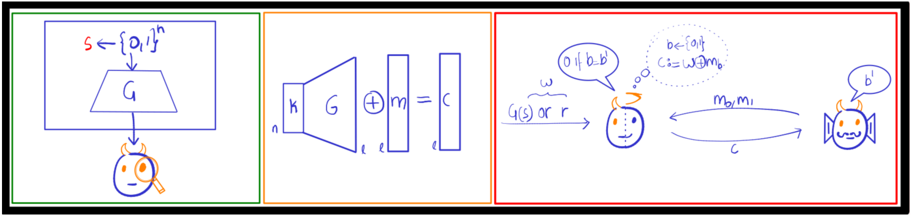


CS409m: Introduction to Cryptography

Lecture 05 (13/Aug/25)

Instructor: Chethan Kamath

Annonuncements

- Grading structure adjusted (as discussed in Lecture 04)

Weightage	Towards
35%	End-sem
25%	Mid-sem
20%	Two (out of three) quizzes
15%	Four lab exercises
5%	Class participation, pop-quizzes

- Lab Exercise 1 (graded)
  - ~~Deadline for submitting flag on CTFd server: 23:59, 11/Aug/25~~
  - Deadline for submitting report on Moodle: 23:59, 13/Aug/25
- Assignment 2 (ungraded) will be uploaded today (13/Aug)

- Reminder: Quiz 1 on 22/Aug, 08:25-09:25 in CC103!

Recall from Previous Lecture...

- Task: secure communication of *long messages* with shared keys
- Problem: $\mathcal{K} \geq \mathcal{M}$ for perfect secrecy against eavesdroppers

Recall from Previous Lecture...

- Task: secure communication of *long messages* with shared keys
- Problem: $\mathcal{K} \geq \mathcal{M}$ for perfect secrecy against eavesdroppers
- Relaxed threat model: computational secrecy against eavesdroppers

Recall from Previous Lecture...

- Task: secure communication of *long messages* with shared keys
- Problem: $\mathcal{K} \geq \mathcal{M}$ for perfect secrecy against eavesdroppers
- Relaxed threat model: computational secrecy against eavesdroppers

Attack Model: Eavesdropping

1. Eve is PPT
2. Knows description of Π (Kerchhoff's principle)
3. Shared key is hidden from Eve
4. Can eavesdrop and learn ciphertext

Recall from Previous Lecture...

- Task: secure communication of *long messages* with shared keys
- Problem: $\mathcal{K} \geq \mathcal{M}$ for perfect secrecy against eavesdroppers
- Relaxed threat model: computational secrecy against eavesdroppers

Attack Model: Eavesdropping

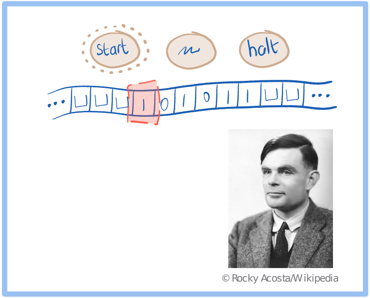
1. Eve is PPT
2. Knows description of Π (Kerchhoff's principle)
3. Shared key is hidden from Eve
4. Can eavesdrop and learn ciphertext

Break Model: Secrecy, w.h.p.

1. Eve breaks with negligible probability
   - Two worlds definition

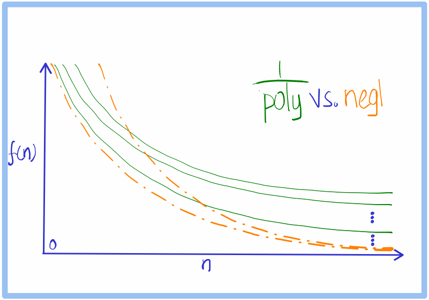
Recall from Previous Lecture...

Probabilistic Poly. Time (PPT)



- "Efficient computation"
- Polynomial-time on probabilistic Turing Machine

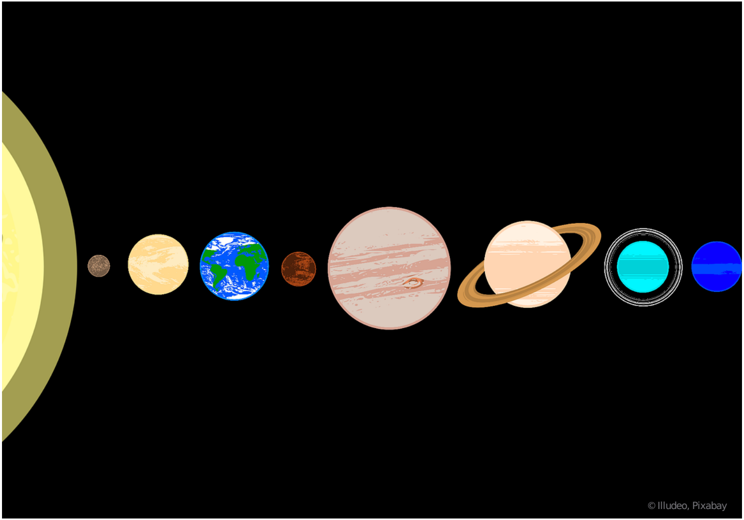
Negligible Function



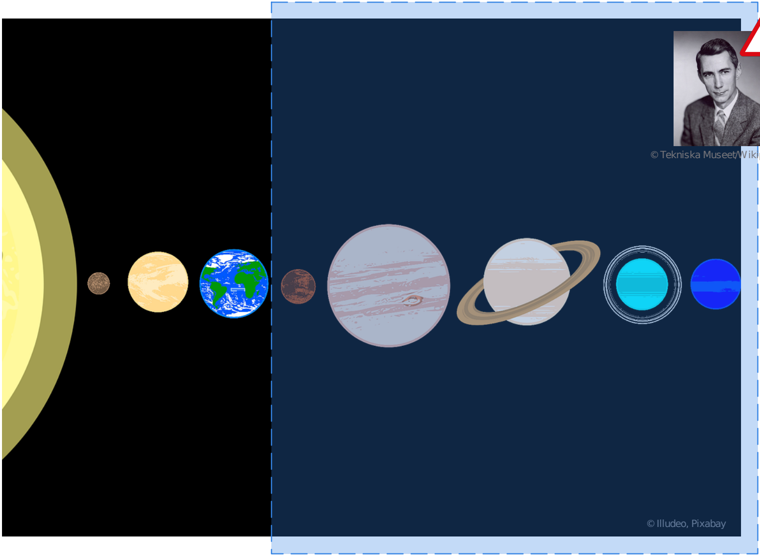
- "Low probability" event
- Decays faster than *any* inverse polynomial function

Recall from Previous Lecture...

- Why PPT and negligible? Goldilocks zone!



- Why PPT and negligible? Goldilocks zone!

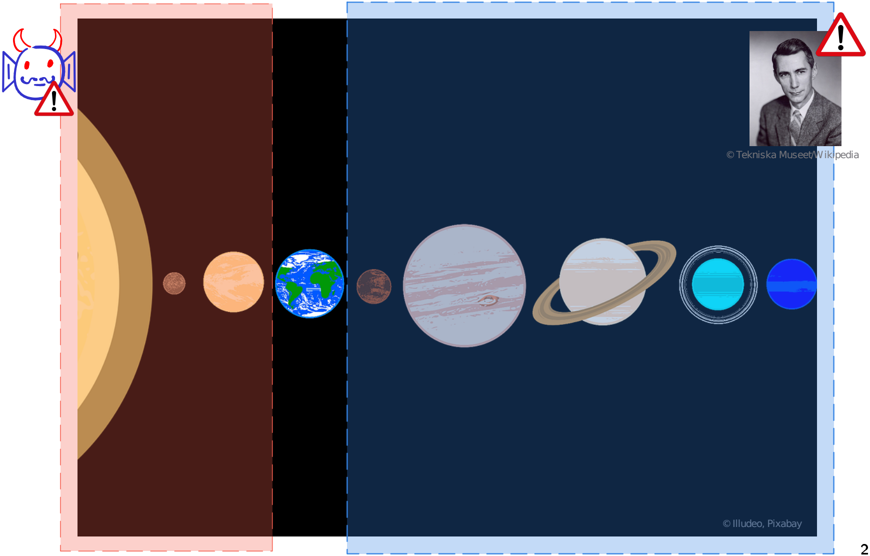


© Illudeo, Pixabay

© Tekniska Museet/Wikipedia

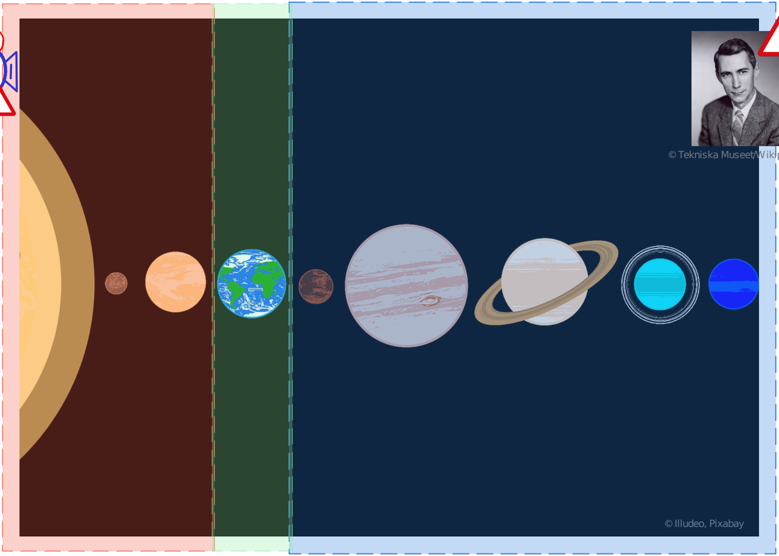
Recall from Previous Lecture...

- Why PPT and negligible? Goldilocks zone!



Recall from Previous Lecture...

- Why PPT and negligible? Goldilocks zone!

