Pseudorandom Generators

Example PRG:

$s \in \{0,1\}^{128} \longrightarrow AES(s, 000\cdots0) \,\|\, AES(s, 00\cdots01)$

128 bits

128 in

128 | 128

= 256 out

- public parameters p prime, g "special" elt of $\mathbb{Z}_p$

input:

$a, b \in \{0, \ldots, p-1\} \longrightarrow g^a \bmod p,\ g^b \bmod p,\ g^{a \cdot b} \bmod p$

Interchangeable vs. Indistinguishable

reading:

QUERY(): $s \leftarrow \{0,1\}^\lambda$ ret $G(s)$	$\approx$	QUERY(): $z \leftarrow \{0,1\}^{\lambda+l}$ ret z

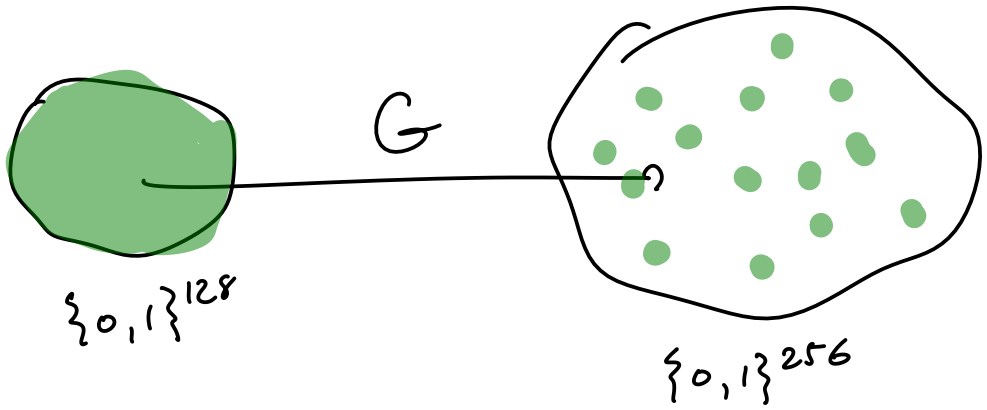
Can these libs be $\equiv$? NO

let's write distinguisher

If calling program runs in poly time, then its advantage can't be very high (negligible)

So, let's write a calling program that runs in exponential time

Obs:



(only 2^{128} green points)

Calling program:

call QUERY(), 2^{256} times
if # of distinct outputs is $\leq 2^{128}$, return 1

A:

Calling program 2:

```
call c = QUERY()
for all s ∈ {0,1}^128
    if c = G(s) return 1
return 0
```

In presence of $\mathcal{L}_{prg\text{-}real}$: $Pr[\text{output } 1] = 1$

(A has "green dot", enumerates all green dots)

In presence of $\mathcal{L}_{prg\text{-}rand}$: $Pr[\text{output } 1] = 1/2^{128}$

($Pr[\text{getting green dot, choosing uniformly}] = \frac{2^{128}}{2^{256}}$)

Can also write exponential-time distinguisher for