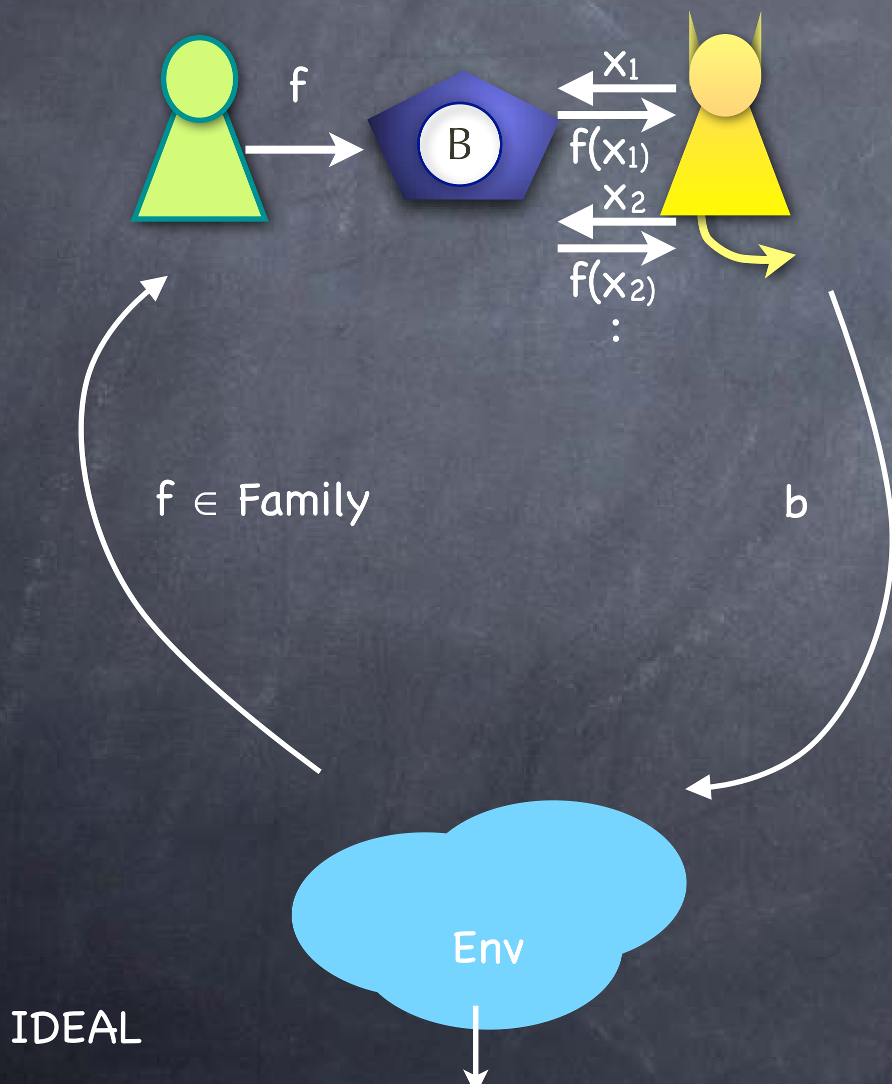
Obfuscation

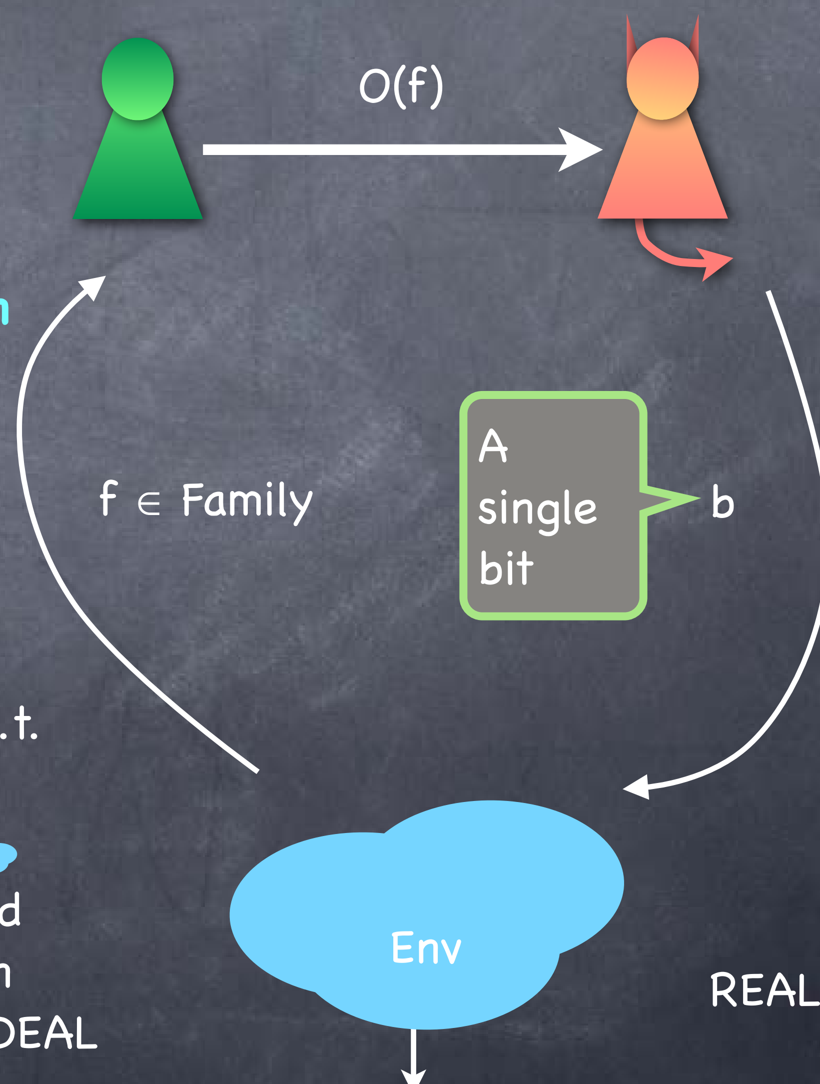
Lecture 26
Different Flavours

VBB Obfuscation

Note: Considers only corrupt receiver



Virtual Black-Box (VBB) Obfuscation



Secure (and correct) if:

∀ PPT

∃ PPT s.t.

∀ PPT

output of

is distributed identically in REAL and IDEAL

Flavours of Obfuscation

VBB Obf.

