6.857 Recitation #4

March 12, 2021

Announcements:

- PSet 2 out; due March 23
- Feedback form

Plan for today:

- Quick recap: goals and constructions
  - "Create" randomness: PRF
  - Hide a message: Encryption
  - Authenticate a message: MAC
  - CPA/CCA: clarification of MACs
- Review hash functions
  - Random Oracle Model (ROM)
  - Families and salts
  - Properties
- Birthday problem

Goals & Constructions: A Quick Recap

"Create randomness":

...actually, "create random-looking" strings

↳ name-brand randomness is too expensive so we counterfeit a knockoff pseudo-name brand instead

Tool: Pseudo-Random Function (PRF)

Idea: take n bits of randomness $k \leftarrow \{0,1\}^n$.

Then a PRF F lets us create poly(n) "random-looking" strings from k.

Hide a message m: Encryption scheme (Gen, Enc, Dec)

"Confidentiality"

Authenticate a message m: Message Authentication Codes (MACs)

"integrity"

aka tags

To try to clarify some c vs m confusion in lecture:

→ MACs are defined independent of Encryption, so we wrote message-tag pairs as

$$(m, MAC(k,m))$$

for an arbitrary input m to authenticate.

→ For CCA security, we authenticate a ciphertext:

shared $k \leftarrow Gen(1^n)$, $c \leftarrow Enc(k,m)$

send $(c, MAC(k,c))$

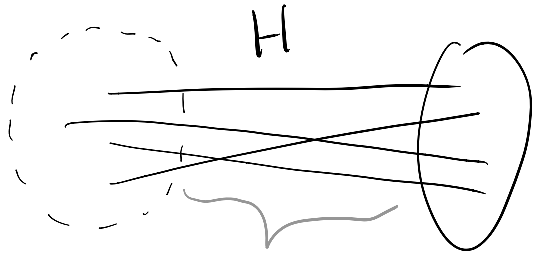
Hash Functions

Why PRFs? Why is randomness "expensive"?

→ A $\underbrace{\text{truly random function}}_{\text{call it } H}$ is not efficiently computable.

In our context, we have a compression:

$$H: \{0,1\}^* \to \{0,1\}^d$$



map is <u>fixed</u> for H

so $H(x) = y$

(Consistently! We don't write $y \leftarrow H(x)$.)

But map is <u>random</u> between distinct $x, x' \in \{0,1\}^*$

ie, $H(x)$ and $H(x')$ are independent

(Can think of H itself as sampled from the set of random maps onto $\{0,1\}^d$)

Back to efficiency: We said H is not efficiently computable.

Suppose we built a circuit for H, C_H.

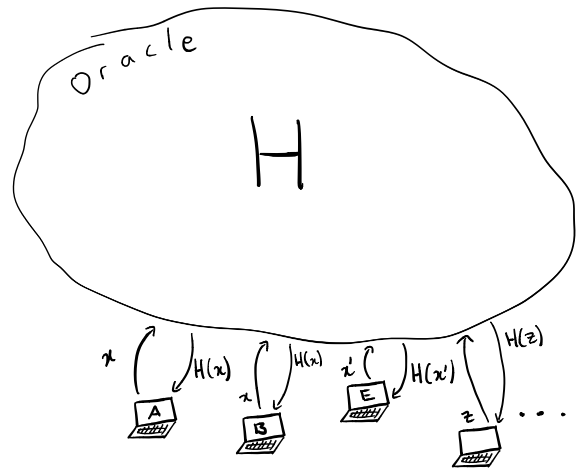
What is the size of C_H?

Recall range of H is $\{0,1\}^d$, size 2^d

So C_H needs to be able to compute $\sim 2^d$ different conditions on its input

Therefore the size of C_H is exponential and can't be computed in poly time.

→ The random oracle model (ROM) is a useful, but controversial, model!



In the ROM, H itself can be used to compute PRFs, even if P = NP!

(Why? H is a OWF even if 3SAT ∈ P, & OWF ⇒ PRF.)

See:
Katz–Lindell book, section called "Is the Random-Oracle Methodology Sound?"

ROM gives us a theoretical arena.

In practice we don't have oracle access to H,

so we use SHA-256 and pray...

Notation:

Function families: $\{h_s\}$ indexed with s

each $h_s : \{0,1\}^* \longrightarrow \{0,1\}^d$

With salts: $h_s(x) = h(s \| x)$

again, commonly SHA-256

What properties of h_s might we want?

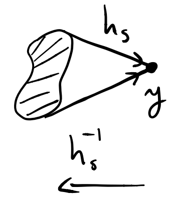
Properties of hash functions

One way:

given s, y (where $y = h_s(\text{something})$),

infeasible to find an input x in $h_s^{-1}(y)$

NB: this is a set:



Collision resistance:

given s,

infeasible to find distinct x, x' s.t. $h_s(x) = h_s(x')$

Target Collision resistance:

given s, x,

infeasible to find x' s.t. $x' \neq x$ and $h_s(x') = h_s(x)$

CR/TCR distinction:

TCR says "$\exists$ hard cases" (ie, I tell you a target x)

CR says "it is hard in general"

$\therefore$ CR is stronger

constrains!
no such constraint for CR

(Think in terms of a game...)

