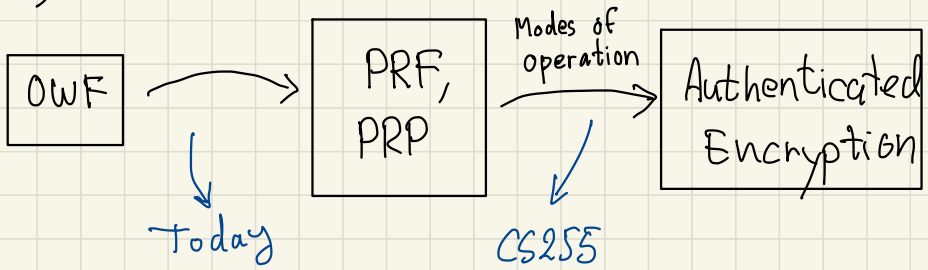
Lecture 2: From OWFs to PRGs and PRFs

Lecture 2 - From OWFs to PRG

Quick logistics reminder

- HW1 is out - due 4/20
- OH - Wed, Thu, Fri x2
- Website, Piazza, Gradescope, Canvas

Symmetric Primitives



The existence of OWF is equivalent to the existence of private-key cryptography.

- Don't know how to build symmetric crypto unconditionally
- But can reduce security to simple, well-studied problems

First course module – Foundations of Crypto

Goals:

(1) Expose to the theoretical point of view

- e.g., hardness assumptions, provably secure primitives relationships between them

(2) Learning proof techniques, basic tools (will be useful even in the more practical later parts of course)

One Way functions

"easy to compute, hard to invert"

Def: $f:\{0,1\}^* \to \{0,1\}^*$ is one-way if:

(1) There exists a poly. time alg. M that computes f. i.e., $\forall x \; M(x)=f(x)$.

(2) $\forall$ PPT alg. A,

probabilistic polynomial time

$$\Pr\left[A(1^n, f(x)) \in f^{-1}(f(x)) : x \xleftarrow{R} \{0,1\}^n\right] \leq \text{negl}(n)$$

randomness over x and the internal randomness of A

A slightly more formal statement of the 2nd condition:

$\forall$ PPT alg. A, $\forall c>0$, $\exists N_c \in \mathbb{N}$, $\forall n > N_c$

$$\Pr\left[f(A(1^n, f(x))) = f(x) : x \xleftarrow{R} \{0,1\}^n\right] \leq \frac{1}{n^c}.$$

Candidates of OWFs:

(1) $f(x,y) = x \cdot y$ for equal-length primes x, y.

(2) $f_{p,g}(x) = g^x \bmod p$ → permutation on $\{1,2,\ldots,p-1\}$

(3) based on subset-sum:

$$f(x_1,\ldots,x_n,S) := (x_1,\ldots,x_n,\sum_{i\in S} x_i)$$

(4) Levin's Universal OWF:

function f_L s.t: $\exists$ OWF $\Rightarrow f_L$ is OWF.

One-way permutation (OWP):

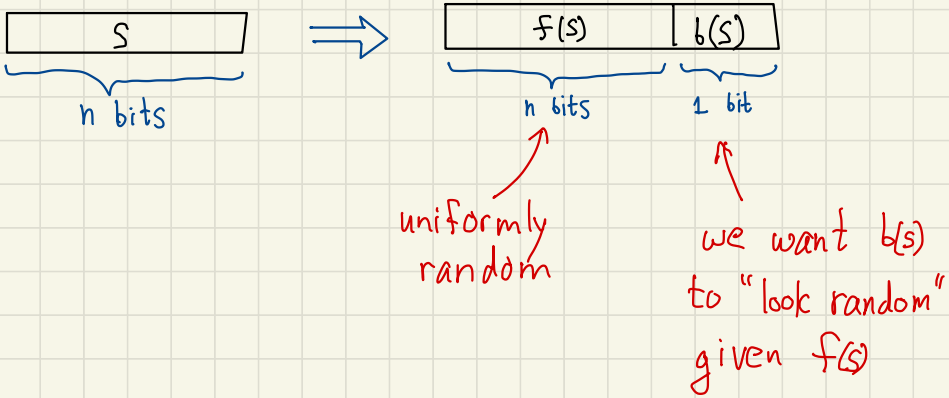
(1) f is a OWF

(2) $\forall n \in \mathbb{N}: f: \{0,1\}^n \to \{0,1\}^n$ is a permution

Plan: OWP $\rightarrow$ stretch 1 PRG $\rightarrow$ Poly. stretch PRG $\rightarrow$ PRF

(The OWF case is harder to prove.)