Adaptive DIO

Differing Inputs Obf.

PC Differing Inputs Obf.

Indistinguishability Obf.

XIO

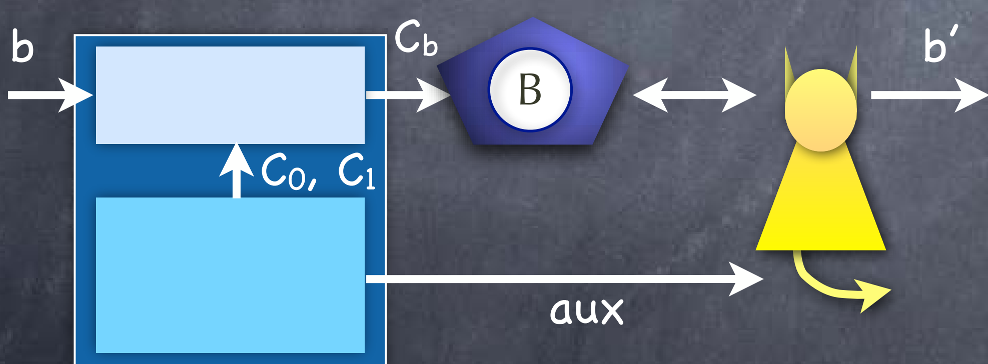
VGB Obf.

IND-PRE Security

Different variants of the definition in this framework

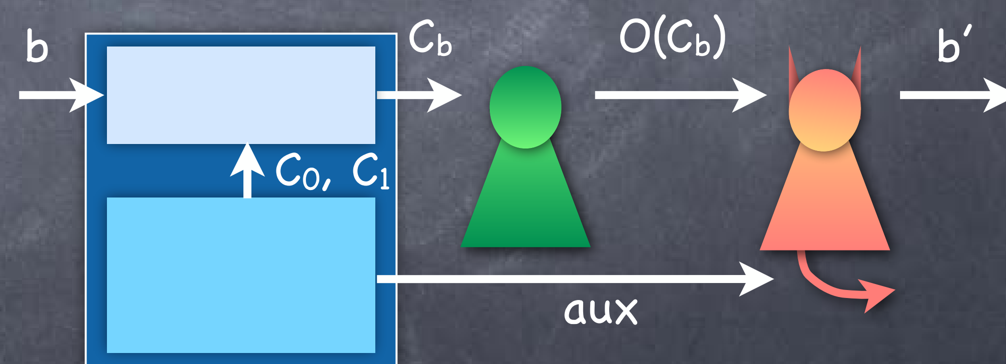
is IDEAL-Hiding if

∀ PPT Pr[b′=b] = ½ ± negl.



is REAL-Hiding if

∀ PPT Pr[b′=b] = ½ ± negl.



