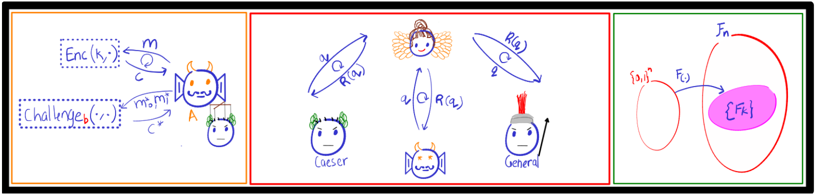
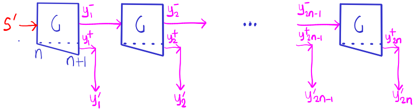


CS783: Theoretical Foundations of Cryptography

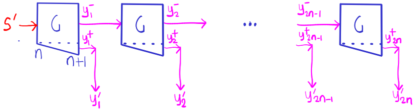
Lecture 5 (13/Aug/24)

Instructor: Chethan Kamath

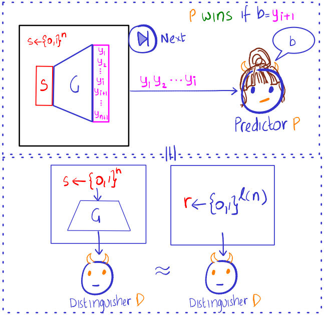
Recall from Last Lecture...

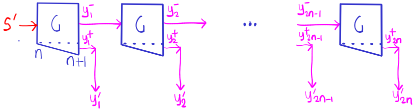


- Length extension of PRG and hybrid argument

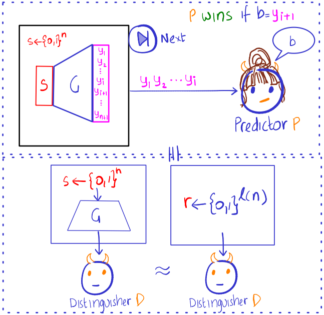


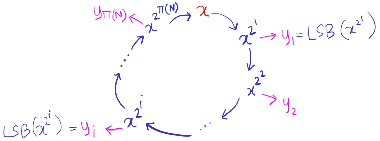
- Length extension of PRG and hybrid argument
- Pseudorandomness vs unpredictability
  - Equivalence between the two notions





- Length extension of PRG and hybrid argument
- Pseudorandomness vs unpredictability
  - Equivalence between the two notions
- Unpredictable sequences from integer factoring





Recall from Last Lecture...

secret communication of longer messages

General *template*:

1. Identify the task
2. Come up with precise threat model M (a.k.a security model)
   - Adversary/Attack: What are the adversary's capabilities? Eavesdropper
   - Security Goal: What does it mean to be secure? Computational secrecy
3. Construct a scheme Π Computational one-time pad from PRG
4. Formally prove that Π in secure in model M

Security reduction from PRG

Plan for This Lecture

multiple

secret communication of ~~longer~~ messages

General *template*:

1. Identify the task
2. Come up with precise threat model M (a.k.a security model)
   - Adversary/Attack: What are the adversary's capabilities? Eavesdropper*
   - Security Goal: What does it mean to be secure? Computational secrecy
3. Construct a scheme Π Computational one-time pad from PRG
4. Formally prove that Π in secure in model M

Security reduction from PRG

Plan for This Lecture

multiple
secret communication of ~~longer~~ messages

General *template*:

1. Identify the task
2. Come up with precise threat model M (a.k.a security model)
   - Adversary/Attack: What are the adversary's capabilities? Eavesdropper*
   - Security Goal: What does it mean to be secure? Computational secrecy
3. Construct a scheme Π Computational one-time pad from PRF
4. Formally prove that Π in secure in model M

Security reduction from PRF

Plan for This Lecture

secret communication of ~~longer~~ multiple messages

General *template*:

1. Identify the task
2. Come up with precise threat model M (a.k.a security model)
   - Adversary/Attack: What are the adversary's capabilities? Eavesdropper* +1 chosen-plaintext attack
   - Security Goal: What does it mean to be secure? Computational secrecy
3. Construct a scheme Π Computational one-time pad from ~~PRG~~ PRF
4. Formally prove that Π in secure in model M

Security reduction from ~~PRG~~ PRF

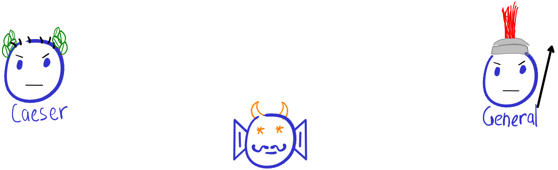
Plan for This Lecture

1. Pseudo-Random Function (PRF)

2. Goldreich-Goldwasser-Micali (GGM) Construction

Let's Encrypt Many Messages Using PRG G...

- Setting: Caeser and his general share a key $k \in \{0,1\}^n$ and want to secretly communicate n messages from $\{0,1\}^n$ in presence of eavesdropper Eve*

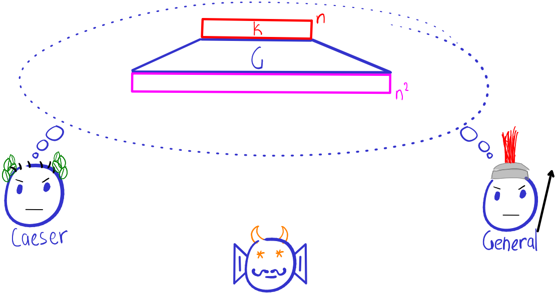


Let's Encrypt Many Messages Using PRG G

$= n$

$n \to n^2$

- Setting: Caeser and his general share a key $k \in \{0,1\}^n$ and want to secretly communicate n messages from $\{0,1\}^n$ in presence of eavesdropper Eve*

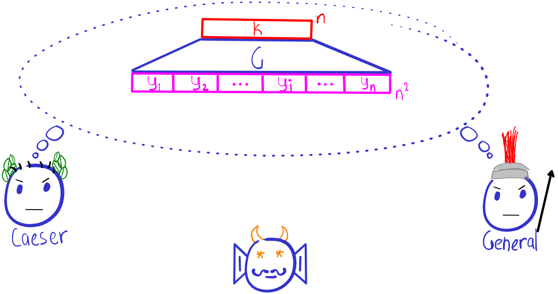


Let's Encrypt Many Messages Using PRG G

$= n$

$n \to n^2$

- Setting: Caeser and his general share a key $k \in \{0,1\}^n$ and want to secretly communicate n messages from $\{0,1\}^n$ in presence of eavesdropper Eve*



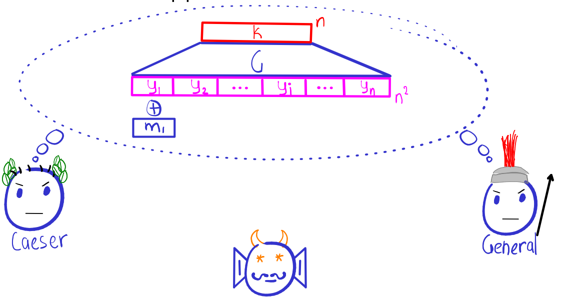
- SKE construction: use output of G as n pseudorandom OTPs

Let's Encrypt Many Messages Using PRG G

$= n$

$n \to n^2$

- Setting: Caeser and his general share a key $k \in \{0,1\}^n$ and want to secretly communicate n messages from $\{0,1\}^n$ in presence of eavesdropper Eve*



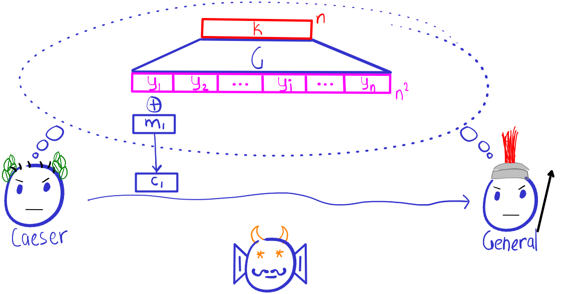
- SKE construction: use output of G as n pseudorandom OTPs

Let's Encrypt Many Messages Using PRG G

$\overset{\shortparallel}{n}$

$n \rightarrow n^2$

- Setting: Caeser and his general share a key $k \in \{0,1\}^n$ and want to secretly communicate n messages from $\{0,1\}^n$ in presence of eavesdropper Eve*

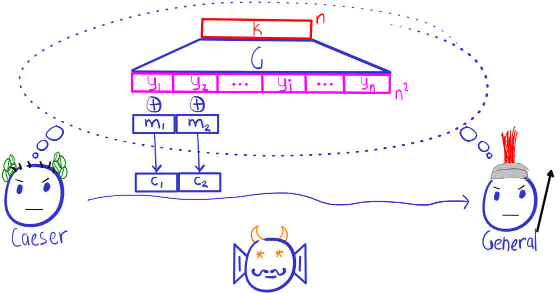


- SKE construction: use output of G as n pseudorandom OTPs

$= n$

$n \to n^2$

- Setting: Caeser and his general share a key $k \in \{0,1\}^n$ and want to secretly communicate n messages from $\{0,1\}^n$ in presence of eavesdropper Eve*



