

# Cryptography

## Lecture 4

Arkady Yerukhimovich

September 9, 2024

- 1 Lecture 3 Review
- 2 Pseudorandom Generator (PRG) (Ch. 3.3.1)
- 3 Proofs by Reduction(Ch. 3.3.2)

# Lecture 3 Review

- Limitations of OTP and perfect secrecy
- Proof techniques
- Defining computationally-secure encryption

- 1 Lecture 3 Review
- 2 Pseudorandom Generator (PRG) (Ch. 3.3.1)
- 3 Proofs by Reduction(Ch. 3.3.2)

# Constructing Private-Key Encryption for Long Messages

- Recall that we encrypted by computing  $\text{Enc}_k(m) = m \oplus k$
- But, if  $|k| < |m|$ , this is not secure

## Key Idea

What if we had a way to stretch key  $k$  into something longer that still looked random?

Construct a deterministic function  
 $G : \{0, 1\}^n \rightarrow \{0, 1\}^{\ell(n)}$  such that:

Construct a deterministic function  
 $G : \{0, 1\}^n \rightarrow \{0, 1\}^{\ell(n)}$  such that:

- $\ell(n) > n$  (Expansion)

Construct a deterministic function

$G : \{0,1\}^n \rightarrow \{0,1\}^{\ell(n)}$  such that:

- $\ell(n) > n$  (Expansion)
- $G(s)$  “looks random” for a random seed  $s$  (Pseudorandomness)



Construct a deterministic function

$G : \{0, 1\}^n \rightarrow \{0, 1\}^{\ell(n)}$  such that:

- $\ell(n) > n$  (Expansion)
- $G(s)$  “looks random” for a random seed  $s$  (Pseudorandomness)

## Observations:

- $G(s)$  is actually (statistically) far from random.  
Consider  $G : \{0, 1\}^n \rightarrow \{0, 1\}^{n+1}$ , then  $G(s)$  can only possibly output at most one half of possible output values

Construct a deterministic function

$G : \{0, 1\}^n \rightarrow \{0, 1\}^{\ell(n)}$  such that:

- $\ell(n) > n$  (Expansion)
- $G(s)$  “looks random” for a random seed  $s$  (Pseudorandomness)

## Observations:

- $G(s)$  is actually (statistically) far from random.  
Consider  $G : \{0, 1\}^n \rightarrow \{0, 1\}^{n+1}$ , then  $G(s)$  can only possibly output at most one half of possible output values
- $G(s)$  cannot “look random” to someone who knows  $s$

Construct a deterministic function

$G : \{0, 1\}^n \rightarrow \{0, 1\}^{\ell(n)}$  such that:

- $\ell(n) > n$  (Expansion)
- $G(s)$  “looks random” for a random seed  $s$  (Pseudorandomness)

## Observations:

- $G(s)$  is actually (statistically) far from random.  
Consider  $G : \{0, 1\}^n \rightarrow \{0, 1\}^{n+1}$ , then  $G(s)$  can only possibly output at most one half of possible output values
- $G(s)$  cannot “look random” to someone who knows  $s$
- $G(s)$  is only required to look random to someone who knows nothing about  $s$  – i.e.,  $s$  is uniformly random

# PRG Definition

Let  $G : \{0, 1\}^n \rightarrow \{0, 1\}^{\ell(n)}$  be a deterministic poly-time function.

# PRG Definition

Let  $G : \{0, 1\}^n \rightarrow \{0, 1\}^{\ell(n)}$  be a deterministic poly-time function.

$PRG_{\mathcal{D}, G}(n)$

- The challenger chooses  $b \leftarrow \{0, 1\}$ .  
If  $b = 0$ , he chooses  $r \leftarrow \{0, 1\}^{\ell(n)}$  – (random)  
if  $b = 1$ , he chooses  $s \leftarrow \{0, 1\}^n$ , and computes  $r = G(s)$  – (PRG)  
He gives  $r$  to  $\mathcal{D}$ .

# PRG Definition

Let  $G : \{0, 1\}^n \rightarrow \{0, 1\}^{\ell(n)}$  be a deterministic poly-time function.

$PRG_{\mathcal{D}, G}(n)$

- The challenger chooses  $b \leftarrow \{0, 1\}$ .  
If  $b = 0$ , he chooses  $r \leftarrow \{0, 1\}^{\ell(n)}$  – (random)  
if  $b = 1$ , he chooses  $s \leftarrow \{0, 1\}^n$ , and computes  $r = G(s)$  – (PRG)  
He gives  $r$  to  $\mathcal{D}$ .
- On input  $r$ , the distinguisher  $\mathcal{D}$  outputs a guess  $b'$

# PRG Definition

Let  $G : \{0, 1\}^n \rightarrow \{0, 1\}^{\ell(n)}$  be a deterministic poly-time function.

$PRG_{\mathcal{D}, G}(n)$

- The challenger chooses  $b \leftarrow \{0, 1\}$ .  
If  $b = 0$ , he chooses  $r \leftarrow \{0, 1\}^{\ell(n)}$  – (random)  
if  $b = 1$ , he chooses  $s \leftarrow \{0, 1\}^n$ , and computes  $r = G(s)$  – (PRG)  
He gives  $r$  to  $\mathcal{D}$ .
- On input  $r$ , the distinguisher  $\mathcal{D}$  outputs a guess  $b'$
- $PRG_{\mathcal{D}, G}(n) = 1$  (i.e.,  $\mathcal{D}$  wins) if  $b' = b$

# PRG Definition

Let  $G : \{0, 1\}^n \rightarrow \{0, 1\}^{\ell(n)}$  be a deterministic poly-time function.

