CIS 5560

Cryptography
Lecture 4

Course website:

pratyushmishra.com/classes/cis-5560-s25

Slides adapted from Dan Boneh and Vinod Vaikuntanathan

Announcements

- **HW 0 is out;** due Friday, Jan 31 at 5PM on Gradescope
- **HW 1** will be released tomorrow
  - OTPs, perfect security/indistinguishability
  - PRGs, computational indistinguishability, negl. fns
- Homework party tomorrow AGH 105A 4:30-6PM
  - Work on HW0 and HW1 with classmates
  - Ask questions to TAs!
- Cryptography related CIS Colloquium today after class
  - See what high level cryptography research looks like!

Recap of last lecture

Computational Indistinguishability

World 0:

$k \leftarrow \mathcal{K}$

$c = \mathsf{Enc}(k, m_0)$

World 1:

$k \leftarrow \mathcal{K}$

$c = \mathsf{Enc}(k, m_1)$

Eve is arbitrary **PPT distinguisher**.

She needs to decide whether c came from World 0 or World 1.

For every **PPT** Eve, there exists a negligible fn ε, st for all m_0, m_1,

$$\left| \Pr\left[\mathsf{Eve}(c) = 0 \,\middle|\, \begin{matrix} k \leftarrow \mathcal{K} \\ c = \mathsf{Enc}(k, m_0) \end{matrix} \right] - \Pr\left[\mathsf{Eve}(c) = 1 \,\middle|\, \begin{matrix} k \leftarrow \mathcal{K} \\ c = \mathsf{Enc}(k, m_1) \end{matrix} \right] \right| = \varepsilon(n)$$

Negligible Functions

Functions that grow slower than 1/p(n) for any polynomial p.

Definition: A function $\varepsilon : \mathbb{N} \to \mathbb{R}$ is **negligible** if
for every polynomial function p,
there exists an n_0 s.t.
for all $n > n_0$:

$$\varepsilon(n) < \frac{1}{p(n)}$$

Question: Let $\varepsilon(n) = 1/n^{\log n}$. Is ε negligible?

Pseudorandom Generators

Informally: **Deterministic** Programs that stretch a "truly random" seed into a (much) longer sequence of **"seemingly random"** bits.

Q1: How to define "seemingly random"?

Q2: Can such a G exist?

PRG Def 1: Indistinguishability

Definition [Indistinguishability]:

A deterministic polynomial-time computable function

$$G : \{0,1\}^n \to \{0,1\}^m \text{ is a } \textbf{PRG} \text{ if:}$$

(a) It is expanding: $m > n$ and

(b) for every PPT algorithm D (called a distinguisher) if there is a negligible function ε such that:

$$\left| \Pr[D(G(U_n)) = 1] - \Pr[D(U_m) = 1] \right| = \varepsilon(n)$$

Notation: U_n (resp. U_m) denotes the random distribution on n-bit (resp. m-bit) strings; m is shorthand for $m(n)$.

PRG Def 1: Indistinguishability

Definition [Indistinguishability]:

A deterministic polynomial-time computable function

$G : \{0,1\}^n \rightarrow \{0,1\}^m$ is a **PRG** if:

(a) It is expanding: $m > n$ and

(b) for every PPT algorithm D (called a distinguisher) if there is a negligible function ε such that:

