Understanding the definition:

Can we learn the least significant bit of a message given only the ciphertext (assuming a semantically-secure cipher)

No! Suppose we could. Then, adversary can choose two messages m_0, m_1 that differ in their least significant bit and distinguish with probability 1.

This generalizes to any efficiently-computable property of the two messages.

How does semantic security relate to perfect secrecy?

Theorem. If a cipher satisfies perfect secrecy, then it is semantically secure.

Proof. Perfect secrecy means that $\forall m_0, m_1 \in \mathcal{M}, c \in \mathcal{C}$:

$$\Pr[k \xleftarrow{R} \mathcal{K} : \text{Encrypt}(k, m_0) = c] = \Pr[k \xleftarrow{R} \mathcal{K} : \text{Encrypt}(k, m_1) = c]$$

Equivalently, the distributions

$$\underbrace{\{k \xleftarrow{R} \mathcal{K} : \text{Encrypt}(k, m_0)\}}_{D_0} \quad \text{and} \quad \underbrace{\{k \xleftarrow{R} \mathcal{K} : \text{Encrypt}(k, m_1)\}}_{D_1}$$

are identical ($D_0 \equiv D_1$). This means that the adversary's output b' is identically distributed in the two experiments, and so

$$\text{SSAdv}[\mathcal{A}, \Pi_{SE}] = |W_0 - W_1| = 0.$$

Corollary. The one-time pad is semantically secure.

encryption key (PRG seed)

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

seems straightforward, but takes some care to prove

Theorem. Let G be a secure PRG. Then, the resulting stream cipher constructed from G is semantically secure.

Proof. Consider the semantic security experiments:

Experiment 0: Adversary chooses m_0, m_1 and receives $c_0 = G(s) \oplus m_0$

Experiment 1: Adversary chooses m_0, m_1 and receives $c_1 = G(s) \oplus m_1$

} Want to show that adversary's output in these two experiments are indistinguishable

Let $W_0 = \Pr[\mathcal{A}$ outputs 1 in Experiment 0$]$

$W_1 = \Pr[\mathcal{A}$ outputs 1 in Experiment 1$]$

Goal: Show that if G is a secure PRG, then for all efficient adversaries $\mathcal{A}$, $|W_0 - W_1| = \text{negl}(\lambda)$.

Idea: If $G(s)$ is uniform random string (i.e., one-time pad), then $W_0 = W_1$. But $G(s)$ is like a one-time pad!

Define Experiment 0': Adversary chooses m_0, m_1 and receives $c_0 = t \oplus m_0$ where $t \xleftarrow{R} \{0,1\}^n$

Experiment 1': Adversary chooses m_0, m_1 and receives $c_1 = t \oplus m_1$ where $t \xleftarrow{R} \{0,1\}^n$

} called "hybrid experiments"

Define W_0', W_1' accordingly.

Now we can write

$$|W_0 - W_1| = |W_0 - W_0' + W_0' - W_1' + W_1' - W_1|$$
$$\leq |W_0 - W_0'| + \underbrace{|W_0' - W_1'|}_{W_0' = W_1' \text{ (for all adversaries } \mathcal{A}\text{)}} + |W_1' - W_1| \quad \text{by triangle inequality}$$

since OTP satisfies perfect secrecy

Suffices to show that for all efficient adversaries, $|W_0 - W_0'| = \text{negl}(\lambda)$ and $|W_1 - W_1'| = \text{negl}(\lambda)$.

Show. If G is a secure PRG, then for all efficient A, $|W_0 - W_0'| = \text{negl}$.

Common proof technique: prove the contrapositive.

Contrapositive: If A can distinguish Experiments 0 and 0′, then G is not a secure PRG.

