PRGs to PRPs!

Outline

- Recap
- Game based Defs
- More Stretch PRGs (BM)
- Hybrid Args
- PRFs (GGM)
- Wrap up / LB * if time!

Recap!

$$\begin{tikzcd}
\text{OWFs} \arrow[r, "\text{Goldreich-Levin}"] & \text{PRGs (1-bit stretch)} \arrow[r, "\text{Blum-Micali}"] & \text{PRGs (poly}(\lambda)\text{ stretch)} \arrow[d, "\text{GGM}"] \\
& \text{Block ciphers} & \text{PRFs} \arrow[l, "\text{Luby-Rackoff}"']
\end{tikzcd}$$

Yesterday!

Today!

<u>Def:</u> A PRG $G: S \to R$ is a deterministic, poly-time algorithm that, given as input a seed $s \in S$ (a seed space), outputs an $r \in R$ (output space).

A PRG G is secure if for all efficient adversaries A,

$$\left| \Pr\left[A(r)=1 : \begin{array}{l} s \xleftarrow{\$} S \\ r \leftarrow G(s) \end{array} \right] - \Pr\left[A(r)=1 : r \xleftarrow{\$} R \right] \right| \leq \text{negl}(\lambda)$$

where the probability space is over the random choice of r, s and randomness of the adversary.

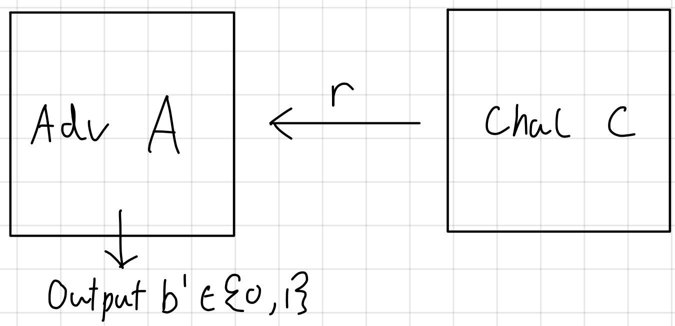
<u>Last Lecture:</u> We saw a construction of a secure PRG $G: \{0,1\}^n \to \{0,1\}^{n+1}$ ($S := \{0,1\}^n$, $R := \{0,1\}^{n+1}$) with a 1-bit stretch from a OWF using Goldreich-Levin.

<u>Today:</u> We will take a PRG $G: \{0,1\}^n \to \{0,1\}^{n+1}$ and construct a PRG $G': \{0,1\}^n \to \{0,1\}^{\ell(n)}$ with arbitrary poly stretch.

Reframing PRG Security as a Game

In the security definition above, the adversary A acts as a distinguisher between two distributions, $\{r \xleftarrow{\$} R\}$ and $\{G(s \xleftarrow{\$} S)\}$. We can equivalently reframe the security definition by treating A as an interactive algorithm which interacts with a challenger C and at the end of the interaction outputs a bit b'.

PRG Security Game



We define two experiments,

- In Experiment 0, the challenger samples $s \xleftarrow{\$} S$, $r \leftarrow G(s)$, then sends r to A.
- In Experiment 1, the challenger samples $r \xleftarrow{\$} R$, sends r to A.

For $b \in \{0,1\}$, let W_b be the event that A outputs 1 in Exp b.

The advantage A has in the PRG Security Game is

$$\mathrm{PRG}[A,G] := \left| \Pr[W_0] - \Pr[W_1] \right|$$

Where the probability space is over the random choices of challenger and A.

A PRG G is secure* if for all efficient adversaries A,

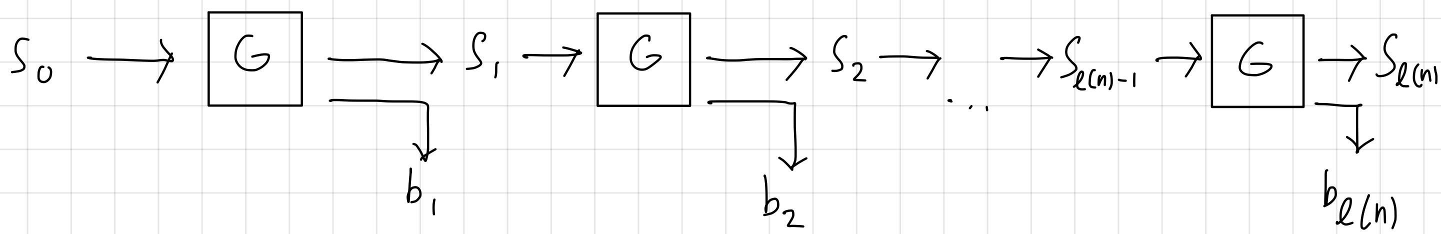
$$\mathrm{PRG}[A,G] \leq \mathrm{negl}(\lambda)$$

* Identical to the prior definition.