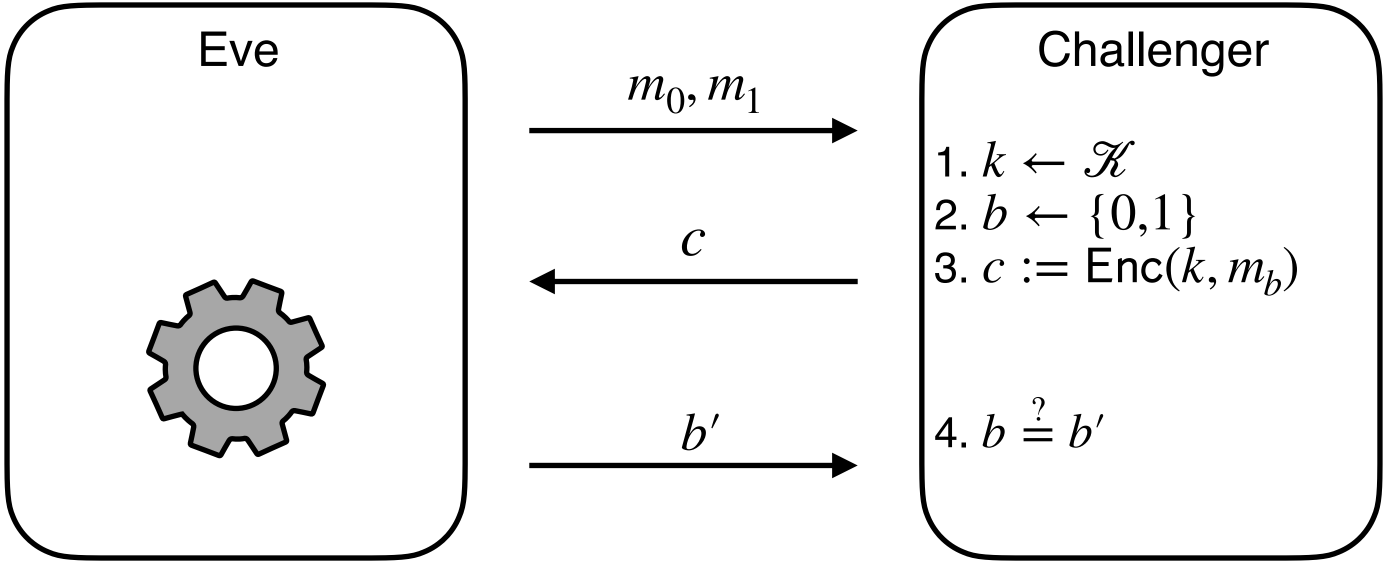
$$\Pr\left[D(y_b) = b \;\middle|\; \begin{array}{r} b \leftarrow \{0,1\} \\ x \leftarrow \{0,1\}^n \\ y_0 = G(x) \\ y_1 \leftarrow \{0,1\}^{n+1} \end{array} \right] \leq 1/2 + \varepsilon(n)$$

Semantic Security

For every **PPT** Eve, there exists a negligible fn ε, st for all m_0, m_1,

$$\Pr\left[\mathsf{Eve}(c) = b \;\middle|\; \begin{array}{r} k \leftarrow \mathcal{K} \\ b \leftarrow \{0,1\} \\ c := \mathsf{Enc}(k, m_b) \end{array}\right] < \frac{1}{2} + \varepsilon(n)$$

Semantic Security



Semantic Security

For every **PPT** Eve, there exists a negligible fn ε such that

$$\Pr\left[\mathsf{Eve}(c) = b \;\middle|\; \begin{array}{r} (m_0, m_1) \leftarrow \mathsf{Eve} \\ k \leftarrow \mathscr{K} \\ b \leftarrow \{0,1\} \\ c := \mathsf{Enc}(k, m_b) \end{array}\right] < \frac{1}{2} + \varepsilon(n)$$

PRGs → Semantically Secure Encryption

PRG $\implies$ 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 $G(k) \oplus c = G(k) \oplus G(k) \oplus m = m$

Today's Lecture

- PRG Indistinguishability → Semantic Security
- One way functions and permutations
- OWPs → PRGs

PRG $\implies$ Semantically Secure Encryption

Security: your first reduction!

Suppose for contradiction that there exists an Eve that breaks our scheme.

That, is assume that there is a p.p.t. Eve, and polynomial function p s.t.

$$\Pr\left[\mathsf{Eve}(c) = b \;\middle|\; \begin{array}{r} (m_0, m_1) \leftarrow \mathsf{Eve} \\ k \leftarrow \mathscr{K} \\ b \leftarrow \{0,1\} \\ c := \mathsf{Enc}(k, m_b) \end{array}\right] > \frac{1}{2} + 1/p(n)$$

PRG $\Longrightarrow$ Semantically Secure Encryption

Security: **your first reduction!**

Assume that there is a p.p.t. Eve, a polynomial function p and m_0, m_1 s.t.

$$\Pr\left[\mathsf{Eve}(c) = b \;\middle|\; \begin{array}{r} (m_0, m_1) \leftarrow \mathsf{Eve} \\ k \leftarrow \{0,1\}^n \\ b \leftarrow \{0,1\} \\ c := G(k) \oplus m_b \end{array}\right] > \frac{1}{2} + 1/p(n)$$