IND-PRE secure if ∀ PPT in <u>Test-Family</u>

IDEAL-hiding ⇒ REAL-hiding

IDEAL

REAL

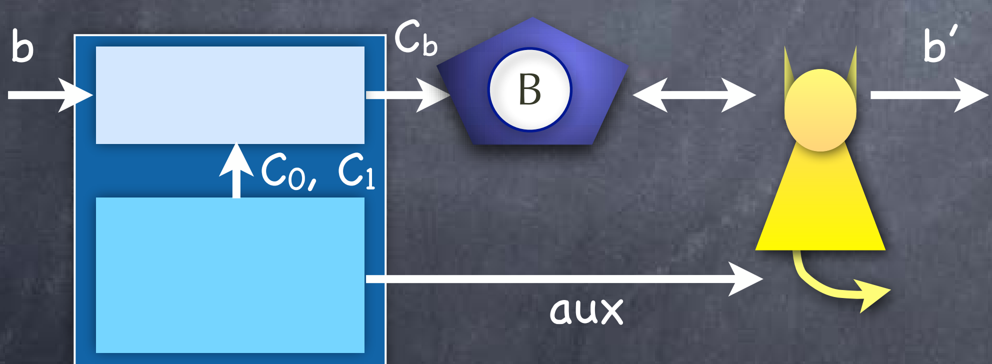
Indistinguishability Obf. (iO)

Test picks functionally equivalent C_0, C_1 (hardwired into it)

Guaranteed to be IDEAL-hiding

is IDEAL-Hiding if

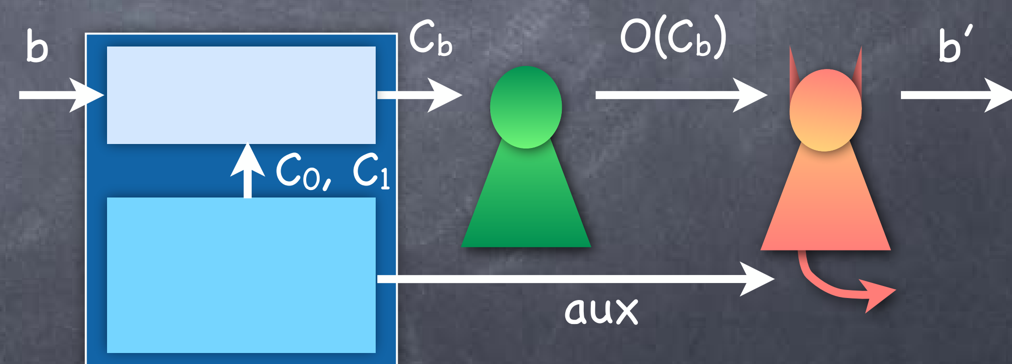
$\forall$ PPT Pr[b′=b] = ½ ± negl.



IDEAL

is REAL-Hiding if

$\forall$ PPT Pr[b′=b] = ½ ± negl.



REAL

iO if $\forall$ PPT in Test-Family

IDEAL-hiding $\Rightarrow$ REAL-hiding

Inefficient iO

XIO: Allows inefficient evaluation, slightly better than truth table

- Write down the truth table of the function? But evaluation not efficient.
- Better solution: Find a canonical circuit for the given circuit (e.g., smallest, lexicographically first)
- Meets every requirement except that of the obfuscator being efficient
- Fact: Can find the canonical circuit in polynomial time if P=NP
  - i.e., P=NP $\Rightarrow$ iO (with efficient obfuscator) exists
  - Cannot rule out the possibility that iO exists but there is no OWF (say), unless we prove P≠NP

Best-Possible Obfuscation

- iO as good at hiding information as any obfuscation
- (aux,iO(O(P))) ≈ (aux,iO(P)), where O is <u>any</u> compiler that perfectly preserves functionality
  - i.e., Any information that can be efficiently learned from (aux,iO(P)) can be efficiently learned from (aux,iO(O(P)))
    - In turn, efficiently learned from (aux,O(P))
  - Note: Only holds when iO is efficient (so not applicable to the canonical encoding construction)

Is iO Any Good?

