Lecture 1: Intro & Merkle Puzzles

MIT – 6.893
Fall 2020
Henry Corrigan-Gibbs

Agenda

- Intro: Merkle puzzles
- Goals of this course
- Stretch break
- Logistics & course info.

Course website: 6893.csail.mit.edu

Me: Henry Corrigan-Gibbs ("Henry")

Two questions we will focus on:

1. What cryptographic tools can we use to protect our privacy?

   ... and how do we even define "protect privacy"?

2. Why do we use so few of these tools in practice?

   ↳ In principle, we can use crypto to build systems that have all sorts of wonderful security & privacy properties.

   What is keeping us from using them in real systems?

   Even if we could, would they solve our privacy problems?

There will be some overlap with 6.857, 6.858, 6.875.

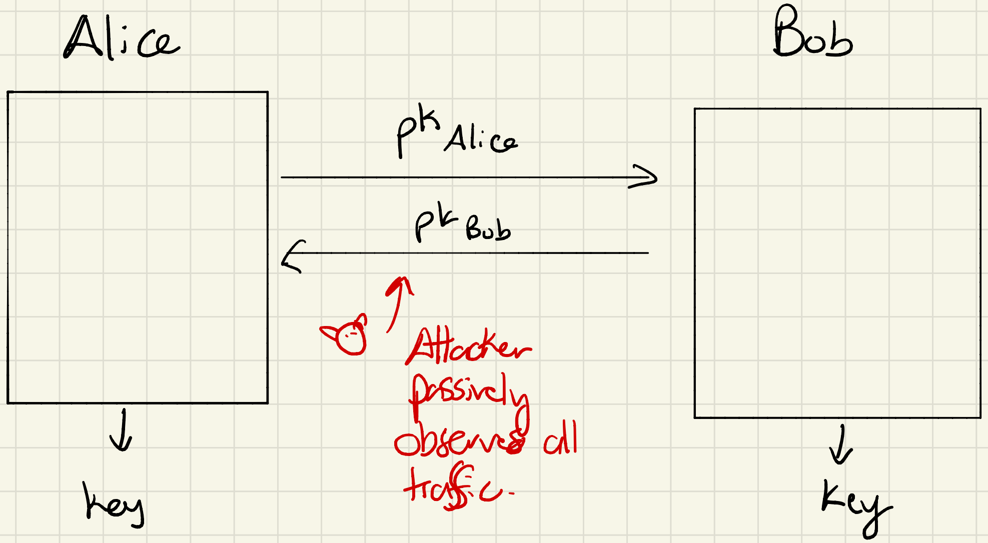
→ Overlap will be greatest at the start.

→ I will try to minimize it.

→ This course should go into more depth on each topic, since we have more time to cover each.

Want to start this course with a nice but simple idea...

Key exchange: The problem that launched modern cryptography.



Properties we want

1. Correctness. Agree on same key.
2. Security. No "efficient" attacker can distinguish true key from random.

In your intro crypto class, you saw how to build key exchange from

- DH problem (discrete log++)
- RSA problem
- ... any public-key encryption system

In these systems, Alice & Bob run in poly time; best attack is super-poly time.

... but these constructions didn't exist until 1976.

Merkle Puzzles (1974)

- Predated DH key exchange (1976)
- Uses only hash fns — no fancy assumptions.
- Conceived by Ralph Merkle as a project for his undergrad CS security class (!!!)
- The catch: Alice & Bob run in time $\approx n$

  Eavesdropper recovers secret in time $\approx n^2$

Quadratic gap, not exponential

Even so, gap b/w 2^{30} and 2^{60} is huge.

Why discuss Merkle puzzles today?

- Beautiful, simple idea
- Good excuse to talk about random-oracle model
- The origins of crypto for privacy
- Reminder that students have fantastic ideas.
- Didn't work in practice...
  ...but led to things that did.

State of many (not all) ideas in crypto that could be useful.