Part 1: OWP $\Rightarrow$ 1-bit stretch PRG



Hard-core bit

Given $f(x)$ for a OWF, finding x is hard.

What about finding the first bit x_1?

This isn't necessary hard. e.g. $f(x_1,\dots,x_n) = x_1 g(x_2 \dots x_n)$

But it cannot be the case that all $x_1,\dots,x_n$ are easy to compute.

Defn: $b: \{0,1\}^* \to \{0,1\}$ is a hard core predicate of a function $f: \{0,1\}^* \to \{0,1\}$ if:

(1) $b(x)$ can be computed in poly. time given x.

(2) $\forall$ PPT alg. A

$$\Pr\left[A(1^n, f(x)) = b(x) : x \xleftarrow{\$} \{0,1\}^n\right] \leq 1/2 + \text{negl}(n)$$

Question: why do we allow $1/2 + \text{negl}(n)$, and not just $\text{negl}(n)$, as for OWF?

Thm: (Goldreich-Levin) Every OWF has a HC bit.

Idea: a random linear combination of the bits of x should be hard to compute.

First extend the function to

$$g(x, r) = (f(x), r) \quad \text{where } |x| = |r|$$

g is still one-way.

Then take: for $x, r \in \{0,1\}^n$

$$b(x,r) = \langle x, r \rangle = \sum_{i=1}^{n} x_i \cdot r_i \bmod 2 \leftarrow \text{inner product mod 2}$$

Details in Homework 1.

Thm: Let f be a OWP with HC bit b.
Then $G(s) := f(s) \| b(s)$ is a 1-bit stretch PRG.

Main idea: if given $f(s)$, can dist. $b(s)$ from random then can also predict $b(s)$ from $f(s)$.

Proof: Suppose there exists a PPT alg. A and a non-negligible $\varepsilon(n)$ s.t.

(1) $\left| \Pr\left[A(\underbrace{G(s)}_{f(s)\|b(s)}) = 1 : s \xleftarrow{R} \{0,1\}^n\right] - \Pr\left[A(r) = 1 : r \xleftarrow{R} \{0,1\}^{n+q}\right] \right| \geq \varepsilon(n)$

Two steps:

($\bar{b} := 1 - b$)

(1) **Claim:** $\frac{1}{2} \cdot \left| \Pr_s\left[A(f(s)\|b(s)) = 1\right] - \Pr_s\left[A(f(s)\|\bar{b}(s)) = 1\right] \right| \geq \varepsilon$

(2) Use claim to construct an alg. that predicts b.

Proof of Claim:

Let's rewrite the distribution $r \xleftarrow{R} \{0,1\}^n$:

$$\left\{ r : r \xleftarrow{R} \{0,1\}^{n+1} \right\} \equiv \left\{ r \leftarrow r_1 \| r_2 : \begin{matrix} r_1 \xleftarrow{R} \{0,1\}^n \\ r_2 \xleftarrow{R} \{0,1\} \end{matrix} \right\}$$

$$\equiv \left\{ r \leftarrow f(s) \| r_2 : \begin{matrix} s \xleftarrow{R} \{0,1\}^n \\ r_2 \xleftarrow{R} \{0,1\} \end{matrix} \right\}$$

f is a permutation

$\Pr_s[f(s)=a] = \Pr_s[s = f^{-1}(a)] = \frac{1}{2^n}$

$$\equiv \left\{ r \leftarrow f(s) \| r_2 : \begin{matrix} s \xleftarrow{R} \{0,1\}^n \\ r_2 \xleftarrow{R} \{b(s), \bar{b}(s)\} \end{matrix} \right\}$$

$\bar{b} := 1 - b$

$$\equiv \text{w.p. } 1/2 : \left\{ r \leftarrow f(s) \| b(s) : s \xleftarrow{R} \{0,1\}^n \right\}$$

$$\text{w.p. } 1/2 : \left\{ r \leftarrow f(s) \| \bar{b}(s) : s \xleftarrow{R} \{0,1\}^n \right\}$$

$$\Rightarrow \Pr_r[A(r)=1] = \frac{1}{2}\Pr_s[A(f(s)\| b(s))=1] + \frac{1}{2}\cdot\Pr_s[A(f(s)\| \bar{b}(s))=1]$$

We now plug this into (1):

$$\varepsilon \leq \left| \Pr_s[A(G(s))=1] - \Pr_r[A(r)=1] \right|$$

$$= \frac{1}{2}\left(\Pr_s[A(f(s)\|b(s))=1] - \Pr_s[A(f(s)\|\bar{b}(s))=1] \right)$$

∎

We now construct an alg. B that breaks the HC bit b (i.e., predicts $b(s)$ from $f(s)$):

Alg. B

On input $y = f(s) \in \{0,1\}^n$:

(1) Choose $c \xleftarrow{R} \{0,1\}$

(2) Run $A(y\|c)$.