- iO does not promise to hide anything about the function (only its representation)
- Can we use iO in cryptographic constructions?
  - Yes (combined with other cryptographic primitives)
  - e.g. PKE from SKE using iO
  - In fact, can get FE (from PKE and NIZK) using iO
    - Recent results: iO "essentially" equivalent to FE for general functions (note: FE doesn't hide function)

With different levels of security

Is iO Any Good?

- PKE from SKE using iO
  - Recall SKE: $\mathrm{Enc}(m) = (\, r, \mathrm{PRF}_K(r) \oplus m \,)$
  - Using obfuscation: $PK = O(\mathrm{PRF}_K(\,\cdot\,))$?
    - But the same key allows decryption also!
    - Need the obfuscated program to carry out the entire encryption, including picking the randomness
      - Or at least, should not allow full freedom in choosing r
    - $PK = O(\, f_K(\cdot))$ where $f_K(s,m) = (\mathrm{PRG}(s), \mathrm{PRF}_K(\mathrm{PRG}(s)) \oplus m)$
    - Problem when using iO: iO may not hide K!

Is iO Any Good?

- PKE from SKE using iO
  - PK = iO($f_K(\cdot)$) where $f_K(s,m) = (PRG(s), PRF_K(PRG(s)) \oplus m)$
  - Problem using iO: iO may not hide K!
  - But the functionality of f_K depends only on PRF_K evaluated on the range of PRG. So it is plausible that there are alternate representations of f_K that does not reveal K fully
  - Idea: Imagine challenge ciphertext is $(r, PRF_K(r) \oplus m)$ where r is <u>not</u> in the range of PRG!
    - Cannot tell the difference by security of PRG
    - Revealing functionality f_K need not reveal $PRF_K(r)$

Is iO Any Good?