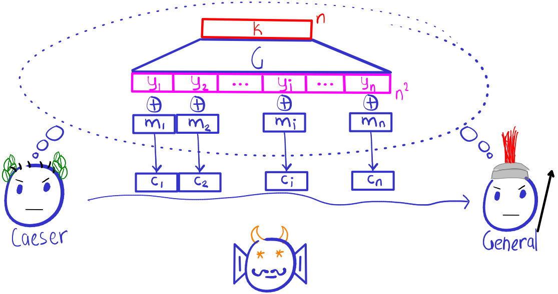
- SKE construction: use output of G as n pseudorandom OTPs

Let's Encrypt Many Messages Using PRG G

$\overset{\shortparallel}{n}$

$n \to n^2$

- Setting: Caeser and his general share a key $k \in \{0,1\}^n$ and want to secretly communicate n messages from $\{0,1\}^n$ in presence of eavesdropper Eve*



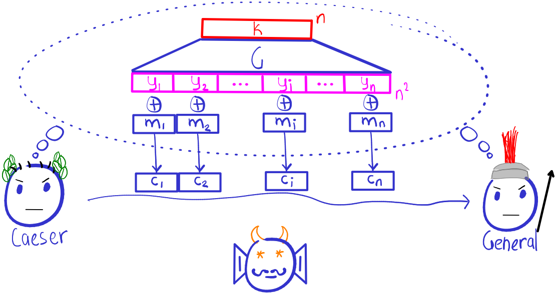
- SKE construction: use output of G as n pseudorandom OTPs

Let's Encrypt Many Messages Using PRG G

$\| \; n$

$n \to n^2$

- Setting: Caeser and his general share a key $k \in \{0,1\}^n$ and want to secretly communicate n messages from $\{0,1\}^n$ in presence of eavesdropper Eve*

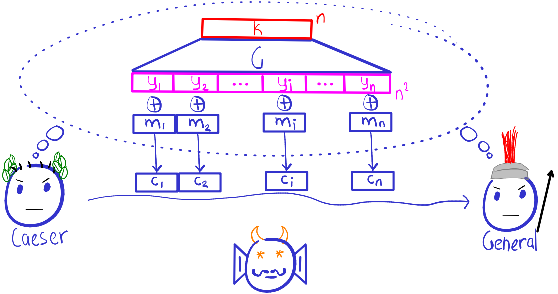


- SKE construction: use output of G as n pseudorandom OTPs
- Problem: construction *stateful*; synchrony must be maintained
  - We lose correctness if (e.g.) ciphertexts delivered out of order

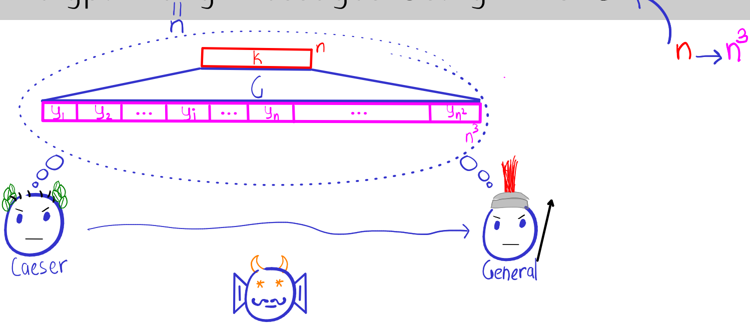
Let's Encrypt Many Messages Using PRG G

$\overset{\|}{n}$ $n \to n^2$

- Setting: Caeser and his general share a key $k \in \{0,1\}^n$ and want to secretly communicate n messages from $\{0,1\}^n$ in presence of eavesdropper Eve*



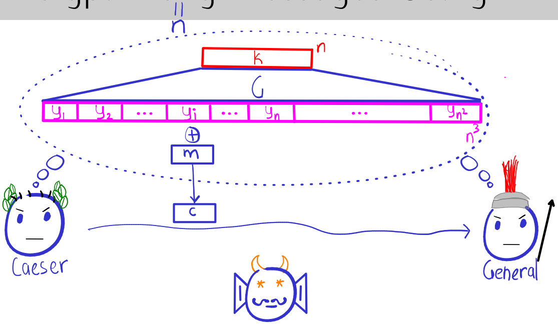
- SKE construction: use output of G as n pseudorandom OTPs
- Problem: construction *stateful*; synchrony must be maintained
  - We lose correctness if (e.g.) ciphertexts delivered out of order
  - ? Come up with a scenario that leads to loss of secrecy



What if the stretch is n^3?

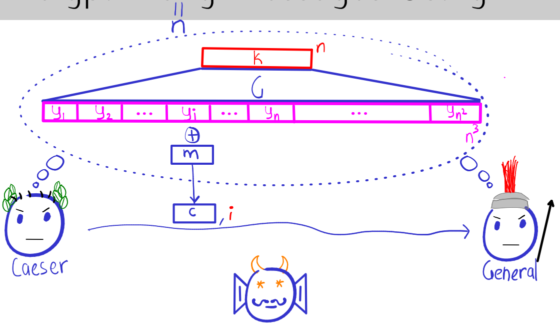
Let's Encrypt Many Messages Using PRG G...

$n \to n^3$



What if the stretch is n^3? Use OTP at random index $i \in [1, n^2]$

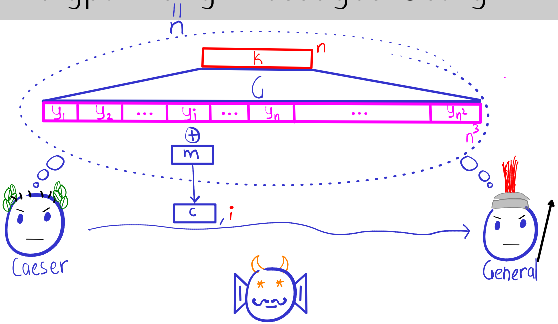
$n \to n^3$



What if the stretch is n^3? Use OTP at random index $i \in [1, n^2]$

Let's Encrypt Many Messages Using PRG G...

$n \to n^3$

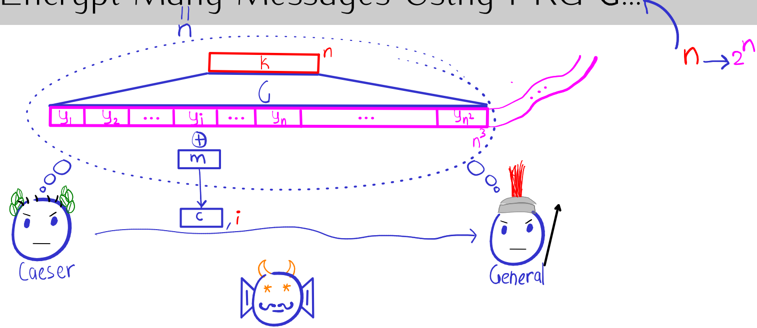


What if the stretch is n^3? Use OTP at random index $i \in [1, n^2]$

- Problem? Collision

- Underlying problem: only poly. pseudorandom OTPs available

Let's Encrypt Many Messages Using PRG G...

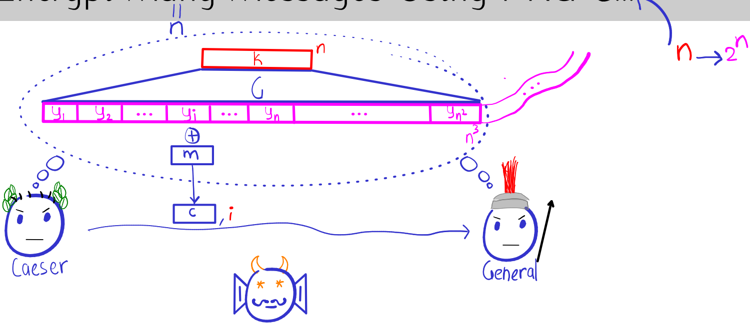


❓ What if the stretch is n^3? Use OTP at random index $i \in [1, n^2]$

- Problem? Collision

- Underlying problem: only poly. pseudorandom OTPs available

❓ What if we stretch the PRG exponentially?



? What if the stretch is n^3? Use OTP at random index $i \in [1, n^2]$

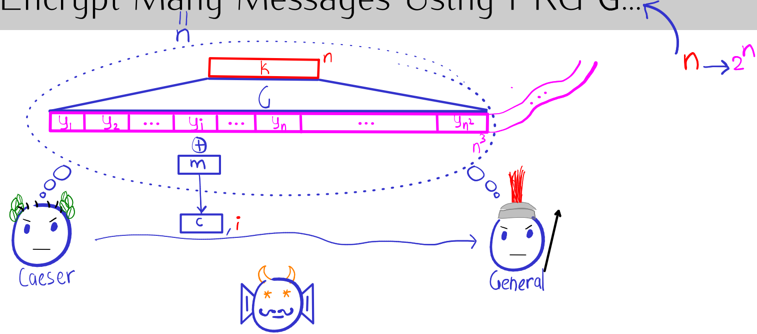
- Problem? Collision

- Underlying problem: only poly. pseudorandom OTPs available

? What if we stretch the PRG exponentially?