$PRG_{\mathcal{D}, G}(n)$

- The challenger chooses  $b \leftarrow \{0, 1\}$ .  
If  $b = 0$ , he chooses  $r \leftarrow \{0, 1\}^{\ell(n)}$  – (random)  
if  $b = 1$ , he chooses  $s \leftarrow \{0, 1\}^n$ , and computes  $r = G(s)$  – (PRG)  
He gives  $r$  to  $\mathcal{D}$ .
- On input  $r$ , the distinguisher  $\mathcal{D}$  outputs a guess  $b'$
- $PRG_{\mathcal{D}, G}(n) = 1$  (i.e.,  $\mathcal{D}$  wins) if  $b' = b$

Definition:  $G$  is a secure PRG if:

- $\ell(n) > n$



# PRG Definition

Let  $G : \{0, 1\}^n \rightarrow \{0, 1\}^{\ell(n)}$  be a deterministic poly-time function.

$PRG_{\mathcal{D}, G}(n)$

- The challenger chooses  $b \leftarrow \{0, 1\}$ .  
If  $b = 0$ , he chooses  $r \leftarrow \{0, 1\}^{\ell(n)}$  – (random)  
if  $b = 1$ , he chooses  $s \leftarrow \{0, 1\}^n$ , and computes  $r = G(s)$  – (PRG)  
He gives  $r$  to  $\mathcal{D}$ .
- On input  $r$ , the distinguisher  $\mathcal{D}$  outputs a guess  $b'$
- $PRG_{\mathcal{D}, G}(n) = 1$  (i.e.,  $\mathcal{D}$  wins) if  $b' = b$

Definition:  $G$  is a secure PRG if:

- $\ell(n) > n$
- For all PPT distinguishers  $\mathcal{D}$ , it holds that

$$\Pr[PRG_{\mathcal{D}, G}(n) = 1] \leq 1/2 + \text{negl}(n)$$

# PRG Definition

Let  $G : \{0, 1\}^n \rightarrow \{0, 1\}^{\ell(n)}$  be a deterministic poly-time function.

$PRG_{\mathcal{D}, G}(n)$

- The challenger chooses  $b \leftarrow \{0, 1\}$ .  
If  $b = 0$ , he chooses  $r \leftarrow \{0, 1\}^{\ell(n)}$  – (random)  
if  $b = 1$ , he chooses  $s \leftarrow \{0, 1\}^n$ , and computes  $r = G(s)$  – (PRG)  
He gives  $r$  to  $\mathcal{D}$ .
- On input  $r$ , the distinguisher  $\mathcal{D}$  outputs a guess  $b'$
- $PRG_{\mathcal{D}, G}(n) = 1$  (i.e.,  $\mathcal{D}$  wins) if  $b' = b$

Definition:  $G$  is a secure PRG if:

- $\ell(n) > n$
- For all PPT distinguishers  $\mathcal{D}$ , it holds that

$$\Pr[PRG_{\mathcal{D}, G}(n) = 1] \leq 1/2 + \text{negl}(n)$$

$\mathcal{D}$  cannot distinguish between  $G(s)$  and  $r \leftarrow \{0, 1\}^{\ell(n)}$

# An Example

Example: Is the following  $G$  a secure PRG?

$$G(s) = s || \bigoplus_{i=1, \dots, n} s_i$$

# An Example

Example: Is the following  $G$  a secure PRG?

$$G(s) = s || \bigoplus_{i=1, \dots, n} s_i$$

Expansion: Output of  $G(s)$  has  $n + 1 > n$  bits.

# An Example

Example: Is the following  $G$  a secure PRG?

$$G(s) = s || \bigoplus_{i=1, \dots, n} s_i$$

Expansion: Output of  $G(s)$  has  $n + 1 > n$  bits.

Pseudorandomness:

- Including  $s$  as part of the output is ok. Since  $\mathcal{D}$  is not given  $s$ , this look random to him.

# An Example

Example: Is the following  $G$  a secure PRG?

$$G(s) = s || \bigoplus_{i=1, \dots, n} s_i$$

Expansion: Output of  $G(s)$  has  $n + 1 > n$  bits.

Pseudorandomness:

- Including  $s$  as part of the output is ok. Since  $\mathcal{D}$  is not given  $s$ , this look random to him.
- However, given  $s$ ,  $\mathcal{D}$  can check whether the last bit of his challenge  $r_{n+1} = \bigoplus_{i=1, \dots, n} s_i$ , output 1 (PRG) if it does, and 0 (random) if not.

# An Example

Example: Is the following  $G$  a secure PRG?

$$G(s) = s || \bigoplus_{i=1, \dots, n} s_i$$

Expansion: Output of  $G(s)$  has  $n + 1 > n$  bits.

