fixed key?

(**Length extension**: If there is a PRG that stretches by one bit, there is one that stretches by polynomially many bits)

(**Pseudorandom functions**: PRGs with exponentially large stretch and "random access" to the output.)

- **So far: PRG with 1-bit expansion**
- Resulting secret-key encryption:
  - Key can be 1 bit shorter than message
  - Not much better than OTP!

**Can we do better?**

**PRG length extension.**

*Theorem*: If there is a PRG that stretches by one bit, there is one that stretches by poly many bits

♦ **New Proof Technique: Hybrid Arguments.**

# Before we go there, a puzzle…

Lemma: Let $p_0, p_1, p_2, \ldots, p_m$ be real numbers s.t.

$$p_m - p_0 \geq \varepsilon \, .$$

Then, there is an index $i$ such that $p_i - p_{i-1} \geq \varepsilon/m$.

Proof:

$$p_m - p_0 = (p_m - p_{m-1}) + (p_{m-1} - p_{m-2}) + \ldots + (p_1 - p_0)$$
$$\geq \varepsilon$$

At least one of the $m$ terms has to be at least $\varepsilon/m$ (averaging).

# Length extension: One bit to Many bits

Let $G : \{0,1\}^n \to \{0,1\}^{n+1}$ be a PRG

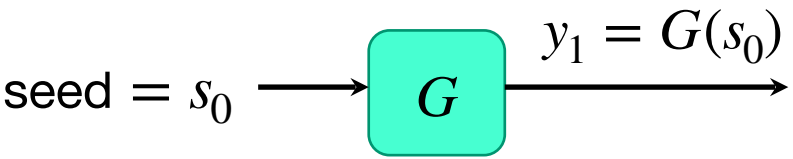
Goal: use $G$ to generate **many** pseudorandom bits.

# Length extension: One bit to Many bits

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

Goal: use $G$ to generate **many** pseudorandom bits.

Construction of $G'(s_0)$

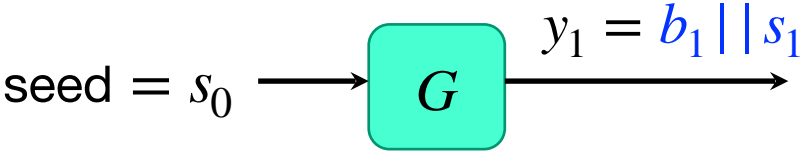


# Length extension: One bit to Many bits

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

Goal: use $G$ to generate **many** pseudorandom bits.

Construction of $G'(s_0)$

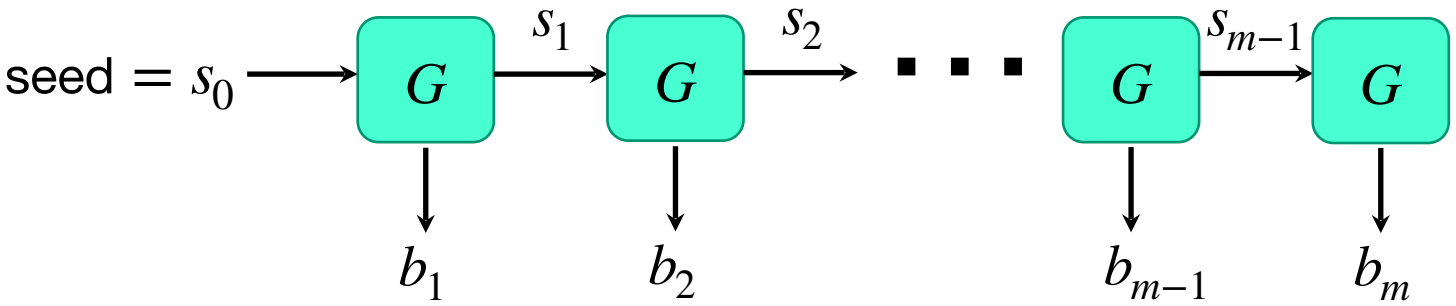


# Length extension: One bit to Many bits

Let $G : \{0,1\}^n \to \{0,1\}^{n+1}$ be a PRG

Goal: use $G$ to generate **many** pseudorandom bits.

Construction of $G'(s_0)$

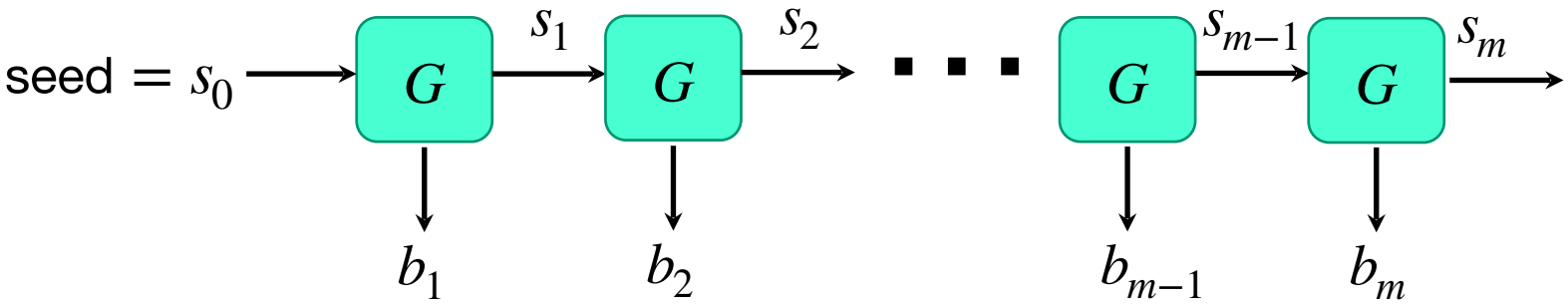


# Length extension: One bit to Many bits

**Proof of Security** (next class):

**Use next-bit (or previous-bit?) unpredictability!**

Construction of $G'(s_0)$



# Next class

- Why does length-extension work?
- PRFs: How to get PRGs with “exponentially-large” output