- Not all pseudorandom OTPs are efficiently "accessible"

Let's Encrypt Many Messages Using PRG G...



? What if the stretch is n^3? Use OTP at random index $i \in [1, n^2]$

- Problem? Collision

- Underlying problem: only poly. pseudorandom OTPs available

? What if we stretch the PRG exponentially?

- Not all pseudorandom OTPs are efficiently "accessible"

- Need "PRG" with
  1. Exponential stretch
  2. Output bits "efficiently" accessible (also called locality)

Let's Encrypt Many Messages Using an Oracle in the Sky

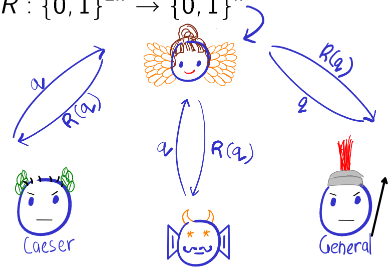
- Setting:
  - Caeser and his general have shared a key $k \in \{0,1\}^n$
  - Everyone (including Eve*) has access to a *random function oracle* $R : \{0,1\}^{2n} \rightarrow \{0,1\}^n$

Let's Encrypt Many Messages Using an Oracle in the Sky

- Setting:
  - Caeser and his general have shared a key $k \in \{0,1\}^n$
  - Everyone (including Eve*) has access to a *random function oracle* $R : \{0,1\}^{2n} \to \{0,1\}^n$

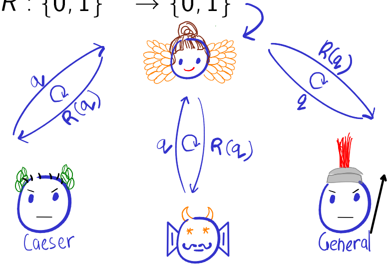
Let's Encrypt Many Messages Using an Oracle in the Sky

- Setting:
  - Caeser and his general have shared a key $k \in \{0,1\}^n$
  - Everyone (including Eve*) has access to a *random function oracle* $R : \{0,1\}^{2n} \to \{0,1\}^n$



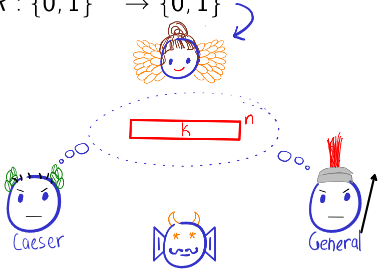
Let's Encrypt Many Messages Using an Oracle in the Sky

- Setting:
  - Caeser and his general have shared a key $k \in \{0,1\}^n$
  - Everyone (including Eve*) has access to a *random function oracle* $R : \{0,1\}^{2n} \to \{0,1\}^n$



Let's Encrypt Many Messages Using an Oracle in the Sky

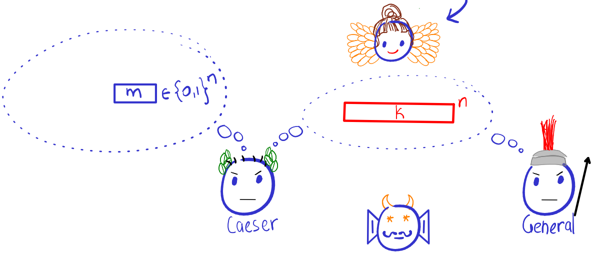
- Setting:
  - Caeser and his general have shared a key $k \in \{0,1\}^n$
  - Everyone (including Eve*) has access to a *random function oracle* $R : \{0,1\}^{2n} \to \{0,1\}^n$



How will you construct a stateless encryption scheme given R?

Let's Encrypt Many Messages Using an Oracle in the Sky

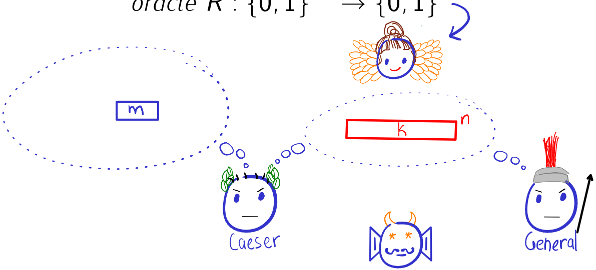
- Setting:
  - Caeser and his general have shared a key $k \in \{0,1\}^n$
  - Everyone (including Eve*) has access to a *random function oracle* $R : \{0,1\}^{2n} \to \{0,1\}^n$



How will you construct a stateless encryption scheme given R?

Let's Encrypt Many Messages Using an Oracle in the Sky

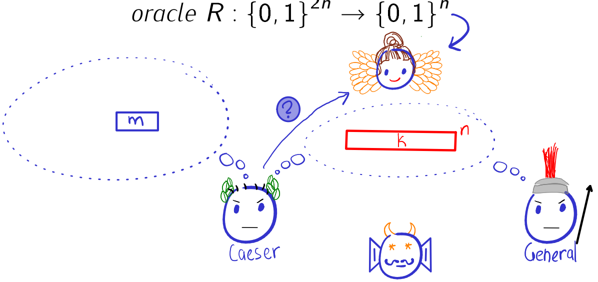
- Setting:
  - Caeser and his general have shared a key $k \in \{0,1\}^n$
  - Everyone (including Eve*) has access to a *random function oracle* $R : \{0,1\}^{2n} \to \{0,1\}^n$



? How will you construct a stateless encryption scheme given R?
  - Hint: R helps generate *exponentially-many* random OTPs

Let's Encrypt Many Messages Using an Oracle in the Sky

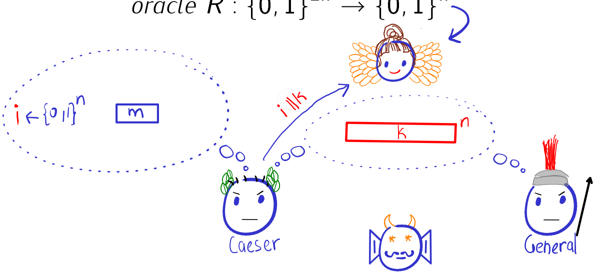
- Setting:
  - Caeser and his general have shared a key $k \in \{0,1\}^n$
  - Everyone (including Eve*) has access to a *random function oracle* $R : \{0,1\}^{2n} \to \{0,1\}^n$



How will you construct a stateless encryption scheme given R?
  - Hint: R helps generate *exponentially-many* random OTPs

Let's Encrypt Many Messages Using an Oracle in the Sky

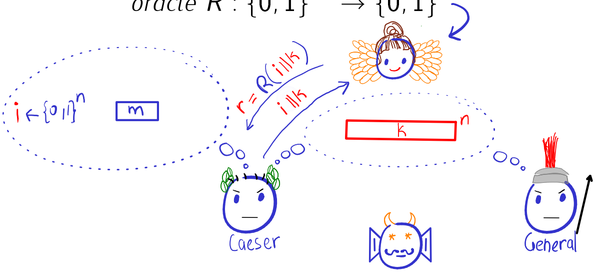
- Setting:
  - Caeser and his general have shared a key $k \in \{0,1\}^n$
  - Everyone (including Eve*) has access to a *random function oracle* $R : \{0,1\}^{2n} \to \{0,1\}^n$



How will you construct a stateless encryption scheme given R?
  - Hint: R helps generate *exponentially-many* random OTPs

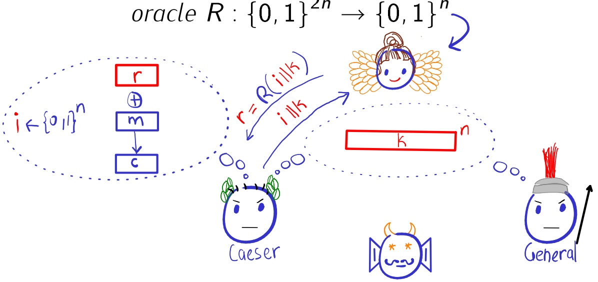
Let's Encrypt Many Messages Using an Oracle in the Sky

- Setting:
  - Caeser and his general have shared a key $k \in \{0,1\}^n$
  - Everyone (including Eve*) has access to a *random function oracle* $R : \{0,1\}^{2n} \to \{0,1\}^n$



How will you construct a stateless encryption scheme given R?
- Hint: R helps generate *exponentially-many* random OTPs