Pseudorandomness:

- Including  $s$  as part of the output is ok. Since  $\mathcal{D}$  is not given  $s$ , this look random to him.
- However, given  $s$ ,  $\mathcal{D}$  can check whether the last bit of his challenge  $r_{n+1} = \bigoplus_{i=1, \dots, n} s_i$ , output 1 (PRG) if it does, and 0 (random) if not.
  - If  $r = G(s)$ , then  $r_{n+1} = \bigoplus_{i=1, \dots, n} s_i$  with probability

# An Example

Example: Is the following  $G$  a secure PRG?

$$G(s) = s || \bigoplus_{i=1, \dots, n} s_i$$

Expansion: Output of  $G(s)$  has  $n + 1 > n$  bits.

Pseudorandomness:

- Including  $s$  as part of the output is ok. Since  $\mathcal{D}$  is not given  $s$ , this look random to him.
- However, given  $s$ ,  $\mathcal{D}$  can check whether the last bit of his challenge  $r_{n+1} = \bigoplus_{i=1, \dots, n} s_i$ , output 1 (PRG) if it does, and 0 (random) if not.
  - If  $r = G(s)$ , then  $r_{n+1} = \bigoplus_{i=1, \dots, n} s_i$  with probability 1



# An Example

Example: Is the following  $G$  a secure PRG?

$$G(s) = s || \bigoplus_{i=1, \dots, n} s_i$$

Expansion: Output of  $G(s)$  has  $n + 1 > n$  bits.

Pseudorandomness:

- Including  $s$  as part of the output is ok. Since  $\mathcal{D}$  is not given  $s$ , this look random to him.
- However, given  $s$ ,  $\mathcal{D}$  can check whether the last bit of his challenge  $r_{n+1} = \bigoplus_{i=1, \dots, n} s_i$ , output 1 (PRG) if it does, and 0 (random) if not.
  - If  $r = G(s)$ , then  $r_{n+1} = \bigoplus_{i=1, \dots, n} s_i$  with probability 1
  - If  $r \leftarrow \{0, 1\}^{n+1}$ , then  $r_{n+1} = \bigoplus_{i=1, \dots, n} s_i$  with probability

# An Example

Example: Is the following  $G$  a secure PRG?

$$G(s) = s || \bigoplus_{i=1, \dots, n} s_i$$

Expansion: Output of  $G(s)$  has  $n + 1 > n$  bits.

Pseudorandomness:

- Including  $s$  as part of the output is ok. Since  $\mathcal{D}$  is not given  $s$ , this look random to him.
- However, given  $s$ ,  $\mathcal{D}$  can check whether the last bit of his challenge  $r_{n+1} = \bigoplus_{i=1, \dots, n} s_i$ , output 1 (PRG) if it does, and 0 (random) if not.
  - If  $r = G(s)$ , then  $r_{n+1} = \bigoplus_{i=1, \dots, n} s_i$  with probability 1
  - If  $r \leftarrow \{0, 1\}^{n+1}$ , then  $r_{n+1} = \bigoplus_{i=1, \dots, n} s_i$  with probability  $1/2$

# An Example

Example: Is the following  $G$  a secure PRG?

$$G(s) = s || \bigoplus_{i=1, \dots, n} s_i$$

Expansion: Output of  $G(s)$  has  $n + 1 > n$  bits.

Pseudorandomness:

- Including  $s$  as part of the output is ok. Since  $\mathcal{D}$  is not given  $s$ , this look random to him.
- However, given  $s$ ,  $\mathcal{D}$  can check whether the last bit of his challenge  $r_{n+1} = \bigoplus_{i=1, \dots, n} s_i$ , output 1 (PRG) if it does, and 0 (random) if not.
  - If  $r = G(s)$ , then  $r_{n+1} = \bigoplus_{i=1, \dots, n} s_i$  with probability 1
  - If  $r \leftarrow \{0, 1\}^{n+1}$ , then  $r_{n+1} = \bigoplus_{i=1, \dots, n} s_i$  with probability  $1/2$
- So,  $\mathcal{D}$  will always output 1 when given  $G(s)$  and output 0 with probability  $1/2$  when given  $r$ .

$$\Pr[\mathcal{D} \text{ WINS}] = \Pr[b = 1] * 1 + \Pr[b = 0] * 1/2 = 3/4 > 1/2$$

# Encryption from a PRG

# Encryption from a PRG

## PRG+OTP Encryption

- $\text{Gen}(1^n): k \leftarrow \{0, 1\}^n$

# Encryption from a PRG

## PRG+OTP Encryption

- $\text{Gen}(1^n): k \leftarrow \{0, 1\}^n$
- $\text{Enc}(k, m): c = G(k) \oplus m$  (for  $m \in \{0, 1\}^{\ell(n)}$ )

# Encryption from a PRG

## PRG+OTP Encryption

- $\text{Gen}(1^n): k \leftarrow \{0, 1\}^n$
- $\text{Enc}(k, m): c = G(k) \oplus m$  (for  $m \in \{0, 1\}^{\ell(n)}$ )
- $\text{Dec}(k, c): m = G(k) \oplus c$

# Encryption from a PRG

## PRG+OTP Encryption

- $\text{Gen}(1^n): k \leftarrow \{0, 1\}^n$
- $\text{Enc}(k, m): c = G(k) \oplus m$  (for  $m \in \{0, 1\}^{\ell(n)}$ )
- $\text{Dec}(k, c): m = G(k) \oplus c$