Let's call this ρ

Compare with $$\Pr\left[\mathsf{Eve}(c) = b \;\middle|\; \begin{array}{r} (m_0, m_1) \leftarrow \mathsf{Eve} \\ k' \leftarrow \{0,1\}^{n+1} \\ b \leftarrow \{0,1\} \\ c := k' \oplus m_b \end{array}\right] = \frac{1}{2}$$

Let's call this ρ'

Clearly, Eve can break security in
PRG case, but not in OTP world!

↓

Eve can distinguish pseudorandom from random!

Idea: Use Eve to break PRG indistinguishability!

PRG Def 1: Indistinguishability

Definition [Indistinguishability]:

A deterministic polynomial-time computable function

$G : \{0,1\}^n \to \{0,1\}^m$ is a **PRG** if:

(a) It is expanding: $m > n$ and

(b) for every PPT algorithm D (called a distinguisher) if there is a negligible function ε such that:

$$\Pr\left[D(y_b) = b \;\middle|\; \begin{array}{r} b \leftarrow \{0,1\} \\ x \leftarrow \{0,1\}^n \\ y_0 = G(x) \\ y_1 \leftarrow \{0,1\}^{n+1} \end{array} \right] \leq 1/2 + \varepsilon(n)$$

Setting: we have 3 parties:

- Eve
- Challenger for PRG game
- Distinguisher D (that we will construct)

Idea: we will "emulate" semantic security game for Eve

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,

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

$\blacksquare$

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)

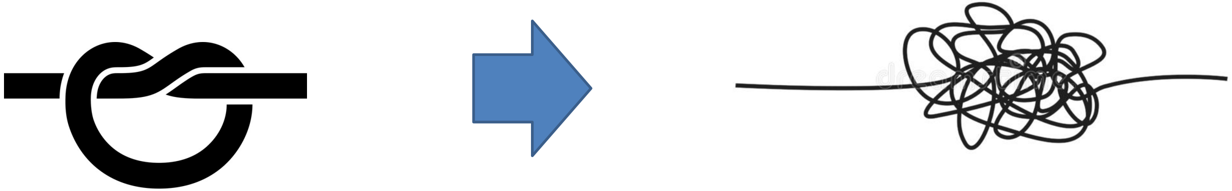
(**Pseudorandom functions**: PRGs with exponentially large stretch and "random access" to the output.)

$\boldsymbol{Q1}$: Do PRGs exist?

Constructing PRGs: Two Methodologies

<u>**The Practical Methodology**</u>

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

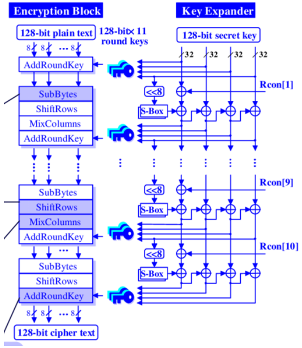


Constructing PRGs: Two Methodologies

<u>**The Practical Methodology**</u>

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)