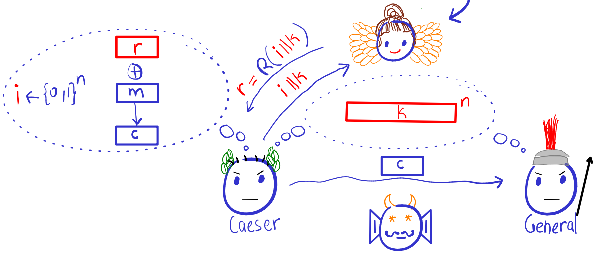
- Setting:
  - Caeser and his general have shared a key $k \in \{0,1\}^n$
  - Everyone (including Eve*) has access to a *random function oracle* $R : \{0,1\}^{2n} \to \{0,1\}^n$



❓ How will you construct a stateless encryption scheme given R?
  - Hint: R helps generate *exponentially-many* random OTPs

Let's Encrypt Many Messages Using an Oracle in the Sky

- Setting:
  - Caeser and his general have shared a key $k \in \{0,1\}^n$
  - Everyone (including Eve*) has access to a *random function oracle* $R : \{0,1\}^{2n} \to \{0,1\}^n$

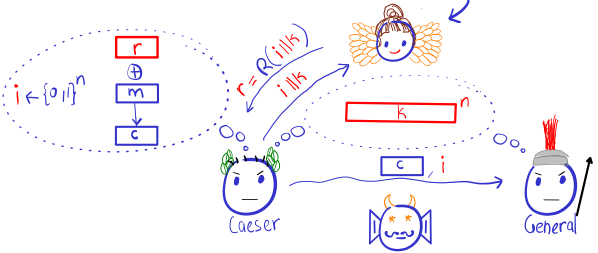


How will you construct a stateless encryption scheme given R?

- Hint: R helps generate *exponentially-many* random OTPs

Let's Encrypt Many Messages Using an Oracle in the Sky

- Setting:
  - Caeser and his general have shared a key $k \in \{0,1\}^n$
  - Everyone (including Eve*) has access to a *random function oracle* $R : \{0,1\}^{2n} \to \{0,1\}^n$

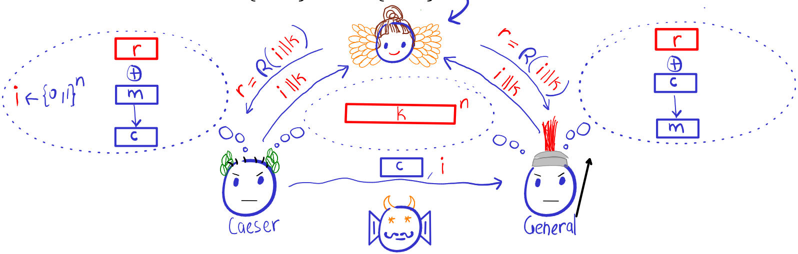


How will you construct a stateless encryption scheme given R?

- Hint: R helps generate *exponentially-many* random OTPs

Let's Encrypt Many Messages Using an Oracle in the Sky

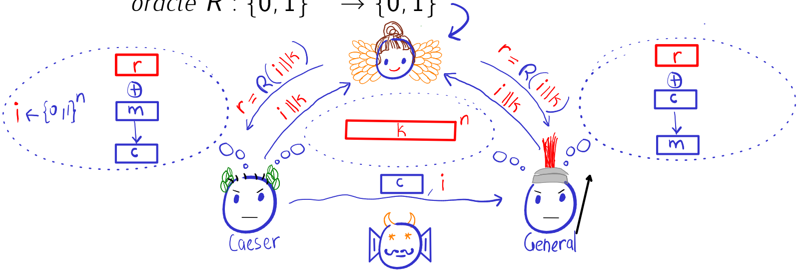
- Setting:
  - Caeser and his general have shared a key $k \in \{0,1\}^n$
  - Everyone (including Eve*) has access to a *random function oracle* $R : \{0,1\}^{2n} \to \{0,1\}^n$



? How will you construct a stateless encryption scheme given R?
  - Hint: R helps generate *exponentially-many* random OTPs

Let's Encrypt Many Messages Using an Oracle in the Sky

- Setting:
  - Caeser and his general have shared a key $k \in \{0,1\}^n$
  - Everyone (including Eve*) has access to a *random function oracle* $R : \{0,1\}^{2n} \to \{0,1\}^n$



How will you construct a stateless encryption scheme given R?
  - Hint: R helps generate *exponentially-many* random OTPs

Exercise 1

What if Caeser and his general did not have the shared key k? Can they still do something given the oracle in the sky?

Plan for This Lecture

Computational analogue of random oracle

1 Pseudo-Random Function (PRF)

2 Goldreich-Goldwasser-Micali (GGM) Construction

PRF: Computational Analogue of Oracle in the Sky

- A function F that "seems like" a random function oracle to PPT distinguishers

PRF: Computational Analogue of Oracle in the Sky

- A function F that "seems like" a random function oracle to PPT distinguishers
- More formally:
  - F_k sampled at random from a (smallish) family of functions $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$
  - $\approx$
  - A random function, sampled from the set of *all* functions $\mathcal{F}_n$

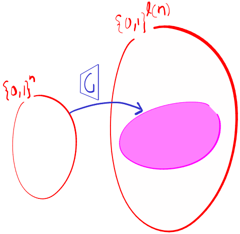
$\{f : \{0,1\}^n \to \{0,1\}^n\}$

PRF: Computational Analogue of Oracle in the Sky

- A function F that "seems like" a random function oracle to PPT distinguishers
- More formally:
  - F_k sampled at random from a (smallish) family of functions $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$

    $\approx$
  - A random function, sampled from the set of *all* functions $\mathcal{F}_n$

$\{f : \{0,1\}^n \to \{0,1\}^n\}$

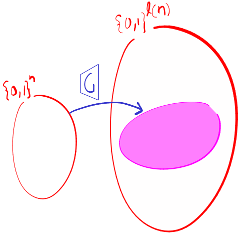


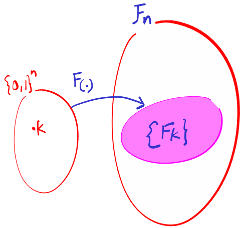
PRF: Computational Analogue of Oracle in the Sky

- A function F that "seems like" a random function oracle to PPT distinguishers
- More formally:
  - F_k sampled at random from a (smallish) family of functions $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$

    $\approx$
  - A random function, sampled from the set of *all* functions $\mathcal{F}_n$

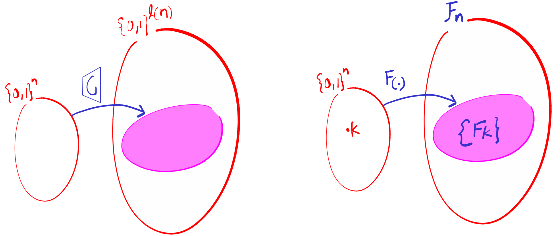
$\{f : \{0,1\}^n \to \{0,1\}^n\}$





- A function F that "seems like" a random function oracle to PPT distinguishers
- More formally:
  - F_k sampled at random from a (smallish) family of functions $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$
  - A random function, sampled from the set of *all* functions $\mathcal{F}_n$

$\{f : \{0,1\}^n \to \{0,1\}^n\}$



? Number of functions in $\{F_k\}$ vs. number of functions $\mathcal{F}_n$?

- Why is it still useful?
  - Helps generate exponentially-many *pseudorandom* OTPs

How Exactly to Define Pseudorandomess for Functions?...

- A function F that "seems like" random function oracle to PPT distinguishers

How Exactly to Define Pseudorandomess for Functions?

- A function F that "seems like" random function oracle to PPT distinguishers
- Recall how we defined pseudorandomness for PRG (Lecture 3)

G is PRG if for every PPT distinguisher D

$$\delta(n) := \left| \Pr_{s \leftarrow \{0,1\}^n}[D(G(s)) = 0] - \Pr_{r \leftarrow \{0,1\}^{\ell(n)}}[D(r) = 0] \right|$$

is negligible.

How Exactly to Define Pseudorandomess for Functions?...

- A function F that "seems like" random function oracle to PPT distinguishers
- Recall how we defined pseudorandomness for PRG (Lecture 3)

G is PRG if for every PPT distinguisher D

$$\delta(n) := \left| \Pr_{s \leftarrow \{0,1\}^n}[D(G(s)) = 0] - \Pr_{r \leftarrow \{0,1\}^{\ell(n)}}[D(r) = 0] \right|$$

is negligible. pseudorandom world random world

How Exactly to Define Pseudorandomess for Functions?...

- A function F that "seems like" random function oracle to PPT distinguishers
- Recall how we defined pseudorandomness for PRG (Lecture 3)

G is PRG if for every PPT distinguisher D

$$\delta(n) := \left| \Pr_{s \leftarrow \{0,1\}^n}[D(G(s)) = 0] - \Pr_{r \leftarrow \{0,1\}^{\ell(n)}}[D(r) = 0] \right|$$

is negligible. pseudorandom world random world

Can we give the distinguisher *full description* of the function (e.g., as a table)?

- No, then it becomes easy to distinguish
- How? (Recall: run-time measured w.r.to size of input)

How Exactly to Define Pseudorandomess for Functions?...