Intuition:

- A PRG stretches a short one-time pad into a longer one-time pad
- Can now encrypt messages longer than key
- Only get computational security due to the use of a PRG



# Encryption from a PRG

## PRG+OTP Encryption

- $\text{Gen}(1^n): k \leftarrow \{0, 1\}^n$
- $\text{Enc}(k, m): c = G(k) \oplus m$  (for  $m \in \{0, 1\}^{\ell(n)}$ )
- $\text{Dec}(k, c): m = G(k) \oplus c$

Intuition:

- A PRG stretches a short one-time pad into a longer one-time pad
- Can now encrypt messages longer than key
- Only get computational security due to the use of a PRG

## Next Step

Prove Security!

- 1 Lecture 3 Review
- 2 Pseudorandom Generator (PRG) (Ch. 3.3.1)
- 3 Proofs by Reduction(Ch. 3.3.2)

# How Do We Prove Security

Goal: Prove that encryption scheme  $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$  is secure

# How Do We Prove Security

Goal: Prove that encryption scheme  $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$  is secure

- Most of the time, cannot prove this directly
- We can't even prove that computationally secure encryption exists (or even that  $\mathcal{P} \neq \mathcal{NP}$ )

# How Do We Prove Security

Goal: Prove that encryption scheme  $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$  is secure

- Most of the time, cannot prove this directly
- We can't even prove that computationally secure encryption exists (or even that  $\mathcal{P} \neq \mathcal{NP}$ )

Instead, we rely on *security assumptions*

- Factoring is hard
- $G$  is a secure PRG

# How Do We Prove Security

Goal: Prove that encryption scheme  $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$  is secure

- Most of the time, cannot prove this directly
- We can't even prove that computationally secure encryption exists (or even that  $\mathcal{P} \neq \mathcal{NP}$ )

Instead, we rely on *security assumptions*

- Factoring is hard
- $G$  is a secure PRG

A security proof shows that  $\Pi$  is secure if the assumption is true

- We say we show a *reduction* from the security of  $\Pi$  to the assumption.

# Proofs by Reduction: Intuition

Recall that, for any events  $A$  and  $B$ :

$$(A \implies B) \iff (\neg B \implies \neg A)$$

# Proofs by Reduction: Intuition

Recall that, for any events  $A$  and  $B$ :

$$(A \implies B) \iff (\neg B \implies \neg A)$$

- Want to prove that assumption  $X$  implies construction  $\Pi$  is secure



# Proofs by Reduction: Intuition

Recall that, for any events  $A$  and  $B$ :

$$(A \implies B) \iff (\neg B \implies \neg A)$$

- Want to prove that assumption  $X$  implies construction  $\Pi$  is secure
- Instead, we prove that if  $\Pi$  is insecure, then assumption  $X$  is false

# Proofs by Reduction: Intuition

Recall that, for any events  $A$  and  $B$ :

$$(A \implies B) \iff (\neg B \implies \neg A)$$

- Want to prove that assumption  $X$  implies construction  $\Pi$  is secure
- Instead, we prove that if  $\Pi$  is insecure, then assumption  $X$  is false
- Essentially, this is a proof by contradiction.

# Proofs by Reduction: Outline

Let  $X$  be a security assumption (e.g.,  $G$  is a PRG), and  $\Pi$  be a construction (e.g., of encryption) we want to prove secure:

# Proofs by Reduction: Outline

Let  $X$  be a security assumption (e.g.,  $G$  is a PRG), and  $\Pi$  be a construction (e.g., of encryption) we want to prove secure:

- Assume there exists a PPT adversary  $\mathcal{A}_c$  breaking  $\Pi$

# Proofs by Reduction: Outline

Let  $X$  be a security assumption (e.g.,  $G$  is a PRG), and  $\Pi$  be a construction (e.g., of encryption) we want to prove secure:

- Assume there exists a PPT adversary  $\mathcal{A}_c$  breaking  $\Pi$
- Construct another adversary  $\mathcal{A}_r$  that solves  $X$  using  $\mathcal{A}_c$

# Proofs by Reduction: Outline

Let  $X$  be a security assumption (e.g.,  $G$  is a PRG), and  $\Pi$  be a construction (e.g., of encryption) we want to prove secure:

- Assume there exists a PPT adversary  $\mathcal{A}_c$  breaking  $\Pi$
- Construct another adversary  $\mathcal{A}_r$  that solves  $X$  using  $\mathcal{A}_c$ 
  - $\mathcal{A}_r$  is given an instance of  $X$  to solve

# Proofs by Reduction: Outline

Let  $X$  be a security assumption (e.g.,  $G$  is a PRG), and  $\Pi$  be a construction (e.g., of encryption) we want to prove secure:

- Assume there exists a PPT adversary  $\mathcal{A}_c$  breaking  $\Pi$
- Construct another adversary  $\mathcal{A}_r$  that solves  $X$  using  $\mathcal{A}_c$ 
  - $\mathcal{A}_r$  is given an instance of  $X$  to solve
  - $\mathcal{A}_r$  *simulates* an instance of  $\Pi$  to  $\mathcal{A}_c$  based on  $X$