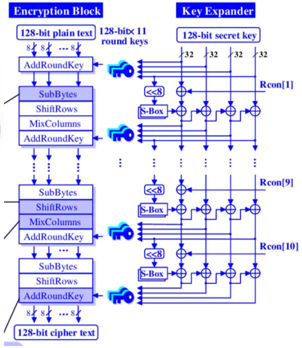
Constructing PRGs: Two Methodologies

<u>The Practical Methodology</u>

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

2. Come up with a candidate construction

3. Do extensive cryptanalysis.



Examples

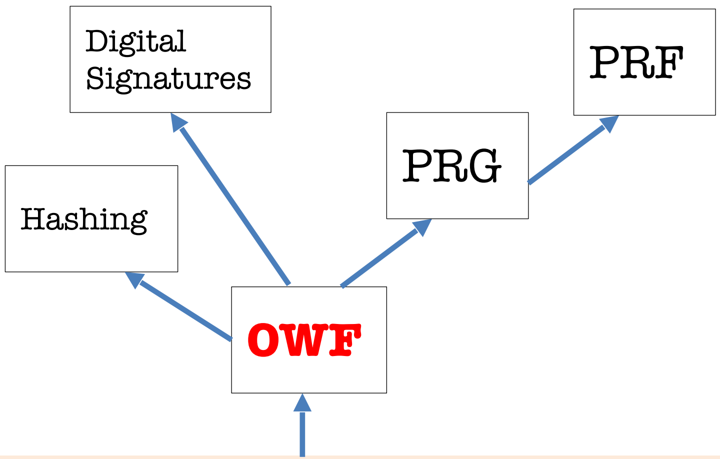
- **RC4: old PRG from 1987**
  - Proposed by Ron Rivest (of RSA fame)
  - **Fast and simple**
  - Used in TLS till 2013
  - However lots of biases
    - e.g. 2nd byte of output has 2/256 chance of being 0.
  - In 2013, attack which made key recovery feasible with just 2^{20} ciphertexts!
  - Finally deprecated in 2015, *28 years* after creation!

Constructing PRGs: Two Methodologies

<u>The Foundational Methodology (much of this course)</u>

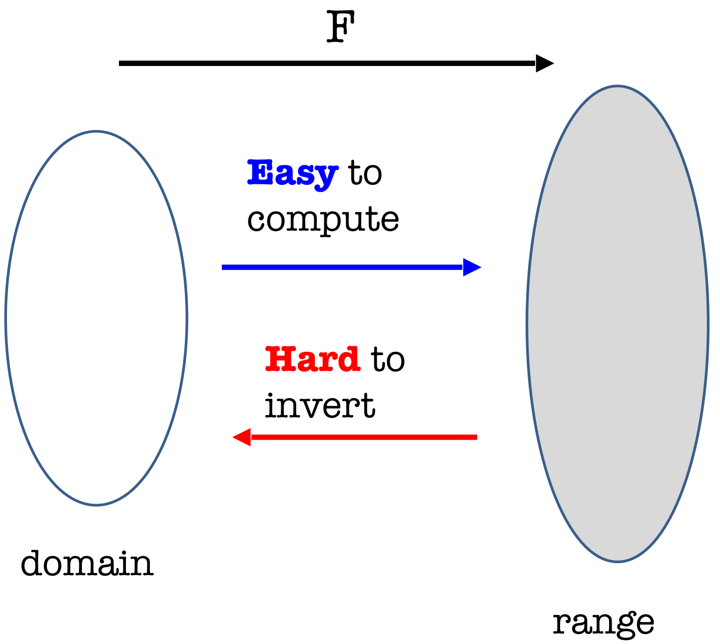
Reduce to simpler primitives.

"Science wins either way" –Silvio Micali



well-studied, average-case hard, problems

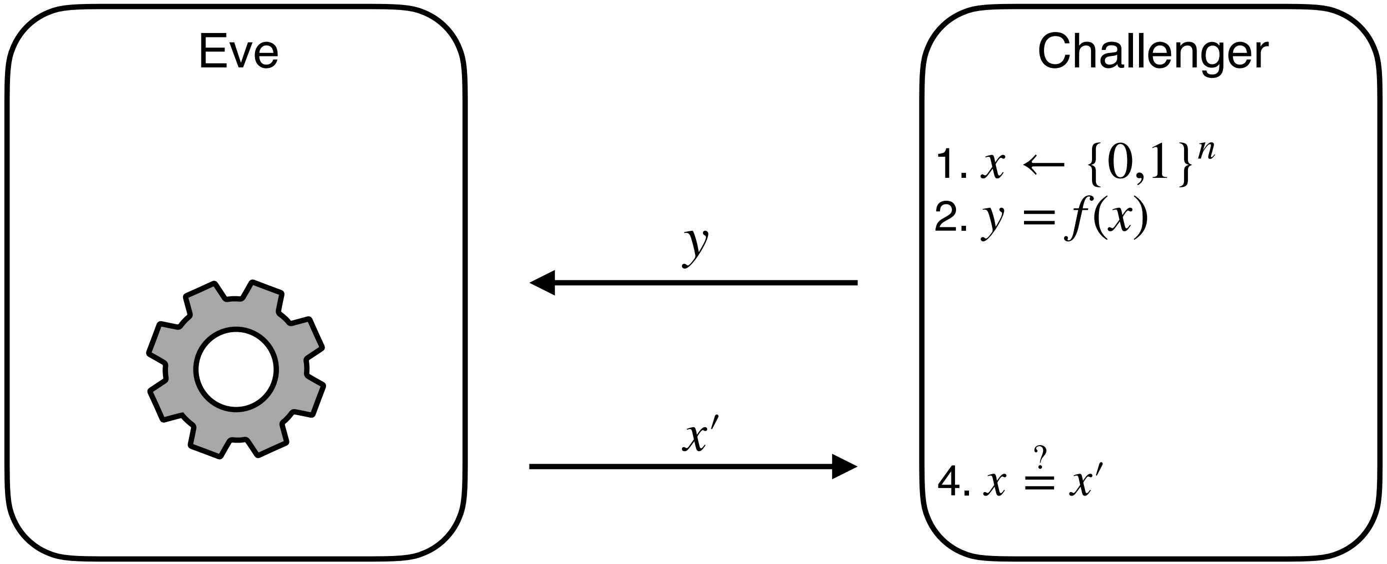
One-way Functions (Informally)