- A function F that "seems like" random function oracle to PPT distinguishers
- Recall how we defined pseudorandomness for PRG (Lecture 3)

G *is PRG if for every PPT distinguisher D*

$$\delta(n) := \left| \Pr_{s \leftarrow \{0,1\}^n}[D(G(s)) = 0] - \Pr_{r \leftarrow \{0,1\}^{\ell(n)}}[D(r) = 0] \right|$$

is negligible. pseudorandom world random world

Can we give the distinguisher *full description* of the function (e.g., as a table)?

- No, then it becomes easy to distinguish
- How? (Recall: run-time measured w.r.to size of input)

- Way around:
  - Distinguisher given *oracle* access to the functions
  - One query=one unit of running time → efficient PPT distinguisher can only make polynomially-many queries

How Exactly to Define Pseudorandomess for Functions?...

Defintion 1 (PRF, via Imitation Game)

A family of functions $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$ is a PRF if for every PPT oracle distinguisher D

$$\delta(n) := \left| \Pr_{k \leftarrow \{0,1\}^n}[D^{F_k(\cdot)}(1^n) = 0] - \Pr_{f \leftarrow \mathcal{F}_n}[D^{f(\cdot)}(1^n) = 0] \right|$$

is negligible.

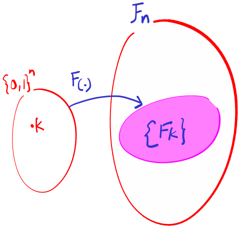
How Exactly to Define Pseudorandomess for Functions?...

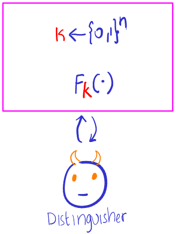
Defintion 1 (PRF, via Imitation Game)

A family of functions $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$ *is a PRF if for every PPT oracle distinguisher* D

$$\delta(n) := \left| \Pr_{k \leftarrow \{0,1\}^n}[D^{F_k(\cdot)}(1^n) = 0] - \Pr_{f \leftarrow \mathcal{F}_n}[D^{f(\cdot)}(1^n) = 0] \right|$$

is negligible. pseudorandom world





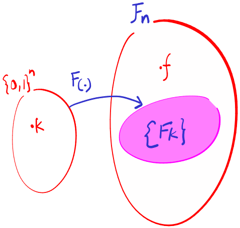
How Exactly to Define Pseudorandomess for Functions?...

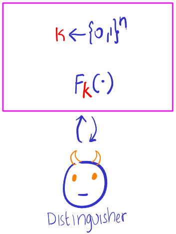
Defintion 1 (PRF, via Imitation Game)

A family of functions $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$ *is a PRF if for every PPT oracle distinguisher* D

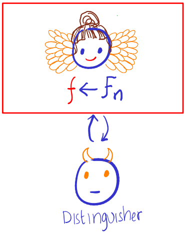
$$\delta(n) := \left| \Pr_{k \leftarrow \{0,1\}^n}[D^{F_k(\cdot)}(1^n) = 0] - \Pr_{f \leftarrow \mathcal{F}_n}[D^{f(\cdot)}(1^n) = 0] \right|$$

is negligible. pseudorandom world random world





$\approx$



Let's Check if You Understood Defintion 1

PRF or not? Below $F^{(1)}$ and $F^{(2)}$ are PRFs

1. $F_k(x) := k \oplus x$
2. $F_{k_1 k_2}(x) := F^{(1)}_{k_1}(x) F^{(2)}_{k_2}(x)$
3. $F_k(x_1 x_2) := F^{(1)}_k(x_1) F^{(2)}_k(x_2)$

Let's Check if You Understood Defintion 1

PRF or not? Below $F^{(1)}$ and $F^{(2)}$ are PRFs

1. $F_k(x) := k \oplus x$
2. $F_{k_1 k_2}(x) := F^{(1)}_{k_1}(x) F^{(2)}_{k_2}(x)$
3. $F_k(x_1 x_2) := F^{(1)}_k(x_1) F^{(2)}_k(x_2)$

PRG or not? Below, F is a PRF

1. $\mathsf{G}(s) := F_s(1) F_s(2) \cdots F_s(n-1) F_s(n)$
2. $\mathsf{G}(s) := F_s(2^0) F_s(2^1) \cdots F_s(2^{n-1}) F_s(2^n)$
3. $\mathsf{G}(s) := F_1(s) F_2(s) \cdots F_{n-1}(s) F_n(s)$

Let's Check if You Understood Defintion 1

PRF or not? Below $F^{(1)}$ and $F^{(2)}$ are PRFs

1. $F_k(x) := k \oplus x$
2. $F_{k_1 k_2}(x) := F^{(1)}_{k_1}(x) F^{(2)}_{k_2}(x)$
3. $F_k(x_1 x_2) := F^{(1)}_k(x_1) F^{(2)}_k(x_2)$

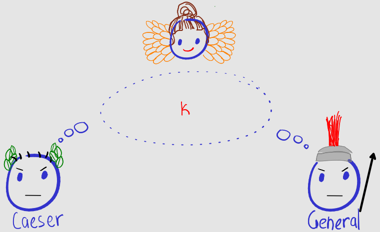
PRG or not? Below, F is a PRF

1. $\mathsf{G}(s) := F_s(1) F_s(2) \cdots F_s(n-1) F_s(n)$
2. $\mathsf{G}(s) := F_s(2^0) F_s(2^1) \cdots F_s(2^{n-1}) F_s(2^n)$
3. $\mathsf{G}(s) := F_1(s) F_2(s) \cdots F_{n-1}(s) F_n(s)$

Exercise 2

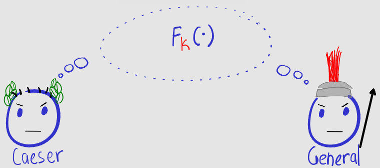
In all the "yes" cases above, formally prove; in all the "no" cases, describe a counter-example.

Construction 1 (Replace random oracle with PRF)



Stateless Symmetric-Key Encryption from PRF

Construction 1 (Replace random oracle with PRF)



Stateless Symmetric-Key Encryption from PRF

Construction 1 (Replace random oracle with PRF)

Construction 1 (Replace random oracle with PRF)

Stateless Symmetric-Key Encryption from PRF

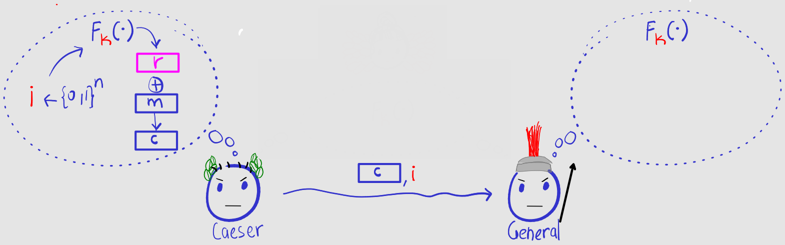
Construction 1 (Replace random oracle with PRF)

Stateless Symmetric-Key Encryption from PRF

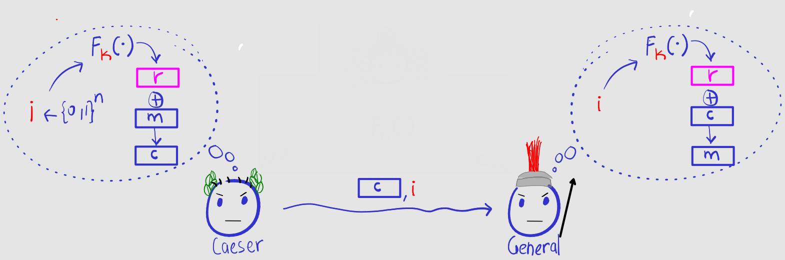
Construction 1 (Replace random oracle with PRF)

Stateless Symmetric-Key Encryption from PRF

Construction 1 (Replace random oracle with PRF)

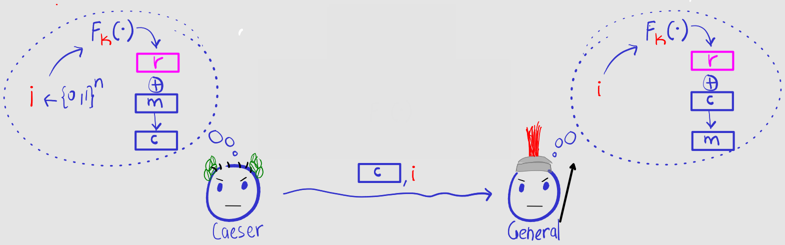


Construction 1 (Replace random oracle with PRF)



Stateless Symmetric-Key Encryption from PRF

Construction 1 (Replace random oracle with PRF)



- Note: encryption is randomised and thus length of ciphertext is longer than plaintext (first such scheme in this course)

Exercise 3 (Hint: reduction similar to pseudorandom OTP)

Prove that Construction 1 is secure against eavesdroppers.

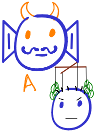
In Fact you Get More: CPA-Secret SKE!