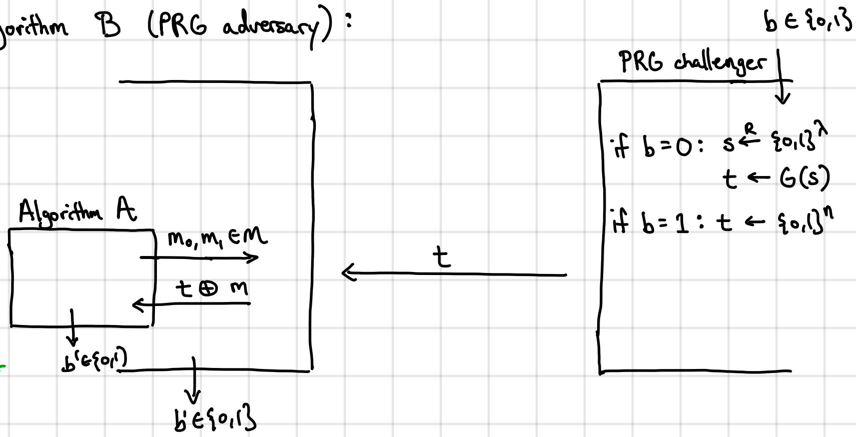
Suppose there exists efficient A that distinguishes Experiment 0 from 0′

$\Rightarrow$ We use A to construct efficient adversary B that breaks security of G.

↳ this step is a reduction

[we show how adversary (i.e., algorithm) for distinguishing Exp. 0 and 0′ $\Rightarrow$ adversary for PRG]

Algorithm B (PRG adversary):



Algorithm A expects to get $t \oplus m$ where $t = G(s)$ or $t \xleftarrow{R} \{0,1\}^n$

Running time of B = running time of A = efficient

Compute $\text{PRGAdv}[B, G]$.

$\Pr[B \text{ outputs } 1 \text{ if } b = 0] = W_0$ ← if $b = 0$, then A gets $G(s) \oplus m$ which is precisely the behavior in Exp. 0

$\Pr[B \text{ outputs } 1 \text{ if } b = 1] = W_0'$ ← if $b = 1$, then A gets $t \oplus m$ which is precisely the behavior in Exp. 0′

$\Rightarrow \text{PRGAdv}[B, G] = |W_0 - W_0'|$, which is non-negligible by assumption. This proves the contrapositive.

Important note: Security of above schemes shown assuming message space is $\{0,1\}^n$ (i.e., all messages are n-bits long)

In practice: We have variable-length messages. In this case, security guarantees indistinguishability from other messages of the same length, but length itself is leaked [inevitable if we want short ciphertexts]

↳ can be problematic — see traffic analysis attacks!

So far, we have shown that if we have a PRG, then we can encrypt messages efficiently (stream cipher).

Do PRGs exist? We don't know! More difficult problem than resolving P vs. NP!

However, it is not hard to see that if PRGs exist, then $P \neq NP$. [Try proving this yourself]

↳ What we can say is that if $\underbrace{\text{"one-way functions"}}$ (OWF) exist, then there exists a PRG that stretches the seed by 1 bit (e.g., λ-bit seed $\to$ $(\lambda+1)$-bit string)

function that is "easy" to compute but "hard" to invert

↳ will define more formally later in the course

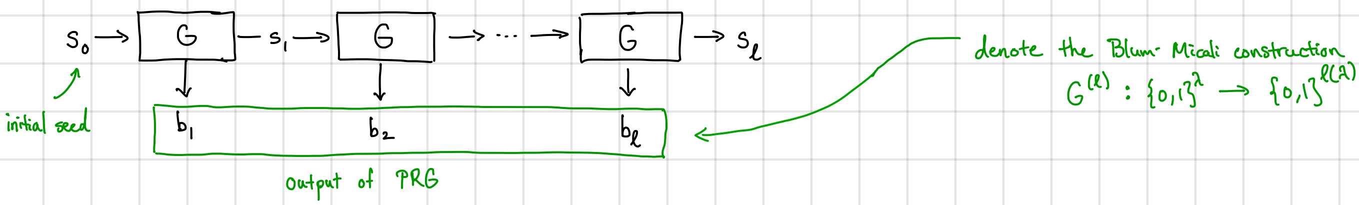
a PRG is an example of such a function

given $s \in \{0,1\}^\lambda$, evaluating $G(s) \in \{0,1\}^n$ is easy

given $G(s) \in \{0,1\}^n$ for random $s \in \{0,1\}^\lambda$, computing s is hard (why?)

But what if we want PRGs with longer stretch? For example, can we build PRGs with stretch $\ell(\lambda) = \text{poly}(\lambda)$ for arbitrary polynomials?

Blum-Micali PRG: Suppose $G: \{0,1\}^\lambda \to \{0,1\}^{\lambda+1}$ is a secure PRG. We build a PRG with stretch $\ell(\lambda) = \text{poly}(\lambda)$ as follows:



Why is this constructing a secure PRG?