Source of all hard problems in cryptography!

What is a good definition?

OWF Security Attempt #1



One-way Functions (Take 1)

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:

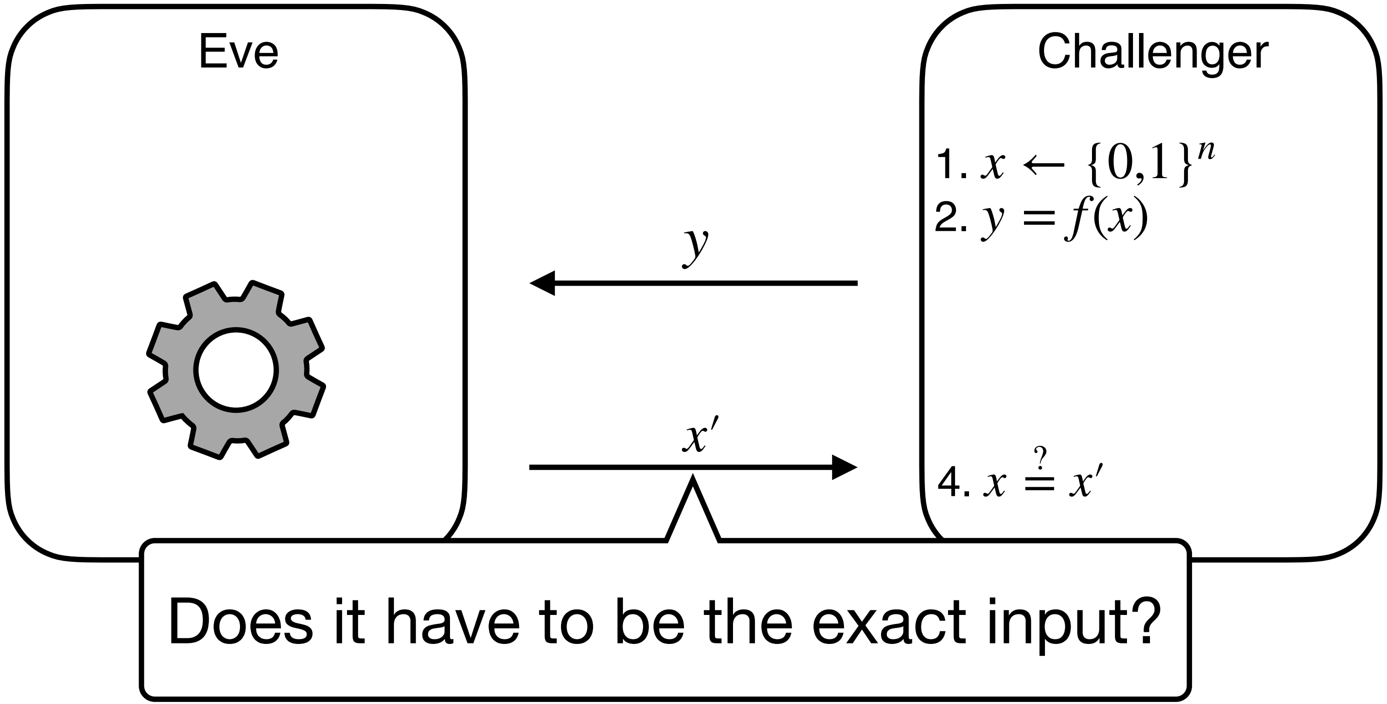
$$\Pr\left[A(1^n, y) = x \,\middle|\, \begin{array}{c} x \leftarrow \{0,1\}^n \\ y := F_n(x) \end{array}\right] = \mathrm{negl}(n)$$

Consider $F_n(x) = 0$ for all x.