- Stronger adversaries who can influence Caeser's messages

In Fact you Get More: CPA-Secret SKE!

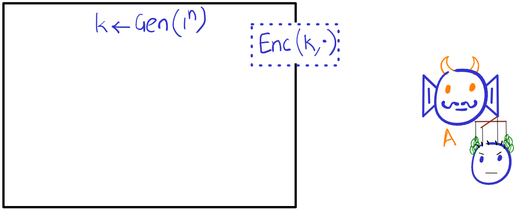
- Stronger adversaries who can influence Caeser's messages

$k \leftarrow Gen(1^n)$



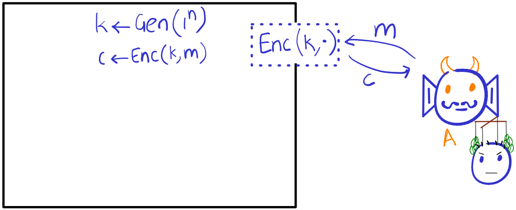
In Fact you Get More: CPA-Secret SKE!

- Stronger adversaries who can influence Caeser's messages



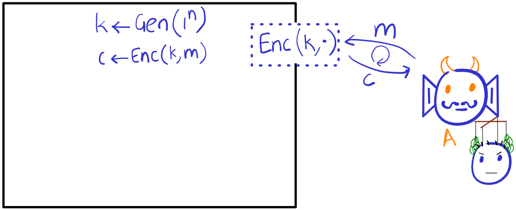
In Fact you Get More: CPA-Secret SKE!

- Stronger adversaries who can influence Caeser's messages



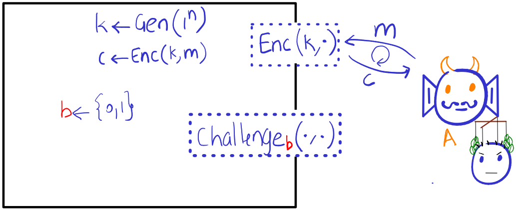
In Fact you Get More: CPA-Secret SKE!

- Stronger adversaries who can influence Caeser's messages



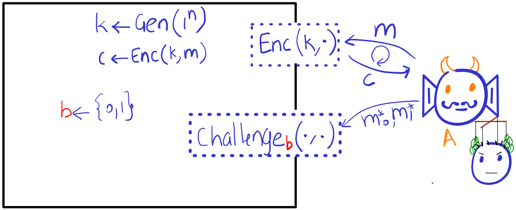
In Fact you Get More: CPA-Secret SKE!

- Stronger adversaries who can influence Caeser's messages

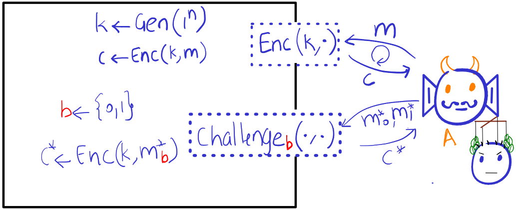


In Fact you Get More: CPA-Secret SKE!

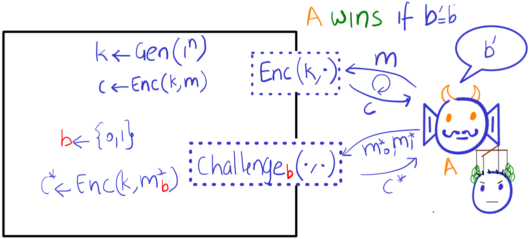
- Stronger adversaries who can influence Caeser's messages



- Stronger adversaries who can influence Caeser's messages



- Stronger adversaries who can influence Caeser's messages



In Fact you Get More: CPA-Secret SKE!

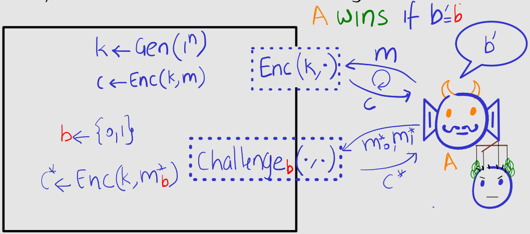
- Stronger adversaries who can influence Caeser's messages

Defintion 2 (Secrecy against chosen-plaintext attack (CPA))

An SKE $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is CPA-secret if for every PPT CPA adversary A

$$\Pr[A \text{ wins}] - 1/2$$

is negligible.



In Fact you Get More: CPA-Secret SKE!

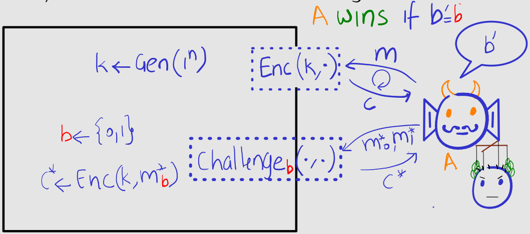
- Stronger adversaries who can influence Caeser's messages

Defintion 2 (Secrecy against chosen-plaintext attack (CPA))

An SKE $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is CPA-secret if for every PPT CPA adversary A

$$\Pr[A \text{ wins}] - 1/2$$

is negligible.



Exercise 4 (CPA model)

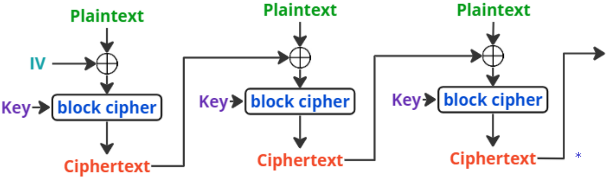
1. *Show that computational OTP (Lecture 3) is not CPA-secret*
2. *Prove that Construction 1 is CPA-secret*

PRFs IRL

- Coming up: theoretical construction, but inefficient for practice
- Practical PRFs: block ciphers like AES, which however only support certain key-sizes (128, 192, 256)
  - Supported by most libraries (e.g., OpenSSL, NaCl) and even implemented on modern processors (AES-NI)

PRFs IRL

- Coming up: theoretical construction, but inefficient for practice
- Practical PRFs: block ciphers like AES, which however only support certain key-sizes (128, 192, 256)
  - Supported by most libraries (e.g., OpenSSL, NaCl) and even implemented on modern processors (AES-NI)
- For encrypting larger messages (e.g., for disk encryption) “modes of operation” used
  - E.g: Cipher block-chaining (CBC) mode

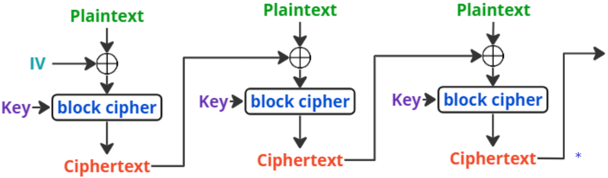


*

*Credit: Wikipedia/Epachamo

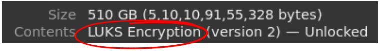
PRFs IRL

- Coming up: theoretical construction, but inefficient for practice
- Practical PRFs: block ciphers like AES, which however only support certain key-sizes (128, 192, 256)
  - Supported by most libraries (e.g., OpenSSL, NaCl) and even implemented on modern processors (AES-NI)
- For encrypting larger messages (e.g., for disk encryption) “modes of operation” used
  - E.g: Cipher block-chaining (CBC) mode



*

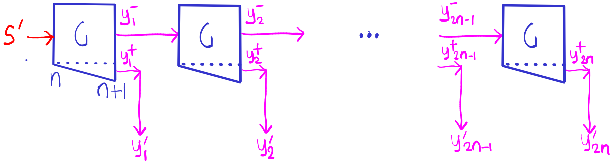
  - My laptop uses LUKS for disk encryption, which uses AES-XTS



Plan for this Lecture

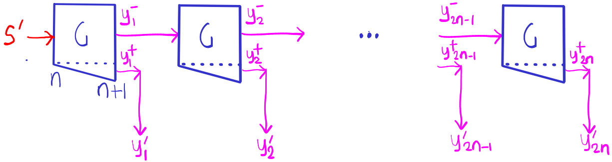
1. Pseudo-Random Function (PRF)
2. Goldreich-Goldwasser-Micali (GGM) Construction

Let's Try to Construct a PRF



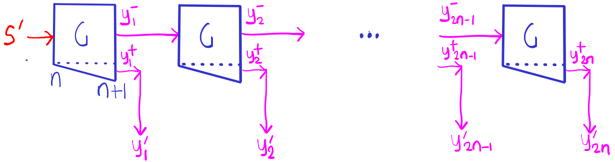
- Recall construction of length-extending PRG from last lecture
- Recall the problem with expanding exponentially:
  - Takes exponential time to access most pseudorandom OTPs

Let's Try to Construct a PRF



- Recall construction of length-extending PRG from last lecture
- Recall the problem with expanding exponentially:
  - Takes exponential time to access most pseudorandom OTPs
- Need "PRG" with
  1. Exponential stretch
  2. Output bits "efficiently" accessible (also called locality)
- How to reconcile the two requirements?
  - Hint: Use length-doubling PRG

Let's Try to Construct a PRF

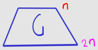


- Recall construction of length-extending PRG from last lecture
- Recall the problem with expanding exponentially:
  - Takes exponential time to access most pseudorandom OTPs
- Need "PRG" with
  1. Exponential stretch
  2. Output bits "efficiently" accessible (also called locality)
- How to reconcile the two requirements?
  - Hint: Use length-doubling PRG
  - Use binary tree instead of chain!