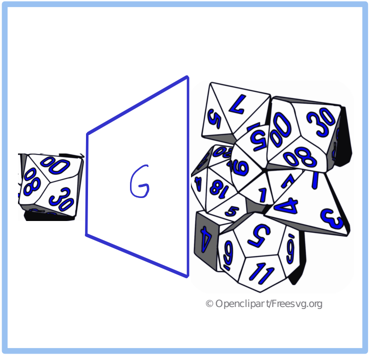


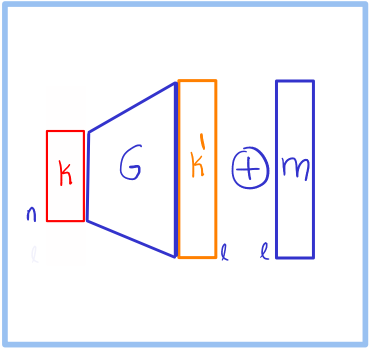
Plan for Today's Lecture...

Goal: construct SKE computationally-secret against eavesdroppers

Goal: construct SKE computationally-secret against eavesdroppers

NEW Pseudorandom Generator (PRG) NEW Computational One-Time Pad

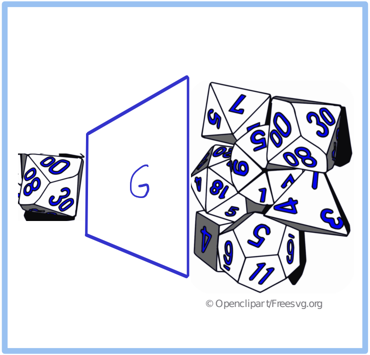


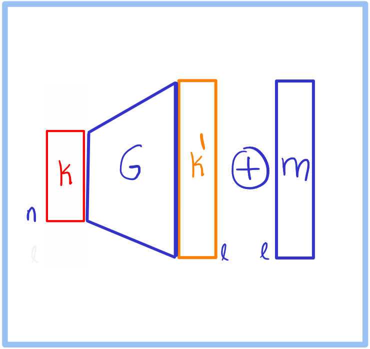


Plan for Today's Lecture...

Goal: construct SKE computationally-secret against eavesdroppers

NEW Pseudorandom Generator (PRG) NEW Computational One-Time Pad



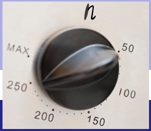


New tool: proof by reduction

Plan for Today's Lecture...

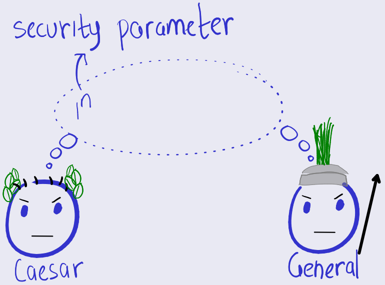
1. Recall: Computational Secrecy Against Eavesdroppers
2. Pseudo-Random Generator (PRG)
3. Computational One-Time Pad

Recall: SKE with Security Parameter

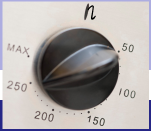


Definition 1 (Shared/Symmetric-Key Encryption (SKE))

An SKE Π is a triple of efficient algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with the following syntax:

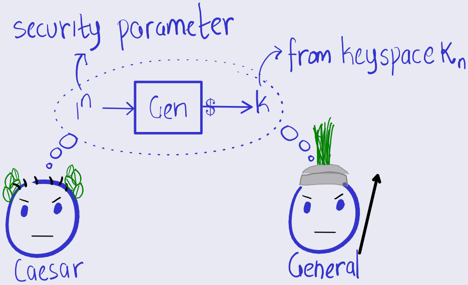


Recall: SKE with Security Parameter

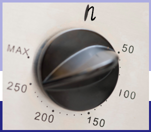


Definition 1 (Shared/Symmetric-Key Encryption (SKE))

An SKE Π is a triple of efficient algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with the following syntax:

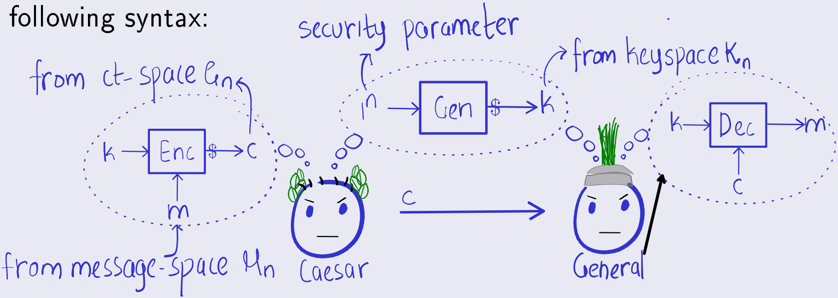


Recall: SKE with Security Parameter

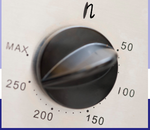


Definition 1 (Shared/Symmetric-Key Encryption (SKE))

An SKE Π is a triple of efficient algorithms (Gen, Enc, Dec) with the following syntax:

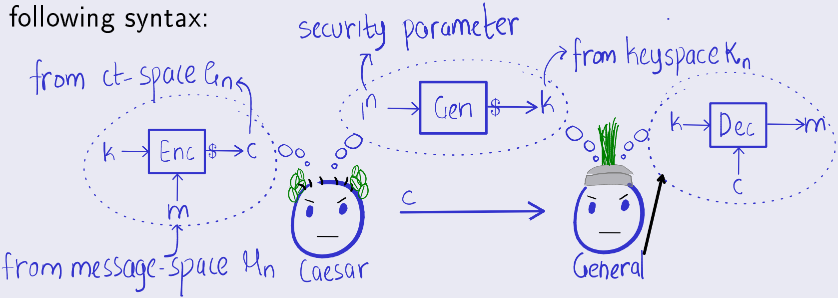


Recall: SKE with Security Parameter



Definition 1 (Shared/Symmetric-Key Encryption (SKE))

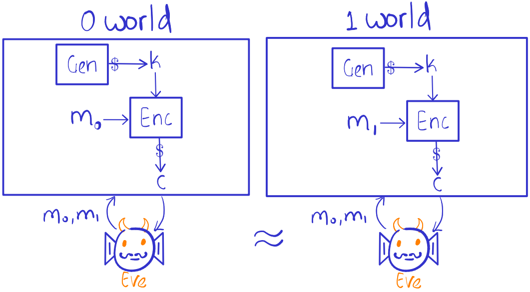
An SKE Π is a triple of efficient algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with the following syntax:



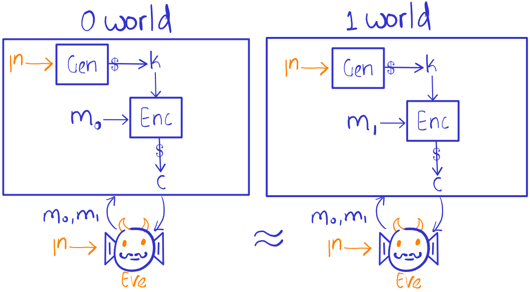
- Correctness of decryption: for every $n \in \mathbb{N}$, message $m \in \mathcal{M}_n$,

$$\Pr_{k \leftarrow \mathsf{Gen}(1^n), c \leftarrow \mathsf{Enc}(k,m)}[\mathsf{Dec}(k, c) = m] = 1$$

Recall: Computational Secrecy



Recall: Computational Secrecy

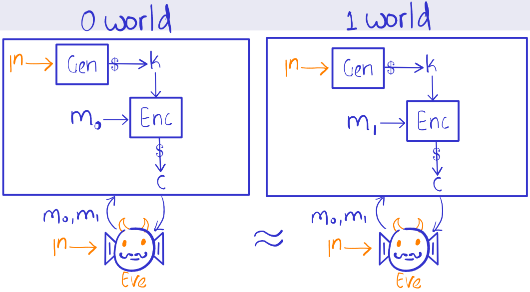


Definition 2 (Two-Worlds Definition)

An SKE $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is computationally-secret against eavesdroppers if for *every* (stateful) PPT Eve

$$\delta(n) := \left| \Pr_{\substack{(m_0,m_1)\leftarrow \mathsf{Eve}(1^n)\\ k\leftarrow \mathsf{Gen}(1^n)\\ c\leftarrow \mathsf{Enc}(k,m_0)}}[\mathsf{Eve}(c)=0] - \Pr_{\substack{(m_0,m_1)\leftarrow \mathsf{Eve}(1^n)\\ k\leftarrow \mathsf{Gen}(1^n)\\ c\leftarrow \mathsf{Enc}(k,m_1)}}[\mathsf{Eve}(c)=0] \right|$$

is negligible.



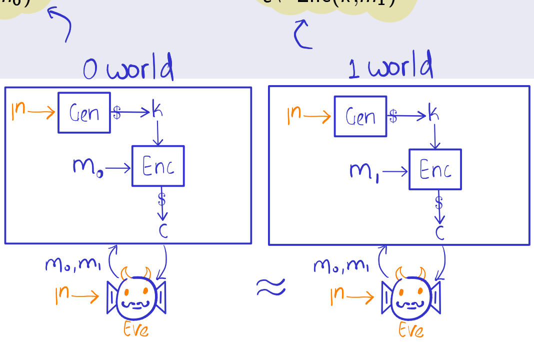
Recall: Computational Secrecy

Definition 2 (Two-Worlds Definition)

An SKE $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is computationally-secret against eavesdroppers if for *every* (stateful) PPT Eve

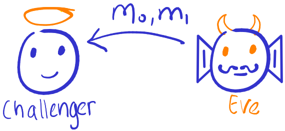
$$\delta(n) := \left| \Pr_{\substack{(m_0,m_1)\leftarrow \mathsf{Eve}(1^n)\\ k\leftarrow \mathsf{Gen}(1^n)\\ c\leftarrow \mathsf{Enc}(k,m_0)}}[\mathsf{Eve}(c)=0] - \Pr_{\substack{(m_0,m_1)\leftarrow \mathsf{Eve}(1^n)\\ k\leftarrow \mathsf{Gen}(1^n)\\ c\leftarrow \mathsf{Enc}(k,m_1)}}[\mathsf{Eve}(c)=0] \right|$$

is negligible.