Merkle's Key Exchange Protocol

Uses hash functions $H: [n^2] \to \{0,1\}^n$ ($[n^2] = \{1, \dots, n^2\}$)

$f: [n^2] \to \{0,1\}^n$

Distinct inputs $\Rightarrow$ distinct outputs

Alice		Bob
1. Pick ints $a_1, \dots, a_n \xleftarrow{R} [n^2]$		
	$\xrightarrow{H(a_1), \dots, H(a_n)}$	
		2. Pick ints $b_1, \dots, b_n \xleftarrow{R} [n^2]$
	$\xleftarrow{H(b_1), \dots, H(b_n)}$	
3. Find least $i, j \in [n]$ s.t. $H(a_i) = H(b_j)$. Output $f(a_i)$ as shared secret.		Do the same as Alice. Output $f(b_j)$ as shared secret.

"Boneh's Law"

Question: What property do we need of hash fn H for this protocol to be secure against a passive eavesdropper?

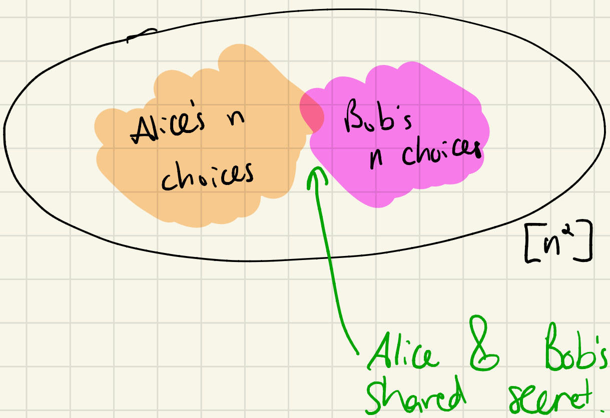
→ This is pretty amazing! No fancy number theory or anything... just hash fns.

Sanity checks:

1. Efficiency: Alice and Bob each invoke H only n times. ✓

Question: What's the true efficiency bottleneck of Merkle's scheme?

2. Correctness: By "Birthday Paradox" (on HW) ✓



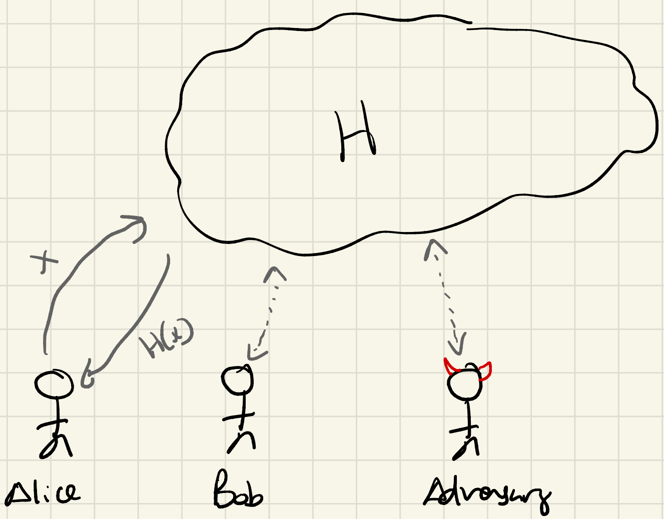
3. Security.

Claim: Passive eavesdropper needs to invoke H, f roughly n^2 times to recover the shared secret.

To analyze Merkle's scheme, we will use the random-oracle model.

Introduced to crypto by Bellare & Rogaway (1993)

Idea: Think of hash fn as a truly random fn to which all parties have oracle access



* In many cases, R.O. model dramatically simplifies the security analysis.

* In practice, replace R.O. with SHA-256 and hope that nothing breaks.
  ↳ This "heuristic" works shockingly well.

The R.O. Model is controversial!

1) SHA256 is not at all a random fn (it has a small description, for one)

2) Unsafe in general

$\exists$ sig schemes that are secure in ROM are insecure when instantiated with any eff hash fn.
(Canetti, Goldreich, Halevi '98)

Not "natural" sig schemes, but still very unsettling.