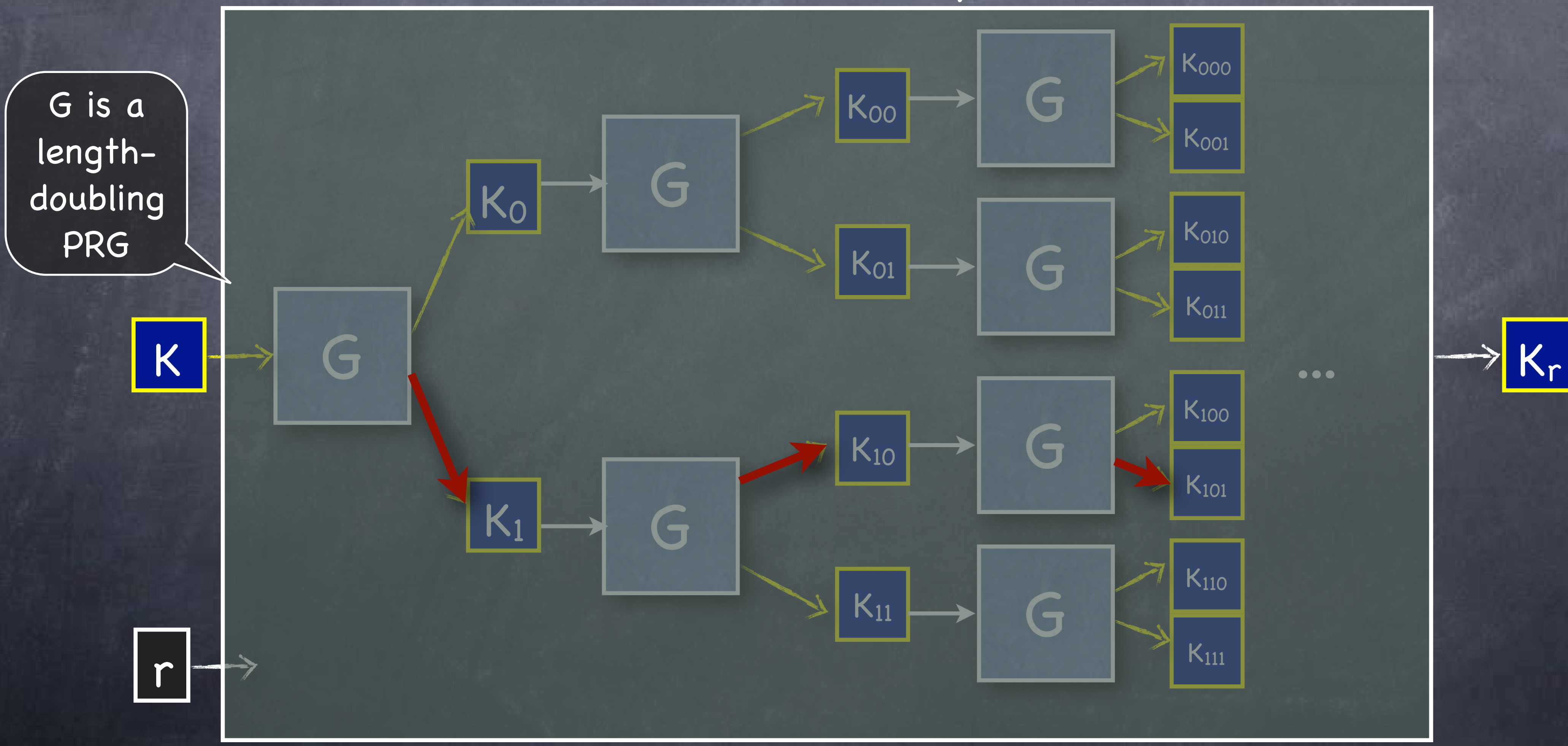
Punctured PRF used only in proof

By modifying the standard construction

- PKE from SKE using iO
  - PK = iO($f_K(\cdot)$) where $f_K(s,m) = (PRG(s), PRF_K(PRG(s)) \oplus m)$
  - Idea: Imagine challenge ciphertext is CT′ = (r, $PRF_K(r) \oplus m$) where r is not in the range of PRG!
    - Cannot tell the difference with real CT by security of PRG
  - Punctured PRF: Key $K^{\bar{r}}$ to evaluate PRF_K on inputs other than r, such that $PRF_K(r)$ is pseudorandom given $K^{\bar{r}}$
  - $f'_{K^{\bar{r}}}(s,m) = (PRG(s), PRF'_{K^{\bar{r}}}(PRG(s)) \oplus m)$, is functionally equivalent to f_K, where PRF′ is the PRF punctured at input r
  - Let PK′ = iO($f'_{K^{\bar{r}}}(\cdot)$). Then (CT,PK) ≈ (CT′,PK′)
    - (CT′,PK′) completely hides m, even if PK′ revealed all of $K^{\bar{r}}$