↳ Intuitively, if s_0 is uniformly random, then $G(s_0) = (b_1, s_1)$ is uniformly random so we can feed s_1 into the PRG and take b_1 as the first output bit of the PRG $\Rightarrow$ iterate until we have ℓ output bits

Theorem. If $G: \{0,1\}^\lambda \to \{0,1\}^{\lambda+1}$ is a secure PRG, then the Blum-Micali generator $G^{(\ell)}: \{0,1\}^\lambda \to \{0,1\}^{\ell(\lambda)}$ is also a secure PRG for all $\ell = \text{poly}(\lambda)$.

Proof. Consider the following experiments:

Experiment H_0: Sample $s_0 \xleftarrow{R} \{0,1\}^\lambda$ and adversary is given $G^{(\ell)}(s_0)$

Experiment H_1: Sample $t \xleftarrow{R} \{0,1\}^{\ell(\lambda)}$ and adversary is given t

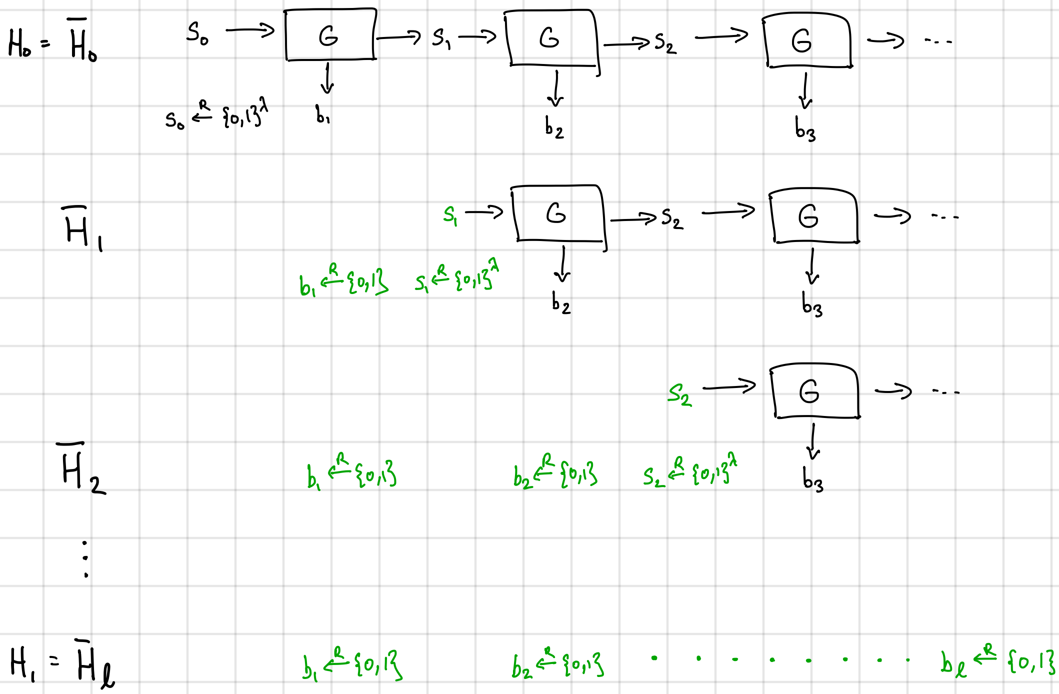
For an adversary A, define

$W_0 := \Pr[A \text{ outputs } 1 \text{ in } H_0]$

$W_1 := \Pr[A \text{ outputs } 1 \text{ in } H_1]$

Goal: Show that if G is secure, then for all efficient adversaries A, $|W_0 - W_1| = \text{negl}(\lambda)$.

We will use a "hybrid" argument. Specifically, we first define a sequence of intermediate experiments, where each adjacent pair of experiments is easy to reason about (i.e., directly reduces to security of G)



<u>Basic idea</u>: in experiment $\bar{H}_i$ the first i bits of output are generated <u>uniformly at random</u> while the remaining bits are generated using the Blum-Micali generator

In each experiment, adversary is given the sequence of bits $b_1 b_2 \cdots b_\ell$