- if A outputs 1, output c
- else, output $\bar{c}$

We analyze the succes probability of B:

$$\Pr_s\left[B(f(s)) = b(s)\right] = \frac{1}{2} \cdot \Pr_{s,c}\left[B(f(s)) = b(s) \mid c = b(s)\right]$$

law of total probability

$$+ \frac{1}{2} \cdot \Pr_{s,c}\left[B(f(s)) = b(s) \mid c = \bar{b}(s)\right]$$

$$= \frac{1}{2} \cdot \left(\Pr_s\left[A(f(s) \| b(s)) = 1\right] + \Pr_s\left[A(f(s) \| \bar{b}(s)) = 0\right]\right)$$

by construction of B

$1 - \Pr\left[A(f(s) \| \bar{b}(s)) = 1\right]$

$$= \frac{1}{2} + \frac{1}{2}\left(\Pr_s\left[A(f(s) \| b(s)) = 1\right] - \Pr_s\left[A(f(s) \| \bar{b}(s)) = 1\right]\right)$$

$$\geq \frac{1}{2} + \varepsilon$$

by Claim

We get that alg. B predicts $b(s)$ with non-negl. prob. $\varepsilon(n)$, which contradict the assumption that b is HC.

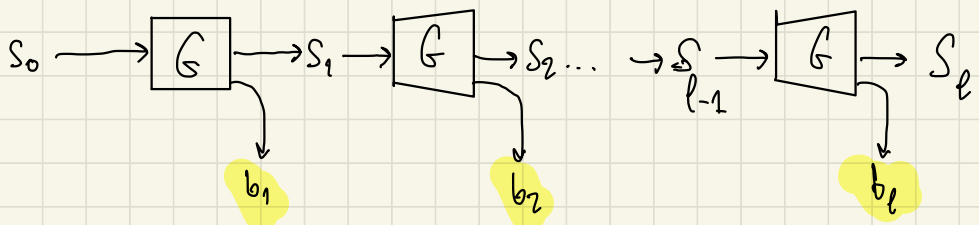
∎

PART 2: From stretch 1 to poly. stretch

The Blum-Micali PRG

Let $G: \{0,1\}^n \to \{0,1\}^{n+1}$ be a PRG,
and let $\ell(n)$ be a polynomial.
We construct $G': \{0,1\}^n \to \{0,1\}^{\ell(n)}$

- on input $s \in \{0,1\}^n$:
  - set $s_0 \leftarrow s$
  - for $i = 1, 2, \ldots, \ell(n)$: $s_i b_i \leftarrow G(s_{i-1})$
  - output $b_1 b_2 \ldots b_{\ell(n)}$



Theorem: G is a secure PRG $\Rightarrow$ G' secure PRG.

- G' is efficient. $t'(n) \approx \ell(n) \cdot t(n) + O(\ell(n))$ ✓
- Security: need to show

$$\{G'(s) : s \xleftarrow{R} \{0,1\}^n\} \approx_c \{y \xleftarrow{R} \{0,1\}^{\ell(n)}\}$$

D_0 — computationally indist. — D_ℓ

That is: $\forall$ PPT alg. A

$$\mathrm{PRGAdv}[A, G'] = \left| \Pr[A(y) = 1 : y \in D_0] - \Pr[A(y) = 1 : y \in D_\ell] \right|$$

$$\leq \mathrm{negl}(n)$$

To this end, we look at a sequence of hybrid distributions:

D_0: $s \to \boxed{G} \to s_1 \to \boxed{G} \to s_2 \to \boxed{G} \to s_3 \ldots$

$\downarrow b_1 \qquad \downarrow b_2 \qquad \downarrow b_3$

D_1: $s \to \boxed{G} \to s_2 \to \ldots$

$b_1 \xleftarrow{R} \{0,1\} \qquad \downarrow b_2$

D_2: $s \to \boxed{G} \to s_3 \to \ldots$

$b_1 \xleftarrow{R} \{0,1\} \qquad b_2 \xleftarrow{R} \{0,1\} \qquad \downarrow b_3$

