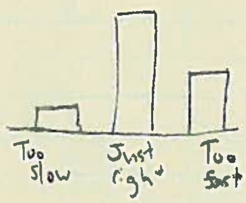
April 26 - Elliptic Curves & DDH

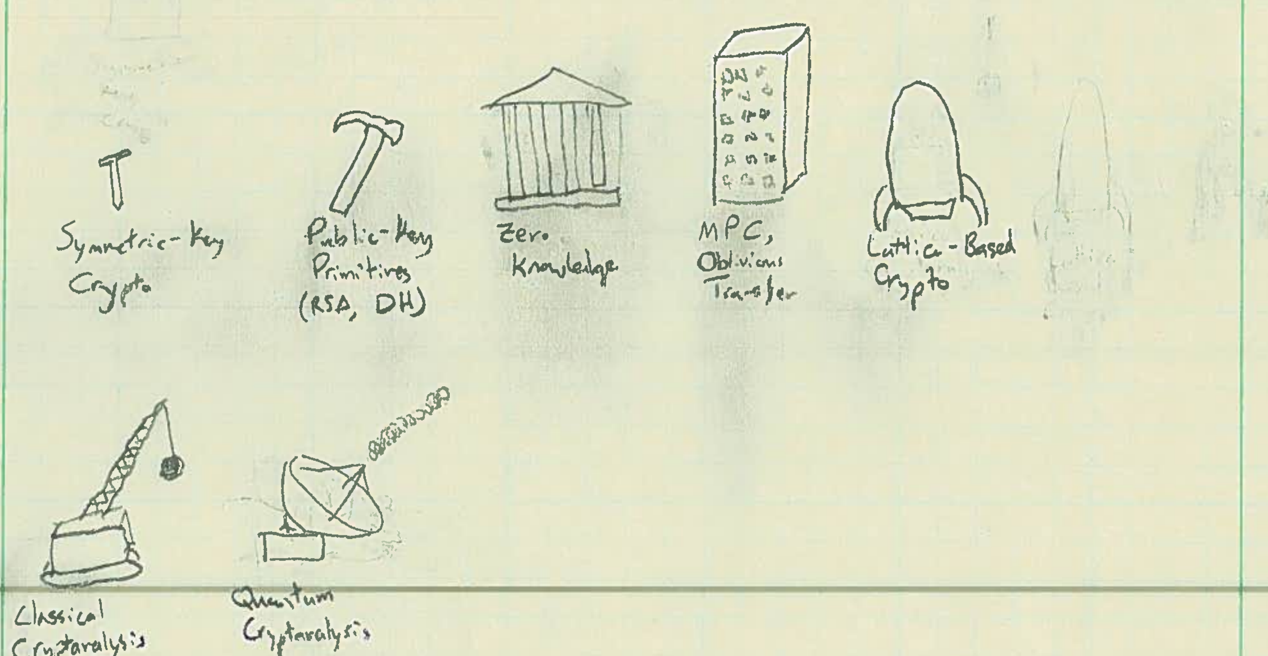
Logistics

- Problem set due Now
- Scribe for lecture
- Feedback summary



* Write bigger, move slower
* relevant modern readings — Historical papers are hard to read!
* Big picture
* History /color commentary
* Bitcoin (probably won't b/c of Bitcoin class)
* Come to OHs if you want clarification

"The Big Picture"



Review of RSA

Hard Problems

Factoring: Given $N = pq$, produce (p, q)

RSA-e : Given $N, a \xleftarrow{R} \mathbb{Z}_N^*, e$ produce x s.t. $x^e = a \pmod N$.

"Find an e-th root mod N"

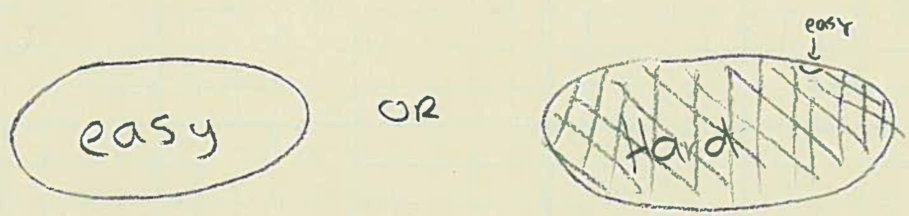
Strong RSA: Given $N, a \xleftarrow{R} \mathbb{Z}_N$ produce (x, e) s.t. $x^e = a \pmod N$ with $e \neq \pm 1$.

"Find <u>any</u> eth root mod N".