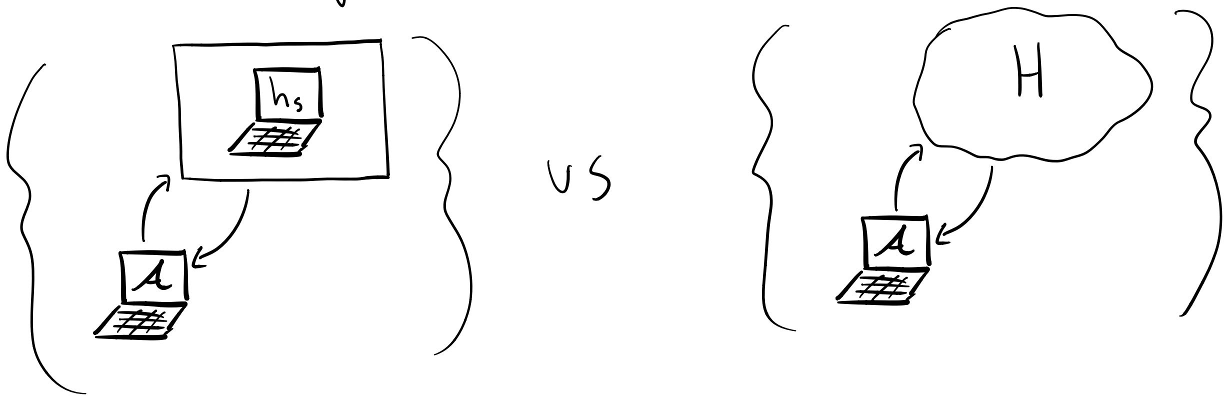
$$CR \Rightarrow TCR \Rightarrow OW$$

$\neg OW \Rightarrow \neg TCR$ bc can take given x and compute y as $h_s(x)$

Pseudo random:

given s,

infeasible to distinguish between



Non-malleable:

given s, $h_s(x)$ for some random x

infeasible to find $h_s(x')$ for a related x'

ie, $x' = f(x)$ for known f, not nec. known x

(this should remind you of the discussion from lecture for motivating CCA security — $(r, F(k,r) \oplus m)$ is CPA, easy to send $(r, F(k,r) \oplus m \oplus 1)$ ... so this encryption scheme is malleable)

Birthday Paradox (appendix of Katz-Lindell)

A faux "paradox"

→ w/ <u>23 people</u> in a room, >50% chance that 2 people share a birthday!

But there are 365 days!

Generalizing:

Sample set of size N — $N = 365$ for birthdays

Take q samples — $q = 23$ above

Probability p of collision — $p = 0.5$ above

Theorem: $p(q, N) = \Theta(\sqrt{N})$

Proof: We will show $\dfrac{q(q-1)}{4N} \leq p(q, N) \leq \dfrac{q^2}{2N}$

First: $p \leq \frac{q^2}{2N}$

Write q samples as x_i for $i = 0, \ldots, q-1$

Union bound over all ways to collide:

$$p \leq \sum_{i,j} \Pr[x_i \text{ and } x_j \text{ collide}]$$

The probability of two x's colliding is at least $\frac{1}{N}$

$$\Rightarrow p \leq \binom{q}{2} \frac{1}{N} \leq \frac{q^2}{2N} \quad \text{because} \quad \binom{q}{2} \leq \frac{q^2}{2}$$

Second: $\quad p \geq \frac{q(q-1)}{4N}$

p is probability of collision.

Pairwise over x's, the probability two x's don't collide is:

$$\prod_{i=1}^{q-1}\left(1-\frac{i}{N}\right) \leq \prod_{i=1}^{q-1} \underbrace{e^{-i/N}}_{\text{Recall Taylor series of } e^{x}=1+x+O(x^2)}$$

Move $\prod$ to sum inside exponential:

$$\Pr[\text{No collision}] \leq \exp\left[-\sum_{i=1}^{q-1}\frac{i}{N}\right] = \exp\left[-\underbrace{q(q-1)}/2N\right]$$

Nice summation result: $\sum_{i=1}^{q-1} i = \frac{q(q-1)}{2}$

Either we have a collision or we don't:

$$p = 1 - \Pr[\text{No collision}]$$

$$\Rightarrow p \geq 1 - \exp\left[-q(q-1)/2N\right]$$

$$\Rightarrow p > \frac{q(q-1)}{4N}$$

(back to Taylor series, $e^{-x} \leq 1 - \frac{x}{2}$ for $|x| \leq 1$)