This is one-way according to the above definition.
In fact, impossible to find *the* inverse even if A has unbounded time.

Conclusion: not a useful/meaningful definition.

OWF Security Attempt #2



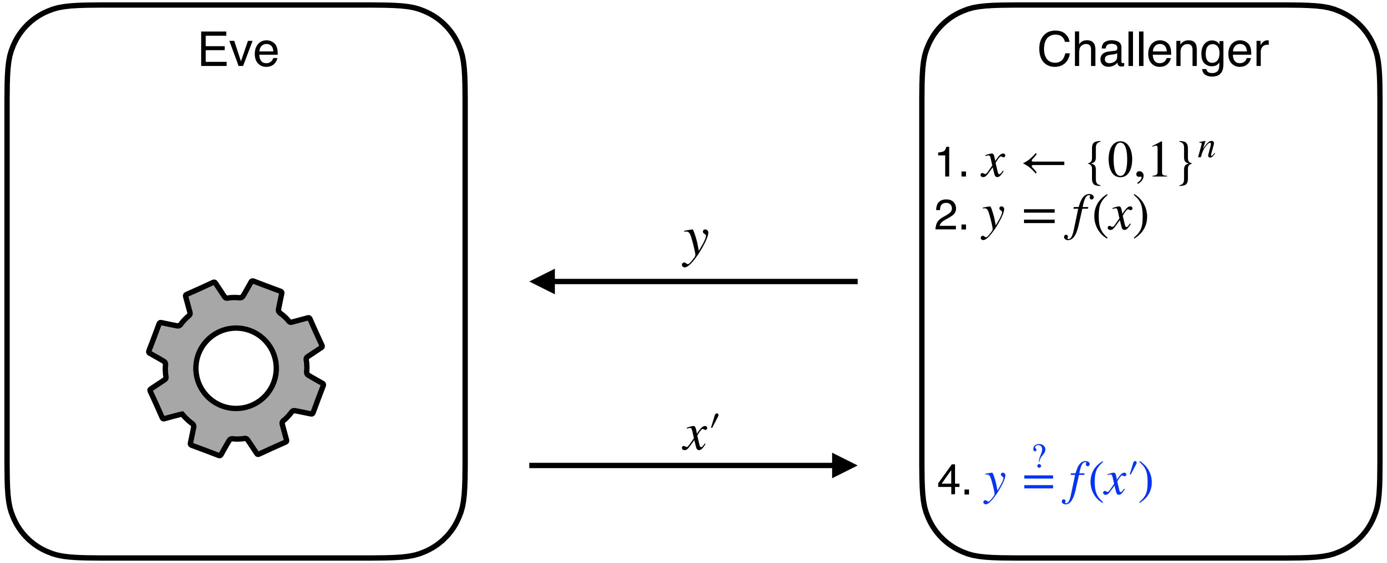
One-way Functions (Take 1)

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:

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

The Right Definition: Impossible to find ***an*** inverse efficiently.

OWF Security Attempt #2



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:

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

- 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 $m(n) = n$.

How to get PRG from OWF?

OWF → PRG, Attempt #1

PRG(k)

1. Output $F_n(k)$

(Assume $m(n) > n$)

Does this work?

OWF → PRG, Attempt #1

Consider $F_n(x)$ constructed from another OWF F'_n:

1. Compute $y := F'_n(x)$
2. Output $y' := (y_0,\ 1, y_1,\ 1, \ldots, y_n,\ 1)$

PRG(k)

1. Output $F_n(k)$

Is F one-way?

Yes!

Is PRG unpredictable?

No!

Our problem:

OWFs don’t tell us anything about how their outputs are distributed.

They are only hard to invert!