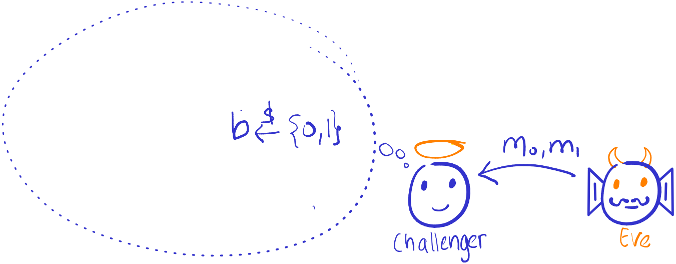


Alternatively: Adversarial Indistinguishability

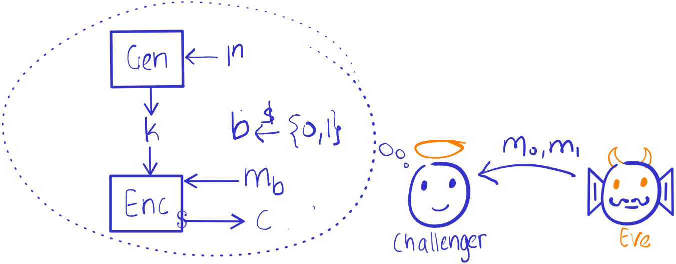
Alternatively: Adversarial Indistinguishability



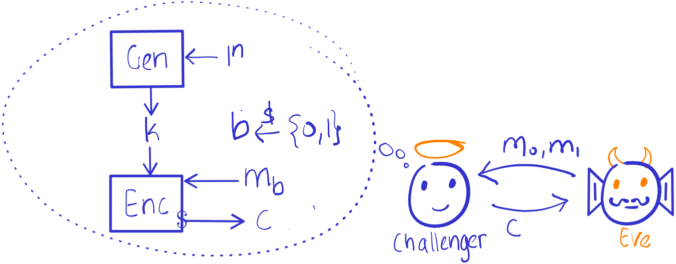
Alternatively: Adversarial Indistinguishability



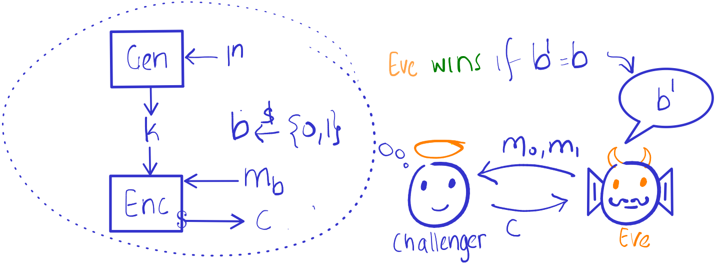
Alternatively: Adversarial Indistinguishability



Alternatively: Adversarial Indistinguishability



Alternatively: Adversarial Indistinguishability



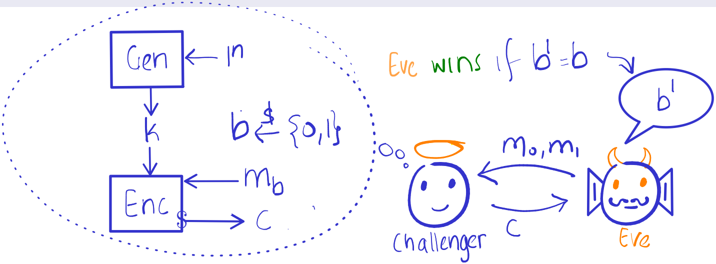
Alternatively: Adversarial Indistinguishability

Definition 3 (Adversarial Indistinguishability)

An SKE $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is computationally-secret against eavesdropper if for *every* (stateful) PPT Eve

$$\delta(n) := \Pr_{\substack{(m_0,m_1)\leftarrow \mathsf{Eve}(1^n)\\ k\leftarrow \mathsf{Gen}(1^n)\\ b\leftarrow\{0,1\}\\ c\leftarrow \mathsf{Enc}(k,m_b)}}[\mathsf{Eve}(c) = b] - \frac{1}{2}$$

is negligible.



The Two Definitions are Equivalent!

Claim 1 (Other direction exercise!)

Definition 3 implies Definition 2.

Proof (using basic probability from Lecture 02).

□

The Two Definitions are Equivalent!

Claim 1 (Other direction exercise!)

Definition 3 implies Definition 2.

Proof (using basic probability from Lecture 02).

Definition 3 implies for every PPT Eve the following is negligible

$$\Pr_{\substack{m_0, m_1 \leftarrow Eve(1^n) \\ k \leftarrow Gen(1^n),\ b \leftarrow \{0,1\} \\ c \leftarrow Enc(k, m_b)}} \left[Eve(c) = b \right] - \frac{1}{2}$$

The Two Definitions are Equivalent!

Claim 1 (Other direction exercise!)

Definition 3 implies Definition 2.

Proof (using basic probability from Lecture 02).

Definition 3 implies for every PPT Eve the following is negligible

$$\Pr\left[\mathsf{Eve}(c)=0, b=0 \text{ or } \mathsf{Eve}(c)=1, b=1\right] - 1/2$$

$m_0, m_1 \leftarrow \mathsf{Eve}(1^n)$
$k \leftarrow \mathsf{Gen}(1^n), b \leftarrow \{0,1\}$
$c \leftarrow \mathsf{Enc}(k, m_b)$

The Two Definitions are Equivalent!

Claim 1 (Other direction exercise!)

Definition 3 implies Definition 2.

Proof (using basic probability from Lecture 02).

Definition 3 implies for every PPT Eve the following is negligible

$$\Pr_{\substack{m_0, m_1 \leftarrow Eve(1^n) \\ k \leftarrow Gen(1^n),\ b \leftarrow \{0,1\} \\ c \leftarrow Enc(k, m_b)}}[Eve(c) = 0, b = 0] + \Pr_{\substack{m_0, m_1 \leftarrow Eve(1^n) \\ k \leftarrow Gen(1^n),\ b \leftarrow \{0,1\} \\ c \leftarrow Enc(k, m_b)}}[Eve(c) = 1, b = 1] - 1/2$$

The Two Definitions are Equivalent!

Claim 1 (Other direction exercise!)

Definition 3 implies Definition 2.

Proof (using basic probability from Lecture 02).

Definition 3 implies for every PPT Eve the following is negligible

$$\Pr_{\substack{m_0,m_1 \leftarrow Eve(1^n) \\ k \leftarrow Gen(1^n),\, b \leftarrow \{0,1\} \\ c \leftarrow Enc(k,m_b)}}[Eve(c)=0, b=0] + \Pr_{\substack{m_0,m_1 \leftarrow Eve(1^n) \\ k \leftarrow Gen(1^n),\, b \leftarrow \{0,1\} \\ c \leftarrow Enc(k,m_b)}}[Eve(c)=1, b=1] - \frac{1}{2}$$

$$\Downarrow$$

$$\frac{1}{2}\left[\Pr_{\substack{m_0,m_1 \leftarrow Eve(1^n) \\ k \leftarrow Gen(1^n) \\ c \leftarrow Enc(k,m_0)}}[Eve(c)=0] + \Pr_{\substack{m_0,m_1 \leftarrow Eve(1^n) \\ k \leftarrow Gen(1^n) \\ c \leftarrow Enc(k,m_1)}}[Eve(c)=1]\right] - \frac{1}{2} \text{ is negl.}$$

The Two Definitions are Equivalent!

Claim 1 (Other direction exercise!)

Definition 3 implies Definition 2.

Proof (using basic probability from Lecture 02).

Definition 3 implies for every PPT Eve the following is negligible

$$\Pr_{\substack{m_0,m_1 \leftarrow Eve(1^n) \\ k \leftarrow Gen(1^n),\, b \leftarrow \{0,1\} \\ c \leftarrow Enc(k, m_b)}}[Eve(c)=0, b=0] + \Pr_{\substack{m_0,m_1 \leftarrow Eve(1^n) \\ k \leftarrow Gen(1^n),\, b \leftarrow \{0,1\} \\ c \leftarrow Enc(k, m_b)}}[Eve(c)=1, b=1] - \frac{1}{2}$$

$$\Downarrow$$

$$\frac{1}{2}\left[\Pr_{\substack{m_0,m_1 \leftarrow Eve(1^n) \\ k \leftarrow Gen(1^n) \\ c \leftarrow Enc(k, m_0)}}[Eve(c)=0] + 1 - \Pr_{\substack{m_0,m_1 \leftarrow Eve(1^n) \\ k \leftarrow Gen(1^n) \\ c \leftarrow Enc(k, m_1)}}[Eve(c)=0]\right] - \frac{1}{2} \text{ is negl.}$$

The Two Definitions are Equivalent!

Claim 1 (Other direction exercise!)

Definition 3 implies Definition 2.

Proof (using basic probability from Lecture 02).

Definition 3 implies for every PPT Eve the following is negligible

$$\Pr_{\substack{m_0,m_1 \leftarrow Eve(1^n) \\ k \leftarrow Gen(1^n),\ b \leftarrow \{0,1\} \\ c \leftarrow Enc(k,m_b)}}\left[Eve(c)=0, b=0\right] + \Pr_{\substack{m_0,m_1 \leftarrow Eve(1^n) \\ k \leftarrow Gen(1^n),\ b \leftarrow \{0,1\} \\ c \leftarrow Enc(k,m_b)}}\left[Eve(c)=1, b=1\right] - \frac{1}{2}$$

$$\Downarrow$$

$$\left| \Pr_{\substack{m_0,m_1 \leftarrow Eve(1^n) \\ k \leftarrow Gen(1^n) \\ c \leftarrow Enc(k,m_0)}}\left[Eve(c)=0\right] - \Pr_{\substack{m_0,m_1 \leftarrow Eve(1^n) \\ k \leftarrow Gen(1^n) \\ c \leftarrow Enc(k,m_1)}}\left[Eve(c)=0\right] \right| \text{ is negl.}$$

The Two Definitions are Equivalent!

