Are we done? We now have a perfectly-secure cipher!

No! Keys are very long! In fact, as long as the message... [if we can share keys of this length, can use same mechanism to share the message itself]

"One-time" restriction

Malleable

Issues with the one-time pad:

- **One-time**: Very important. Never reuse the one-time pad to encrypt two messages. Completely broken!

Suppose $c_1 = k \oplus m_1$ and $c_2 = k \oplus m_2$

Then, $c_1 \oplus c_2 = (k \oplus m_1) \oplus (k \oplus m_2)$

$= m_1 \oplus m_2$ ← learn the xor of two messages! ← can leverage this to recover messages

One-time pad reuse:

- Project Verona (U.S. counter-intelligence operation against U.S.S.R during Cold War)
  - ↳ Soviets reused some pages in codebook ~ led to decryption of ~3000 messages sent by Soviet intelligence over 37-year period [notably exposed espionage by Julius and Ethel Rosenberg]
- Microsoft Point-to-Point Tunneling (MS-PPTP) in Windows 98/NT (used for VPN)
  - ↳ Same key (in stream cipher) used for both server → client communication AND for client → server communication
    - ↳ (RC4)
- 802.11 WEP: both client and server use same key to encrypt traffic

  Many problems just beyond one-time pad reuse (can even recover key after observing small number of frames!)

- **Malleable**: one-time pad provides no integrity; anyone can modify the ciphertext:

$m \leftarrow k \oplus c$

↖ replace c with $c \oplus m'$

$\Rightarrow \quad k \oplus (c \oplus m') = m \oplus m'$ ← adversary's change now xored into original message

Theorem (Shannon). If a cipher satisfies perfect secrecy, then $|\mathcal{K}| \geq |\mathcal{M}|$.

Intuition: Every ciphertext can decrypt to at most $|\mathcal{K}| < |\mathcal{M}|$ messages. This means that ciphertext leaks information about the message (not all messages equally likely). Cannot be perfectly secret.

Proof. We will use a "counting" argument. Suppose $|\mathcal{K}| < |\mathcal{M}|$. Take any ciphertext $c \leftarrow \text{Encrypt}(k,m)$ for some $k \in \mathcal{K}$, $m \in \mathcal{M}$. This ciphertext can only decrypt to at most $|\mathcal{K}|$ possible messages (one for each choice of key). Since $|\mathcal{K}| < |\mathcal{M}|$, there is some message $m' \in \mathcal{M}$ such that

$$\forall k \in \mathcal{K} : \text{Decrypt}(k,c) \neq m'$$

By correctness of the cipher,

$$\forall k \in \mathcal{K} : \text{Encrypt}(k,m') \neq c$$

This means that

$$\left.\begin{aligned} &\Pr[k \xleftarrow{R} \mathcal{K} : \text{Encrypt}(k,m') = c] = 0 \\ &\Pr[k \xleftarrow{R} \mathcal{K} : \text{Encrypt}(k,m) = c] > 0 \end{aligned}\right\} \text{Cannot be perfectly secret!}$$

Take-away: Perfect secrecy requires long keys. Very impractical (except in the most critical scenarios — exchanging daily codebooks)

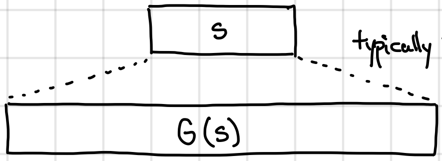
If we want something efficient / usable, we need to compromise somewhere.

- Observe: Perfect secrecy is an information-theoretic (i.e., a mathematical) property

  Even an infinitely-powerful (computationally-unbounded) adversary cannot break security

  We will relax this property and only require security against computationally-bounded (efficient) adversaries

Idea: "compress" the one-time pad: we will generate a long random-looking string from a <u>short</u> seed (e.g., $s \in \{0,1\}^{128}$).



typically: $s \in \{0,1\}^\lambda$ (λ is the seed length or <u>security parameter</u>)

$G(s) \in \{0,1\}^n$ where $n \gg \lambda$

n is the "stretch" of a PRG

Stream cipher: $\mathcal{K} = \{0,1\}^\lambda$

$\mathcal{M} = \mathcal{C} = \{0,1\}^n$

Encrypt(k, m): $c \leftarrow m \oplus \boxed{G(k)}$

Decrypt(k, c): $m \leftarrow c \oplus \boxed{G(k)}$

Instead of xor-ing with the key, we use the key to derive a "stream" of random-looking bits and use that in place of the one-time pad

If $\lambda < n$, then this scheme cannot be perfectly secure! So we need a <u>different</u> notion of security

<u>Intuitively</u>: Want a stream cipher to function "like" a one-time pad to any "reasonable" adversary.

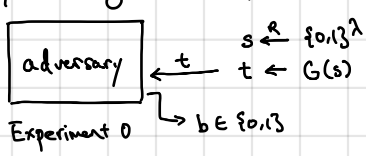
$\Rightarrow$ Equivalently: output of a PRG should "look" like uniformly-random string

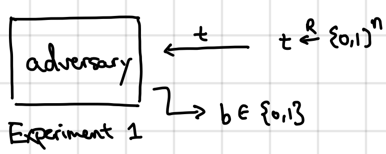
What is a "reasonable" adversary?