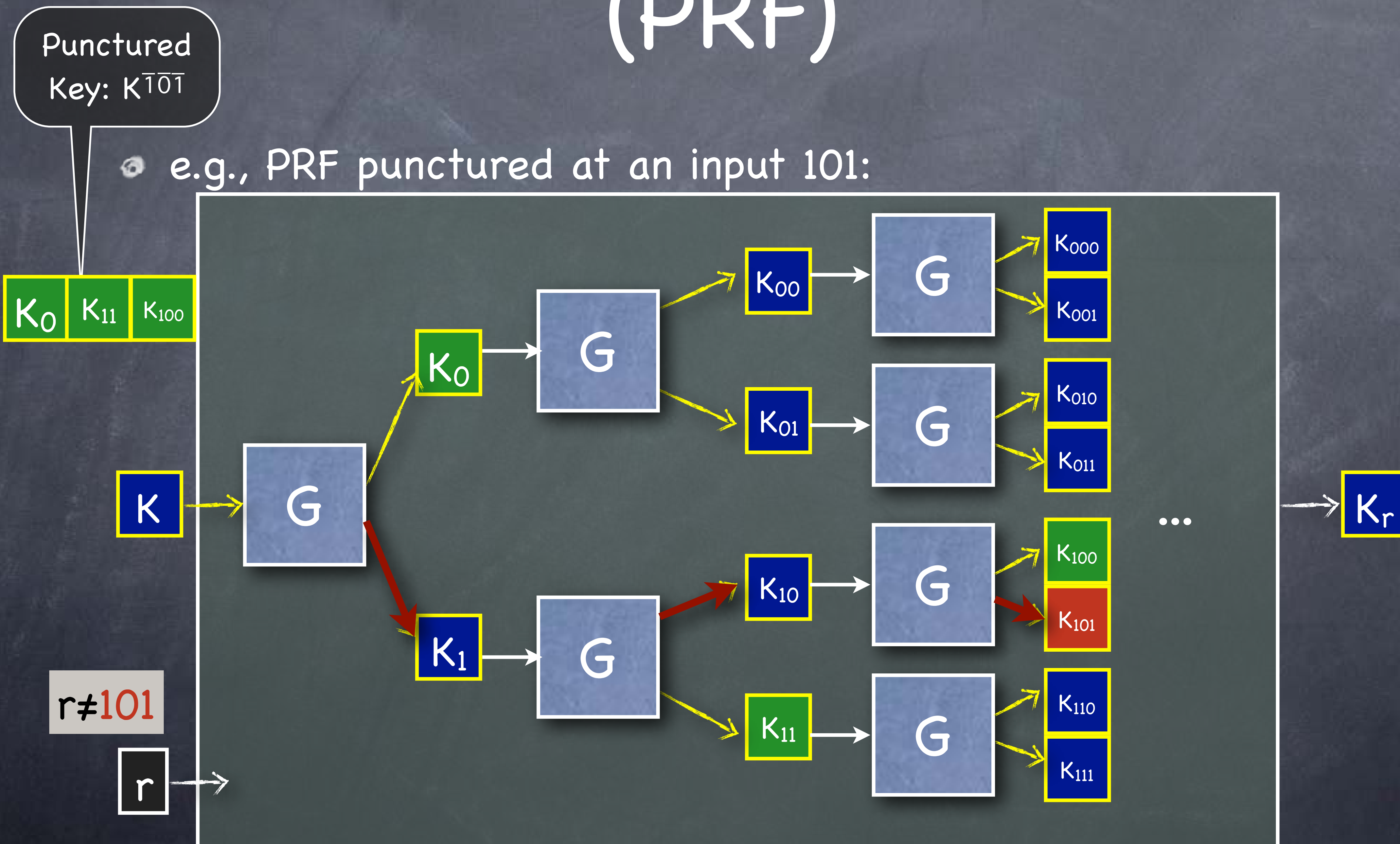
Pseudorandom Function (PRF)

- A PRF can be constructed from any PRG



Pseudorandom Function (PRF)

- e.g., PRF punctured at an input 101:



Constructing IO

- Last lecture: iO from (idealized) multi-linear maps
  - State-of-the-art: Can base on L-linear maps under assumptions in the standard model, for L as low as 3
    - Result does not extend to basing iO on bilinear maps
  - Exploits connections with Functional Encryption
- iO is quite useful if we can construct it
  - But stronger obfuscation would be even more powerful

Differing Input Obf.

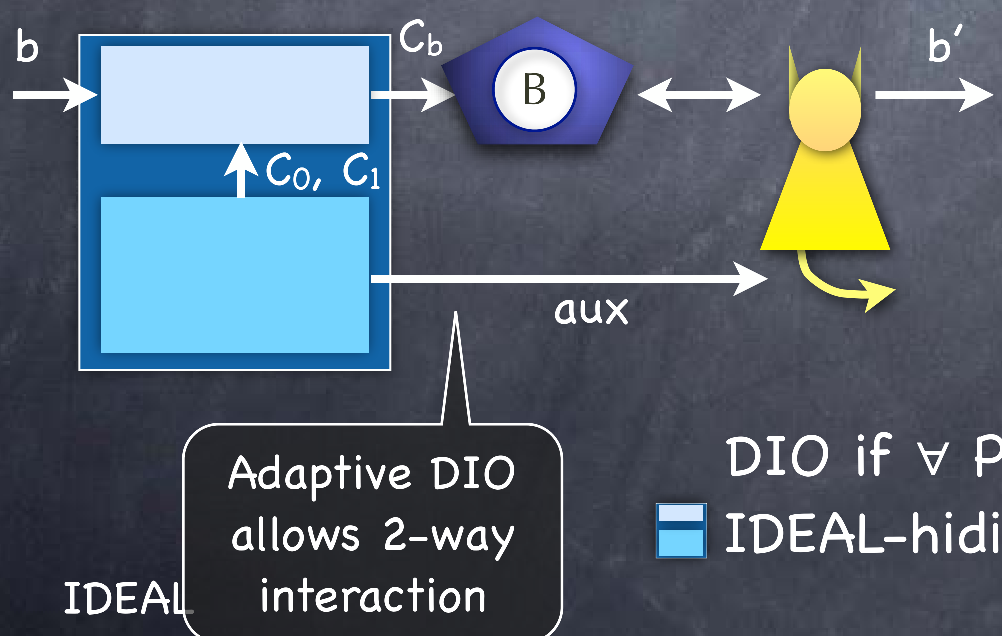
Any PPT Test that includes (C_0,C_1) in aux