Claim 1 (Other direction exercise!)

Definition 3 implies Definition 2.

Proof (using basic probability from Lecture 02).

Definition 3 implies for every PPT Eve the following is negligible

$$\Pr_{\substack{m_0,m_1 \leftarrow Eve(1^n)\\ k \leftarrow Gen(1^n),\, b \leftarrow \{0,1\}\\ c \leftarrow Enc(k,m_b)}}[Eve(c)=0, b=0] + \Pr_{\substack{m_0,m_1 \leftarrow Eve(1^n)\\ k \leftarrow Gen(1^n),\, b \leftarrow \{0,1\}\\ c \leftarrow Enc(k,m_b)}}[Eve(c)=1, b=1] - \frac{1}{2}$$

$$\Downarrow$$

$$\left[\Pr_{\substack{m_0,m_1 \leftarrow Eve(1^n)\\ k \leftarrow Gen(1^n)\\ c \leftarrow Enc(k,m_0)}}[Eve(c)=0] - \Pr_{\substack{m_0,m_1 \leftarrow Eve(1^n)\\ k \leftarrow Gen(1^n)\\ c \leftarrow Enc(k,m_1)}}[Eve(c)=0]\right] \text{ is negl.}$$

? Where did I cheat?

More Generally: Computational Indistinguishability

- Ciphertext distributions $\mapsto$ *any* two distributions

More Generally: Computational Indistinguishability

- Ciphertext distributions $\mapsto$ *any* two distributions

Definition 4 (computational indistinguishability)

Two distributions $\mathrm{X}_0 := (\mathrm{X}_{0,n})_{n \in \mathbb{N}}$ and $\mathrm{X}_1 := (\mathrm{X}_{1,n})_{n \in \mathbb{N}}$ are *computationally indistinguishable* if for every PPT *distinguisher* D,

$$\delta(n) := \left| \Pr_{x \leftarrow \mathrm{X}_{0,n}}[\mathsf{D}(x) = 0] - \Pr_{x \leftarrow \mathrm{X}_{1,n}}[\mathsf{D}(x) = 0] \right|$$

is negligible.

More Generally: Computational Indistinguishability

- Ciphertext distributions $\mapsto$ *any* two distributions

Definition 4 (computational indistinguishability)

Two distributions $\mathrm{X}_0 := (\mathrm{X}_{0,n})_{n\in\mathbb{N}}$ and $\mathrm{X}_1 := (\mathrm{X}_{1,n})_{n\in\mathbb{N}}$ are *computationally indistinguishable* if for every PPT *distinguisher* D,

$$\delta(n) := \left| \Pr_{x\leftarrow \mathrm{X}_{0,n}}[\mathsf{D}(x)=0] - \Pr_{x\leftarrow \mathrm{X}_{1,n}}[\mathsf{D}(x)=0] \right|$$

is negligible.

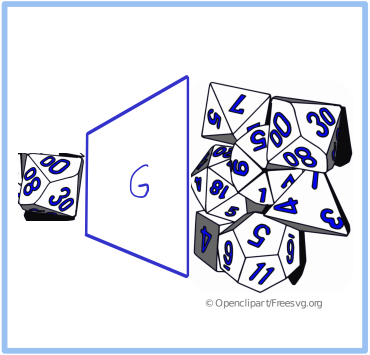
- Computational secrecy against eavesdroppers: the ciphertext distribution of m_0 and m_1 are computationally indistinguishable

Exercise 1

Formally write down the two distributions

Goal: construct SKE computationally-secret against eavesdroppers

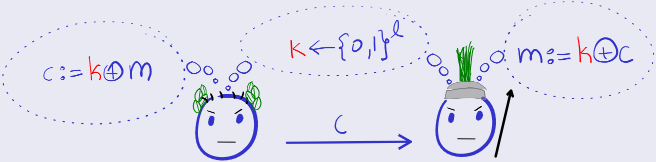
NEW Pseudorandom Generator (PRG) NEW Computational One-Time Pad



New tool: proof by reduction

Recall One-Time Pad (Vernam's Cipher)

Construction 1 (Message space $\{0,1\}^\ell$)

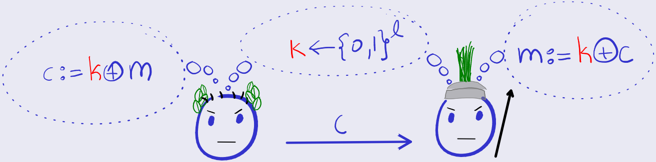


Pseudocode 1 (Message space $\{0,1\}^\ell$)

- Key generation $\mathsf{Gen}(1^\ell)$: output $k \leftarrow \{0,1\}^\ell$
- Encryption $\mathsf{Enc}(k, m)$: output $c := k \oplus m$
- Decryption $\mathsf{Dec}(k, c)$: output $m := k \oplus c$

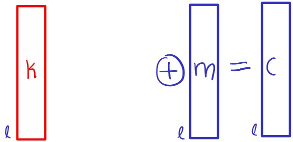
Recall One-Time Pad (Vernam's Cipher)

Construction 1 (Message space $\{0,1\}^\ell$)



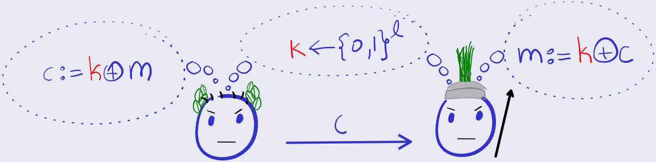
Pseudocode 1 (Message space $\{0,1\}^\ell$)

- Key generation $\mathsf{Gen}(1^\ell)$: output $k \leftarrow \{0,1\}^\ell$
- Encryption $\mathsf{Enc}(k,m)$: output $c := k \oplus m$
- Decryption $\mathsf{Dec}(k,c)$: output $m := k \oplus c$



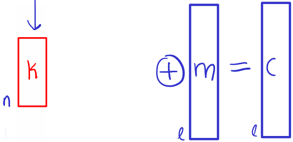
Recall One-Time Pad (Vernam's Cipher)

Construction 1 (Message space $\{0,1\}^\ell$)

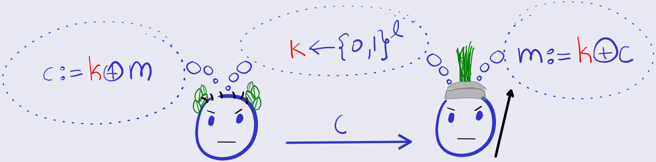


Pseudocode 1 (Message space $\{0,1\}^\ell$)

- Key generation $\mathsf{Gen}(1^\ell)$: output $k \leftarrow \{0,1\}^\ell$
- Encryption $\mathsf{Enc}(k,m)$: output $c := k \oplus m$
- Decryption $\mathsf{Dec}(k,c)$: output $m := k \oplus c$

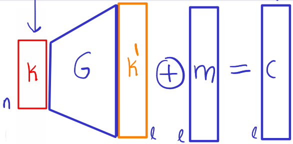


Construction 1 (Message space $\{0,1\}^\ell$)

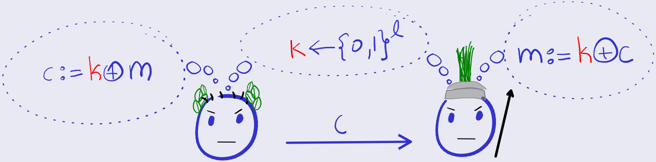


Pseudocode 1 (Message space $\{0,1\}^\ell$)

- Key generation $\mathsf{Gen}(1^\ell)$: output $k \leftarrow \{0,1\}^\ell$
- Encryption $\mathsf{Enc}(k, m)$: output $c := k \oplus m$
- Decryption $\mathsf{Dec}(k, c)$: output $m := k \oplus c$



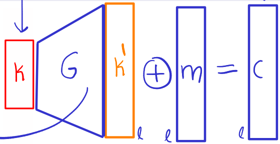
Construction 1 (Message space $\{0,1\}^{\ell}$)



Pseudocode 1 (Message space $\{0,1\}^{\ell}$)

- Key generation $\mathsf{Gen}(1^{\ell})$: output $k \leftarrow \{0,1\}^{n}$, $n<\ell$
- Encryption $\mathsf{Enc}(k,m)$: output $c := k' \oplus m$
- Decryption $\mathsf{Dec}(k,c)$: output $m := k \oplus c$

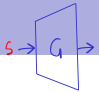
Requirement: G expands $k \to k'$ such that k' "seems random" to ☺



Pseudo-Random Generator (PRG)...

Intuitive definition: expanding function whose output (on uniformly random input) "seems random" to PPT *distinguishers*.

Pseudo-Random Generator (PRG)...

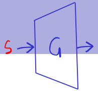


Intuitive definition: expanding function whose output (on uniformly random input) "seems random" to PPT *distinguishers*.

Definition 5 (Two worlds definition)

Let G be an efficient deterministic algorithm that for any $n \in \mathbb{N}$ and input $s \in \{0,1\}^n$, outputs a string of length $\ell(n) > n$.

Pseudo-Random Generator (PRG)...



Intuitive definition: expanding function whose output (on uniformly random input) "seems random" to PPT *distinguishers*.

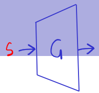
Definition 5 (Two worlds definition)

Let G be an efficient deterministic algorithm that for any $n \in \mathbb{N}$ and input $s \in \{0,1\}^n$, outputs a string of length $\ell(n) > n$.
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.

Pseudo-Random Generator (PRG)...



Intuitive definition: expanding function whose output (on uniformly random input) "seems random" to PPT *distinguishers*.