- Theoretical answer: algorithm runs in (probabilistic) polynomial time (denote poly(λ) where λ is a security parameter)
  e.g., length of secret key
- Practical answer: runs in time $< 2^{80}$ and space $< 2^{64}$ (can use larger numbers as well)

<u>Goal</u>: Construct a PRG so no efficient adversary can distinguish output from random.

Captured by defining two experiments or games:





the input to the adversary (t) is often called the <u>challenge</u>

Adversary's goal is to <u>distinguish</u> between Experiment 0 (pseudorandom string) and Experiment 1 (truly random string)

↳ It is given as input a string t of length n (either $t \leftarrow G(s)$ or $t \xleftarrow{R} \{0,1\}^n$)

↳ It outputs a guess (a single bit $b \in \{0,1\}$)

Remember: adversary knows the algorithm G; only seed is hidden!

Let $W_0 := \Pr[\text{adversary outputs } 1 \text{ in Experiment } 0]$
$W_1 := \Pr[\text{adversary outputs } 1 \text{ in Experiment } 1]$
} define the distinguishing advantage of $\mathcal{A}$ as

$\text{PRGAdv}[\mathcal{A}, G] := |W_0 - W_1|$

DO NOT RELY ON SECURITY BY OBSCURITY!

Definition. A PRG $G: \{0,1\}^\lambda \to \{0,1\}^n$ is <u>secure</u> if for all efficient adversaries $\mathcal{A}$,

probabilistic polynomial time

$$\text{PRGAdv}[\mathcal{A}, G] = \underline{\text{negl}}(\lambda)$$

↳ negligible function (in the input length)

smaller than any inverse polynomial

e.g., $\frac{1}{2^\lambda}$, $\lambda^{\log \lambda}$

- Theoretical definition: $f(\lambda)$ is negligible if $f \in o(\lambda^c)$ for all $c \in \mathbb{N}$
- Practical "definition": quantity $\leq 2^{-80}$ or $\leq 2^{-128}$

Understanding the definition:

1. Can we ask for security against all adversaries (when $n \gg \lambda$)?

   No! Consider inefficient adversary that outputs 1 if t is the image of G and 0 otherwise.
   - $W_0 = 1$
   - $W_1 = \Pr[t \xleftarrow{R} \{0,1\}^n : \exists s \in \{0,1\}^\lambda : G(s) = t] = \frac{1}{2^{n-\lambda}}$

   $\Big\}$ $\mathrm{PRGAdv}[A,G] = 1 - \frac{1}{2^{n-\lambda}} \approx 1$ if $n \gg \lambda$

2. Can the output of a PRG be biased (e.g., first bit of PRG output is 1 w.p. $\frac{2}{3}$)?

   No! Consider efficient adversary that outputs 1 if first bit of challenge is 1.
   - $W_0 = \frac{2}{3}$
   - $W_1 = \frac{1}{2}$

   $\Big\}$ $\mathrm{PRGAdv}[A,G] = \frac{1}{6}$ NOT NEGLIGIBLE!

   More generally, no efficient statistical test can distinguish output of a secure PRG from random.

3. Can the output of a PRG be predictable (e.g., given first 10 bits, predict the 11^{th} bit)?

   No! If the bits are predictable w.p. $\frac{1}{2} + \varepsilon$, can distinguish with advantage ε (since random string is unpredictable)

   In fact: unpredictable $\Rightarrow$ pseudorandom

Take-away: A secure PRG has the same statistical properties as the one-time pad to any efficient adversary.

$\Longrightarrow$ Should be able to use it in place of one-time pad to obtain a secure encryption scheme (against efficient adversaries)

Need to define security of an encryption scheme.

Goal is to capture property that no efficient adversary can learn any information about the message given only the ciphertext. Suffices to argue that no efficient adversary can distinguish encryption of message m_0 from m_1, even if m_0, m_1 are adversarially-chosen.

Let (Encrypt, Decrypt) be a cipher. We define two experiments (parameterized by $b \in \{0,1\}$):

adversary		challenger ($b \in \{0,1\}$)
	$m_0, m_1 \in \mathcal{M}$ $\longrightarrow$	$k \xleftarrow{R} \mathcal{K}$
	$\longleftarrow$ $c_b \leftarrow \mathrm{Encrypt}(k, m_b)$	
$\downarrow$ $b' \in \{0,1\}$		

$\Big\}$ semantic security experiment

Adversary chooses two messages and receives encryption of one of them. Needs to guess which one (i.e., distinguish encryption of m_0 from encryption of m_1)

Let $W_0 := \Pr[b' = 1 \mid b = 0]$
$W_1 := \Pr[b' = 1 \mid b = 1]$

$\Big\}$ probability that adversary guesses 1 (if adversary is good distinguisher, these two should be very different)

Define semantic security advantage of adversary A for cipher $\Pi_{SE} = (\mathrm{Encrypt}, \mathrm{Decrypt})$

$$\mathrm{SSAdv}[A, \Pi_{SE}] = |W_0 - W_1|$$

Definition. A cipher $\Pi_{SE} = (\mathrm{Encrypt}, \mathrm{Decrypt})$ is semantically secure if for all efficient adversaries A,

$$\mathrm{SSAdv}[A, \Pi_{SE}] = \mathrm{negl}(\lambda)$$

λ is a security parameter (here, models the bit-length of the key)