# Proofs by Reduction: Outline

Let  $X$  be a security assumption (e.g.,  $G$  is a PRG), and  $\Pi$  be a construction (e.g., of encryption) we want to prove secure:

- Assume there exists a PPT adversary  $\mathcal{A}_c$  breaking  $\Pi$
- Construct another adversary  $\mathcal{A}_r$  that solves  $X$  using  $\mathcal{A}_c$ 
  - $\mathcal{A}_r$  is given an instance of  $X$  to solve
  - $\mathcal{A}_r$  *simulates* an instance of  $\Pi$  to  $\mathcal{A}_c$  based on  $X$ 
    - $\mathcal{A}_c$  thinks it's attacking a real instance of  $\Pi$ , so by assumption, it succeeds

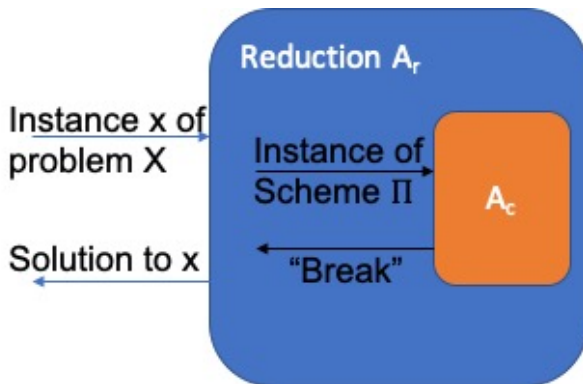


# Proofs by Reduction: Outline

Let  $X$  be a security assumption (e.g.,  $G$  is a PRG), and  $\Pi$  be a construction (e.g., of encryption) we want to prove secure:

- Assume there exists a PPT adversary  $\mathcal{A}_c$  breaking  $\Pi$
- Construct another adversary  $\mathcal{A}_r$  that solves  $X$  using  $\mathcal{A}_c$ 
  - $\mathcal{A}_r$  is given an instance of  $X$  to solve
  - $\mathcal{A}_r$  *simulates* an instance of  $\Pi$  to  $\mathcal{A}_c$  based on  $X$ 
    - $\mathcal{A}_c$  thinks it's attacking a real instance of  $\Pi$ , so by assumption, it succeeds
  - If  $\mathcal{A}_c$  succeeds in breaking the simulated  $\Pi$ ,  $\mathcal{A}_r$  uses this to solve  $X$

# Proofs by Reduction: Outline



# Proofs by Reduction: Example

Assumption:  $G : \{0, 1\}^n \rightarrow \{0, 1\}^{n+2}$  is PRG

Goal: Prove that  $G' = G(s)_{1,\dots,n+1}$  is a PRG

# Proofs by Reduction: Example

Assumption:  $G : \{0, 1\}^n \rightarrow \{0, 1\}^{n+2}$  is PRG

Goal: Prove that  $G' = G(s)_{1,\dots,n+1}$  is a PRG

$PRG_{\mathcal{D},G}(n)$

- The challenger chooses  $b \leftarrow \{0, 1\}$ .  
If  $b = 0$ , he chooses  $r \leftarrow \{0, 1\}^{l(n)}$ ;  
if  $b = 1$ , he chooses  $s \leftarrow \{0, 1\}^n$ , and computes  $r = G(s)$ .  
He gives  $r$  to  $\mathcal{D}$ .
- On input  $r$ , the distinguisher  $\mathcal{D}$  outputs a guess  $b'$
- $PRG_{\mathcal{D},G}(n) = 1$  (i.e.,  $\mathcal{D}$  wins) if  $b' = b$

# Proofs by Reduction: Example

Assumption:  $G : \{0, 1\}^n \rightarrow \{0, 1\}^{n+2}$  is PRG

Goal: Prove that  $G' = G(s)_{1,\dots,n+1}$  is a PRG

Proof:

- $G'$  expands from  $n$  bits to  $n + 1$  bits

$PRG_{\mathcal{D},G}(n)$

- The challenger chooses  $b \leftarrow \{0, 1\}$ .  
If  $b = 0$ , he chooses  $r \leftarrow \{0, 1\}^{l(n)}$ ;  
if  $b = 1$ , he chooses  $s \leftarrow \{0, 1\}^n$ , and computes  $r = G(s)$ .  
He gives  $r$  to  $\mathcal{D}$ .
- On input  $r$ , the distinguisher  $\mathcal{D}$  outputs a guess  $b'$
- $PRG_{\mathcal{D},G}(n) = 1$  (i.e.,  $\mathcal{D}$  wins) if  $b' = b$

# Proofs by Reduction: Example

Assumption:  $G : \{0, 1\}^n \rightarrow \{0, 1\}^{n+2}$  is PRG

Goal: Prove that  $G' = G(s)_{1,\dots,n+1}$  is a PRG

Proof:

- $G'$  expands from  $n$  bits to  $n + 1$  bits
- Assume there exists PPT  $\mathcal{A}_c$  that breaks  $G'$   
 $\Pr[PRG_{\mathcal{A}_c, G'}(n) = 1] > 1/2 + 1/\text{poly}(n)$ .  
Construct  $\mathcal{A}_r$  that breaks  $G$ :