Tree-Based Construction from Length-Doubling PRG G

$n \to n^2$

Construction 2 (GGM PRF $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$)



Tree-Based Construction from Length-Doubling PRG G

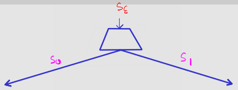
$n \to n^2$

Construction 2 (GGM PRF $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$)

Tree-Based Construction from Length-Doubling PRG G

$n \to n^2$

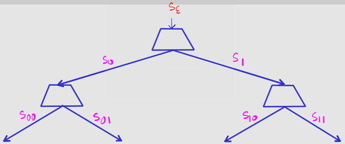
Construction 2 (GGM PRF $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$)



Tree-Based Construction from Length-Doubling PRG G

$n \rightarrow n^2$

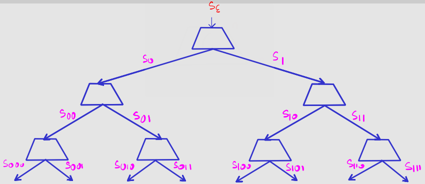
Construction 2 (GGM PRF $\{F_k : \{0,1\}^n \rightarrow \{0,1\}^n\}_{k \in \{0,1\}^n}$)



Tree-Based Construction from Length-Doubling PRG G

$n \to n^2$

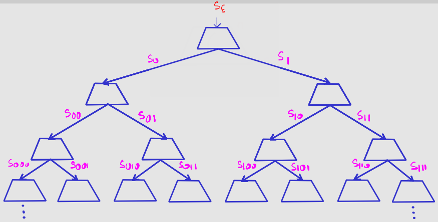
Construction 2 (GGM PRF $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$)



Tree-Based Construction from Length-Doubling PRG G

$n \to n^2$

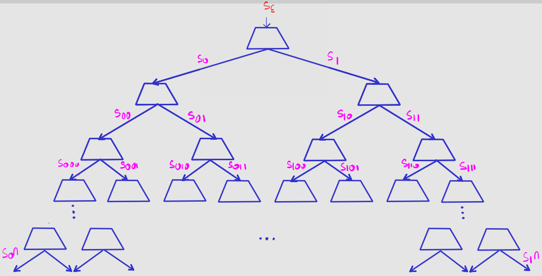
Construction 2 (GGM PRF $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$)



Tree-Based Construction from Length-Doubling PRG G

$n \rightarrow n^2$

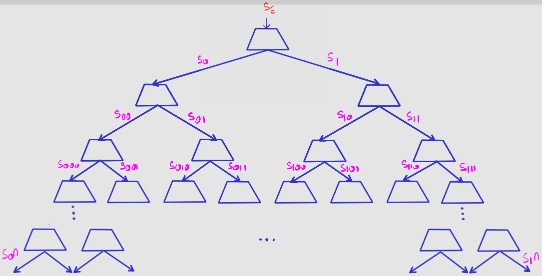
Construction 2 (GGM PRF $\{F_k : \{0,1\}^n \rightarrow \{0,1\}^n\}_{k \in \{0,1\}^n}$)



Tree-Based Construction from Length-Doubling PRG G

$n \to n^2$

Construction 2 (GGM PRF $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$)

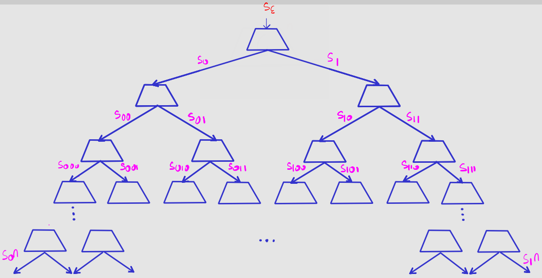


- *Define* $F_k(x) =$

Tree-Based Construction from Length-Doubling PRG G

$n \to n^2$

Construction 2 (GGM PRF $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$)



- *Define* $F_k(x) = s_x$ *with* $s_\varepsilon := k$

Exercise 5

1. *Write down the construction formally.*
2. *What if we use* d*-ary tree instead of binary tree?*

How do We Prove that Construction 2 is a PRF?...

Theorem 1

If G *is a length-doubling PRG, then Construction 2 is a PRF.*

Proof. First attempt: off-the-shelf hybrid argument.

Strategy: replace, breadth-first, pseudorandom by random

How do We Prove that Construction 2 is a PRF?...

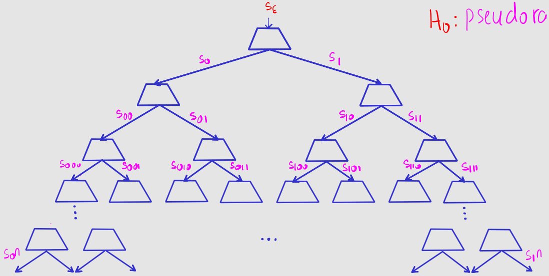
Theorem 1

If G *is a length-doubling PRG, then Construction 2 is a PRF.*

Proof. First attempt: off-the-shelf hybrid argument.

Strategy: replace, breadth-first, pseudorandom by random

H_0: pseudorandom world



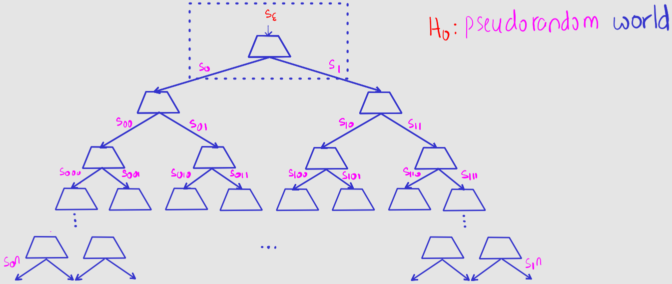
How do We Prove that Construction 2 is a PRF?...

Theorem 1

If G *is a length-doubling PRG, then Construction 2 is a PRF.*

Proof. First attempt: off-the-shelf hybrid argument.

Strategy: replace, breadth-first, pseudorandom by random



How do We Prove that Construction 2 is a PRF?...

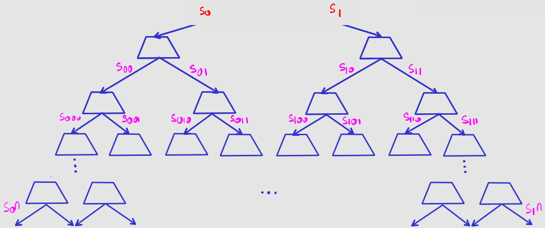
Theorem 1

If G is a length-doubling PRG, then Construction 2 is a PRF.

Proof. First attempt: off-the-shelf hybrid argument.

Strategy: replace, breadth-first, pseudorandom by random

H_1 : hybrid world



How do We Prove that Construction 2 is a PRF?...

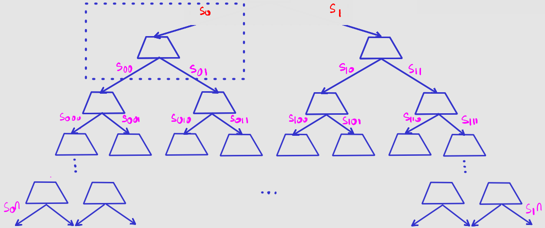
Theorem 1

If G is a length-doubling PRG, then Construction 2 is a PRF.

Proof. First attempt: off-the-shelf hybrid argument.

Strategy: replace, breadth-first, pseudorandom by random

H_1: hybrid world



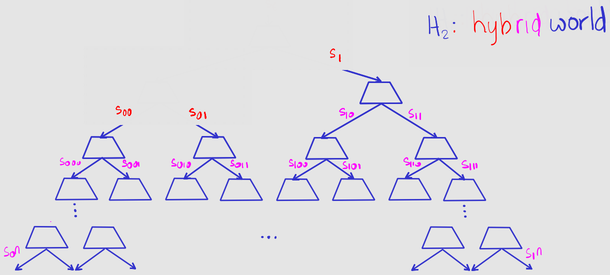
How do We Prove that Construction 2 is a PRF?...

Theorem 1

If G is a length-doubling PRG, then Construction 2 is a PRF.

Proof. First attempt: off-the-shelf hybrid argument.

Strategy: replace, breadth-first, pseudorandom by random



How do We Prove that Construction 2 is a PRF?