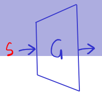
Definition 5 (Two worlds definition)

Let G be an efficient deterministic algorithm that for any $n \in \mathbb{N}$ and input $s \in \{0,1\}^n$, outputs a string of length $\ell(n) > n$.
G is PRG if for every PPT *distinguisher* D

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

pseudorandom world   random world

is negligible.



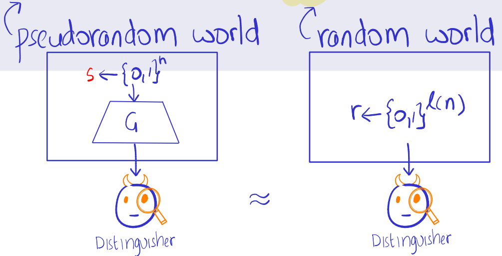
Intuitive definition: expanding function whose output (on uniformly random input) "seems random" to PPT *distinguishers*.

Definition 5 (Two worlds definition)

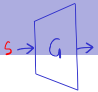
Let G be an efficient deterministic algorithm that for any $n \in \mathbb{N}$ and input $s \in \{0,1\}^n$, outputs a string of length $\ell(n) > n$.
G is PRG if for every PPT *distinguisher* D

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

is negligible.



Pseudo-Random Generator (PRG)



Intuitive definition: expanding function whose output (on uniformly random input) "seems random" to PPT *distinguishers*.

Definition 5 (Two worlds definition)

Let G be an efficient deterministic algorithm that for any $n \in \mathbb{N}$ and input $s \in \{0,1\}^n$, outputs a string of length $\ell(n) > n$.
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.

Exercise 2

1. Write up "adversarial indistinguishability" definition of PRG
2. Show that the two definitions are equivalent

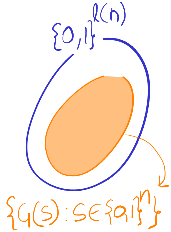
Definition 5 (Two worlds definition)

Let G be an efficient deterministic algorithm that for any $n \in \mathbb{N}$ and input $s \in \{0,1\}^n$, outputs a string of length $\ell(n) > n$.
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.

- Let's check your understanding of Definition 5
  - How can an *unbounded* distinguisher break PRG?



Pseudo-Random Generator...

Definition 5 (Two worlds definition)

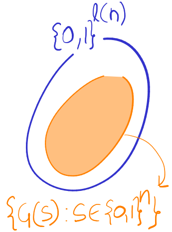
Let G be an efficient deterministic algorithm that for any $n \in \mathbb{N}$ and input $s \in \{0,1\}^n$, outputs a string of length $\ell(n) > n$.
G is PRG if for every PPT *distinguisher* D

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

is negligible.

Let's check your understanding of Definition 5

- How can an *unbounded* distinguisher break PRG?
- Is G a PRG or not? Below G_1 and G_2 are PRGs
  1. $G(s) := G_1(s) \| 0$



Definition 5 (Two worlds definition)

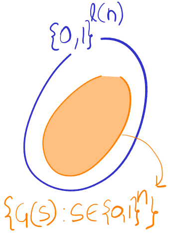
Let G be an efficient deterministic algorithm that for any $n \in \mathbb{N}$ and input $s \in \{0,1\}^n$, outputs a string of length $\ell(n) > n$.
G is PRG if for every PPT *distinguisher* D

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

is negligible.

Let's check your understanding of Definition 5

- How can an *unbounded* distinguisher break PRG?
- Is G a PRG or not? Below G_1 and G_2 are PRGs
  1. $G(s) := G_1(s)\|0$
  2. $G(s) := s\|s_1 \oplus \ldots \oplus s_n$



Pseudo-Random Generator...

Definition 5 (Two worlds definition)

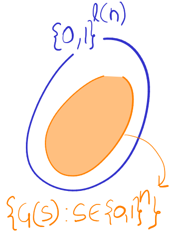
Let G be an efficient deterministic algorithm that for any $n \in \mathbb{N}$ and input $s \in \{0,1\}^n$, outputs a string of length $\ell(n) > n$.
G is PRG if for every PPT *distinguisher* D

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

is negligible.

Let's check your understanding of Definition 5

- How can an *unbounded* distinguisher break PRG?
- Is G a PRG or not? Below G_1 and G_2 are PRGs
  1. $G(s) := G_1(s) \| 0$
  2. $G(s) := s \| s_1 \oplus \ldots \oplus s_n$
  3. $G(s_1 \| s_2) := G_1(s_1) \| G_2(s_2)$



Pseudo-Random Generator...

Definition 5 (Two worlds definition)

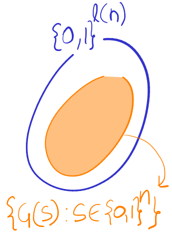
Let G be an efficient deterministic algorithm that for any $n \in \mathbb{N}$ and input $s \in \{0,1\}^n$, outputs a string of length $\ell(n) > n$.
G is PRG if for every PPT *distinguisher* D

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

is negligible.

Let's check your understanding of Definition 5

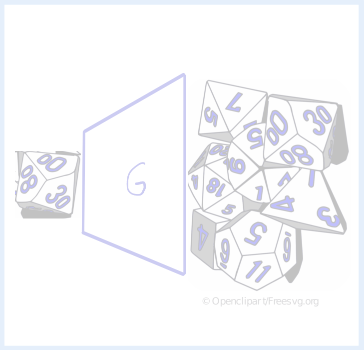
- How can an *unbounded* distinguisher break PRG?
- Is G a PRG or not? Below G_1 and G_2 are PRGs
  1. $G(s) := G_1(s) \| 0$
  2. $G(s) := s \| s_1 \oplus \ldots \oplus s_n$
  3. $G(s_1 \| s_2) := G_1(s_1) \| G_2(s_2)$
  4. $G(s) := G_1(s) \| G_2(s)$
  5. $G(s) := G_1(s) \oplus G_2(s)$



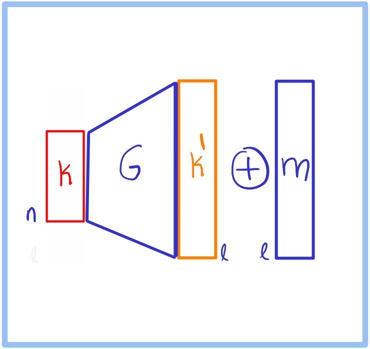
Plan for Today's Lecture...

Goal: construct SKE computationally-secret against eavesdroppers

Pseudorandom Generator (PRG)



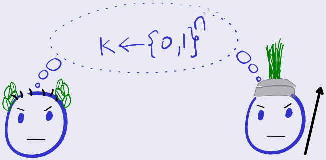
Computational One-Time Pad



New tool: proof by reduction

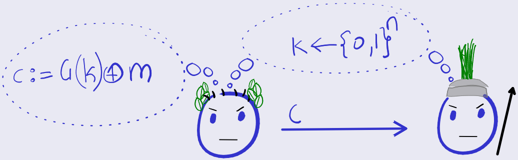
"Computational" One-Time Pad from PRG G

Construction 2 (Message space $\{0,1\}^{\ell(n)}$)



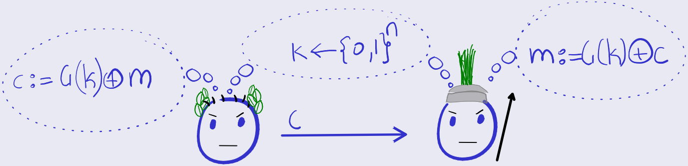
"Computational" One-Time Pad from PRG G

Construction 2 (Message space $\{0,1\}^{\ell(n)}$)



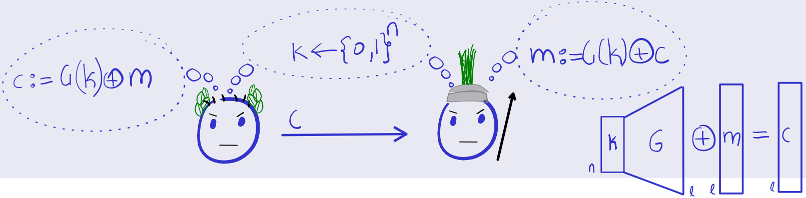
"Computational" One-Time Pad from PRG G

Construction 2 (Message space $\{0,1\}^{\ell(n)}$)



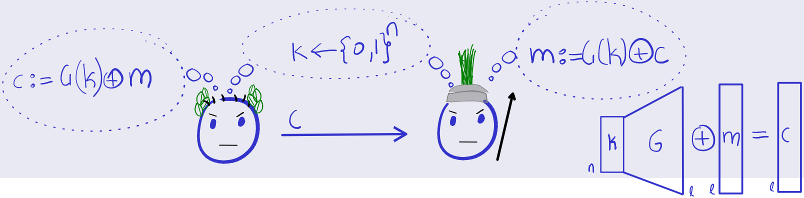
"Computational" One-Time Pad from PRG G

Construction 2 (Message space $\{0,1\}^{\ell(n)}$)



"Computational" One-Time Pad from PRG G

Construction 2 (Message space $\{0,1\}^{\ell(n)}$)

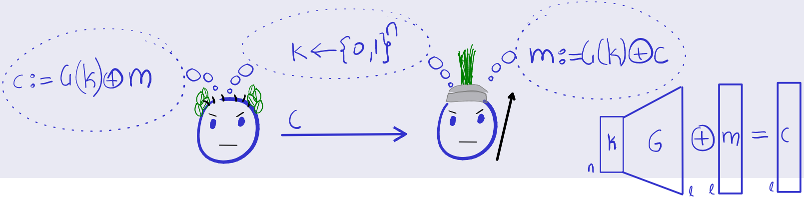


Pseudocode 2 (Message space $\{0,1\}^{\ell(n)}$)