$PRG_{\mathcal{D}, G}(n)$

- The challenger chooses  $b \leftarrow \{0, 1\}$ .  
If  $b = 0$ , he chooses  $r \leftarrow \{0, 1\}^{l(n)}$ ;  
if  $b = 1$ , he chooses  $s \leftarrow \{0, 1\}^n$ , and computes  $r = G(s)$ .  
He gives  $r$  to  $\mathcal{D}$ .
- On input  $r$ , the distinguisher  $\mathcal{D}$  outputs a guess  $b'$
- $PRG_{\mathcal{D}, G}(n) = 1$  (i.e.,  $\mathcal{D}$  wins) if  $b' = b$

# Proofs by Reduction: Example

Assumption:  $G : \{0, 1\}^n \rightarrow \{0, 1\}^{n+2}$  is PRG

Goal: Prove that  $G' = G(s)_{1,\dots,n+1}$  is a PRG

Proof:

- $G'$  expands from  $n$  bits to  $n + 1$  bits
- Assume there exists PPT  $\mathcal{A}_c$  that breaks  $G'$   
 $\Pr[PRG_{\mathcal{A}_c, G'}(n) = 1] > 1/2 + 1/\text{poly}(n)$ .  
Construct  $\mathcal{A}_r$  that breaks  $G$ :
  - $\mathcal{A}_r$  gets  $r \in \{0, 1\}^{n+2}$  as its challenge

$PRG_{\mathcal{D}, G}(n)$

- The challenger chooses  $b \leftarrow \{0, 1\}$ .  
If  $b = 0$ , he chooses  $r \leftarrow \{0, 1\}^{l(n)}$ ;  
if  $b = 1$ , he chooses  $s \leftarrow \{0, 1\}^n$ , and computes  $r = G(s)$ .  
He gives  $r$  to  $\mathcal{D}$ .
- On input  $r$ , the distinguisher  $\mathcal{D}$  outputs a guess  $b'$
- $PRG_{\mathcal{D}, G}(n) = 1$  (i.e.,  $\mathcal{D}$  wins) if  $b' = b$

# Proofs by Reduction: Example

Assumption:  $G : \{0, 1\}^n \rightarrow \{0, 1\}^{n+2}$  is PRG

Goal: Prove that  $G' = G(s)_{1,\dots,n+1}$  is a PRG

Proof:

- $G'$  expands from  $n$  bits to  $n + 1$  bits
- Assume there exists PPT  $\mathcal{A}_c$  that breaks  $G'$   
 $\Pr[PRG_{\mathcal{A}_c, G'}(n) = 1] > 1/2 + 1/\text{poly}(n)$ .

Construct  $\mathcal{A}_r$  that breaks  $G$ :

- $\mathcal{A}_r$  gets  $r \in \{0, 1\}^{n+2}$  as its challenge
- $\mathcal{A}_r$  computes  $r' = r_{1,\dots,n+1}$  and gives this as the challenge to  $\mathcal{A}_c$

$PRG_{\mathcal{D}, G}(n)$

- The challenger chooses  $b \leftarrow \{0, 1\}$ .  
If  $b = 0$ , he chooses  $r \leftarrow \{0, 1\}^{l(n)}$ ;  
if  $b = 1$ , he chooses  $s \leftarrow \{0, 1\}^n$ , and computes  $r = G(s)$ .  
He gives  $r$  to  $\mathcal{D}$ .
- On input  $r$ , the distinguisher  $\mathcal{D}$  outputs a guess  $b'$
- $PRG_{\mathcal{D}, G}(n) = 1$  (i.e.,  $\mathcal{D}$  wins) if  $b' = b$



# Proofs by Reduction: Example

Assumption:  $G : \{0, 1\}^n \rightarrow \{0, 1\}^{n+2}$  is PRG

Goal: Prove that  $G' = G(s)_{1,\dots,n+1}$  is a PRG

Proof:

- $G'$  expands from  $n$  bits to  $n + 1$  bits
- Assume there exists PPT  $\mathcal{A}_c$  that breaks  $G'$

$\Pr[PRG_{\mathcal{A}_c, G'}(n) = 1] > 1/2 + 1/\text{poly}(n)$ .

Construct  $\mathcal{A}_r$  that breaks  $G$ :

- $\mathcal{A}_r$  gets  $r \in \{0, 1\}^{n+2}$  as its challenge
- $\mathcal{A}_r$  computes  $r' = r_{1,\dots,n+1}$  and gives this as the challenge to  $\mathcal{A}_c$
- $\mathcal{A}_c$  outputs its guess  $b'$  and  $\mathcal{A}_r$  outputs this as its guess

$PRG_{\mathcal{D}, G}(n)$

- The challenger chooses  $b \leftarrow \{0, 1\}$ .  
If  $b = 0$ , he chooses  $r \leftarrow \{0, 1\}^{l(n)}$ ;  
if  $b = 1$ , he chooses  $s \leftarrow \{0, 1\}^n$ , and computes  $r = G(s)$ .  
He gives  $r$  to  $\mathcal{D}$ .
- On input  $r$ , the distinguisher  $\mathcal{D}$  outputs a guess  $b'$
- $PRG_{\mathcal{D}, G}(n) = 1$  (i.e.,  $\mathcal{D}$  wins) if  $b' = b$

