# Roadmap

- Getting rid of the first restriction imposed by perfect security
  - Encrypting long messages using short keys
- Pseudorandom generators
  - Various equivalent definitions

# Encrypting Long Messages Using Short Keys : The Basic Idea

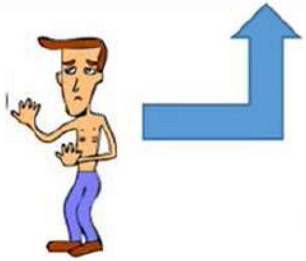
$\mathcal{M} = \mathcal{K} = \mathcal{C} = \{0,1\}^L \qquad G: \{0,1\}^\ell \rightarrow \{0,1\}^L, \ell < L$

$m \in \mathcal{M}$

$s \in_r \{0,1\}^\ell$

$c \coloneqq m \oplus G(s)$

$s \in_r \{0,1\}^\ell$



Computationally
bounded

- A computationally unbounded adversary cannot distinguish whether $c$ is an encryption of $m_0$ or $m_1$, since the pad $k$ is a uniformly random $L$-bit string

# Encrypting Long Messages Using Short Keys : The Basic Idea

$\mathcal{M} = \mathcal{K} = \mathcal{C} = \{0,1\}^L$  $G: \{0,1\}^\ell \to \{0,1\}^L, \ell < L$