FACTORING $\geq$ RSA $\geq$ Strong RSA

Random Self-Reduction

→ If RSA-e is hard for any $a \in \mathbb{Z}_N$, it is hard for almost <u>all</u> $a \in \mathbb{Z}_N$.



Rabin TDOWF

- $f(x) = x^2 \bmod N$
- We argued that inverting $f(x)$ for $x \xleftarrow{R} \mathbb{Z}_N$ is as hard as factoring N.
- Can build PKE from Rabin's function (also signatures)
  ↳ we didn't explain how (<5255)

⟹ Taking square roots mod N is as hard as factoring N!

Review II - RSA Applications

Hash-and-Sign Signatures

$\text{Sign}(sk, m) = H(m)^d \bmod N$

$\text{Verify}(vk, m, \sigma) = \{\sigma^e \stackrel{?}{=} H(m)\} \bmod N$

$\Rightarrow$ Without the hash, it's broken!

Threshold RSA Signatures

↳ split the signing key into p parts
need all p parts to sign

Blind Signatures

- Can sign a message without knowing what it is
- e.g. anonymous survey w/ extra credit

	Student		Us
Anonymously receive token		$\xrightarrow{r^e H(\text{"henrycg"})}$	
		$\xleftarrow{r \cdot H(\text{"henrycg"})^d}$	
Claim extra credit		$\xrightarrow{\sigma,\ H(\text{"henrycg"})}$	

RSA Accumulator

- Shorter Merkle tree

In this lecture

1) Elliptic curves: what & why?
2) Hard problems related to EC.
3) Application: PRF from DDH
4) Pairings & applications

Elliptic Curves

First, why: * an RSA signature is an element of $\mathbb{Z}_N^*$

* $\underbrace{\text{for 128-bit security}}_{\text{best attack cost} \approx 2^{128} \text{ work}}$, need $\approx$ 3072-bit modulus

Reason: best alg factors a n-bit int in time $\approx 2^{2.78 n^{1/3} (\log n)^{2/3}}$

You want to choose s.t.

$$2^{2.78 n^{1/3} (\log n)^{2/3}} > 2^{\lambda}$$

$\Rightarrow n$ grows like λ^3

* In contrast, these fancy attacks do not apply to EC systems

Best known attack on n-bit keys: $2^{n/2}$ time

$\Rightarrow n$ grows like 2λ.

$\Rightarrow$ For 128-bit security, 256-bit key

Signatures are nearly as short (10x shorter!)

Proposed in 1980s by Koblitz & Miller

- Certicom (Canadian company) pushed ECC starting in 90s
- RSA Corp lobbied heavily against ECC

2005 - NSA pushed industry to switch to ECC

... took until 2011 for Google to make ECC default in TLS

Why took so long?

* suspicion of NSA backdoors — many magic constants!
  - ↳ later proved valid Dual EC
* Math is harder - RSA is relatively easy for a programmer to see
* Lots of subtle special-case attacks
  - → # points on $E/\mathbb{F}_p = p$ "anomalous curves"
  - → ECDL → Dlog $\mathbb{F}_p$ "Weil descent"
  - → EC over $\mathbb{F}_{p^m}$ when m large
  - → Hyper-elliptic curves

EC: What Changed

1988 - NFS (Pollard) → Bigger RSA modulus

2004 - IBE from Pairings (DAN)

↳ Positive applications

↳ Lots of academics have a stake in ECC success

2010s - TLS becomes widespread

↳ Saves on bandwidth — key resource.

Elliptic Curves

A group is $(G, *)$

set of elements — binary operation.

1. Closure
2. Associativity
3. Identity
4. Inverse

In "Classic DH"

$G = \{\text{integers mod prime } p\}$

$* =$ "multiplication mod p"

In ECDH

$G = \{\text{set of } (x,y) \text{ points on curve } E\}$

$* =$ "addition of EC points"

Elliptic Curve

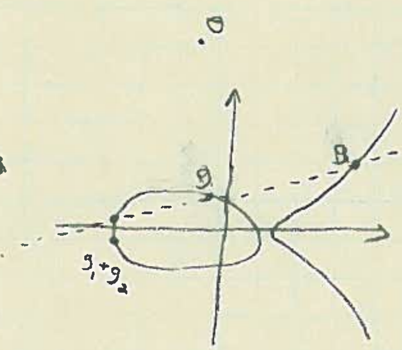
- Work mod prime ($p \approx 256$ bits)

$y^2 = x^3 + Ax + B \pmod p$ ← No repeated roots

$= x(x-1)(x-\lambda)$, $\lambda > 1$

- Over $\mathbb{R} \to$

$y^2 = x^3 - 5x + 4$



- Why EC?