- Key generation $\mathsf{Gen}(1^n)$: output $k \leftarrow \{0,1\}^n$
- Encryption $\mathsf{Enc}(k, m)$: output $c := G(k) \oplus m$
- Decryption $\mathsf{Dec}(k, c)$: output $m := G(k) \oplus c$

"Computational" One-Time Pad from PRG G

Construction 2 (Message space $\{0,1\}^{\ell(n)}$)



Pseudocode 2 (Message space $\{0,1\}^{\ell(n)}$)

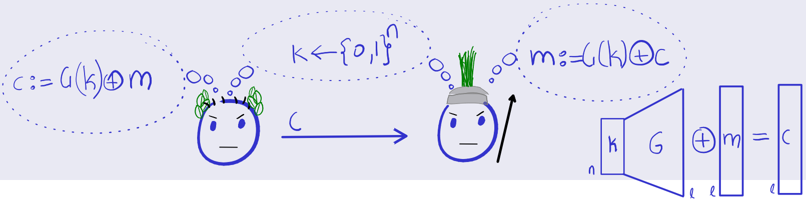
- Key generation $\mathsf{Gen}(1^n)$: output $k \leftarrow \{0,1\}^n$
- Encryption $\mathsf{Enc}(k,m)$: output $c := G(k) \oplus m$
- Decryption $\mathsf{Dec}(k,c)$: output $m := G(k) \oplus c$

- Correctness of decryption: for every $n \in \mathbb{N}$ and $m \in \{0,1\}^{\ell(n)}$,

$$\Pr_{k \leftarrow \{0,1\}^n}\left[G(k) \oplus \left(G(k) \oplus m\right) = m\right]$$

"Computational" One-Time Pad from PRG G

Construction 2 (Message space $\{0,1\}^{\ell(n)}$)



Pseudocode 2 (Message space $\{0,1\}^{\ell(n)}$)

- Key generation $\mathsf{Gen}(1^n)$: output $k \leftarrow \{0,1\}^n$
- Encryption $\mathsf{Enc}(k,m)$: output $c := G(k) \oplus m$
- Decryption $\mathsf{Dec}(k,c)$: output $m := G(k) \oplus c$

- Correctness of decryption: for every $n \in \mathbb{N}$ and $m \in \{0,1\}^{\ell(n)}$,

$$\Pr_{k \leftarrow \{0,1\}^n}\left[G(k) \oplus \left(G(k) \oplus m\right) = m\right] = \Pr_{k \leftarrow \{0,1\}^n}\left[m = m\right] = 1$$

Proof of Computational Secrecy...

Theorem 1

If G is a PRG, then Construction 2 is comp. secret against eavesdroppers

Proof by reduction. $\exists$D for G $\Leftarrow$ $\exists$Eve breaking Construction 2.

Proof of Computational Secrecy...

Theorem 1

If G is a PRG, then Construction 2 is comp. secret against eavesdroppers

Proof by reduction. ∃D for G ⇐ ∃Eve breaking Construction 2.

Intuition: consider the following four worlds

World 0

$$\Pr_{s \leftarrow \{0,1\}^n}[Eve(G(s) \oplus m_0) = 0]$$

World 1

$$\Pr_{s \leftarrow \{0,1\}^n}[Eve(G(s) \oplus m_1) = 0]$$

$$\Pr_{r \leftarrow \{0,1\}^{\ell(n)}}[Eve(r \oplus m_0) = 0]$$

World 0'

$$\Pr_{r \leftarrow \{0,1\}^{\ell(n)}}[Eve(r \oplus m_1) = 0]$$

World 1'

Proof of Computational Secrecy...

Theorem 1

If G is a PRG, then Construction 2 is comp. secret against eavesdroppers

Proof by reduction. $\exists$D for G $\Leftarrow$ $\exists$Eve breaking Construction 2.

Intuition: consider the following four worlds

World 0

$$\Pr_{s \leftarrow \{0,1\}^n}[\text{Eve}(G(s) \oplus m_0) = 0]$$

Target ≈

World 1

$$\Pr_{s \leftarrow \{0,1\}^n}[\text{Eve}(G(s) \oplus m_1) = 0]$$

$$\Pr_{r \leftarrow \{0,1\}^{\ell(n)}}[\text{Eve}(r \oplus m_0) = 0]$$

World 0'

$$\Pr_{r \leftarrow \{0,1\}^{\ell(n)}}[\text{Eve}(r \oplus m_1) = 0]$$

World 1'

Proof of Computational Secrecy...

Theorem 1

If G is a PRG, then Construction 2 is comp. secret against eavesdroppers

Proof by reduction. $\exists$D for G $\Leftarrow$ $\exists$Eve breaking Construction 2.

Intuition: consider the following four worlds

World 0

$$\Pr_{s \leftarrow \{0,1\}^n}[\mathrm{Eve}(G(s) \oplus m_0) = 0]$$

Target

World 1

$$\Pr_{s \leftarrow \{0,1\}^n}[\mathrm{Eve}(G(s) \oplus m_1) = 0]$$

$$\Pr_{r \leftarrow \{0,1\}^{\ell(n)}}[\mathrm{Eve}(r \oplus m_0) = 0]$$

World 0'

$$\Pr_{r \leftarrow \{0,1\}^{\ell(n)}}[\mathrm{Eve}(r \oplus m_1) = 0]$$

World 1'

Proof of Computational Secrecy...

Theorem 1

If G is a PRG, then Construction 2 is comp. secret against eavesdroppers

Proof by reduction. $\exists$D for G $\Leftarrow$ $\exists$Eve breaking Construction 2.

Intuition: consider the following four worlds

World 0 (PRG, Target)

$$\Pr_{s \leftarrow \{0,1\}^n}[\mathrm{Eve}(G(s) \oplus m_0) = 0]$$

World 1 (Target)

$$\Pr_{s \leftarrow \{0,1\}^n}[\mathrm{Eve}(G(s) \oplus m_1) = 0]$$

World 0' (PRG)

$$\Pr_{r \leftarrow \{0,1\}^{\ell(n)}}[\mathrm{Eve}(r \oplus m_0) = 0]$$

World 1'

$$\Pr_{r \leftarrow \{0,1\}^{\ell(n)}}[\mathrm{Eve}(r \oplus m_1) = 0]$$

Proof of Computational Secrecy...

Theorem 1

If G is a PRG, then Construction 2 is comp. secret against eavesdroppers

Proof by reduction. $\exists$D for G $\Leftarrow$ $\exists$Eve breaking Construction 2.

Intuition: consider the following four worlds

World 0

$$\Pr_{s \leftarrow \{0,1\}^n}\left[\mathrm{Eve}(G(s) \oplus m_0) = 0\right]$$

$\approx$ Target

World 1

$$\Pr_{s \leftarrow \{0,1\}^n}\left[\mathrm{Eve}(G(s) \oplus m_1) = 0\right]$$

$\approx$ PRG

$$\Pr_{r \leftarrow \{0,1\}^{\ell(n)}}\left[\mathrm{Eve}(r \oplus m_0) = 0\right]$$

World 0'

PRG $\approx$

$$\Pr_{r \leftarrow \{0,1\}^{\ell(n)}}\left[\mathrm{Eve}(r \oplus m_1) = 0\right]$$

World 1'

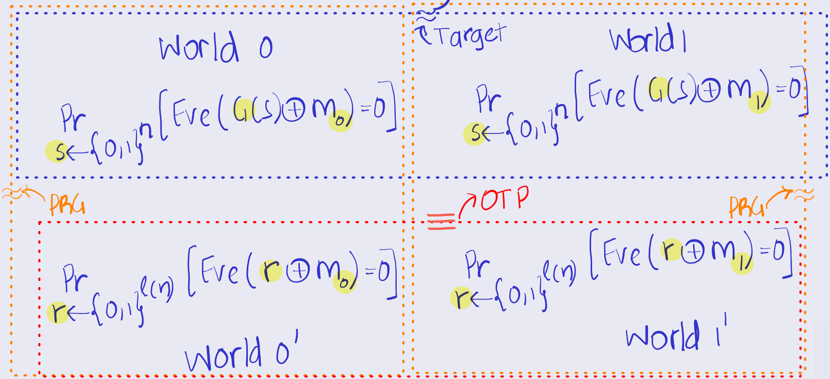
Proof of Computational Secrecy...

Theorem 1

If G is a PRG, then Construction 2 is comp. secret against eavesdroppers

Proof by reduction. ∃D for G ⇐ ∃Eve breaking Construction 2.

Intuition: consider the following four worlds



Proof of Computational Secrecy...

Theorem 1

If G is a PRG, then Construction 2 is comp. secret against eavesdroppers

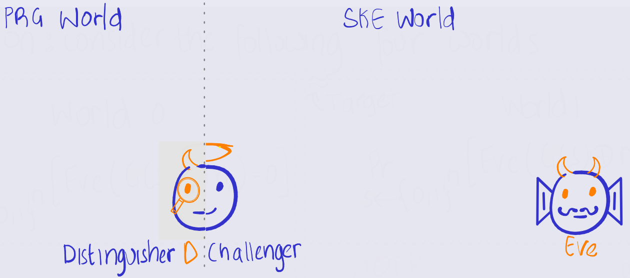
Proof by reduction. $\exists$D for G $\Leftarrow$ $\exists$Eve breaking Construction 2.

Proof of Computational Secrecy...

Theorem 1

If G is a PRG, then Construction 2 is comp. secret against eavesdroppers

Proof by reduction. $\exists$D for G $\Leftarrow$ $\exists$Eve breaking Construction 2.

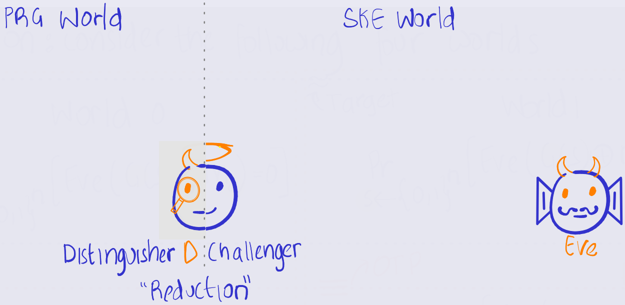


Proof of Computational Secrecy...

