CIS 5560

Cryptography
Lecture 9

Course website:

pratyushmishra.com/classes/cis-5560-s25/

Slides adapted from Dan Boneh and Vinod Vaikuntanathan

Announcements

- **HW 3 due next Friday**
- **HW2 due tomorrow!**

Recap of last lecture

Pseudorandom Functions

Collection of functions $\mathcal{F}_\ell = \{F_k : \{0,1\}^\ell \to \{0,1\}^m\}_{k \in \{0,1\}^n}$

- indexed by a key k
- n: key length, ℓ: input length, m: output length.
- Independent parameters, all poly(sec-param) = poly(n)
- #functions in $\mathcal{F}_\ell \leq 2^n$ (singly exponential in n)

$\mathbf{Gen}(1^n)$: Generate a random n-bit key k.

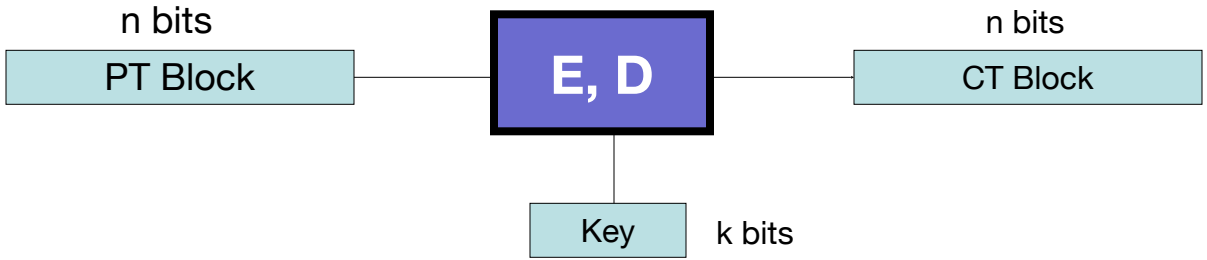
$\mathbf{Eval}(k, x)$ is a poly-time algorithm that outputs $F_k(x)$

Security: Cannot distinguish from random function

$$\left| \Pr\left[A^{f_k}(1^n) = 1 \mid k \leftarrow \{0,1\}^{\ell}\right] - \Pr\left[A^{F}(1^n) = 1 \mid F \leftarrow \mathsf{Fns}\right] \right| \leq \mathsf{negl}(n)\,.$$

PRP/Block Cipher

A **block cipher** is a pair of efficient algs. (E, D):

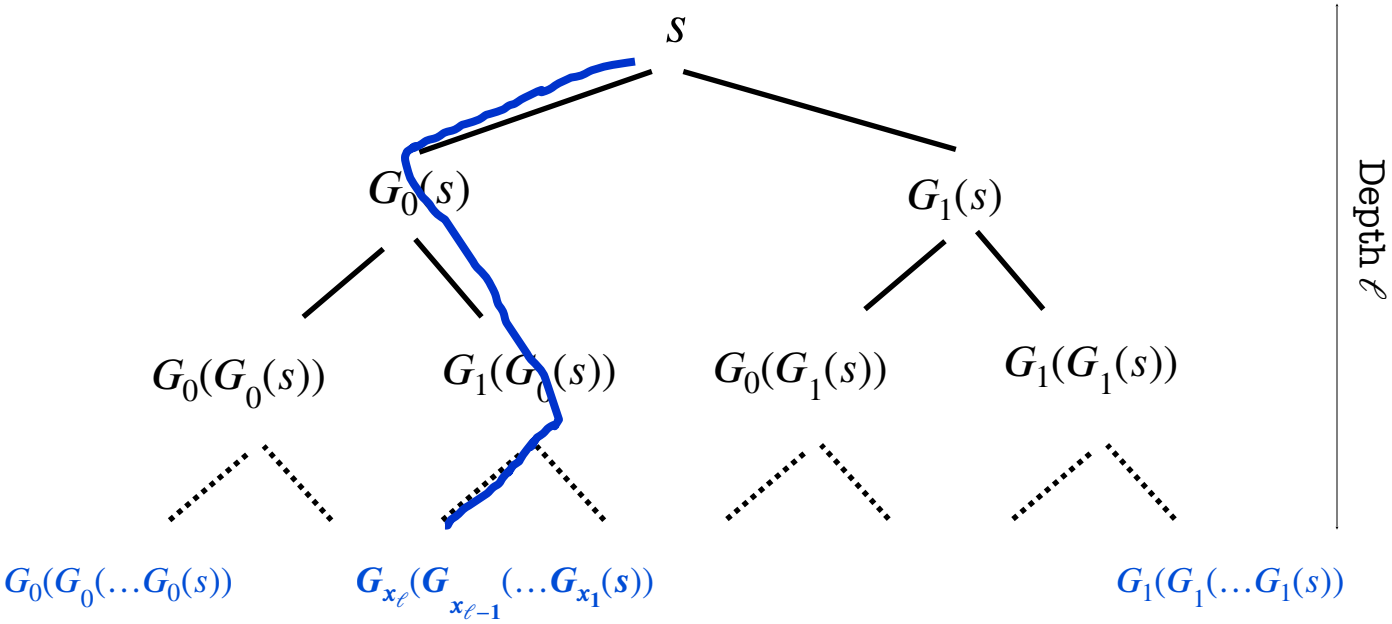


Canonical examples:

1. **AES**: n=128 bits, k = 128, 192, 256 bits
2. **3DES**: n= 64 bits, k = 168 bits (historical)

Goldreich-Goldwasser-Micali PRF

Construction: Let G(s) = $G_0(s)||G_1(s)$ where $G_0(s)$ and $G_1(s)$ are both n bits each.