$m \in \mathcal{M}$

$s \in_r \{0,1\}^\ell$

$c := m \oplus G(s)$

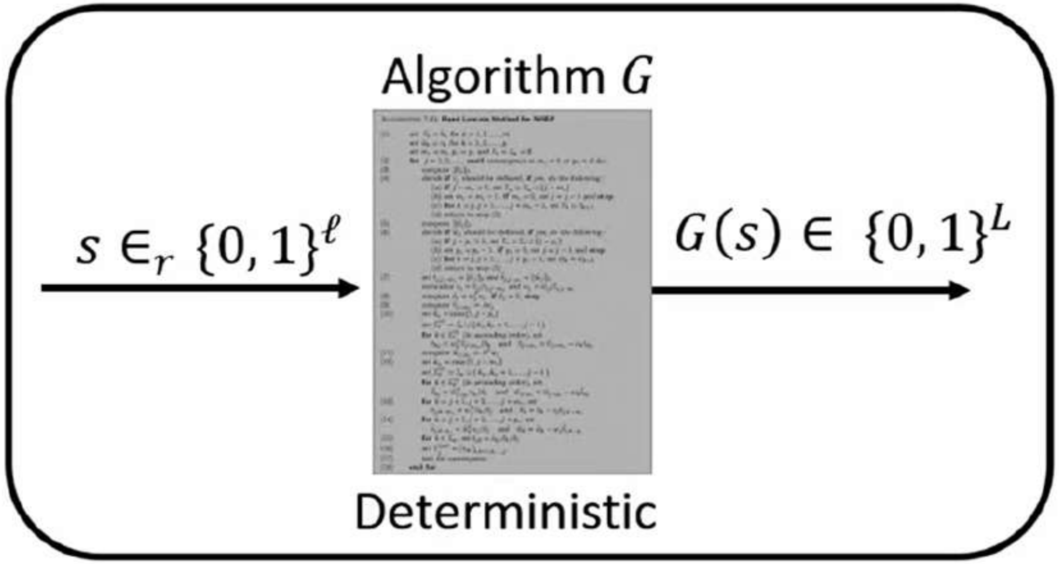
Computationally bounded

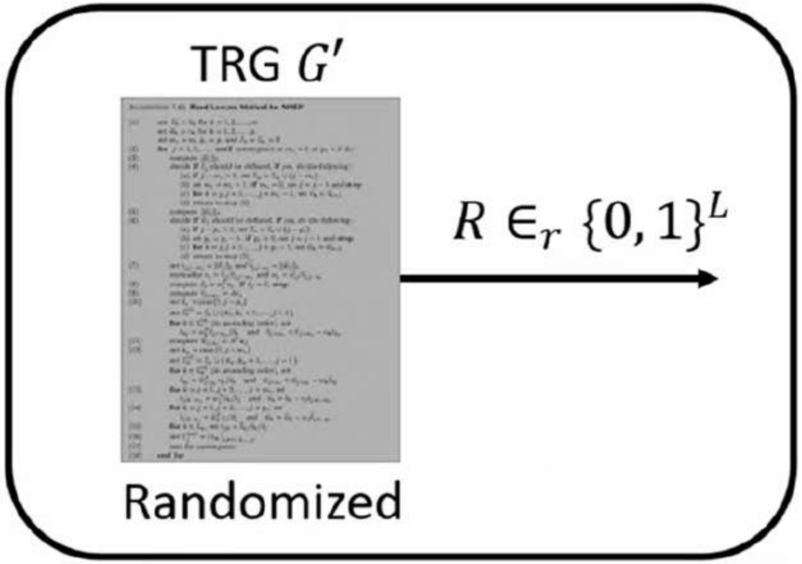
$s \in_r \{0,1\}^\ell$

- A computationally bounded adversary cannot distinguish between $G(s)$ and a uniformly random string from $\{0,1\}^L$

A computationally bounded adversary cannot distinguish whether $c$ is an encryption of $m_0$ or $m_1$

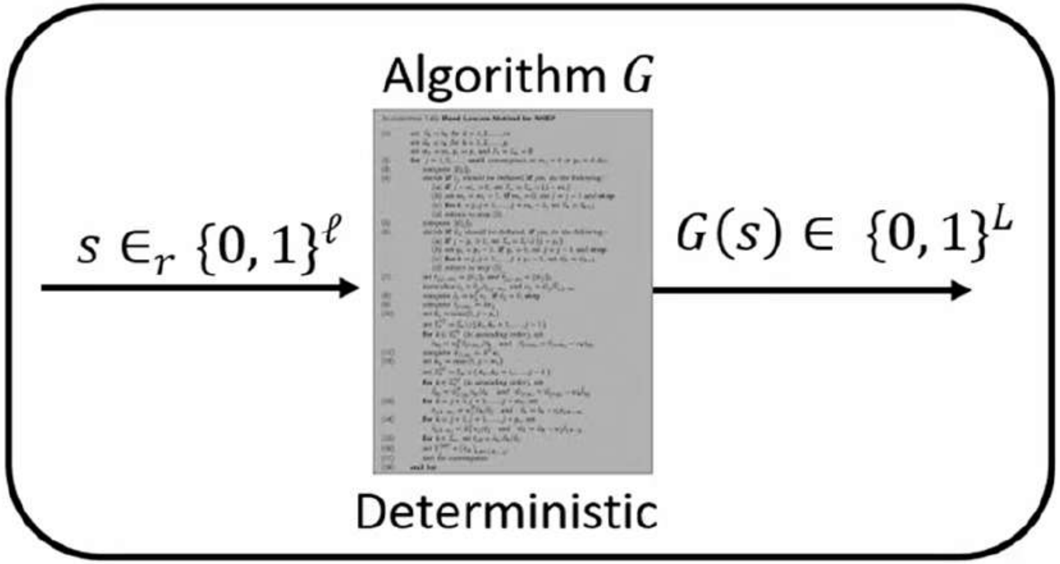
# Pseudorandom Generator (PRG)

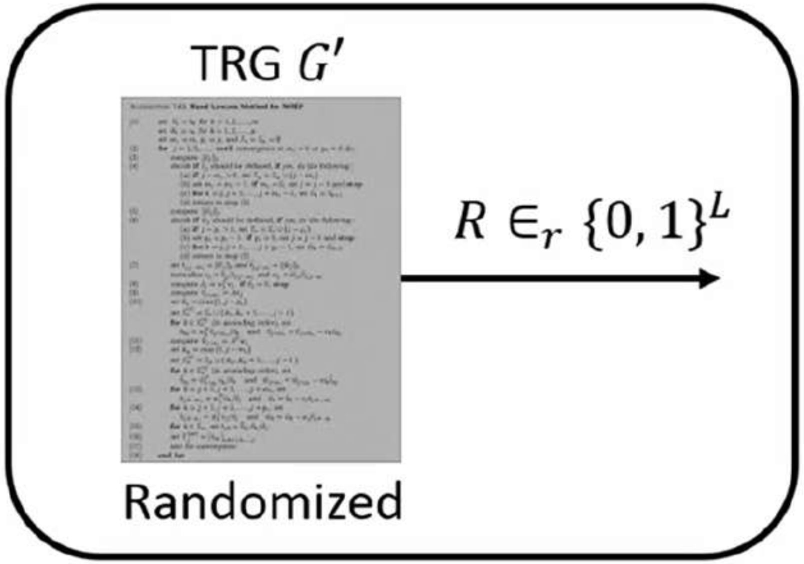



- Requirements from algorithm $G$:
  - $G$ should be an efficient algorithm
  - Expansion : $L > \ell$
  - Pseudo randomness (informal): no efficient statistical test should significantly separate apart an output of $G$ from the output of an $L$-bit truly random generator (TRG)