$\vdots$

D_ℓ $\quad b_1 \xleftarrow{R} \{0,1\} \qquad b_2 \xleftarrow{R} \{0,1\} \qquad \ldots \qquad b_{\ell(n)} \xleftarrow{R} \{0,1\}$

$$p_i := \Pr\left[A(y)=1 : y \leftarrow D_i\right]$$

want to show $|p_0 - p_\ell| = \mathrm{negl}(n)$

$$|p_0 - p_\ell| = |p_0 - p_1 + p_1 - p_2 + p_2 + \ldots + p_{\ell-1} - p_\ell| \leq \sum_{i \in [\ell]} |p_{i+1} - p_i|$$

Claim: $\forall i \in [\ell] \quad |p_{i+1} - p_i| \leq negl(n)$

First, note that Claim $\Rightarrow$ Theorem, since

$$|p_0 - p_\ell| \leq \sum_{i \in [\ell]} |p_i - p_{i+1}| \leq \ell(n) \cdot negl(n) = negl(n)$$

$\ell(n)$ is poly.

Proof of Claim: Suppose for the sake of contradiction that:

$|p_{i+1} - p_i| = \varepsilon(n)$ is <u>not</u> negligible.

We construct a distinguisher $\mathcal{B}$ for G,

and we need to analyze:

$$PRGAdv[B, G] = \left| Pr[B(G(s)) : s \xleftarrow{R} \{0,1\}^n] - Pr[B(y) : y \xleftarrow{R} \{0,1\}^{n+1}] \right|$$

Distinguisher B:

on input $z \in \{0,1\}^{n+1}$:

(1) parse z as (b_{i+1}, s_{i+1}) where $b_{i+1} \in \{0,1\}$, $s_{i+1} \in \{0,1\}^n$

(2) choose $b_1, \ldots, b_i \xleftarrow{R} \{0,1\}$

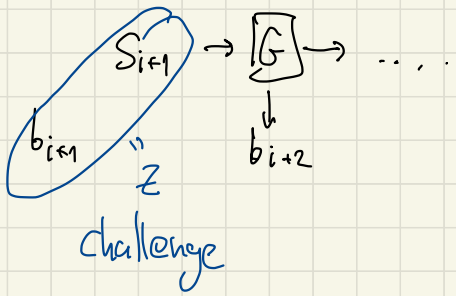
(3) compute $b_{i+2}, s_{i+2} \leftarrow G(s_{i+1}), \ldots \; b_\ell s_\ell \leftarrow G(s_{\ell-1})$

(4) set $y \leftarrow b_1 \ldots b_\ell$

(5) run and output $A(y)$

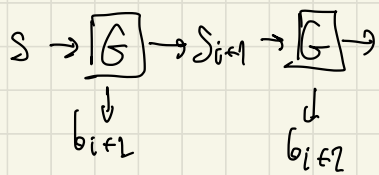
Picture of B:

$b_1 \leftarrow \{0,1\}$ ... $b_i \leftarrow \{0,1\}$



Two cases:

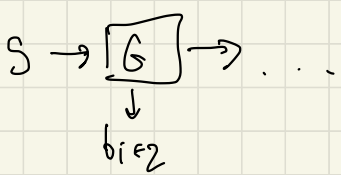
(1) $z = G(s)$ for $s \xleftarrow{R} \{0,1\}^n$



$b_1 \leftarrow \{0,1\}$ ... $b_i \leftarrow \{0,1\}$

This is exactly D_i

(2) $z \xleftarrow{R} \{0,1\}^{n+1}$



$b_1 \xleftarrow{R} \{0,1\}$ ... $b_{i+1} \xleftarrow{R} \{0,1\}$

This is exactly D_{i+1}

$$PRGAdv[B,G] = \left|Pr[B(G(s))=1 : s \xleftarrow{R} \{0,1\}^n] - Pr[B(z)=1 : z \xleftarrow{R} \{0,1\}^{n+1}]\right|$$

$$= \left|Pr[A(y)=1 : y \in D_i] - Pr[A(y)=1 : y \in D_{i+1}]\right| = |p_{i+1} - p_i| = \varepsilon(n)$$

Since, by our assumption, G is a secure PRG, the advantage $\varepsilon(n)$ must be negligible.