Formally,

Non-interactive Key Exchange

$$1^n = \underbrace{111\cdots1}_{n \text{ times}}$$

Three efficient algs:

$\text{Setup}(1^n) \longrightarrow pp$ — Output public params

$\text{Publish}(pp) \longrightarrow (sk, pk)$ — Output secret part, public part

$\text{KeyGen}(sk, pk) \to \text{key} \in \mathcal{K}$ — Generate shared secret key.

Properties

1. Correctness $\forall\ pp \leftarrow \text{Setup}(1^n)$

$$(sk_A, pk_A) \leftarrow \text{Publish}(pp)$$
$$(sk_B, pk_B) \leftarrow \text{Publish}(pp)$$

$$\Pr[\underbrace{\text{KeyGen}(sk_A, pk_B)}_{\text{Alice's output}} = \underbrace{\text{KeyGen}(sk_B, pk_B)}_{\text{Bob's output}}] \geq 1 - \text{negl}(n).$$

Recall: A "negligible" function $f(n)$ is one s.t.

$f(n)$ is $O(\frac{1}{n^c})$ for all $c \in \mathbb{N}$.

Or, its inverse grows faster than any fixed poly.

e.g. 2^{-n}, $2^{-\sqrt{n}}$, $n^{-\log n}$, $n^{-\log\log\log\log n}$, $n^{\sqrt{\log n}}$, ...

Useful b/c $\text{negl}(n) \cdot \text{poly}(n)$ is negligible.

Properties

2. Security: "Efficient" adv shouldn't be able to distinguish shared secret from random value.

For $b \in \{0,1\}$, let W_b denote the event that the following experiment outputs "1".

$$
\boxed{
\begin{array}{l}
pp \leftarrow \mathrm{Setup}(1^n) \\
(sk_A, pk_A) \leftarrow \mathrm{Publish}(pp) \\
(sk_B, pk_B) \leftarrow \mathrm{Publish}(pp) \\
key_0 \leftarrow \mathrm{KeyGen}(sk_A, pk_B) \\
key_1 \xleftarrow{R} \mathcal{K} \\
\text{output } \mathcal{A}(pp, pk_A, pk_B, key_b)
\end{array}
}
$$

Then define the advantage of $\mathcal{A}$ at breaking our key ex scheme as

$$\mathrm{Adv}[\mathcal{A}] := |\Pr[W_0] - \Pr[W_1]|.$$

We say that a key ex scheme is "secure" if for all efficient advs $\mathcal{A}$,

$$\mathrm{Adv}[\mathcal{A}] \leq \mathrm{negl}(n).$$

Can't distinguish true secret from random.

We will show that adversary running in time $o(n^2)$ has advantage $o(1)$.

Run scheme n times in parallel and take the XOR / hash of all keys to drive this advantage down to $\mathrm{negl}(n)$.

Security Intuition

Unless adversary can query H or f at the special point (call it x^*) at on which Alice & Bob agree, adv has no information on shared secret.

↳ Can't even distinguish it from a random value.

Making these arguments precise is surprisingly tedious and error-prone.

World 0:

World 1:

Challenger

$a_1, \dots, a_n \leftarrow [n^2]$

$b_1, \dots, b_n$

Let x^* be first collision b/w a_i & b_j.

On H or f queries at x^*, reply with fresh random value.

$H(a_1), \dots, H(a_n)$

$H(b_1), \dots, H(b_n)$

$\xrightarrow{f(x^*)}$

$\xleftarrow{x}$

$\xrightarrow{H(x)}$

$\xleftarrow{x}$

$\xrightarrow{f(x)}$

Adversary

$\downarrow$

$b \in \{0,1\}$

In this interaction, event that adv outputs 1 is exactly event W_0 in our sec defn.

Strategy: Modify experiment.

→ In world 1, challenger responds to adv's hash queries at special point x^* using fresh random value

→ In world 1, shared secret $f(x^*)$ is indep. of adv's view

Now, define a failure event F:

Let F = event that adv queries H or f on point x^*

Claim: $\text{Adv}[\mathcal{A}] = \Pr[F]$.