Theorem 1

If G is a PRG, then Construction 2 is comp. secret against eavesdroppers

Proof by reduction. $\exists$D for G $\Leftarrow$ $\exists$Eve breaking Construction 2.

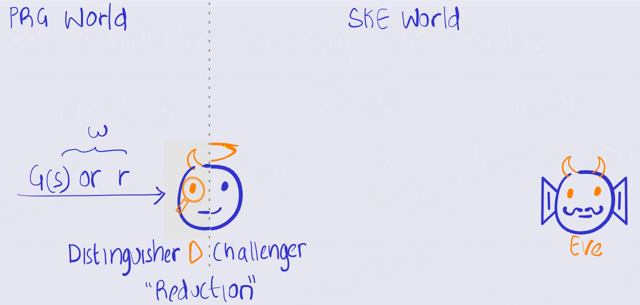


Proof of Computational Secrecy...

Theorem 1

If G is a PRG, then Construction 2 is comp. secret against eavesdroppers

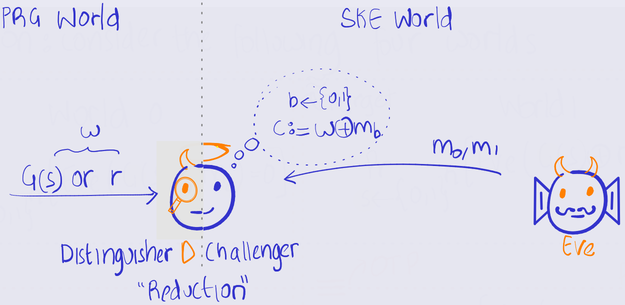
Proof by reduction. $\exists$D for G $\Leftarrow$ $\exists$Eve breaking Construction 2.



Theorem 1

If G is a PRG, then Construction 2 is comp. secret against eavesdroppers

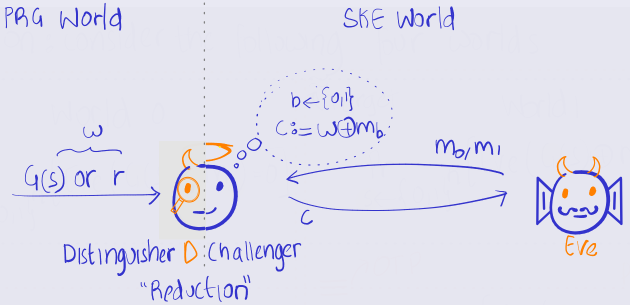
Proof by reduction. $\exists$D for G $\Leftarrow$ $\exists$Eve breaking Construction 2.



Theorem 1

If G is a PRG, then Construction 2 is comp. secret against eavesdroppers

Proof by reduction. $\exists$D for G $\Leftarrow$ $\exists$Eve breaking Construction 2.

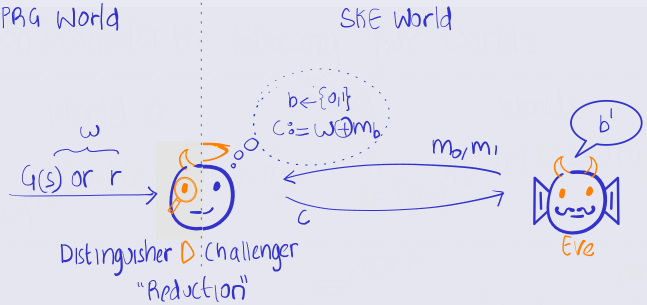


Proof of Computational Secrecy...

Theorem 1

If G is a PRG, then Construction 2 is comp. secret against eavesdroppers

Proof by reduction. ∃D for G ⇐ ∃Eve breaking Construction 2.

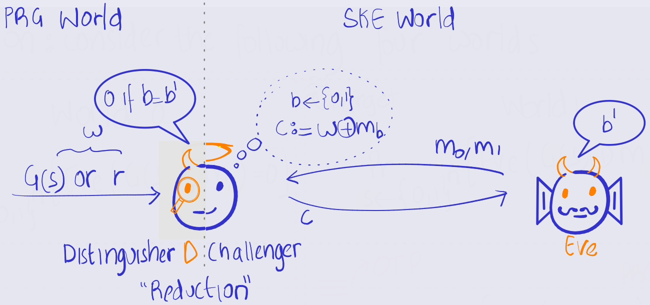


Proof of Computational Secrecy...

Theorem 1

If G is a PRG, then Construction 2 is comp. secret against eavesdroppers

Proof by reduction. ∃D for G ⇐ ∃Eve breaking Construction 2.

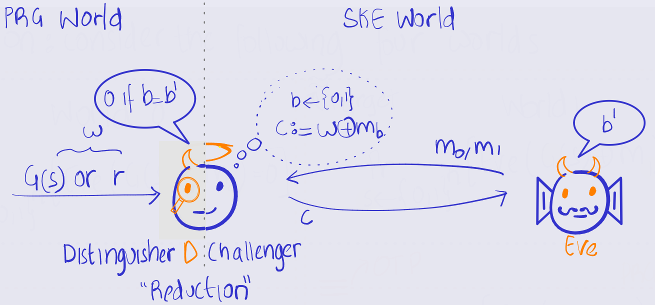


Proof of Computational Secrecy...

Theorem 1

If G is a PRG, then Construction 2 is comp. secret against eavesdroppers

Proof by reduction. ∃D for G ⇐ ∃Eve breaking Construction 2.



Analysis

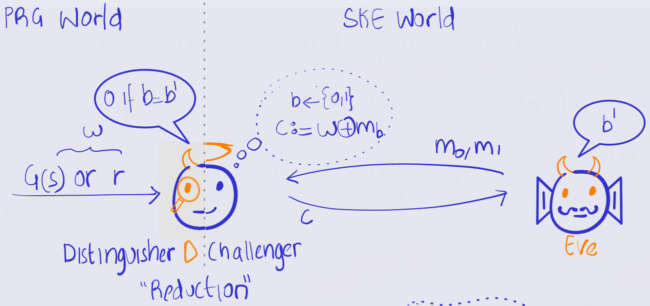
We want to show: $\left| \Pr_{s \leftarrow \{0,1\}^n}[D(G(s))=0] - \Pr_{r \leftarrow \{0,1\}^{\ell(n)}}[D(r)=0] \right|$ not negl

Proof of Computational Secrecy...

Theorem 1

If G is a PRG, then Construction 2 is comp. secret against eavesdroppers

Proof by reduction. $\exists$D for G $\Leftarrow$ $\exists$Eve breaking Construction 2.



Analysis

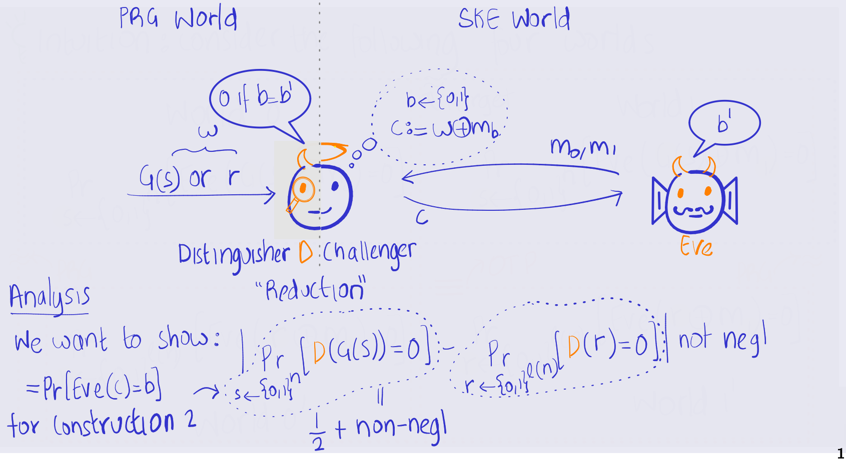
We want to show: $\left|\Pr_{s\leftarrow\{0,1\}^n}[D(G(s))=0] - \Pr_{r\leftarrow\{0,1\}^{\ell(n)}}[D(r)=0]\right|$ not negl

$=\Pr[Eve(c)=b]$ for Construction 2

Theorem 1

If G is a PRG, then Construction 2 is comp. secret against eavesdroppers

Proof by reduction. $\exists$D for G $\Leftarrow$ $\exists$Eve breaking Construction 2.

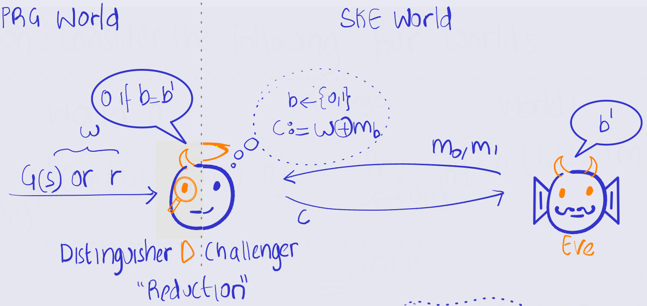


Proof of Computational Secrecy...

Theorem 1

If G is a PRG, then Construction 2 is comp. secret against eavesdroppers

Proof by reduction. ∃D for G ⇐ ∃Eve breaking Construction 2.



