

lec 02: Hybrid arguments

0 Recap

1 1st security pf.

2 Stream ciphers + PRG extension
(2nd security pf)

-indistinguishability

3 Bonus (if time): PRFs

□ Recall: PRGs

$$G(s \in \{0,1\}^L) \rightarrow r \in \{0,1\}^L$$

security: $\forall$ PPT A ,

$$\text{PRGadv}[A, G] = |\Pr_r[A(r) = 1] - \Pr_s[A(G(s)) = 1]| \\ \leq \text{negl}(\lambda)$$

□ First PRG proof

If $G(s) \rightarrow r$ is a PRG, so is $G'(s) = G(s) \oplus 1^L$

pf:

Goal: G is a PRG $\rightarrow G'$ is a PRG

It suffices to show:

G' isn't a PRG $\rightarrow G$ isn't a PRG.

So, let A' be an adv. for G' with

$$\text{PRGadv}[A', G'] \neq \text{negl}$$

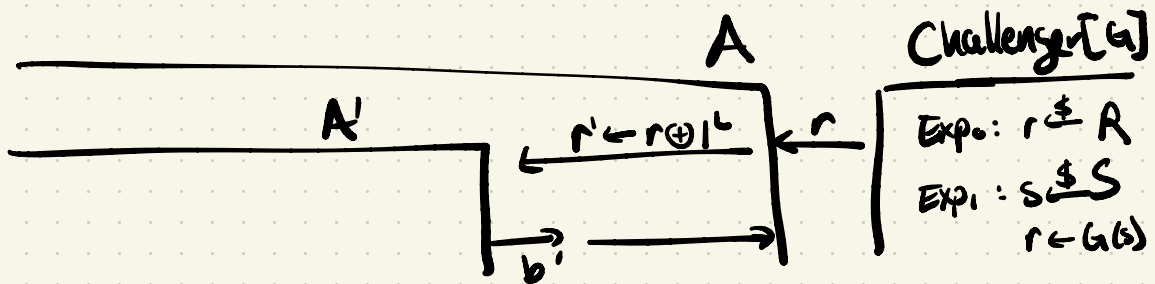
Now, we want: G isn't a PRG.

So, we define an adv. A for G

First, let's define A mathematically:

$$A(r) \triangleq A'(r \oplus 1^L)$$

Now, in pictures:



Notice that A' sees:

$$\begin{aligned} & \leftarrow r' = r \oplus 1^L, \quad r \leftarrow R \text{ (Exp. 0)} \\ & \leftarrow r' = G(s) \oplus 1^L, \quad s \leftarrow S \text{ (Exp. 1)} \end{aligned}$$

so, it sees $\text{Challenger}[G]!$

To make this more precise:

$$\Pr_{r \leftarrow R} [A(r) = 1] = \Pr_{r \leftarrow R} [A'(r \oplus 1^L) = 1] = \Pr_{r' \leftarrow R} [A'(r') = 1]$$

since r is uniform,
so is $r' = r \oplus 1^L!$

A' 's Exp_0 is coupled to A 's Exp_0 .

$$\Pr_{s \leftarrow S} [A(G(s)) = 1] = \Pr_{s \leftarrow S} [A'(G(s) \oplus 1^L) = 1] = \Pr_{s \leftarrow S} [A'(G'(s)) = 1]$$

$G'(s)$

A' 's Exp_1 is coupled to A 's Exp_1 .

$$\begin{aligned} \text{So: } \text{PRGadv}[A', G'] &= |\Pr_{r'} [A'(r') = 1] - \Pr_s [A'(G'(s)) = 1]| \\ &= |\Pr_r [A(r) = 1] - \Pr_s [A(G(s)) = 1]| \\ &= \text{PRGadv}[A, G] \\ &\neq \text{negl}. \end{aligned}$$



② Stream Ciphers + PRG extension

Problem:

PRGs are used in Stream Ciphers

$$\text{Stream Cipher: } \text{Enc}(k, m) = \overset{\text{PRG}}{\downarrow} G(k) \oplus m$$

Key size is 256 bits - 256 b

→ G 's input is $\{0, 1\}^{256}$, $l = 256$

Stream ciphers apply to messages of many sizes.