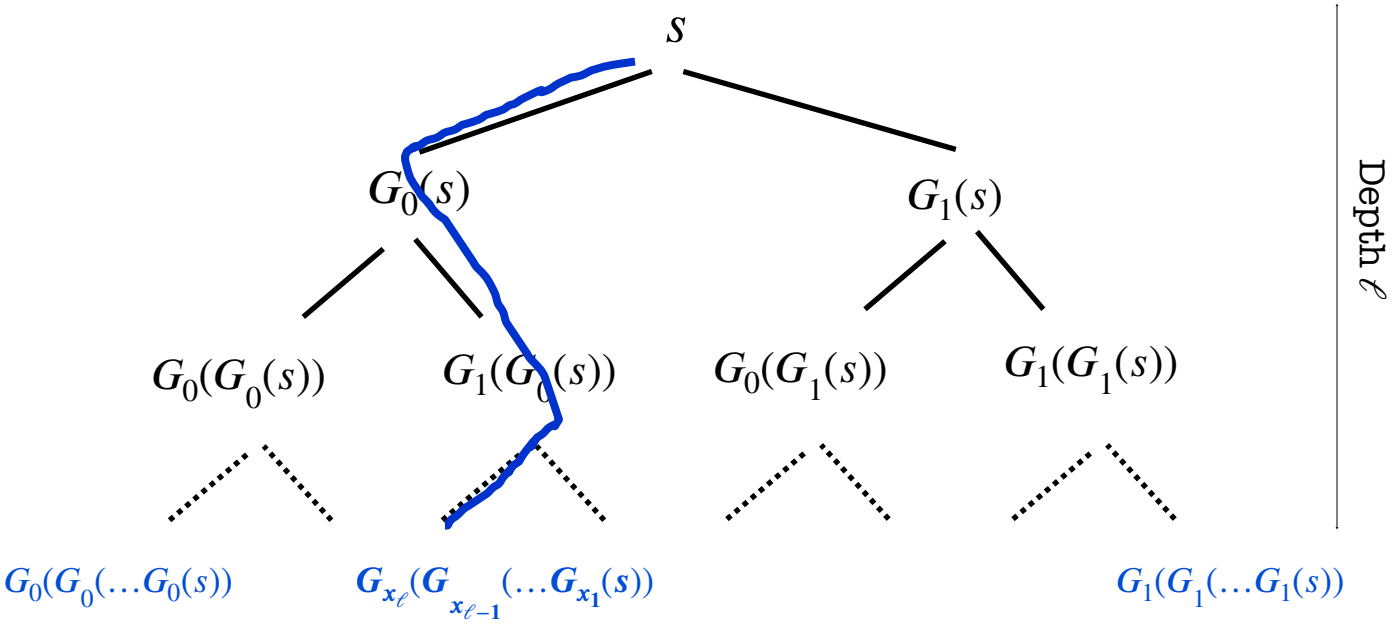
Each path/leaf labeled by $x \in \{0,1\}^\ell$ corresponds to $f_s(x)$.

Today's Lecture

- Proof of security for MAC
- Short MAC → Long MACs

Goldreich-Goldwasser-Micali PRF

Construction: Let G(s) = $G_0(s)||G_1(s)$ where $G_0(s)$ and $G_1(s)$ are both n bits each.



Each path/leaf labeled by $x \in \{0,1\}^\ell$ corresponds to $f_s(x)$.

Goldreich-Goldwasser-Micali PRF

Construction: Let G(s) = $G_0(s)||G_1(s)$ where $G_0(s)$ and $G_1(s)$ are both n bits each.

The pseudorandom function family $\mathcal{F}_\ell$ is defined by a collection of functions f_s where:

$$f_s(x_1 x_2 \ldots x_\ell) = \boldsymbol{G}_{x_\ell}(\underbrace{\boldsymbol{G}_{x_{\ell-1}}(\ldots \boldsymbol{G}_{x_1}(s))}_{\ell\text{-bit input}})$$

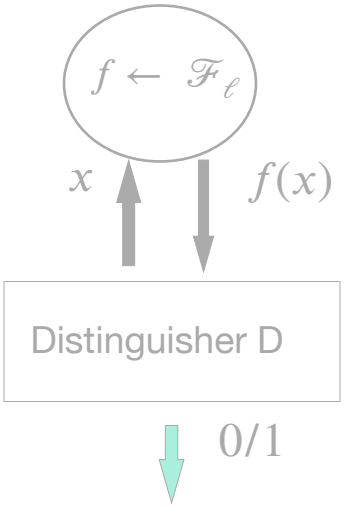
- f_s defines 2^ℓ pseudorandom bits.
- The x^{th} bit can be computed using ℓ evaluations of the PRG G (as opposed to $x \approx 2^\ell$ evaluations as before.)