Building a PRG with more stretch

Let $G:\{0,1\}^n \to \{0,1\}^{n+1}$ be a secure PRG. We construct a PRG $G':\{0,1\}^n \to \{0,1\}^{\ell(n)}$, where ℓ is a poly, as follows:

☆ Idea: Let's sequentially compose PRG evaluations and output the extra bits!



$G'(s \in \{0,1\}^n)$:

$s_0 \leftarrow s$

for $i \in \{1, 2, \ldots, \ell(n)\}$:

$(s_i, b_i) \leftarrow G(s_{i-1})$

Output $b_1 b_2 \ldots b_{\ell(n)}$

Theorem: If G is a secure PRG, then G' is a secure PRG.

Lemma 1: G' is polytime.

Since G is a PRG, let $t(n)$ be the poly runtime of G. G' runtime is $\ell(n) \cdot t(n) + O(\ell(n))$. Since $\ell(n)$ is poly, G' is poly time.

Lemma 2: For every PRG Adv A that plays the PRG Security Game with respect to G', there exists a PRG adv B that plays the PRG security game with respect to G, such that

$$\text{PRGadv}[A, G'] = \ell(n) \cdot \text{PRGadv}[B, G]$$

Hybrid Arguments

Issue: informally, the definition of a PRG tells us that $G(s)$ looks random if s is random. But here we evaluate $s_i \leftarrow G(s_{i-1})$ where s_{i-1} is a PRG eval (not random). How do we leverage that G is secure PRG?

Solution: Apply the def of a PRG one execution at a time!

We define a sequence of "Hybrid Games" for an efficient PRG adversary A for G' such that each game behaves identically to the PRG security game, except that $r := b_1 b_2 \ldots b_{\ell(n)}$ is sampled differently by the challenger.

Hybrid 0 := Exp 0

$$\begin{tikzcd} s_0 \xleftarrow{\$} \{0,1\} \arrow[r] & \boxed{G} \arrow[r] \arrow[d] & s_1 \arrow[r] & \boxed{G} \arrow[r] \arrow[d] & s_2 \ldots \arrow[r] & s_{\ell(n)-1} \arrow[r] & \boxed{G} \arrow[r] \arrow[d] & s_{\ell(n)} \\ & b_1 & & b_2 & & & b_{\ell(n)} & \end{tikzcd}$$

Hybrid 1

$$\begin{tikzcd} b_1 \xleftarrow{\$} \{0,1\},\ s_1 \xleftarrow{\$} \{0,1\} \arrow[r] & \boxed{G} \arrow[r] \arrow[d] & s_2 \ldots \arrow[r] & s_{\ell(n)-1} \arrow[r] & \boxed{G} \arrow[r] \arrow[d] & s_{\ell(n)} \\ & b_2 & & & b_{\ell(n)} & \end{tikzcd}$$

⋮

Hybrid j

$$\begin{tikzcd} b_1, \ldots, b_j \xleftarrow{\$} \{0,1\}^j,\ s_j \xleftarrow{\$} \{0,1\}^{n+1} \arrow[r] & \boxed{G} \arrow[r] \arrow[d] & s_{j+1} \ldots \arrow[r] & s_{\ell(n)-1} \arrow[r] & \boxed{G} \arrow[r] \arrow[d] & s_{\ell(n)} \\ & b_{j+1} & & & b_{\ell(n)} & \end{tikzcd}$$

⋮

Hybrid $\ell(n)$:= Exp 1

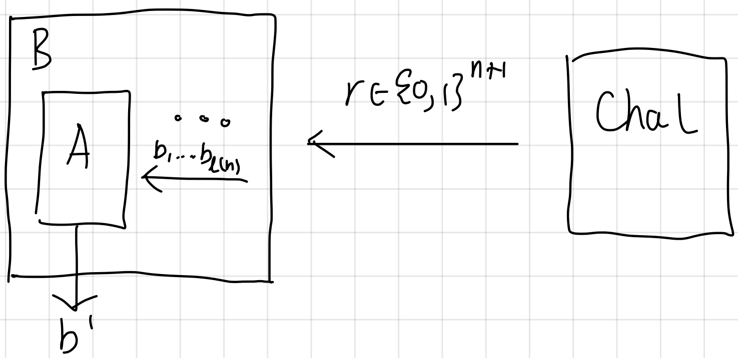
$$b_1, \ldots, b_{\ell(n)} \leftarrow \{0,1\}^{\ell(n)}$$