# Indistinguishability Based Definition of PRG




- The output behavior of $G$ and $G'$ should be almost identical
- No efficient algorithm (distinguisher) should be able to distinguish apart a random sample generated by $G$, from a random sample generated by $G'$, with a significant probability
  - Modeled as an indistinguishability game

# PRG Indistinguishability Game

- $b = 0 : y \in_r \{0,1\}^L$
- $b = 1$:

$b \leftarrow \{0, 1\}$

$G$

$\xrightarrow{s \in_r \{0,1\}^\ell} G \xrightarrow{y = G(s)}$

$\xrightarrow{y \in \{0,1\}^L}$

(How $y$ is generated?)

$\xleftarrow{b'}$

PPT Distinguisher $\mathcal{D}$

- Algorithm $G$ is a PRG, if for every PPT distinguisher $\mathcal{D}$ participating in the above experiment:

$$\Pr(\mathcal{D} \text{ outputs } b' = b) \leq {}^1/_2 + \mathrm{negl}(n)$$

$$\approx$$

- Algorithm $G$ is a PRG, if for every PPT distinguisher $\mathcal{D}$ participating in the above experiment:

$$\Big| \underbrace{\Pr[\mathcal{D} \text{ outputs } b'=1 \mid b = 0]}_{\text{Prob. of } \mathcal{D} \text{ labeling } y \text{ as outcome of PRG, given that } y \text{ is generated by TRG}} - \underbrace{\Pr[\mathcal{D} \text{ outputs } b'=1 \mid b = 1]}_{\text{Prob. of } \mathcal{D} \text{ labeling } y \text{ as outcome of PRG, given that } y \text{ is generated by PRG}} \Big| \leq \mathrm{negl}(n)$$

# PRG : An Example

$G$

$s \in_r \{0,1\}^\ell$  $G(s) = ss'$

$$s' \stackrel{\text{def}}{=} s_1 \oplus s_2 \oplus \ldots \oplus s_\ell$$

- Does there exist an efficient statistical test to distinguish a random sample of $G$, from a uniformly random $(\ell+1)$-bit length string, with a significant probability ?
  - For any $y = (y_1, \ldots, y_{\ell+1})$ where $y = G(s)$, $y_{\ell+1} = y_1 \oplus \ldots \oplus y_\ell$ always holds
  - For any $y \in_r \{0,1\}^{\ell+1}$, $y_{\ell+1} = y_1 \oplus \ldots \oplus y_\ell$ holds with probability ½

- $b = 0$ : $y \in_r \{0,1\}^{\ell+1}$
- $b = 1$: $y = G(s)$, where $s \in_r \{0,1\}^\ell$

$b \leftarrow \{0, 1\}$

$y = (y_1, \ldots, y_{\ell+1})$

(How I generated it?)

$b' = 1$, iff $y_{\ell+1} = y_1 \oplus \ldots \oplus y_\ell$

Distinguisher $\mathcal{D}$

Pr[$\mathcal{D}$ outputs $b'$=1 | $b$ = 0] = ½

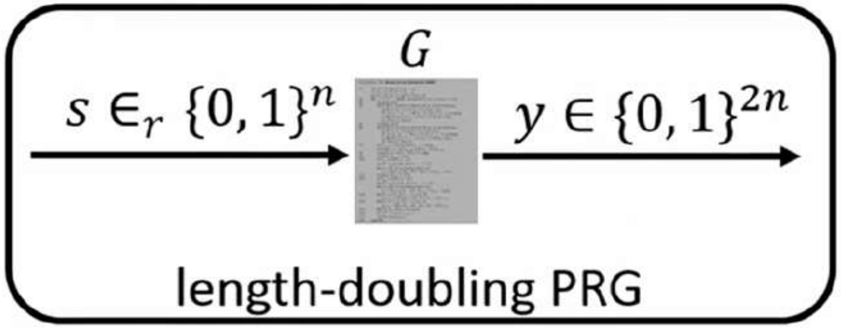
Pr[$\mathcal{D}$ outputs $b'$=1 | $b$ = 1] = 1