GGM PRF: Proof of Security

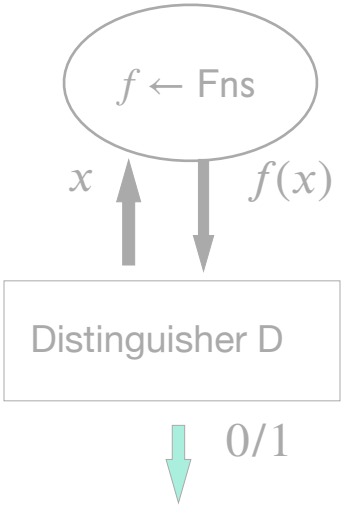
By contradiction. Assume there is a ppt D and a poly function p s.t.

$$\left| \Pr\left[A^{f_k}(1^n) = 1 \mid k \leftarrow \{0,1\}^\ell\right] - \Pr\left[A^F(1^n) = 1 \mid F \leftarrow \text{Fns}\right] \right| \geq 1/p(n)\,.$$

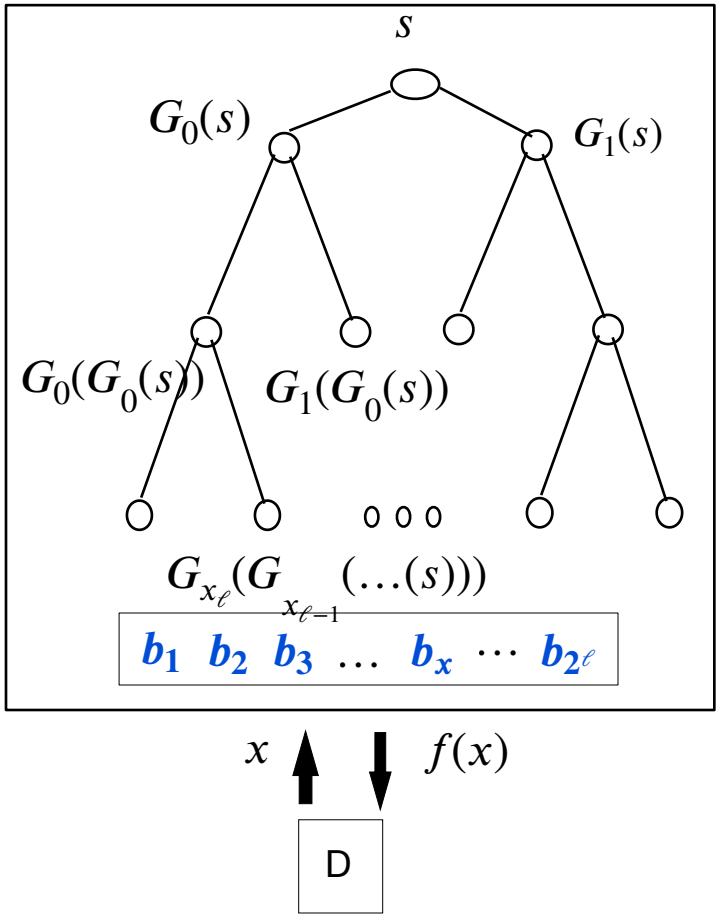
The pseudorandom world



The random world

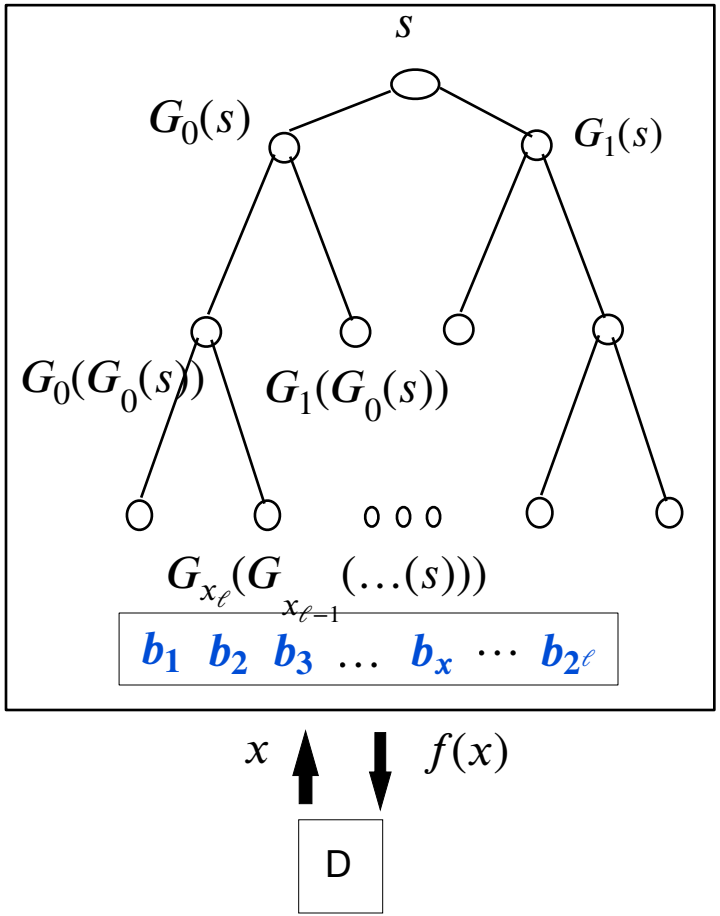


The pseudorandom world: Hybrid 0



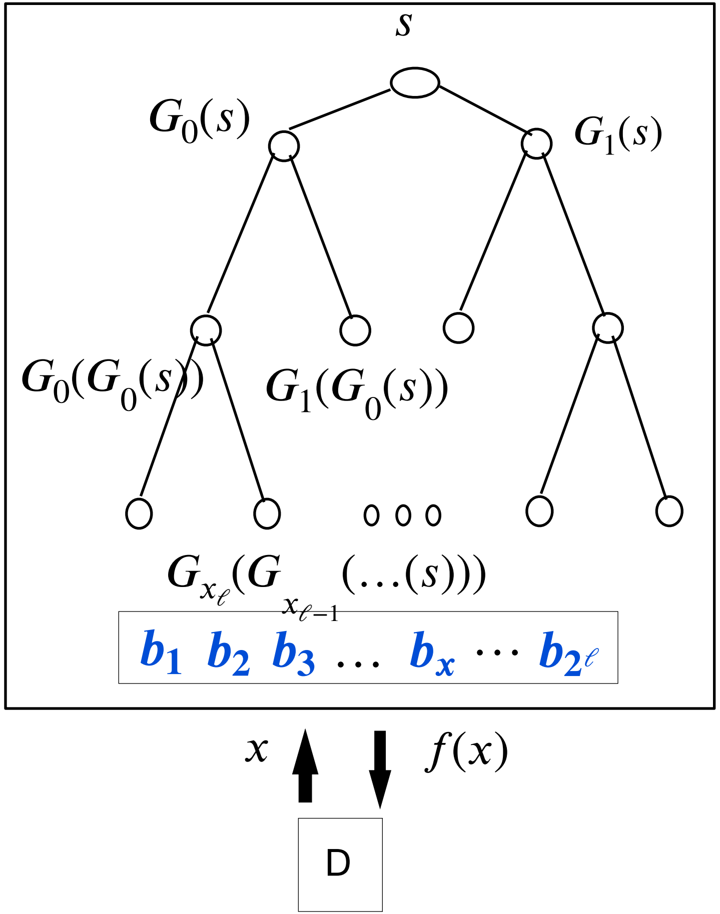
Problem:
Hybrid argument on leaves doesn't work. Why?