Define for $i \in \{0, \dots, \ell(n)\}$, p_i as the probability A outputs 1 in hybrid game i. Notice that

$$\text{PRGadv}[A, G'] := |\Pr[W_0] - \Pr[W_1]| = |p_0 - p_{\ell(n)}|$$

Construction

We will construct an efficient adv B that plays the PRG Security Game with respect to G that is a wrapper around A



B

- Recieve $r \in \{0,1\}^{n+1}$ from Challenger
- Sample $w \xleftarrow{\$} \{1, \dots, \ell(n)\}$
- Sample $b_1, \dots, b_{w-1} \xleftarrow{\$} \{0,1\}$
- Parse r as (S_w, b_w)
- for $i \in \{w+1, \ell(n)\}$

  $(S_i, b_i) \leftarrow G(S_{i-1})$
- send $b_1 b_2 \dots b_{\ell(n)}$ to A and output what A outputs.

Analysis:

Conditioned on $w = j$ for $j \in \{1, \dots, \ell(n)\}$. In Exp 0 of B's PRG Game, $r \leftarrow G(s \xleftarrow{\$} S)$. Thus, B identically simulates the challenger in Hybrid $j-1$ to A. In Exp 1 of B's PRG Game, $r \xleftarrow{\$} \{0,1\}^{n+1}$. Thus, B identically simulates the challenger in Hybrid j to A.

Therefore, $\Pr[W_0 \mid w = j] = p_{j-1}$ and $\Pr[W_1 \mid w = j] = p_j$

Thus, $\text{PRGadv}[B, G] = \left| \Pr[W_0] - \Pr[W_1] \right|$

$$= \left| \sum_{j=1}^{\ell(n)} \Pr[W_0 \mid w=j] \Pr[w=j] - \sum_{j=1}^{\ell(n)} \Pr[W_1 \mid w=j] \Pr[w=j] \right|$$

$$= \frac{1}{\ell(n)} \left| \sum_{j=1}^{\ell(n)} p_{j-1} - \sum_{j=1}^{\ell(n)} p_j \right| = \frac{1}{\ell(n)} \left| p_0 - p_{\ell(n)} \right| = \text{PRGadv}[A, G']$$

Since we assumed G is a secure PRG, $\text{PRGadv}[B, G] \leq \text{negl}(\lambda)$

$$\Rightarrow \text{PRGadv}[A, G'] \leq \underbrace{\ell(n) \cdot \text{negl}(\lambda)}_{\text{must be negl}(\lambda)}$$

Thus, G' is a secure PRG (Lemma 1 + Lemma 2) $\square$

Returning to our diagram, we have shown

$$\text{PRG with 1 bit stretch} \xrightarrow{\text{Blum-Micali}} \text{PRG with poly stretch} \xrightarrow{\text{GGM}} \text{PRF}$$

Now, we will construct PRFs from a PRGs.

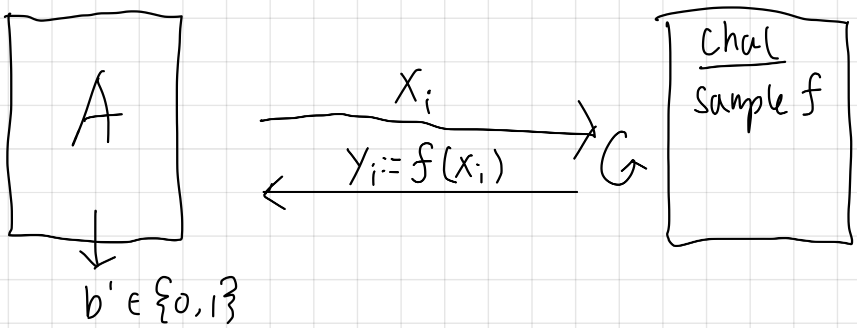
PRFs

A PRF is a deterministic algorithm $F: K \times X \to Y$ that takes as input a key $k \in K$, x in an input space X and outputs y from an output space Y.

Intuitively, for a random key k, $F(K, \cdot): X \to Y$ should "behave like" a random function from the space of functions $Func[X, Y]$.