# Proofs by Reduction: Example

Assumption:  $G : \{0, 1\}^n \rightarrow \{0, 1\}^{n+2}$  is PRG

Goal: Prove that  $G' = G(s)_{1,\dots,n+1}$  is a PRG

Proof:

- $G'$  expands from  $n$  bits to  $n + 1$  bits
- Assume there exists PPT  $\mathcal{A}_c$  that breaks  $G'$   
 $\Pr[PRG_{\mathcal{A}_c, G'}(n) = 1] > 1/2 + 1/\text{poly}(n)$ .  
Construct  $\mathcal{A}_r$  that breaks  $G$ :
  - $\mathcal{A}_r$  gets  $r \in \{0, 1\}^{n+2}$  as its challenge
  - $\mathcal{A}_r$  computes  $r' = r_{1,\dots,n+1}$  and gives this as the challenge to  $\mathcal{A}_c$
  - $\mathcal{A}_c$  outputs its guess  $b'$  and  $\mathcal{A}_r$  outputs this as its guess
- Analysis:
  - If  $b = 0$ , then  $r \leftarrow \{0, 1\}^{n+2}$  so  $r' \leftarrow \{0, 1\}^{n+1}$  (same as  $b = 0$  for  $\mathcal{A}_c$ )

$PRG_{\mathcal{D}, G}(n)$

- The challenger chooses  $b \leftarrow \{0, 1\}$ .  
If  $b = 0$ , he chooses  $r \leftarrow \{0, 1\}^{n+2}$ ;  
if  $b = 1$ , he chooses  $s \leftarrow \{0, 1\}^n$ , and computes  $r = G(s)$ .  
He gives  $r$  to  $\mathcal{D}$ .
- On input  $r$ , the distinguisher  $\mathcal{D}$  outputs a guess  $b'$
- $PRG_{\mathcal{D}, G}(n) = 1$  (i.e.,  $\mathcal{D}$  wins) if  $b' = b$

# Proofs by Reduction: Example

Assumption:  $G : \{0, 1\}^n \rightarrow \{0, 1\}^{n+2}$  is PRG

Goal: Prove that  $G' = G(s)_{1,\dots,n+1}$  is a PRG

Proof:

- $G'$  expands from  $n$  bits to  $n + 1$  bits
- Assume there exists PPT  $\mathcal{A}_c$  that breaks  $G'$   
 $\Pr[PRG_{\mathcal{A}_c, G'}(n) = 1] > 1/2 + 1/\text{poly}(n)$ .  
Construct  $\mathcal{A}_r$  that breaks  $G$ :
  - $\mathcal{A}_r$  gets  $r \in \{0, 1\}^{n+2}$  as its challenge
  - $\mathcal{A}_r$  computes  $r' = r_{1,\dots,n+1}$  and gives this as the challenge to  $\mathcal{A}_c$
  - $\mathcal{A}_c$  outputs its guess  $b'$  and  $\mathcal{A}_r$  outputs this as its guess
- Analysis:
  - If  $b = 0$ , then  $r \leftarrow \{0, 1\}^{n+2}$  so  $r' \leftarrow \{0, 1\}^{n+1}$  (same as  $b = 0$  for  $\mathcal{A}_c$ )
  - If  $b = 1$ , then  $r = G(s)$ , so  $r' = G'(s)$  (same as  $b = 1$  for  $\mathcal{A}_c$ )

$PRG_{\mathcal{D}, G}(n)$

- The challenger chooses  $b \leftarrow \{0, 1\}$ .  
If  $b = 0$ , he chooses  $r \leftarrow \{0, 1\}^{n+2}$ ;  
if  $b = 1$ , he chooses  $s \leftarrow \{0, 1\}^n$ , and computes  $r = G(s)$ .  
He gives  $r$  to  $\mathcal{D}$ .
- On input  $r$ , the distinguisher  $\mathcal{D}$  outputs a guess  $b'$
- $PRG_{\mathcal{D}, G}(n) = 1$  (i.e.,  $\mathcal{D}$  wins) if  $b' = b$

# Proofs by Reduction: Example

Assumption:  $G : \{0, 1\}^n \rightarrow \{0, 1\}^{n+2}$  is PRG

Goal: Prove that  $G' = G(s)_{1,\dots,n+1}$  is a PRG

Proof:

- $G'$  expands from  $n$  bits to  $n + 1$  bits
- Assume there exists PPT  $\mathcal{A}_c$  that breaks  $G'$   
 $\Pr[\text{PRG}_{\mathcal{A}_c, G'}(n) = 1] > 1/2 + 1/\text{poly}(n)$ .  
Construct  $\mathcal{A}_r$  that breaks  $G$ :
  - $\mathcal{A}_r$  gets  $r \in \{0, 1\}^{n+2}$  as its challenge
  - $\mathcal{A}_r$  computes  $r' = r_{1,\dots,n+1}$  and gives this as the challenge to  $\mathcal{A}_c$
  - $\mathcal{A}_c$  outputs its guess  $b'$  and  $\mathcal{A}_r$  outputs this as its guess
- Analysis:
  - If  $b = 0$ , then  $r \leftarrow \{0, 1\}^{n+2}$  so  $r' \leftarrow \{0, 1\}^{n+1}$  (same as  $b = 0$  for  $\mathcal{A}_c$ )
  - If  $b = 1$ , then  $r = G(s)$ , so  $r' = G'(s)$  (same as  $b = 1$  for  $\mathcal{A}_c$ )
  - If  $\mathcal{A}_c$  outputs  $b=b'$ , then  $\mathcal{A}_r$  outputs  $b = b'$