The pseudorandom world: Hybrid 0

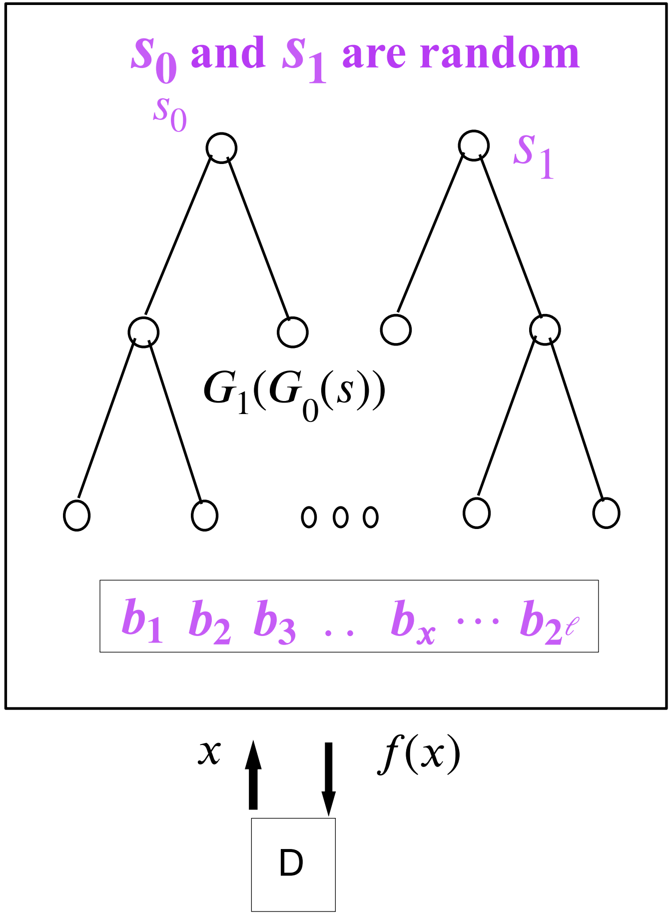


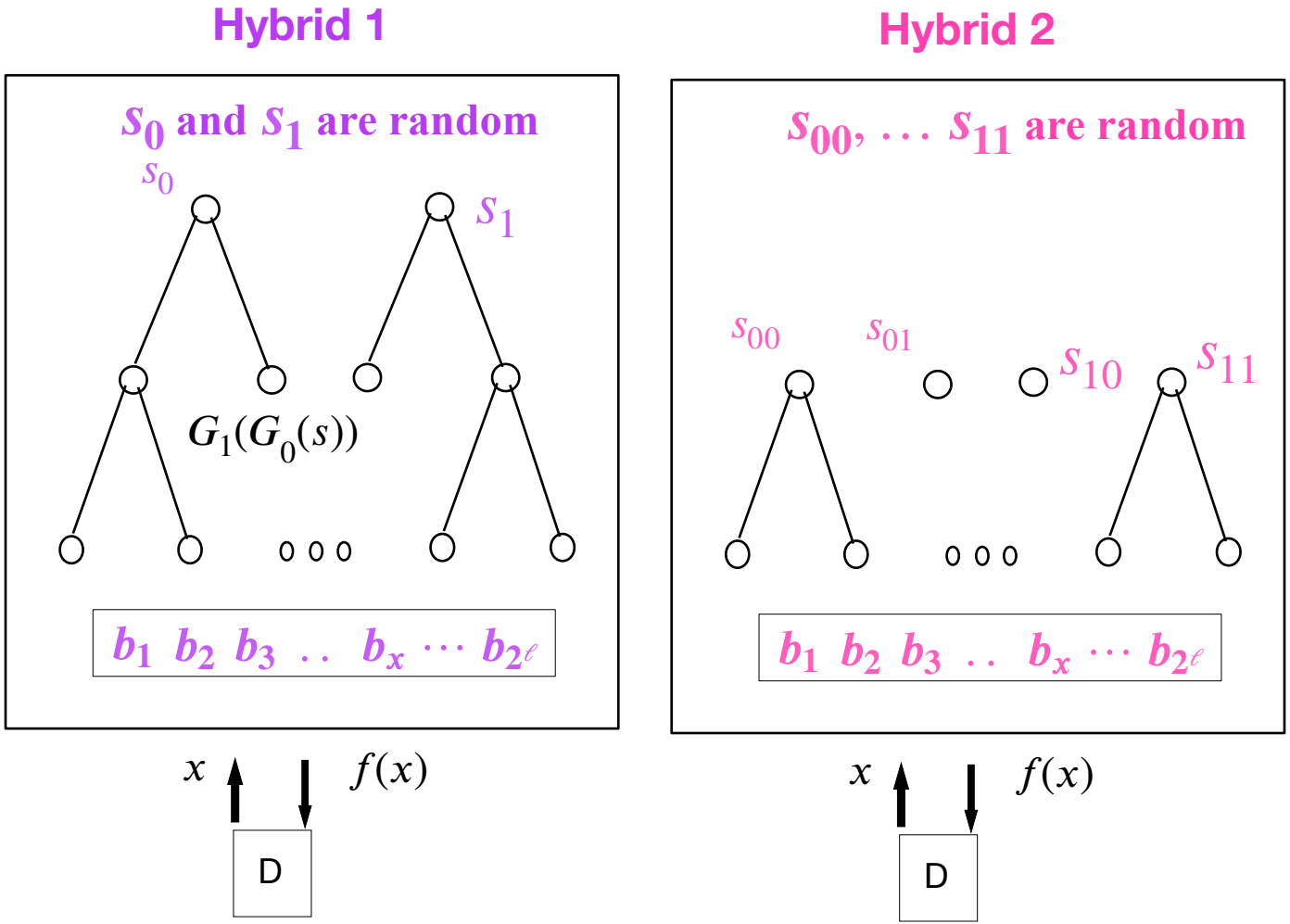
Key Idea:
Hybrid argument by levels of the tree