PRF Security Game

Picture



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

- Challenger samples $f \in Func[X,Y]$ as follows
  - If $b = 0$: $k \xleftarrow{\$} K$, $f \leftarrow F(K, \cdot)$
  - If $b = 1$: $f \xleftarrow{\$} Func[X, Y]$
- For $i \in \{1, \dots, Q\}$:, a polynomial number of queries
  - Adversary sends a query $x_i \in X$ to the chal
  - Chal responds with $y_i := f(x_i)$
- Adversary outputs a bit $b' \in \{0,1\}$

Similarly, define W_b be the event that A outputs 1 in Exp b.
We define the advantage of an efficient adversary A with respect to F as

$$PRFadv[A, F] := \left| Pr[W_0] - Pr[W_1] \right|$$

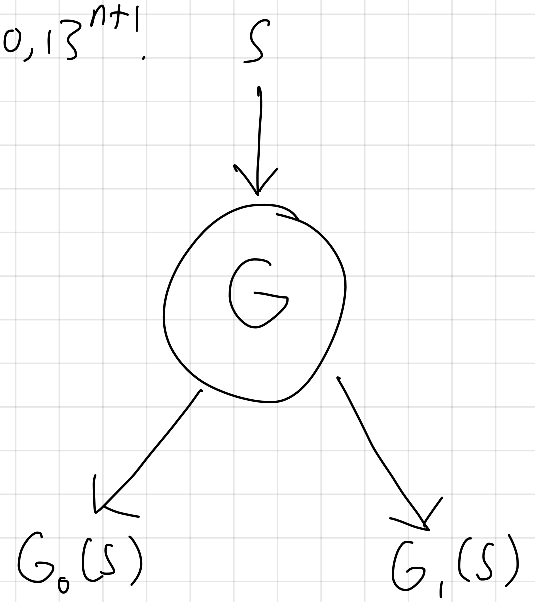
An adversary is a Q-query PRF adversary if A makes at most Q queries.

A PRF F is secure if for all efficient adv A,

$$PRFadv[A, F] \leq negl(\lambda)$$

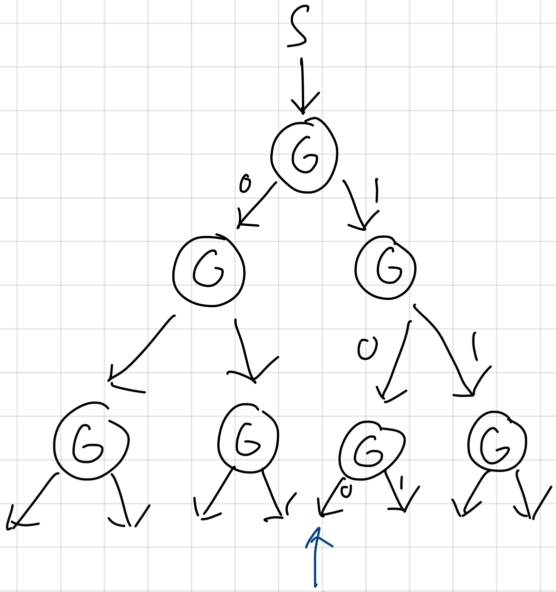
PRGs → PRFs

Here we give a construction of a PRF and provide intuition about it's security proof (see 4.6 of GCAC for more details).
We are given a PRG $G: S \to S^2$. For example, we can use the previous construction to obtain $G': \{0,1\}^n \to \{0,1\}^{2n}$ from a PRG $G: \{0,1\}^n \to \{0,1\}^{n+1}$.



Looks a node in a binary tree

We can construct a PRF $F: S \times \{0,1\}^\ell \to S$ as follows:



$F(s, 100)$

GGM Construction

Construct an evaluation tree by selectively composing the PRG evals. For an input $x := b_1 b_2 \ldots b_\ell$, evaluate the path selected by the bits.

More formally,

$F(s, b_1 \ldots b_\ell)$:

$t \leftarrow s$
for $i \in \{1, \ldots, \ell\}$:
  $t \leftarrow G_{b_i}(t)$
output t