OW**P** → PRG, Attempt #1

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

Consider the following PRG candidate

PRG(k)

1. Output $F(k)$

Does this work?

No, it's not expanding!

But how are outputs distributed?

Claim: Output of F is uniformly distributed

Claim: Output of OWP is uniformly distributed

Proof: Assume for contradiction that this is not the case.

This means that there exists some y such that

$$\Pr[F(x) = y \,|\, x \leftarrow \{0,1\}^n] > 1/2^n$$

This means that $\dfrac{\left|\{x \,|\, F(x) = y\}\right|}{2^n} > \dfrac{1}{2^n}$,

which in turn means that F is not a permutation!

Our problem:

OWFs don't tell us anything about how their outputs are distributed.

Solution: use OWP

Problem: no expansion

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 $F(k)$

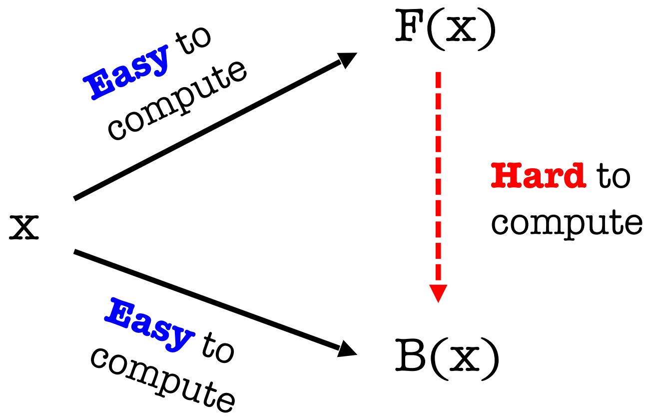
Hardcore Bits

HARDCORE PREDICATE

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

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

Hardcore Predicate (in pictures)



Existence of hardcore predicates

Goldreich-Levin Theorem

Let $F : \{0,1\}^n \to \{0,1\}^n$ be a one-way function.
Define $H(x || r) := F(x) || r$.

Then $B(x || r) := \langle x, r \rangle$ is a hardcore predicate for H

Existence of hardcore predicates

Hardcore predicate for RSA

Define $F_{N,e}(x) := x^e \mod N$ to be the **RSA** OWF.