The pseudorandom world:
Hybrid 0



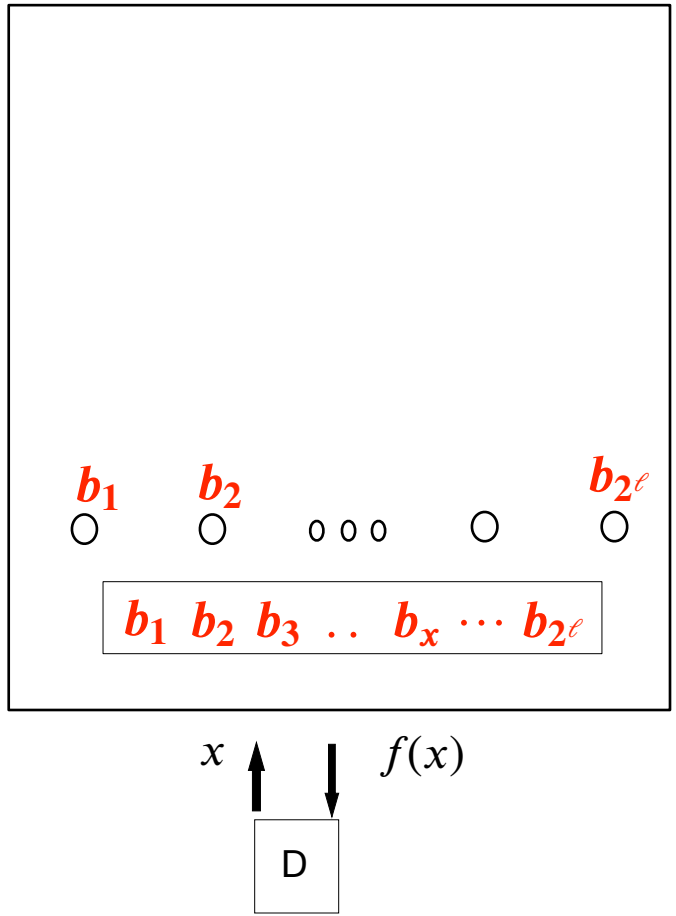
Hybrid 1

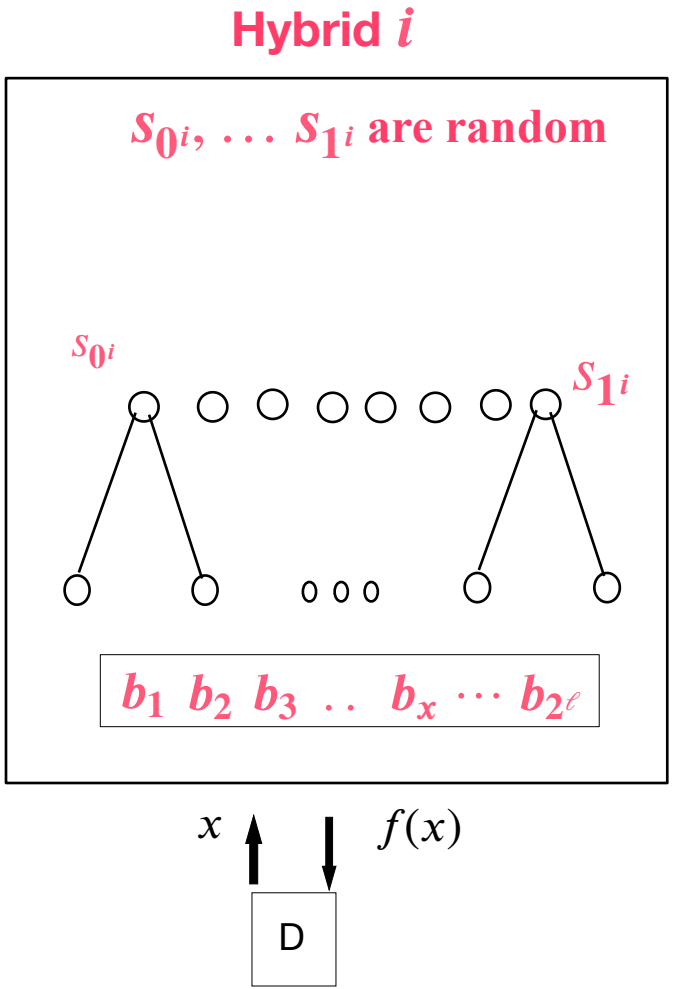



Hybrid 1
s_0 and s_1 are random
s_0
s_1
$G_1(G_0(s))$
$b_1\ b_2\ b_3\ \ldots\ b_x\ \cdots\ b_{2^\ell}$
x
$f(x)$
D
Hybrid 2
$s_{00}, \ldots s_{11}$ are random
s_{00}
s_{01}
s_{10}
s_{11}
$b_1\ b_2\ b_3\ \ldots\ b_x\ \cdots\ b_{2^\ell}$
x
$f(x)$
D

The random world:

Hybrid ℓ





Q: **Is the function in the hybrid efficiently computable?**

A: **Yes! Lazy Evaluation.**

GGM PRF

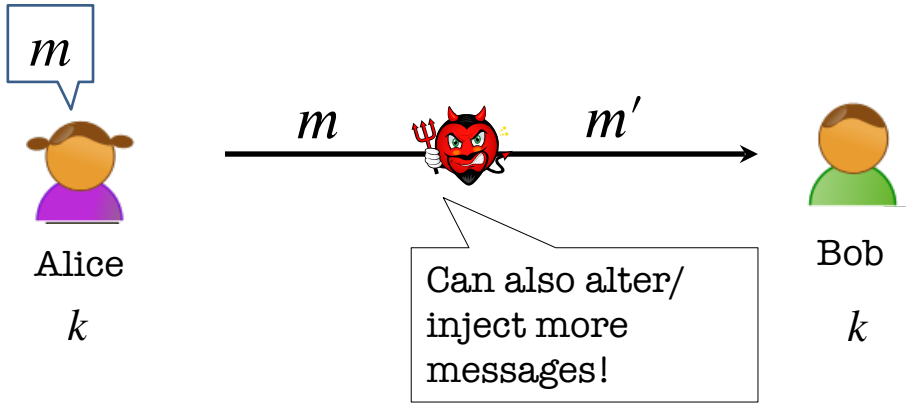
Theorem: Let G be a PRG. Then, for every polynomials ℓ, m, there exists a PRF family $\mathcal{F}_\ell = \{f_s : \{0,1\}^\ell \to \{0,1\}^m\}_{s \in \{0,1\}^n}$.