Signal messages are $\sim 1 \text{ kb}$

→ output is $L = 1024$

Netflix movies are $\sim 6 \text{ GB}$ (2hr. HD)

→ do we need a huge PRG?

→ does every app. need its own PRG?

→ No.

→ Idea: PRG extension (Blum-Micali)

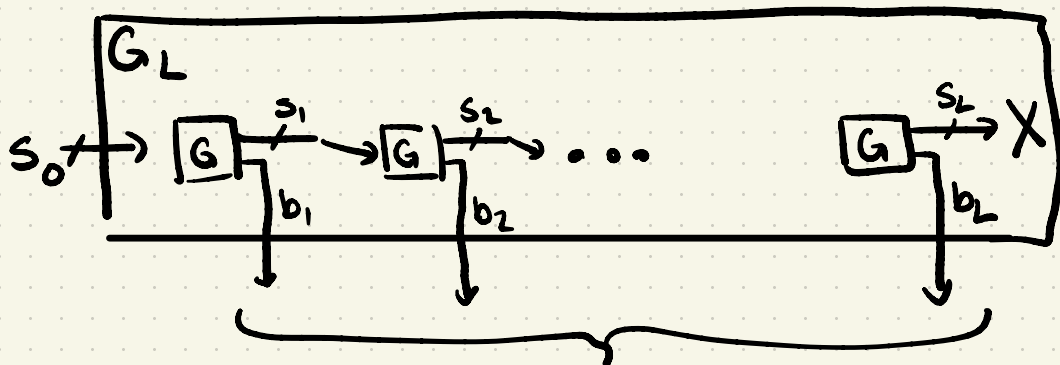
PRG extension (Blum - Micali)

Turn a small PRG $\square \rightarrow \square$ into a huge one $\square \rightarrow \boxed{}$

Let $G: \{0,1\}^L \rightarrow \{0,1\}^{L+1}$ be a PRG

$$s \in \{0,1\}^L \mapsto \boxed{G} \begin{matrix} \nearrow s' \in \{0,1\}^L \\ \rightarrow b \in \{0,1\} \end{matrix}$$

Define $G_L: \{0,1\}^L \rightarrow \{0,1\}^L$, as:
 $\uparrow$ parameter



$$G_L(s) \in \{0,1\}^L$$

We will prove that $\forall L = \text{poly}(\lambda)$, G_L is a PRG, assuming G is.

But first, a broader perspective on $\text{PRGadv}[A, G]$:

PRGadv as distributional indistinguishability.

Recall $\text{PRGadv}[A, G] =$

$$\left| \Pr_{r \in R} [A(r) = 1] - \Pr_{s \in S} [A(G(s)) = 1] \right|$$

↑
here, A gets
a sample from
uniform
distr.
on R $\rightarrow U(R)$
a.k.a.
 $\{r\}_r$

↑
here,
 $G(U(S))$
a.k.a.
 $\{G(s)\}_s$

for distributions D, D'

$$\text{Distadv}[A, D, D'] =$$

↑
"distinguishing" $\left| \Pr_{d \in D} [A(d) = 1] - \Pr_{d' \in D'} [A(d') = 1] \right|$

$$\text{so, } \text{PRGadv}[A, G] = \text{Distadv}[A, U(R), G(U(S))]$$

also, Distadv has a Δ -inequality:

$$\text{Distadv}[A, D, D''] \leq \text{Distadv}[A, D, D'] + \text{Distadv}[A, D', D'']$$

Gives 3 notions of distributional closeness:

- computational:

$$D \approx_c D' \text{ if } \forall \text{ PPT } A, \text{Distadv}[A, D, D'] \leq \text{negl}(\lambda)$$

- statistical:

$$D \approx_s D' \text{ if } \forall A, \text{Distadv}[A, D, D'] \leq \text{negl}(\lambda)$$

- perfect

$$D = D' \text{ if } \forall A, \text{Distadv}[A, D, D'] = 0$$

→ same probability for all samples

So:

$$G \text{ is a PRG} \iff U(R) \approx_c \{G(s)\}_s$$

Back to Blum-Micali:

goal: G_L is a PRG.