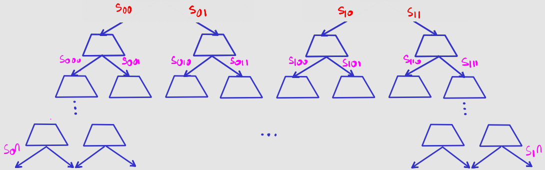
Theorem 1

If G is a length-doubling PRG, then Construction 2 is a PRF.

Proof. First attempt: off-the-shelf hybrid argument.

Strategy: replace, breadth-first, pseudorandom by random

H_3: hybrid world



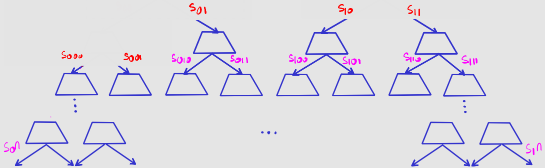
How do We Prove that Construction 2 is a PRF?...

Theorem 1

If G is a length-doubling PRG, then Construction 2 is a PRF.

Proof. First attempt: off-the-shelf hybrid argument.

Strategy: replace, breadth-first, pseudorandom by random



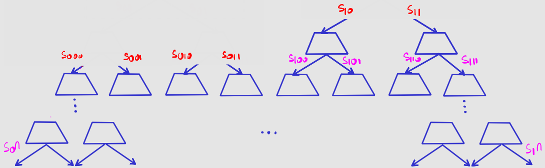
How do We Prove that Construction 2 is a PRF?

Theorem 1

If G is a length-doubling PRG, then Construction 2 is a PRF.

Proof. First attempt: off-the-shelf hybrid argument.

Strategy: replace, breadth-first, pseudorandom by random



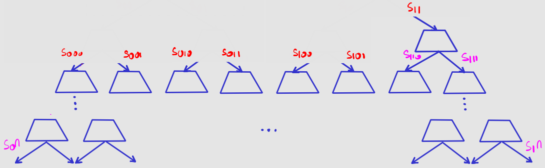
How do We Prove that Construction 2 is a PRF?...

Theorem 1

If G is a length-doubling PRG, then Construction 2 is a PRF.

Proof. First attempt: off-the-shelf hybrid argument.

Strategy: replace, breadth-first, pseudorandom by random

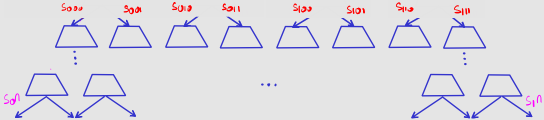


Theorem 1

If G is a length-doubling PRG, then Construction 2 is a PRF.

Proof. First attempt: off-the-shelf hybrid argument.

Strategy: replace, breadth-first, pseudorandom by random



How do We Prove that Construction 2 is a PRF?...

Theorem 1

If G *is a length-doubling PRG, then Construction 2 is a PRF.*

Proof. First attempt: off-the-shelf hybrid argument.

Strategy: replace, breadth-first, pseudorandom by random

How do We Prove that Construction 2 is a PRF?

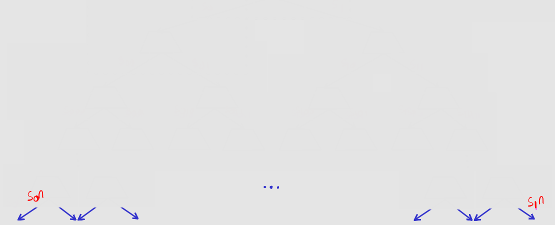
Theorem 1

If G is a length-doubling PRG, then Construction 2 is a PRF.

Proof. First attempt: off-the-shelf hybrid argument.

Strategy: replace, breadth-first, pseudorandom by random

H_{2^n+1}: random world



How do We Prove that Construction 2 is a PRF?

Theorem 1

If G is a length-doubling PRG, then Construction 2 is a PRF.

Proof. First attempt: off-the-shelf hybrid argument.

Strategy: replace, breadth-first, pseudorandom by random

Hybrid argument: if D can distinguish H_0 from $H_{2^{n+1}}$

$\Downarrow$

$\exists i \in [0, 2^{n+1}-1]$ such that D distinguishes H_i from H_{i+1}

How do We Prove that Construction 2 is a PRF?

Theorem 1

If G is a length-doubling PRG, then Construction 2 is a PRF.

Proof. First attempt: off-the-shelf hybrid argument.

Strategy: replace, breadth-first, pseudorandom by random

Hybrid argument: if D can distinguish H_0 from $H_{2^{n+1}}$ w/ pr. δ

$\Downarrow$

$\exists i \in [0, 2^{n+1}-1]$ such that D distinguishes H_i from H_{i+1}

w/ pr. $\delta/2^{n+1}$

Theorem 1

If G is a length-doubling PRG, then Construction 2 is a PRF.

Proof. First attempt: off-the-shelf hybrid argument.

Strategy: replace, breadth-first, pseudorandom by random

Hybrid argument: if D can distinguish H_0 from $H_{2^{n+1}}$ w/ pr. δ

$\Downarrow$

$\exists i \in [0, 2^{n+1}-1]$ such that D distinguishes H_i from H_{i+1}

w/ pr. $\delta / 2^{n+1}$

Problem: exponential number of hybrids ×

How do We Prove that Construction 2 is a PRF?...

Theorem 1

If G is a length-doubling PRG, then Construction 2 is a PRF.

Proof. Idea: hybrid argument with on-the-fly/lazy sampling!

- Switching every single value to random is overkill
- Only switch values required to answer distinguisher's queries

How do We Prove that Construction 2 is a PRF?...

Theorem 1

If G *is a length-doubling PRG, then Construction 2 is a PRF.*

Proof. Idea: hybrid argument with on-the-fly/lazy sampling!

- Switching every single value to random is overkill
- Only switch values required to answer distinguisher's queries
  - Distinguisher makes at most Q queries $\Rightarrow$ number of switches per level of the tree is at most Q

How do We Prove that Construction 2 is a PRF?...

Theorem 1

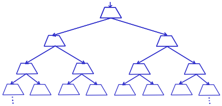
If G is a length-doubling PRG, then Construction 2 is a PRF.

Proof. Idea: hybrid argument with on-the-fly/lazy sampling!

- Switching every single value to random is overkill
- Only switch values required to answer distinguisher's queries
  - Distinguisher makes at most Q queries $\Rightarrow$ number of switches per level of the tree is at most Q
- The hybrid worlds:
  - Each level $i \in [1, n]$ has *at most* Q hybrid worlds
  - Hybrid worlds at level $i \in [1, n]$ (think of $2^i \gg Q$):
    - $H_{i,0}, \cdots, H_{i,Q}$, where in $H_{i,q}$ the values used to answer first q queries are switched from pseudorandom to random

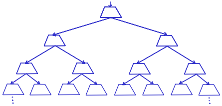
□

To Recap



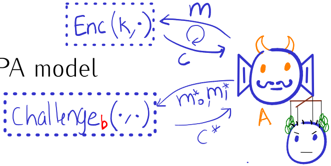
- Defined and constructed PRFs
  - GGM tree-based construction from length-doubling PRGs
  - Another application of hybrid argument 

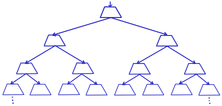
To Recap



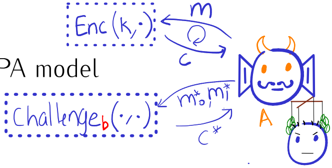
- Defined and constructed PRFs
  - GGM tree-based construction from length-doubling PRGs
  - Another application of hybrid argument 

- Constructed a stateless SKE from PRF
  - It is actually secret in the stronger CPA model

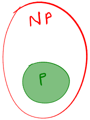




- Defined and constructed PRFs
    - GGM tree-based construction from length-doubling PRGs
    - Another application of hybrid argument

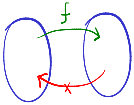


- Constructed a stateless SKE from PRF
    - It is actually secret in the stronger CPA model
- Other applications of PRFs
    - Authentication (coming up: Lecture 7)
    - Natural proofs: barrier to resolving the $\mathsf{P} \overset{?}{=} \mathsf{NP}$ question



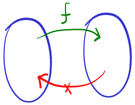
Next Lecture

- Hardness vs. pseudorandomness
- One-way function and one-way permutation
- Hardcore predicates



Next Lecture

- Hardness vs. pseudorandomness
- One-way function and one-way permutation
- Hardcore predicates



More Questions?

Scott Aaronson.
Is P versus NP formally independent?
Bull. EATCS, 81:109–136, 2003.

Timothy Y Chow.
What is… a natural proof?
Notices of the AMS, 58(11):1586–1587, 2011.

Oded Goldreich, Shafi Goldwasser, and Silvio Micali.
How to construct random functions (extended abstract).
In *25th FOCS*, pages 464–479. IEEE Computer Society Press, October 1984.

Oded Goldreich.
The Foundations of Cryptography - Volume 1: Basic Techniques.
Cambridge University Press, 2001.

Jonathan Katz and Yehuda Lindell.
Introduction to Modern Cryptography (3rd ed.).
Chapman and Hall/CRC, 2014.