This completes proof of the claim. ∎

Part 3: PRG → PRF (Goldreich-Goldwasser-Micali)

Suppose we have a length-doubling PRG $G: \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$

We write $G(s) \to (s_0, s_1) = (G_0(s), G_1(s))$

Claim: G can be viewed as a secure PRF $F: \underbrace{\{0,1\}^\lambda}_{\text{key}} \times \underbrace{\{0,1\}}_{\text{single-bit domain}} \to \underbrace{\{0,1\}^\lambda}_{\text{range}}$

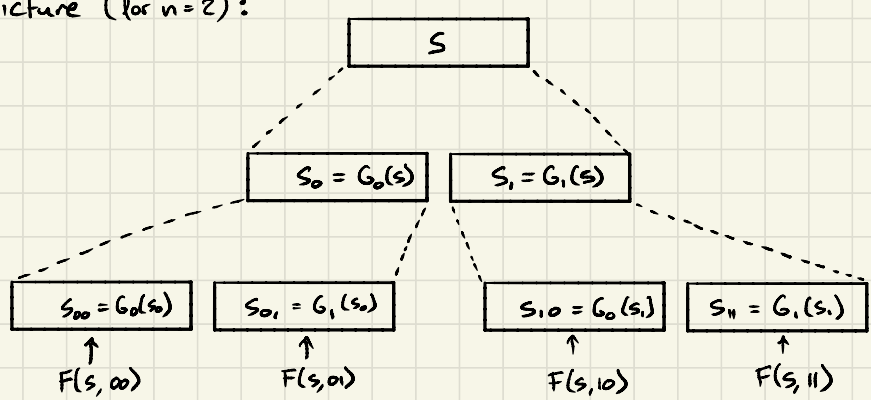
$$F(s, 0) = G_0(s) \quad , \quad F(s, 1) = G_1(s)$$

(key: s; input bit: 0 / 1)

How do we generalize this to an arbitrary domain $X_\lambda = \{0,1\}^{n(\lambda)}$?

Construction: $F(s, \overbrace{x_1 x_2 \dots x_n}^{\text{binary decomposition of input}}) = G_{x_n}(G_{x_{n-1}}(\cdots G_{x_1}(s) \cdots))$

Picture (for $n = 2$):

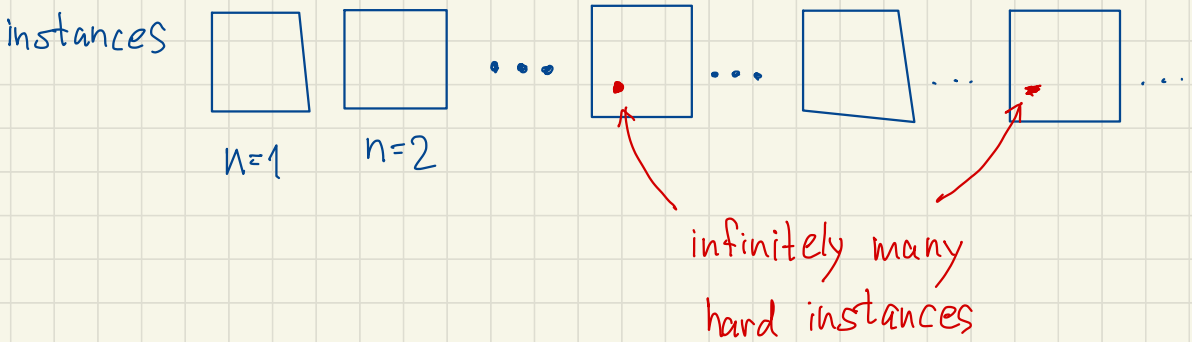


How do we prove this is a secure PRF? (See Section 4.6 of the cryptobook if interested)

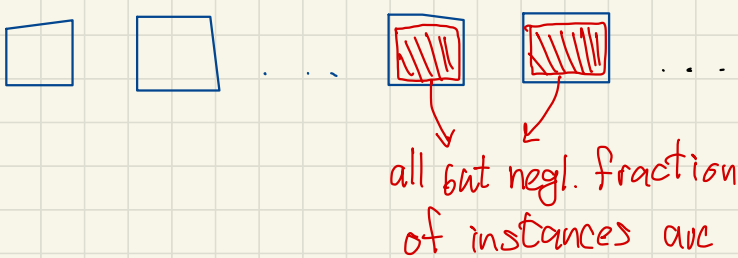
* Assume G is a secure PRG
* Show that if an adversary has non-negligible advantage on the PRF, then there exists an adversary with non-negligible advantage on the PRG G ↯ contradiction
  ↳ this step uses a hybrid argument

Worst-case vs. average-case hardness:

NP hardness:



OWF average case hardness:



Note: can define "mild" average case hardness where some poly $\varepsilon(n)$ fraction of instances are hard.

Turns out: mild OWF $\Rightarrow$ (strong) OWF ("hardness amplification").