$$\text{pt. want: } U(\{0,1\}^L) \approx_c \{G_L(s)\}_s$$

fix a PPT A

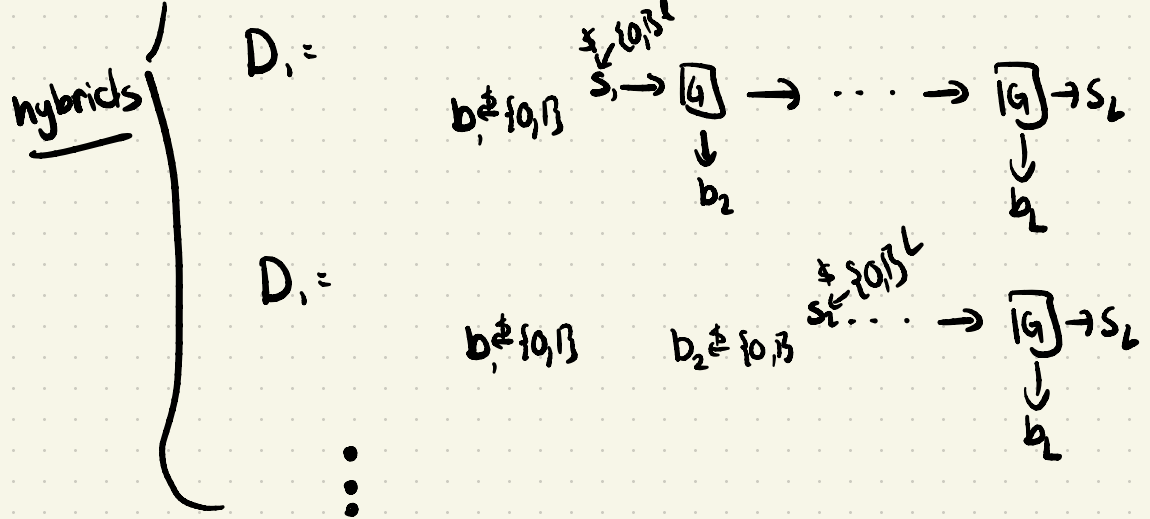
$$\text{want: } \text{PRGadv}[A, U(\{0,1\}^L), \{G(s)\}_s] \leq \text{negl}$$

intuitively, since s is random, $s \cdot 2$ are random, so $s \cdot 4$ are random... , so $s \cdot 8$ are ~ random, so all b_i are random

$$\{G(s)\}_s = D_0 = s_0 \rightarrow \boxed{G} \rightarrow s_1 \rightarrow \boxed{G} \rightarrow \dots \rightarrow \boxed{G} \rightarrow s_L$$

$\downarrow \quad \quad \downarrow \quad \quad \downarrow$
 $b_1 \quad \quad b_2 \quad \quad b_L$

in crypto, "multi-step arguments" like this are formalized as hybrid arguments.
we will now do one.



$$U(R) = D_L =$$

$$b_1 \neq \{0, \beta\} \quad b_2 \neq \{0, \beta\} \quad \dots \quad b_L \neq \{0, \beta\} \quad s_L \leftarrow \{0, \beta\}^L$$

These distributions (D_i) are hybrids between D_0 and D_L . This is a hybrid argument.

By Δ -inequality:

$$\begin{aligned}
 \text{PRGadv}[A, D_0, D_L] &\leq \text{PRGadv}[A, D_0, D_1] \\
 &\quad + \text{PRGadv}[A, D_1, D_2] \\
 &\quad + \dots \\
 &\quad + \text{PRGadv}[A, D_{L-1}, D_L]
 \end{aligned}$$

Choose hybrids s.t. D_i & D_{i+1} are "close", so that bounding $D_i \text{ v. } D_{i+1}$ is easy.

→ here, they differ by one G call

Now, we'll show $\forall i$ $\text{Dist adv}[A, D_i, D_{i+1}] \leq \text{negl}$.

→ implies $\text{Dist adv}[A, D_0, D_L] \leq \text{negl}$, if $L \leq \text{poly}$.

Suppose not, then we'll build PPT B s.t.

$\text{PRG adv}[B, G] \not\leq \text{negl}$.