C_0, C_1 need not be functionally equivalent

To be not IDEAL-hiding, need a PPT which can find a "differing input"

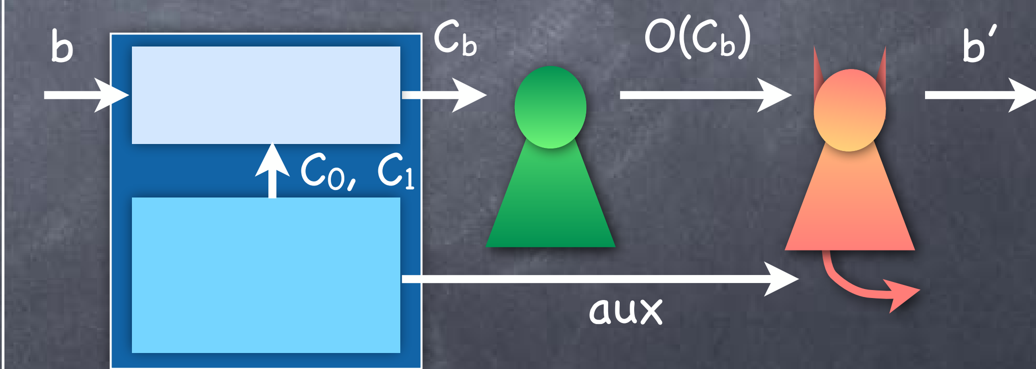
is IDEAL-Hiding if

∀ PPT Pr[b'=b] = ½ ± negl.



is REAL-Hiding if

∀ PPT Pr[b'=b] = ½ ± negl.



DIO if ∀ PPT in Test-Family

IDEAL-hiding ⇒ REAL-hiding

REAL

Implausibility of DIO?

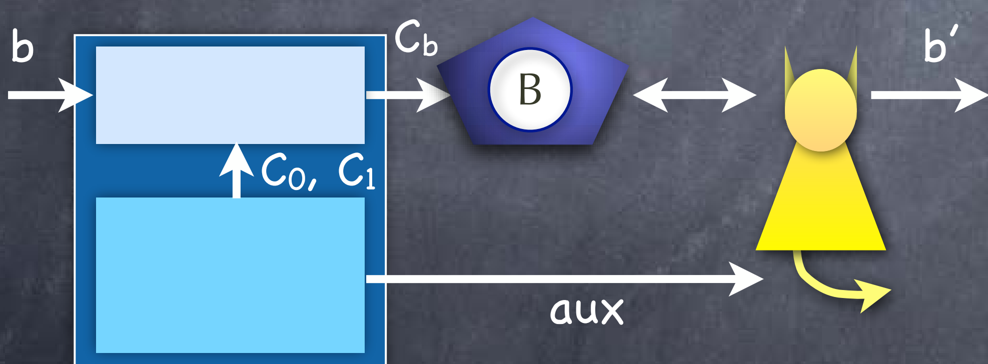
- Is DIO (im)possible?
- Open
- Constructions from multi-linear maps under strong (or idealized) assumptions
- Implausibility results
  - If highly secure ("sub-exponentially secure") one-way functions exist, then highly secure DIO for Turing machines cannot exist!
- Problem is the auxiliary information
  - Let aux be an obfuscated program which can extract secrets from the obfuscated program. But in the ideal world, aux would be useless (as it is obfuscated).