$\text{PRG}_{\mathcal{D}, G}(n)$

- The challenger chooses  $b \leftarrow \{0, 1\}$ .  
If  $b = 0$ , he chooses  $r \leftarrow \{0, 1\}^{l(n)}$ ;  
if  $b = 1$ , he chooses  $s \leftarrow \{0, 1\}^n$ , and computes  $r = G(s)$ .  
He gives  $r$  to  $\mathcal{D}$ .
- On input  $r$ , the distinguisher  $\mathcal{D}$  outputs a guess  $b'$
- $\text{PRG}_{\mathcal{D}, G}(n) = 1$  (i.e.,  $\mathcal{D}$  wins) if  $b' = b$

# Proofs by Reduction: Example

Assumption:  $G : \{0, 1\}^n \rightarrow \{0, 1\}^{n+2}$  is PRG

Goal: Prove that  $G' = G(s)_{1,\dots,n+1}$  is a PRG

Proof:

- $G'$  expands from  $n$  bits to  $n + 1$  bits
- Assume there exists PPT  $\mathcal{A}_c$  that breaks  $G'$

$$\Pr[\text{PRG}_{\mathcal{A}_c, G'}(n) = 1] > 1/2 + 1/\text{poly}(n).$$

Construct  $\mathcal{A}_r$  that breaks  $G$ :

- $\mathcal{A}_r$  gets  $r \in \{0, 1\}^{n+2}$  as its challenge
- $\mathcal{A}_r$  computes  $r' = r_{1,\dots,n+1}$  and gives this as the challenge to  $\mathcal{A}_c$
- $\mathcal{A}_c$  outputs its guess  $b'$  and  $\mathcal{A}_r$  outputs this as its guess
- Analysis:
  - If  $b = 0$ , then  $r \leftarrow \{0, 1\}^{n+2}$  so  $r' \leftarrow \{0, 1\}^{n+1}$  (same as  $b = 0$  for  $\mathcal{A}_c$ )
  - If  $b = 1$ , then  $r = G(s)$ , so  $r' = G'(s)$  (same as  $b = 1$  for  $\mathcal{A}_c$ )
  - If  $\mathcal{A}_c$  outputs  $b=b'$ , then  $\mathcal{A}_r$  outputs  $b = b'$
  - Since  $\Pr[\text{PRG}_{\mathcal{A}_c, G'}(n) = 1] > 1/2 + 1/\text{poly}(n)$ , we get that  $\Pr[\text{PRG}_{\mathcal{A}_r, G}(n) = 1] > 1/2 + 1/\text{poly}(n)$

$\text{PRG}_{\mathcal{D}, G}(n)$

- The challenger chooses  $b \leftarrow \{0, 1\}$ .  
If  $b = 0$ , he chooses  $r \leftarrow \{0, 1\}^{n+2}$ ;  
if  $b = 1$ , he chooses  $s \leftarrow \{0, 1\}^n$ , and computes  $r = G(s)$ .  
He gives  $r$  to  $\mathcal{D}$ .
- On input  $r$ , the distinguisher  $\mathcal{D}$  outputs a guess  $b'$
- $\text{PRG}_{\mathcal{D}, G}(n) = 1$  (i.e.,  $\mathcal{D}$  wins) if  $b' = b$

# Proof by Reduction: Observations

- Construction only uses  $G$  as a black-box, does not look at how  $G$  works.

# Proof by Reduction: Observations

- Construction only uses  $G$  as a black-box, does not look at how  $G$  works.
  - So, construction must work starting from any PRG  $G$  (even a really “weird” one)

# Proof by Reduction: Observations

- Construction only uses  $G$  as a black-box, does not look at how  $G$  works.
  - So, construction must work starting from any PRG  $G$  (even a really “weird” one)
  - Common way of proving a construction insecure is to find such a weird  $G$  for which reduction fails.



# Proof by Reduction: Observations

- Construction only uses  $G$  as a black-box, does not look at how  $G$  works.
  - So, construction must work starting from any PRG  $G$  (even a really “weird” one)
  - Common way of proving a construction insecure is to find such a weird  $G$  for which reduction fails.
- Reduction ( $\mathcal{A}_r$ ) only uses  $\mathcal{A}_c$  as a black-box, does not look at how  $\mathcal{A}_c$  works

# Proof by Reduction: Observations

- Construction only uses  $G$  as a black-box, does not look at how  $G$  works.
  - So, construction must work starting from any PRG  $G$  (even a really “weird” one)
  - Common way of proving a construction insecure is to find such a weird  $G$  for which reduction fails.
- Reduction ( $\mathcal{A}_r$ ) only uses  $\mathcal{A}_c$  as a black-box, does not look at how  $\mathcal{A}_c$  works

## Fully Black-Box Reductions

- Such reductions are called *fully black-box*
- (Almost) all reductions in cryptography are fully black-box