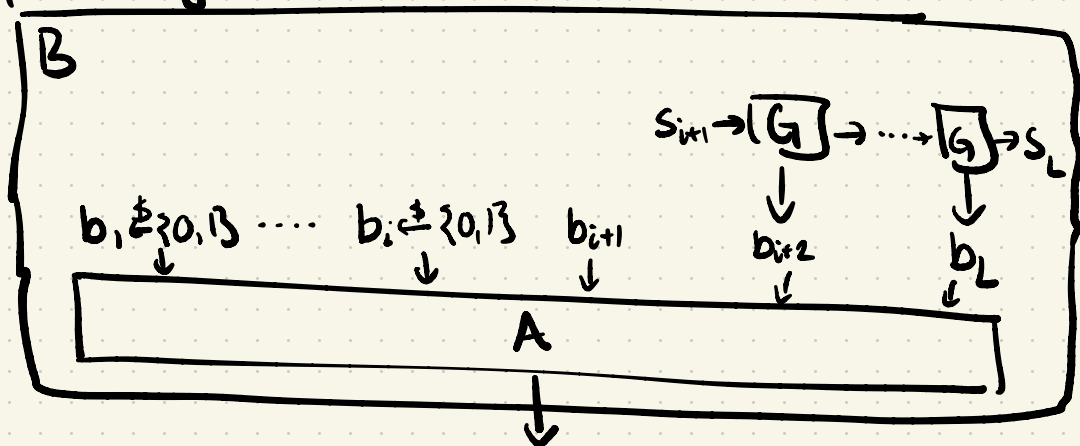
On input (s_{i+1}, b_{i+1}) , B :
← G -output or random

samples $b_1, \dots, b_i \xleftarrow{\$} \{0,1\}^{i-1}$

compute $b_{i+2}, \dots, b_L$ by daisy-chaining s_{i+1}

give $b_1, \dots, b_L$ to A and return its output.

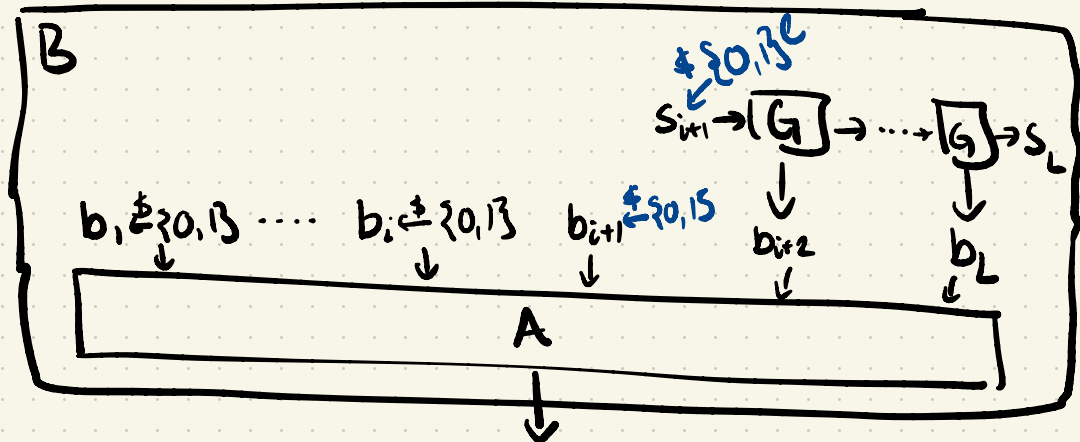
pictorially:



Two experiments that B could be in:

$(b_{i+1}, s_{i+1}) \xleftarrow{\$} \{0,1\}^{L+1}$

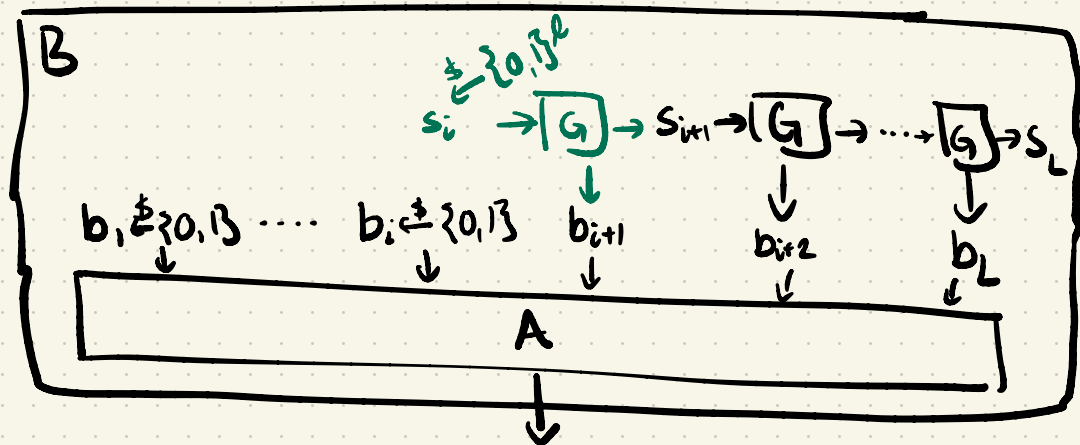
(Expo)