Some nits:

- *Expensive*: ℓ invocations of a PRG.
- *Sequential*: bit-by-bit, ℓ sequential invocations of a PRG.
- *Loss in security reduction*: break PRF with advantage $\varepsilon \implies$ break PRG with advantage $\varepsilon/q\ell$, where q is an arbitrary polynomial = #queries of the PRF distinguisher.

  Tighter reduction? Avoid the loss?

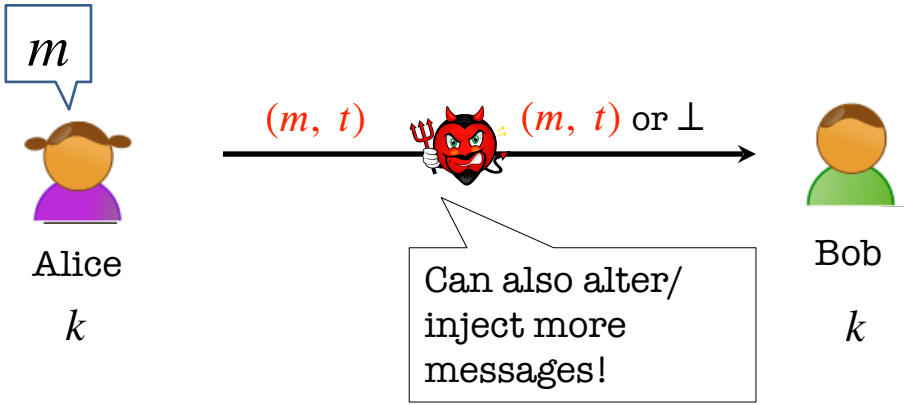
The authentication problem



This is known as a **man-in-the-middle attack.**

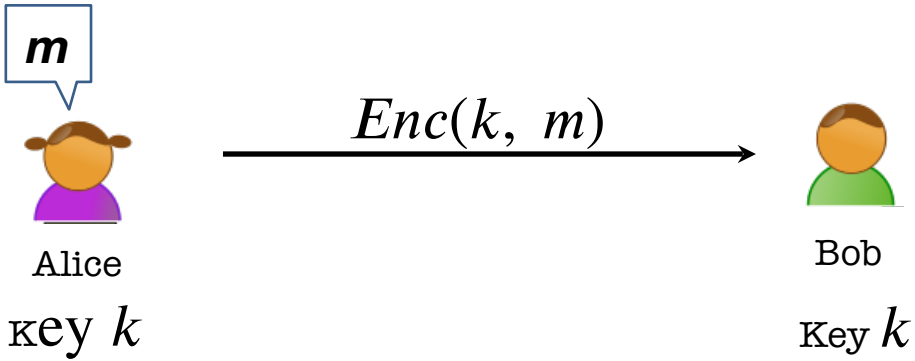
How can Bob check if the message is indeed from Alice?

The authentication problem

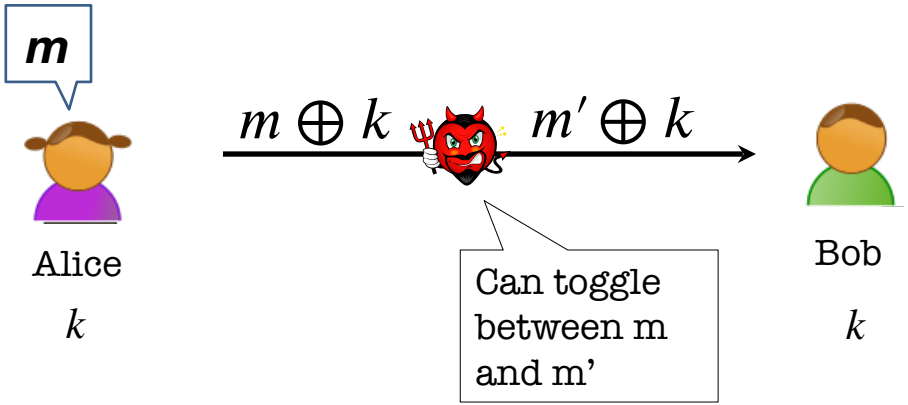


We want Alice to generate a tag for the message *m* which is **hard to generate** without the secret key *k*.

Wait… Does encryption not solve this?