Let A be an efficient distinguisher. Define $\bar{W}_i := \Pr[\text{A outputs 1 in experiment } \bar{H}_i]$

$$\begin{aligned}
\text{Then, PRGAdv}[A,G] &= |W_0 - W_1| \\
&= |\bar{W}_0 - \bar{W}_\ell| \quad \text{(by definition)} \\
&= |\bar{W}_0 - \bar{W}_1 + \bar{W}_1 - \bar{W}_2 + \cdots + \bar{W}_{\ell-1} - \bar{W}_\ell| \\
&\leq |\bar{W}_0 - \bar{W}_1| + |\bar{W}_1 - \bar{W}_2| + \cdots + |\bar{W}_{\ell-1} - \bar{W}_\ell| \quad \text{(by triangle equality)}
\end{aligned}$$

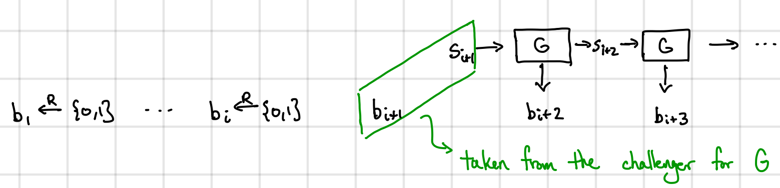
<u>Claim</u>. If G is a secure PRG, then for all efficient adversaries A, $|\bar{W}_i - \bar{W}_{i+1}| = \text{negl}(\lambda)$.

<u>Proof</u>. We will show the <u>contrapositive</u>: if A can distinguish experiments $\bar{H}_i$ and $\bar{H}_{i+1}$, then A can break pseudorandomness of G.

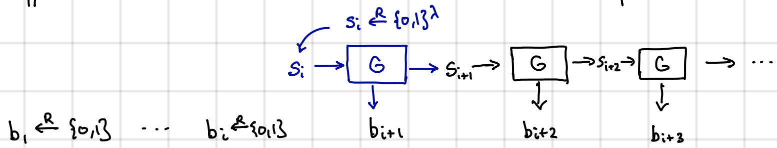
Suppose $|\bar{W}_i - \bar{W}_{i+1}| = \varepsilon$. We use A to build a distinguisher B for G. Algorithm B works as follows:

1. On input a string $z \in \{0,1\}^{\lambda+1}$, algorithm B parses z as (b_{i+1}, s_{i+1}) where $b_{i+1} \in \{0,1\}$ and $s_{i+1} \in \{0,1\}^\lambda$
2. Sample $b_1, \ldots, b_i \xleftarrow{R} \{0,1\}$.
3. Compute $b_{i+2}, \ldots, b_\ell$ using Blum-Micali with seed s_{i+1}. Give $b_1 \cdots b_\ell$ to A and output whatever A outputs.

In pictures:

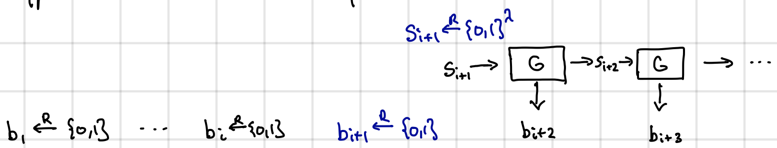


Two possibilities: 1. Suppose $z = G(s_i)$ for some $s_i \xleftarrow{R} \{0,1\}^\lambda$. Then, above picture looks like this:



In this case, $b_1, \ldots, b_\ell$ is distributed exactly as in experiment H_i and so $\mathcal{A}$ outputs 1 with prob. $\overline{W}_i$

2. Suppose $z \xleftarrow{R} \{0,1\}^{\lambda+1}$. Then above picture looks like this:



In this case, $b_1, \ldots, b_\ell$ is distributed exactly as in experiment H_{i+1} and so $\mathcal{A}$ outputs 1 with prob. $\overline{W}_{i+1}$

Very important to argue that $\mathcal{B}$ "simulates" the correct view for $\mathcal{A}$. Otherwise, behavior of $\mathcal{A}$ is unknown!