$\Rightarrow A$ gets a sample from D_{i+1}

$\Rightarrow B(\text{random}) \sim A(D_{i+1})$

Or, $(b_{i+1}, s_{i+1}) = G(s_i), \quad s_i \in \{0,1\}^l \quad (\text{Exp}_1)$



$\Rightarrow A$ gets a sample from D_i

$\Rightarrow B(G(s)) \sim A(D_i)$

$$\text{PRGadv}[B, G]$$

$$= | \Pr_r [B(s) = 1] - \Pr_s [B(G(s)) = 1] |$$

$$= | \Pr_{d \in D_i} [A(d) = 1] - \Pr_{d' \in D_{i+1}} [A(d') = 1] |$$

$$= \text{Distadv}[A, D_i, D_{i+1}]$$

$$\neq \text{negl}$$

so, B has non-negl. advantage against G .

$\Rightarrow \Leftarrow$, since G is a PRG.

$$\text{so, } \text{Distadv}[A, D_i, D_{i+1}] \leq \text{negl}$$

$$\text{so, } \text{Distadv}[A, D_0, D_L] \leq \text{negl}$$

$$\text{so, } \text{PRGadv}[A, G_L] \leq \text{negl}$$



Other uses of hybrid arguments

(Let G, G' be PRGs)

Fact: $G'(G(s))$ is a PRG.

Fix A .

Want $\{r\}_r \approx_{c,A} \{G'(G(s))\}_s$

use: $\{r\}_r \approx_{c,A} \{G'(s')\}_{s'}$

← shorthand $\text{DistAdv}[A, -, \cdot] \leq \epsilon$

Fact: $(s, s') \mapsto (G(s), G'(s'))$

is a PRG

$\Rightarrow$ proving this is a
homework problem. ü

Bonus: Pseudorandom Functions (PRFs) from PRGs.

defⁿ PRF : $f(k, x) \rightarrow y$
 $\uparrow$ key $\uparrow$ input $\uparrow$ output.

security (informal): No PPT A can distinguish between having an oracle to

$\{f(k, \cdot) \rightarrow \cdot\}_{k \in K}$ or $\{f(\cdot) \rightarrow \cdot\}_{f \in F_n[X, Y]}$

$\uparrow$
 f is fixed,
 but k is random
 $\log |K|$ random bits.

$\uparrow$
 random function
 (each output is UID)