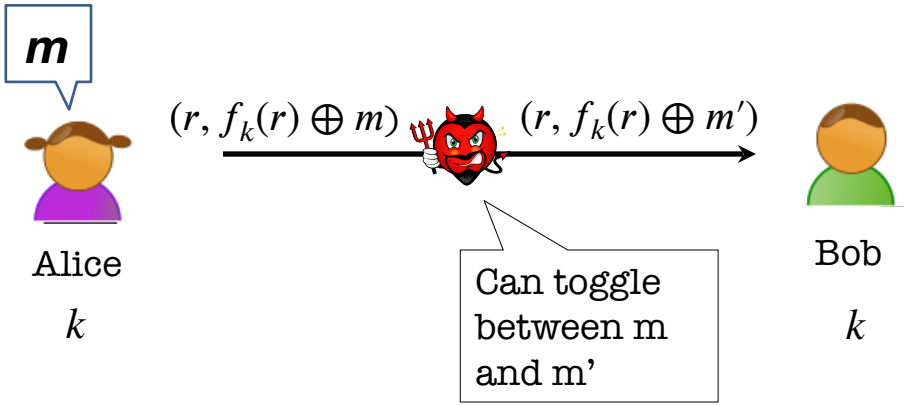


Wait… Does encryption not solve this?



One-time pad (and encryption schemes in general) are ***malleable***.

Wait… Does encryption not solve this?



One-time pad (and encryption schemes in general) are ***malleable***.

Privacy and Integrity are very **different goals!**

Message Authentication Codes (MACs)

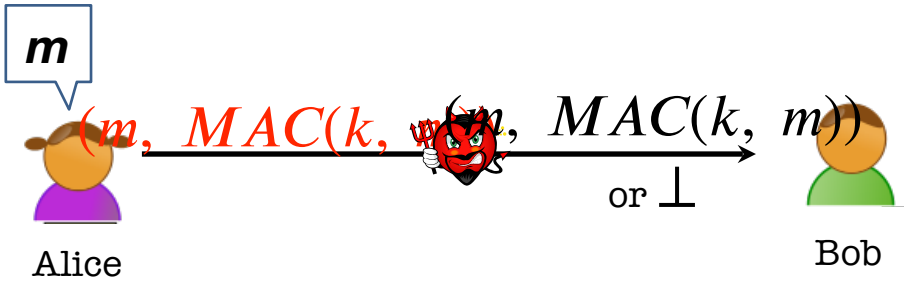
A triple of algorithms (Gen, MAC, Ver):

- Gen(1^n): Produces a key $k \leftarrow \mathcal{K}$.
- MAC(k, m): Outputs a tag t (may be deterministic).
- Ver(k, m, t): Outputs Accept or Reject.

Correctness: $\Pr[\text{Ver}(k, m, \text{MAC}(k, m) = 1] = 1$

Security: *Hard to forge.* Intuitively, it should be hard to come up with a new pair *(m', t')* such that Ver accepts.

What is the power of the adversary?



- Can see many pairs $(m,\ MAC(k,\ m))$.
- Can access a MAC oracle $MAC(k,\ \bullet\ )$
  - Obtain tags for message of choice.

This is called a *chosen message attack (CMA).*

Defining MAC Security

- **Total break:** The adversary should not be able to recover the key *k*.
- **Universal break:** The adversary can generate a valid tag for every message.
- **Existential break:** The adversary can generate a new valid tag *t* for some message *m.*

We will require MACs to be secure against the existential break!!

EUF-CMA Security

Existentially **U**n**f**orgeable against **C**hosen **M**essage **A**ttacks

$k \leftarrow K$

m_1

$t_1 = \text{MAC}(k, m_1)$

m_2

$t_2 = \text{MAC}(k, m_2)$

$\vdots$

$(m,\ t)$

Accept if $(m,\ t) \neq (m_i,\ t_i)$ for all i, and $\text{Ver}(k, m, t) = 1$

Want: $\Pr((m,\ t) \leftarrow A^{MAC(k,\ \cdot)}(1^n),\ Ver(k,\ m,\ t) = 1,\ (m,\ t) \notin Q)) = negl(n)$.

where Q is the set of queries $\left\{ (m_i,\ t_i) \right\}_i$ that A makes.

Let I = (S,V) be a MAC.

Suppose an attacker is able to find $m_0 \neq m_1$ such that

$$MAC(k, m_0) = MAC(k, m_1) \quad \text{for } \tfrac{1}{2} \text{ of the keys } k \text{ in } K$$

Can this MAC be secure?

Yes, the attacker cannot generate a valid tag for m_0 or m_1

No, this MAC can be broken using a chosen msg attack

It depends on the details of the MAC

$Adv[A, I] = 1/2$

Let I = (S,V) be a MAC.

Suppose MAC(k,m) is always 5 bits long

Can this MAC be secure?

⟹ No, an attacker can simply guess the tag for messages

It depends on the details of the MAC

Yes, the attacker cannot generate a valid tag for any message

$Adv[A,I] = 1/32$

Dealing with Replay Attacks

- The adversary could send an old valid *(m, tag)* at a later time.
  - In fact, our definition of security does not rule this out.

- **In practice:**
  - Append a time-stamp to the message. Eg. (m, T, MAC(m, T)) where T = 21 Sep 2022, 1:47pm.
  - Sequence numbers appended to the message (this requires the MAC algorithm to be *stateful*).