Public-Coin DIO

Test as in DIO, but aux includes all the randomness used by Test

is IDEAL-Hiding if

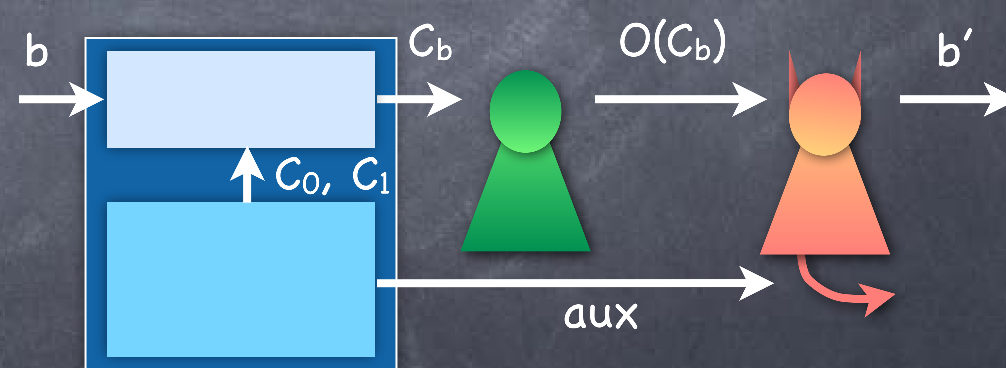
$\forall$ PPT Pr[b′=b] = ½ ± negl.



IDEAL

is REAL-Hiding if

$\forall$ PPT Pr[b′=b] = ½ ± negl.



REAL

PC-DIO if $\forall$ PPT in <u>Test-Family</u>

IDEAL-hiding $\Rightarrow$ REAL-hiding

Virtual Grey Box Obf.

Arbitrary PPT Test, with arbitrary aux (C_0, C_1 not given).
Allow computationally unbounded adversaries in the ideal world.

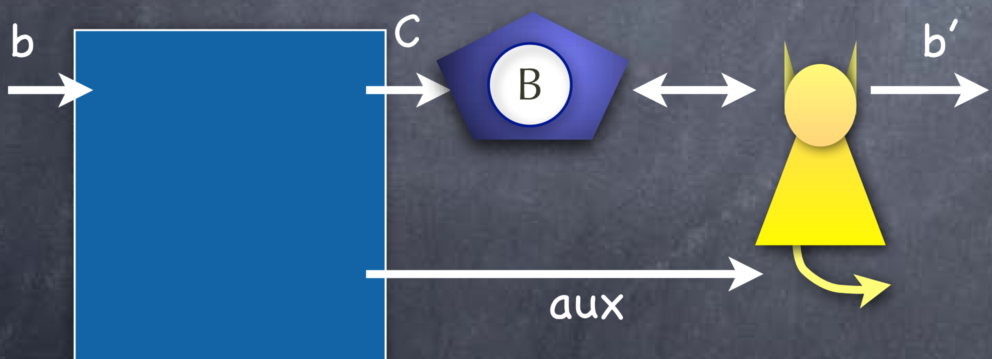
Original definition is simulation-based a la VBB Obfuscation

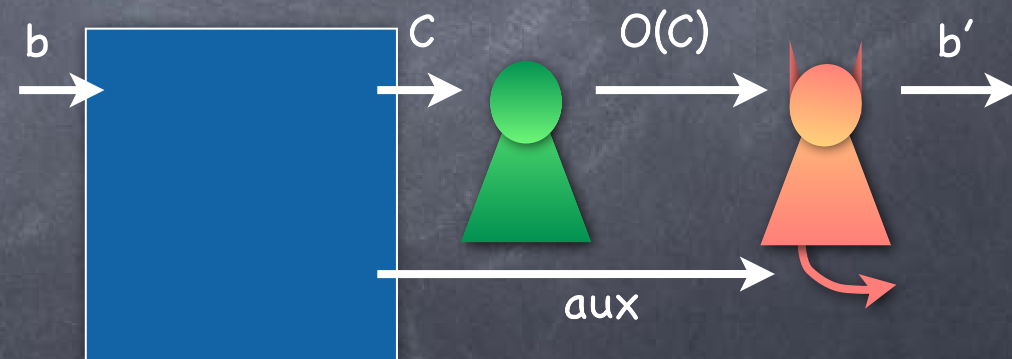
is IDEAL-Hiding if

∀ Pr[b′=b] = ½ ± negl.

is REAL-Hiding if

∀ PPT Pr[b′=b] = ½ ± negl.





VGB Obf. if ∀ PPT in <u>Test-Family</u>

IDEAL-hiding ⇒ REAL-hiding

IDEAL

REAL