Why? If adv outputs 1 in world 0 and $\bar{F}$, adv outputs 1 in world 1 and $\bar{F}$.

$$W_0 \wedge \bar{F} \iff W_1 \wedge \bar{F}$$

Then

$$\underbrace{|\Pr[W_0] - \Pr[W_1]|}_{\mathcal{A}\text{'s advantage}} \leq \Pr[F]$$

One version of this is the "Difference Lemma" of Boneh & Shoup Thm 4.7!

Now we just need to bound $\Pr[F]$.

Claim: $\Pr[F] \leq o(1)$.

Subconstant in n.

Pf idea: * Say that adv makes $T = o(n^2)$ queries total.

* The value x^* is indep of adv's view initially.

* On the ith query $\Pr[A \text{ queries } x^*] \leq \frac{1}{n^2 - T}$.

Then by union bound

$$\Pr[F] \leq T \cdot \frac{2}{n^2 - T} = o(1)$$

So, we've shown that adversary running in time $o(n^2)$ has advantage $o(1)$ at distinguishing shared key from random.

$\Longrightarrow$ Amplify by running n times in parallel to drive down adv's advantage.

Stretch
Break!

Logistics

This is an exceptionally stressful and confusing time for all of us.

My goals:

* that you look forward to coming to class,
* that the psets are challenging, but not frustrating, and
* that you leave this class with the knowledge & motivation to bring some new privacy tech into the world.

We will have five truly exceptional guest speakers.

↳ ACLU, FTC, Google, Distinguished cryptographer, Columbia Ctr Digital Journalism,

Logistics

Communication: Most questions → Piazza

(easiest for me to track
+ other people will have same Q)

HW questions that might reveal answer
↳ Piazza, private Q

Course feedback: especially constructive criticism but also things you liked. (Or requests!)
↳ Anonymous feedback form hosted on Qualtrics. See link on course site.

"Any time, any reason"

Individual questions → Email

Office hours: Wed 3–4:30pm on Zoom

(See Piazza for link)

↳ If you want to talk 1-on-1 about something (potential research idea, ask advice about ___, ...) feel free to email me.

(Students are #1 priority for me. I will try to make time.)

Logistics

Problem sets: Publication & due dates posted

HW #1 posted now
↳ Due 9/18 5pm Boston via Gradescope.

6 HW over semester.

This is a 3-0-9 corse...
≈ 9 hrs of outside work per week
≈ 18 hrs per problem set.

I'm going to try to keep the problem sets in the 10-20 hr range. First few will be uncalibrated

Everyone gets three free "late days". See website for details.

Collaboration: Allowed in groups of ≤ 3. You must declare your collaborators on problem sets.

Bugs: I will try my best to write unambiguous and bug-free problem sets. But I will fail sometimes. If a Q looks unclear and/or impossible, please ask on Piazza.

(I apologize in advance!)

Logistics

Attendance:

Required & important always. This is a small class and interaction is key. (Also fun!)

ESPECIALLY IMPORTANT when we have guest speakers.

These people are taking time from their hectic lives to share their knowledge with us. Respect their time by showing up with your questions and enthusiasm.

Grading

1/6 for each of six psets.

(Unexcuse absences $\Rightarrow$ Grade $\downarrow$)

Grades at MIT are not curved. I reserve the right to increase your grades so that the letter grade matches my perception of the mastery of the material.

Resources: For your time at MIT

We are all going through a tough time.
If you need help with school

Places to go for help:

For anything

In EECS: UG: Katrine Lacurts
Grad: Leslie Kolodziejski

Institute: UG: Student Support Services (S3)

Essentially external to MIT: All: MIT Ombuds Office, "Any time, any reason"
* Confidential

For school stuff:

Institute: Grad: Grad Support

Mental health
* Confidential

(anxious? isolated? sleeping too much? don't enjoy things you used to? ....)

UG/Grad: MIT Medical Student Mental Health
(free for you! use it while you can)

You can always ask me if you're not sure where to go... I'll try to point you to the right place.

Closing thoughts?

→ HW1 out now

→ Sign up for Piazza & Gradescope.