$G_{5,4} = \{(x,y) \in \mathbb{Z}_p^2 \mid y^2 = x^3 - 5x + 4 \bmod p\}$
$\cup \{\text{point at infinity}\}$

What about group op $*$?

$g_1 * g_2 =$ draw line & reflect

$g * g =$ draw tangent & reflect

$g * \mathcal{O} =$ do nothing

⇒ Lots of deep theory here

* 4 properties of group hold.

* Point compression:
  - For every x, only ≤ 2 (x,y) points on $E_{A,B}$
  - Send $(x, \pm)$ instead of (x,y)

EC Notation

g = point on curve

$g^2 = g * g$ = point composed w/ itself

$g^3 = g * (g^2)$

$g^4 = g * (g^3)$

⋮

g^a = point composed w/ itself "a" times.

If there are q points on E, q prime $\exists$ point g s.t. "generator"

$G = \{g, g^2, g^3, \ldots, g^{q-1}, g^q = O\}$

G works like our normal DH group!
Order is q.

[Sometimes you'll see the confusing notation
$G = \{P, 2P, 3P, \ldots, (q-1)P, qP\}$]

When working in E mod p, $q \neq p$.

Computational Problems in G:

Fix EC G, generator g of order q.

Discrete Log: $a \xleftarrow{R} \mathbb{Z}_q$

Given (g, g^a) produce a.

CDH: $a, b \xleftarrow{R} \mathbb{Z}_q$

Given (g, g^a, g^b) produce g^{ab}.

DDH $a, b, c \xleftarrow{R} \mathbb{Z}_q$

Given (g, g^a, g^b, g^{ab}) or (g, g^a, g^b, g^c)
identify which you've been given

EC Hard Problems

Factoring $\geq$ RSA $\geq$ Strong RSA

Dlog $\geq$ CDH $\geq$ DDH

→ Best algorithm for DDH is "brute force" dlog

when $p = n$ bits, $2^{n/2}$ time.

For 128-bit security, need ~256-bit prime

↳ Mysteriously in Aug 2015, NSA changed to require ~384-bit prime

→ Do you have a better alg for ECDlog? ←

DDH is useful (see Dan's paper)

— It's a qualitatively different type of assumption

Search: factor N, recover e-th root, find dlog

Decision: distinguish these distributions

$$\{(g, g^a, g^b, g^{ab})\} \stackrel{c}{\approx} \{(g, g^a, g^b, g^c)\}$$

— For building crypto, decision is useful!

— DDH has a randomized self-reduction!

↳ Either easy everywhere or hard almost everywhere

↳ DDH is <u>easy</u> in $\mathbb{Z}_p^*$!

Application: PRG from DDH

Recall PRG

$f: \mathcal{K} \longrightarrow \{0,1\}^n$ (s.t. $n > \lg|\mathcal{K}|$)

$\{k \xleftarrow{R} \mathcal{K} : f(k)\} \overset{c}{\approx} \{z \xleftarrow{R} \{0,1\}^n : z\}$

Simple PRG

Fix g, g^a (for random $a \in \mathbb{Z}_q$)

$f: \mathbb{Z}_q \to G$

$f_{g,g^a}(k) = \langle g^k, g^{ak} \rangle$

Security is immediate under DDH!

$\{(g^k, g^{ak})\} \overset{c}{\approx} \{(g^k, g^r)\} \longleftarrow$ Random over G!

By DDH

N.B. CDH/Dlog not enough!

Application: PRF from DDH

Recall: a PRF $f: \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$

$$\{k \xleftarrow{R} \mathcal{K} : f(k, \cdot)\} \stackrel{c}{\approx} \{F \xleftarrow{R} \text{Funs}[\mathcal{X}, \mathcal{Y}] : F(\cdot)\}$$

Naor - $\underbrace{\text{Reingold}}_{\text{Omer}}$ PRF

$\mathcal{K} = \mathbb{Z}_q^{n+1} \leftarrow$ n+1 field elements

$\mathcal{X} = \{0,1\}^n$

$\mathcal{Y} = \mathbb{G}$

$$\vec{k} = (k_0, k_1, \dots, k_n) \in \mathbb{Z}_q^{n+1}$$

$$\vec{x} = x_1 x_2 \cdots x_n$$

$$f_{\vec{k}}(\vec{x}) = g^{k_0 \prod_{i=1}^{n} k_i^{x_i}} \in \mathbb{G}.$$

Thm If DDH holds, f is a secure PRF.