x	y
0000	
0001	
0010	
...	
1111	

$\uparrow$
 all of these
 are
 random

$\log |Y|$
 rand. bits

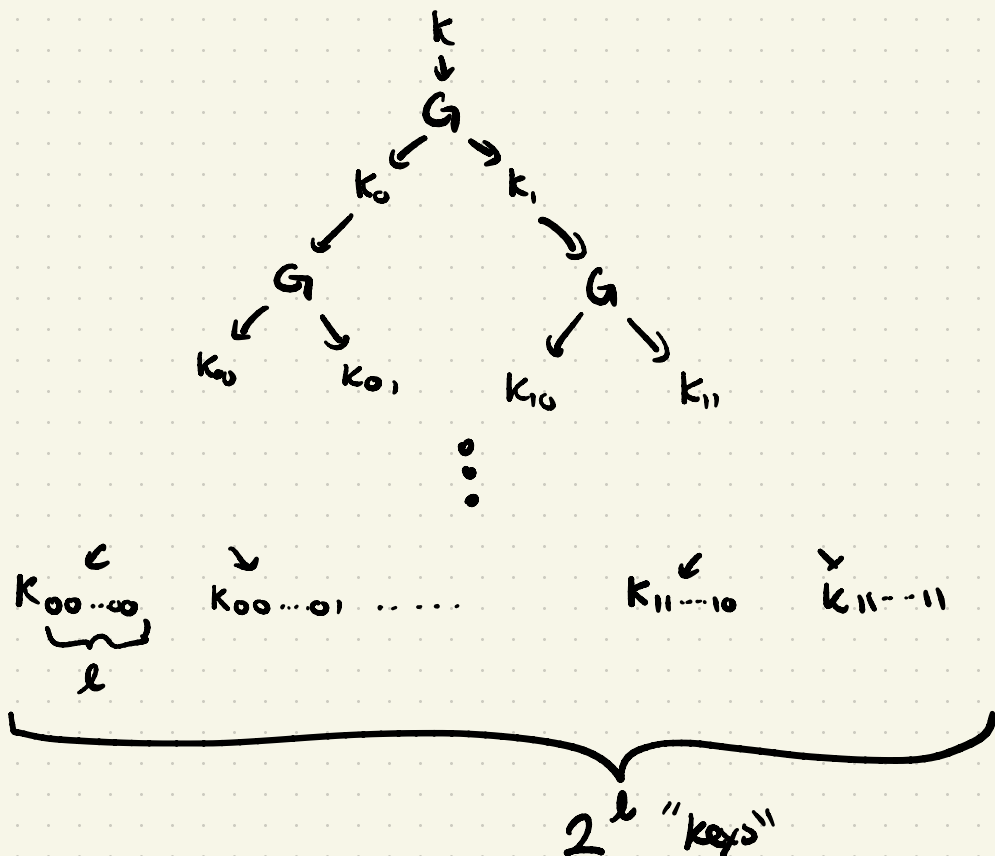
note A is PPT,
 so it gets poly(λ)
 queries.

Building a PRF from a PRG: [GGM'84]

Let $G(s \in \{0, 1\}^\lambda) \rightarrow (r_0, r_1) \in \{0, 1\}^{2\lambda}$ be a PRG

$f_{\text{GGM}}(k \in \{0, 1\}^\lambda, x \in \{0, 1\}^L) \rightarrow y \in \{0, 1\}^\lambda$

is defined by a tree

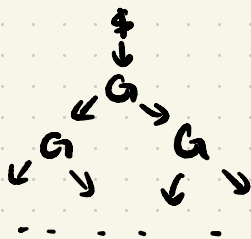


these will be the PRF's outputs

$$f_{\text{GGM}}(k, x \in \{0,1\}^l) \triangleq k_x$$

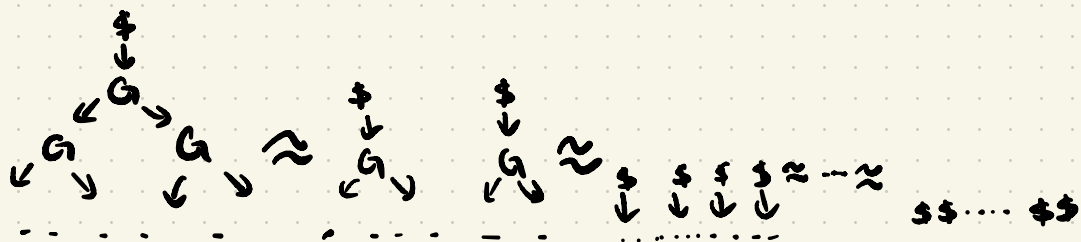
Q: how to prove f_{GGM} is a PRF?

Goal: $\circ \approx \circ$



$\$ \ \$ \ \dots \ \$ \$$

[Failed] approach: layer-by-layer hybrids



fails if $\ell = \Omega(L)$, since there are $(2^L - 1)$ G's.

$\Rightarrow$ Recall $\approx_c$ is only transitive for poly(1) steps!

better approach: adversary A can only query f_{GGM} $\text{poly}(n)$ times

$\Rightarrow$ only replace G's along the paths to A's queries.

$$\# \text{ replacements} \leq \underbrace{\text{poly}(\lambda)}_{\text{query count}} \cdot \underbrace{\ell}_{\text{path depth}}$$

also poly(λ)