<u>Efficiency:</u> ℓ evals of PRG G

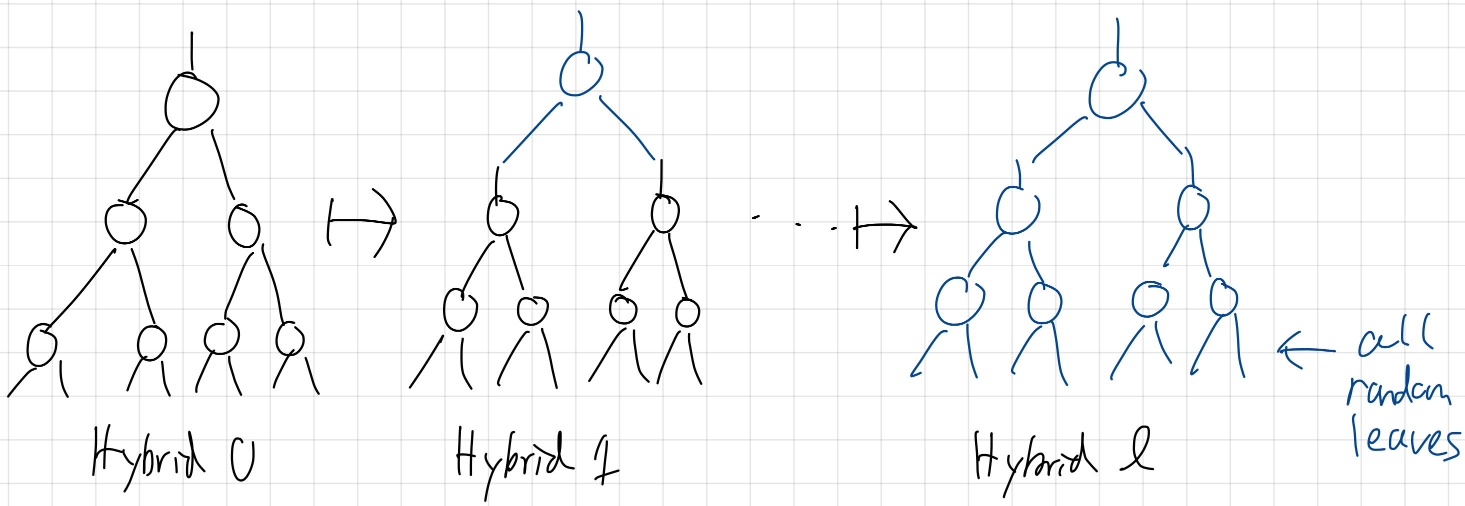
<u>Security:</u> If G is a secure PRG, then F constructed above is a secure PRF.

- for every PRF adv A, we can construct a Q-query adv B s.t.

$$\text{PRFadv}[A, F] = \ell Q \cdot \text{PRGadv}[B, G]$$

Sketch of Argument

- Given an adversary A that plays the PRF game, we want to construct an adversary B that plays the PRG game.* *with respect to the parallel PRG
- We will proceed with a Hybrid Argument:
- In the hybrid games, we can progressively replace each level of the tree with random seeds.



- The PRG adversary B will need to simulate the challenger in the PRF Game when interacting with the PRF adversary A.
- However, each level of the tree is exponentially size. So, how can B remain efficient?
- Since A is Q-query bounded, B only needs to simulate at max Q (a polynomial) number of paths of the tree!
- For more details, read the section in the book!

Symmetric Lecture Conclusion

$$\begin{tikzcd}[column sep=large]
\text{OWF} \arrow[r, bend left, "\text{GL}"] & \text{PRG}_{+1} \arrow[l, bend left, "\text{By definition}"] \arrow[r, bend left, "\text{BM}"] & \text{PRG}_{\text{tpoly}} \arrow[l, bend left, "\text{Truncate}"] \arrow[r, bend left, "\text{GGM}"] & \text{PRF} \arrow[l, bend left, "\text{CTR Mode}"] \arrow[r, bend left, "\text{Luby Rackoff / Fiestal Networks}"] & \text{PRPs / Block Ciphers} \arrow[l, bend left, "\text{Switching Lemma}"]
\end{tikzcd}$$

Just a PRF but a permutation

☆ Bonus if time

PRF → PRP/BC (LR / Fiestal Network)

- Let $F: K \times \mathcal{X} \to \mathcal{X}$ be a secure PRF where $\mathcal{X} = \{0,1\}^n$ (hence $x_1 \oplus x_2$ is a valid op)
- Define a round fxn $\phi_K : \mathcal{X}^2 \to \mathcal{X}^2$, $(a,b) \mapsto (b, a \oplus F(K,b))$
- flip: $\mathcal{X}^2 \to \mathcal{X}^2$, $(a,b) \mapsto (b,a)$

- Note ϕ_K is a permutation, $\phi_K^{-1} = \text{flip} \circ \phi_K \circ \text{flip}$

We can construct a PRP $P: K^3 \times \mathcal{X}^2 \to \mathcal{X}^2$ as follows:

$$P((K_1, K_2, K_3), \cdot) := \phi_{K_3} \circ \phi_{K_2} \circ \phi_{K_1}$$

The inverse is also eff computable given the key!

$$P^{-1}((K_1, K_2, K_3), \cdot) := \phi_{K_1}^{-1} \circ \phi_{K_2}^{-1} \circ \phi_{K_1}^{-1} = \text{flip} \circ \phi_{K_1} \circ \phi_{K_2} \circ \phi_{K_3} \circ \text{flip}$$