Thus, $\mathrm{PRGAdv}[\mathcal{B}, G] = |\overline{W}_i - \overline{W}_{i+1}|$
$= \varepsilon$ since $\mathcal{B}$ outputs whatever $\mathcal{A}$ outputs

Since $\mathcal{B}$ is efficient (assuming $\mathcal{A}$ is efficient), by security of G, $\mathrm{PRGAdv}[\mathcal{B}, G] = \mathrm{negl}(\lambda)$. Thus, $\varepsilon = |p_i - p_{i+1}| = \mathrm{negl}(\lambda)$, and the claim follows. ■

To complete the proof of the main theorem, we have that

$$
\begin{aligned}
|\overline{W}_0 - \overline{W}_\ell| &\leq |\overline{W}_0 - \overline{W}_1| + \cdots + |\overline{W}_{\ell-1} - \overline{W}_\ell| \\
&\leq \ell \cdot \mathrm{negl}(\lambda) \\
&= \mathrm{negl}(\lambda) \quad \text{since } \ell = \mathrm{poly}(\lambda).
\end{aligned}
$$

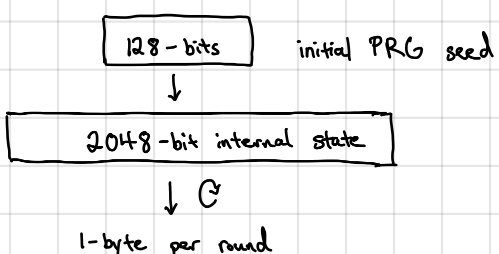
<u>Proof strategy recap</u>: 1. Hybrid arguments: to argue indistinguishability of a pair of distributions, begin by identifying a <u>simple</u> set of intermediate distributions, and argue that each pair of adjacent distributions is indistinguishable

2. Security reduction (proof by contrapositive): To show a statement of the form "If X is secure, then Y is secure," show instead the the statement "If Y is not secure, then X is not secure." In the proof, show that if there exists an adversary for Y (i.e. Y is not secure), then there exists an adversary for X.

↳ When constructing this adversary, it is important to show that it simulates the <u>correct distribution</u> of inputs to the underlying adversary (i.e., this is essentially showing correctness of the <u>reduction algorithm</u>)

Stream ciphers in practice:

- RC4 (1987) stream cipher (widely used - SSL/TLS protocol, 802.11b)



Numerous problems:

- Bias in initial output: $\Pr[\text{second byte} = 0] = \frac{2}{256} > \frac{1}{256}$
  - ↳ When using RC4, recommendation is to ignore first 256 bytes due to potential bias
  - ↳ Correlations in output: probability of seeing $(0,0)$ in output is $\frac{1}{256^2} + \frac{1}{256^3} > \frac{1}{256^2}$
  - ↳ Given outputs of RC4 with related keys (e.g., keys sharing common suffix), possible to recover keys after seeing few blocks of output
    - ↳ Can be very problematic on weak devices (who may not have good sources of entropy)

- Modern stream ciphers (eSTREAM project: 2004-2008)
  - Salsa20 (2005) ⇝ ChaCha (2008)
    ↳ core design maps 256-bit key, 64-bit nonce, 64-bit counter onto a 512-bit output

    (64-bit nonce:) enables using same key (and different nonces) to encrypt multiple messages (will discuss later)

    (64-bit counter:) allows random access into the stream

    Design is more complex:
    - relies on a sequence of rounds
    - each round consists of 32-bit additions, xors, and bit-shifts

    ↳ very fast even in software (4-14 CPU cycles/output byte) — used to encrypt TLS traffic between Android and Google services

Recall: the one-time pad is not reusable (i.e., the two-time pad is totally broken)

NEVER REUSE THE KEY TO A STREAM CIPHER!

But wait... we "proved" that a stream cipher was secure, and yet, there is an attack?

Recall security game:

adversary		challenger ($b \in \{0,1\}$)
	m_0, m_1 →	$k \xleftarrow{R} K$
	← c_b	$c_b \leftarrow \text{Encrypt}(k, m_b)$
$b' \in \{0,1\}$	

Observe: adversary only sees one ciphertext; key is only used once

⇒ Security in this model says nothing about multiple messages/ciphertexts

Problem: If we want security with multiple ciphertexts, we need a different or stronger definition (CPA security)

Reusable security: security against chosen-plaintext attacks $\boxed{\text{(CPA-security)}}$

↳ semantic security should hold even if adversary sees multiple encrypted messages of its choosing

adversary does not just passively observe, it can choose the messages to be encrypted!

↳ captures many settings where adversary might know the message that is encrypted (e.g., predictable headers or site content in web traffic) or be able to influence it (e.g., client replies to an email sent by adversary)

↳ goal is to capture as broad of a range of attacks as possible