Analysis

We want to show: $\left|\Pr_{s\leftarrow\{0,1\}^n}[D(G(s))=0] - \Pr_{r\leftarrow\{0,1\}^{\ell(n)}}[D(r)=0]\right|$ not negl

$\Pr_{s\leftarrow\{0,1\}^n}[D(G(s))=0] = \Pr[Eve(c)=b]$ for Construction 2 $= \frac{1}{2} +$ non-negl

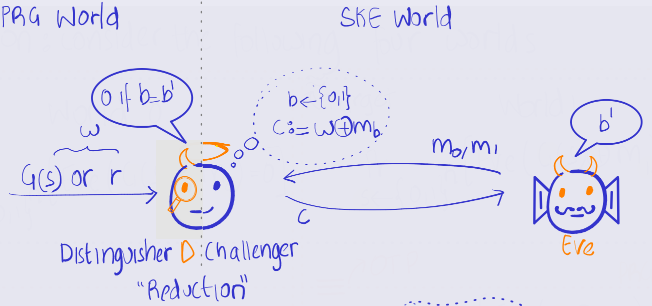
$\Pr_{r\leftarrow\{0,1\}^{\ell(n)}}[D(r)=0] = \frac{1}{2} = \Pr[Eve(c)=b]$ for OTP

Proof of Computational Secrecy...

Theorem 1

If G is a PRG, then Construction 2 is comp. secret against eavesdroppers

Proof by reduction. $\exists$D for G $\Leftarrow$ $\exists$Eve breaking Construction 2.



<u>Analysis</u>

We want to show: $\left| \Pr_{s \leftarrow \{0,1\}^n}[D(G(s))=0] - \Pr_{r \leftarrow \{0,1\}^{\ell(n)}}[D(r)=0] \right|$ not negl

$= \Pr[Eve(c)=b]$ for Construction 2

$= \left| \frac{1}{2} + \text{non-negl} - \frac{1}{2} \right|$, where $\frac{1}{2} = \Pr[Eve(c)=b]$ for OTP

Proof of Computational Secrecy...

Exercise 3 (Formalise proof of Theorem 1)

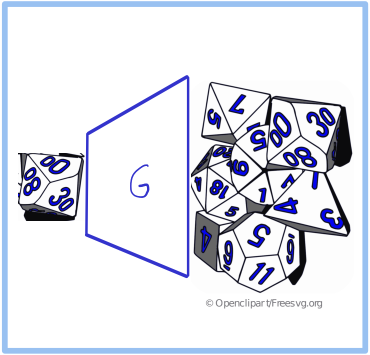
Write down the proof formally:

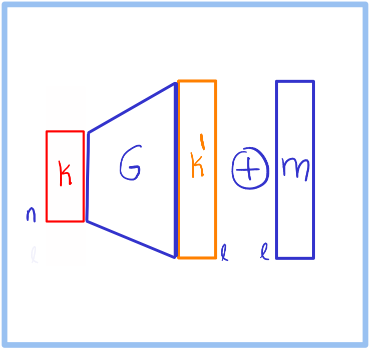
1. Analyse why the reduction works
2. In the analysis, explicitly write down expression for "not negligible"

Plan for Today's Lecture...

Goal: construct SKE computationally-secret against eavesdroppers

NEW Pseudorandom Generator (PRG) NEW Computational One-Time Pad





New tool: proof by reduction

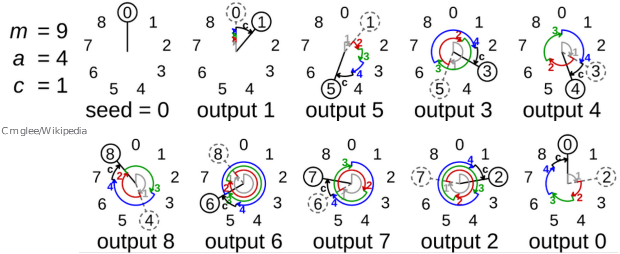
Do PRGs Exist?

- What about Linear Congruential Generator (LCG)?
  - Useful for physics simulation

Do PRGs Exist?

- What about Linear Congruential Generator (LCG)?
  - Useful for physics simulation
- Defined by following recurrence relation, with "seed" $x_0 \in [0, m-1]$:

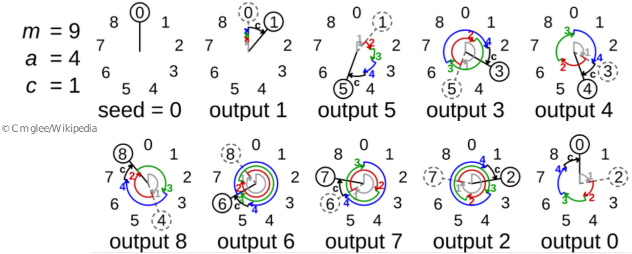
$$x_{n+1} = ax_n + c \bmod m$$



Do PRGs Exist?

- What about Linear Congruential Generator (LCG)?
  - Useful for physics simulation
- Defined by following recurrence relation, with "seed" $x_0 \in [0, m-1]$:

$$x_{n+1} = ax_n + c \bmod m$$



- But insecure for cryptographic purposes: "non-cryptographic" PRG

Exercise 4

1) Think of why is LCG insecure 2) Look up LFSR

Do Cryptographic PRGs Exist?...

Theoretical constructions

- Rely on well-studied hard problems

Do Cryptographic PRGs Exist?...

Theoretical constructions

- Rely on well-studied hard problems
- Subset-sum problem:
  - Input: prime p, $a_1, \ldots, a_n \in \mathbb{Z}_p$
  - Solution: $I \subseteq [1, n] : \sum_{i \in I} a_i = 0 \bmod p$

Do Cryptographic PRGs Exist?...

Theoretical constructions

- Rely on well-studied hard problems
- Subset-sum problem:
  - Input: prime p, $a_1, \ldots, a_n \in \mathbb{Z}_p$
  - Solution: $I \subseteq [1, n] : \sum_{i \in I} a_i = 0 \bmod p$
- Believed to be "hard" (even for $a_1, \ldots, a_n \leftarrow \mathbb{Z}_p$)

Do Cryptographic PRGs Exist?...

Theoretical constructions

- Rely on well-studied hard problems
- Subset-sum problem:
  - Input: prime p, $a_1, \ldots, a_n \in \mathbb{Z}_p$
  - Solution: $I \subseteq [1, n] : \sum_{i \in I} a_i = 0 \bmod p$
- Believed to be "hard" (even for $a_1, \ldots, a_n \leftarrow \mathbb{Z}_p$)

- E.g., PRG from subset-sum problem:

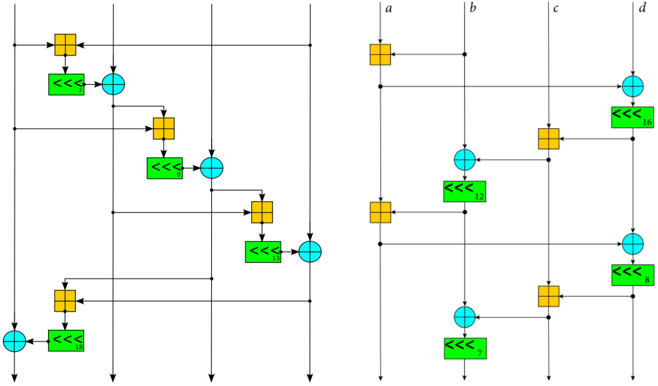
$$G(x_1 \| \ldots \| x_n) := \sum_{i \in [1,n]} x_i a_i \bmod p$$

  - On selecting $p \approx n^2$, G is expanding
  - Pseudorandomness based on hardness of subset-sum problem

Do Cryptographic PRGs Exist?...

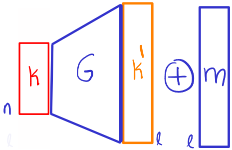
Practical constructions

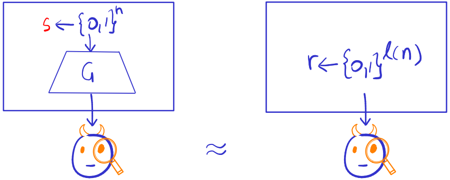
- “Complex” functions, repeated “many times” look random
- Build a candidate construction and do extensive cryptanalysis
- E.g., Stream ciphers like Salsa20 and ChaCha



© Sissssou/Wikipedia © Tony Arcieri/Wikipedia

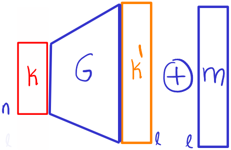
- To recap:
  - Defined PRG
  - Constructed computational OTP from PRG
    - New tool: proof by reduction
  - Constructions of PRG

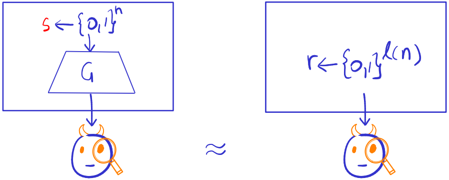




Recap/Next Lecture...

- To recap:
  - Defined PRG
  - Constructed computational OTP from PRG
    - New tool: proof by reduction
  - Constructions of PRG

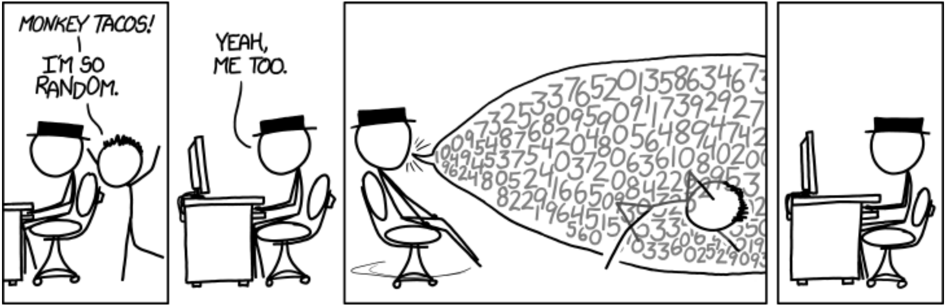




- Next lecture:
  - Encrypting longer messages!
  - Extending the length of a PRG

    - New tool: hybrid argument

Recap/Next Lecture...



©xkcd.com

More Questions?

References

1. §3.2 and §3.3 in [KL14] for more details on computational secrecy and computational OTP
2. To read more about stream ciphers, refer to §4 in [BS23]

Dan Boneh and Victor Shoup.
A Graduate Course in Applied Cryptography, Version 0.6.
2023.

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