Then $\mathsf{lsb}(x)$ is a hardcore predicate for F

OWP → PRG

OWP $\Rightarrow$ PRG

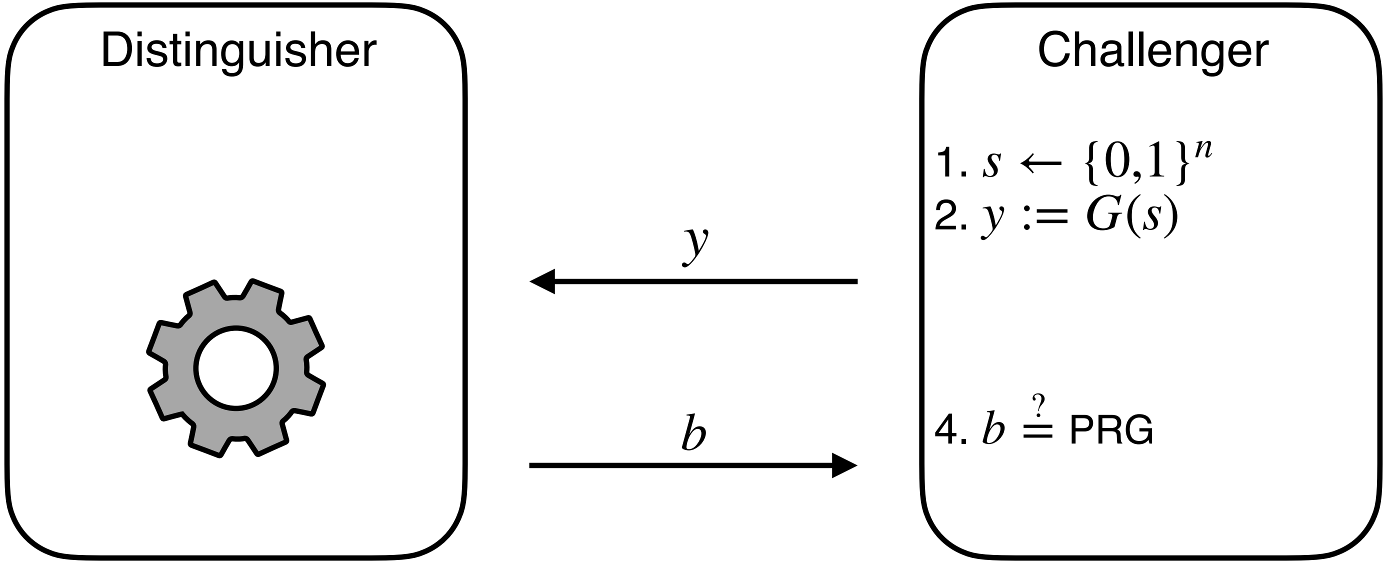
Theorem

Let F be a one-way permutation, and let B be a hardcore predicate for F.

Then, $G(x) := F(x) \mid\mid B(x)$ is a PRG.

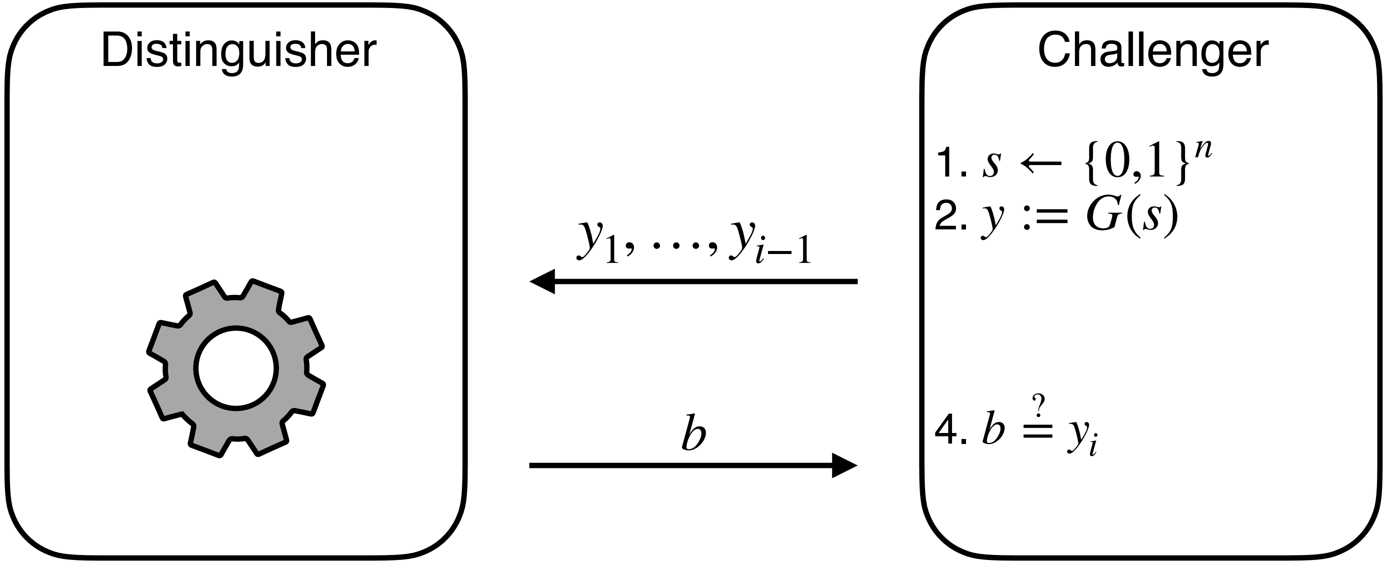
Proof (next slide): Use next-bit unpredictability.

PRG Indistinguishability



$$\left| \Pr[D(G(U_n)) = 1] - \Pr[D(U_m) = 1] \right| = \varepsilon(n)$$

PRG Next-Bit Unpredictability



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

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* μ *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.

Def 1 and Def 2 are Equivalent

Theorem:
A PRG G passes all PPT distinguishers if and only if it passes PPT *next-bit* distinguishers.

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 $\Rightarrow$ 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

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

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

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

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

Next class

- Indistinguishability <=> Unpredictability
- How to extend the length of PRGs
- How to get PRGs with “exponentially-large” output