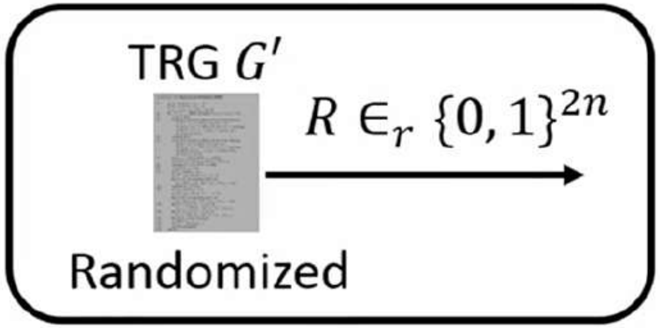
| Pr[$\mathcal{D}$ outputs $b'$=1 | $b$ = 0] − Pr[$\mathcal{D}$ outputs $b'$=1 | $b$ = 1] |

$= \frac{1}{2} =$ non- negl($n$)

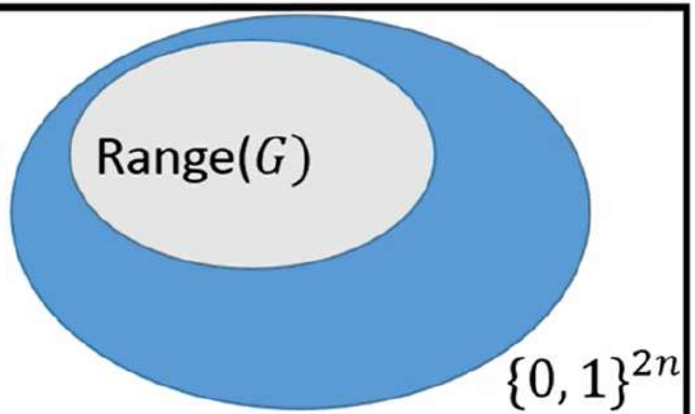
# PRG Can be Always Distinguished by a Brute force Distinguisher

- Any PRG has to deterministically expand its input
  - Consequence : output of PRG is far away from a uniformly random string




- Most strings of length $2n$ do not occur in the range of $G$
  - Range of $G$ --- a proper subset of $\{0,1\}^{2n}$ of size atmost $2^n$
  - Prob. that a uniformly random $2n$-bit string occurs in the range of $G$ is atmost $2^n/2^{2n} = 2^{-n}$



# PRG Can be Always Distinguished by a Brute force Distinguisher

$G$

$s \in_r \{0,1\}^n$  $y \in \{0,1\}^{2n}$

length-doubling PRG

TRG $G'$

 $R \in_r \{0,1\}^{2n}$

Randomized

- ❑ $b = 0 : y \in_r \{0,1\}^{2n}$
- ❑ $b = 1$: $y = G(s)$, where $s \in_r \{0,1\}^n$

$b \leftarrow \{0, 1\}$

$y = (y_1, \ldots, y_{2n})$

(How $y$ is generated?)

Distinguisher $\mathcal{D}$

$b' = 1$, iff $y = G(s)$, for some $s \in \{0,1\}^n$

- ❑ Running time of $\mathcal{D} = \mathcal{O}(2^n)$ --- inefficient

| Pr[$\mathcal{D}$ outputs $b'$=1 | $b$ = 0] − Pr[$\mathcal{D}$ outputs $b'$=1 | $b$ = 1] | $= 1 - 2^{-n}$

# PRG : An Alternate Definition

TRG $G'$

- For $i = 1, \dots, L$
  - Generate $y_i \in_r \{0, 1\}$
- Output $y = (y_1, \dots, y_L)$

$(y_1, \dots, y_i)$

Partial output

$\mathcal{A}$

$\Pr[\mathcal{A}(y_1, \dots, y_i) = y_{i+1}] \leq \frac{1}{2}$

- For any $i \in \{1, \dots, L-1\}$, given the output bits $y_1, \dots, y_i$ of $y$, no algorithm can predict the next output bit $y_{i+1}$, with probability better than ½
- PRG alternate definition (Next-bit predictor test) : the above should also hold for a PRG

Publicly known $G: \{0,1\}^\ell \Rightarrow \{0,1\}^L$

Experiment $\text{Next-Bit}_{\mathcal{A}}^{G}$

$G$

$s \in_r \{0,1\}^\ell$

$y = (y_1, \dots, y_L)$

$i \in \{1, \dots, L-1\}$

$(y_1, \dots, y_i)$

$b \in \{0, 1\}$

PPT $\mathcal{A}$

$G$ is called unpredictable if

$\Pr[\mathcal{A} \text{ outputs } b = y_{i+1}] \leq \frac{1}{2} + \mathsf{negl}(n)$