Last time:

✓ • computational hardness of learning

$\mathcal{C}$ = all poly(n)-size Boolean circuits

based on existence of pseudorandom function families

✓ • mapping the boundary of efficient learnability

✓ • start hardness of learning based on public-key cryptography (trapdoor 1-way permutations)

Today: • more hardness of learning based on public-key cryptography (trapdoor 1-way permutations)

→ our hardness assumption: "discrete cube roots" are hard to compute

• using this to show that even "simple" poly(n)-size Boolean circuits are hard to learn

→ more precisely: poly(n) size Boolean formulas; equivalently, $O(\log n)$-depth Boolean circuits

Questions?

(Reminder: final) → next Fri: 12/15 9:00 am – 10:30 closed book/notes.

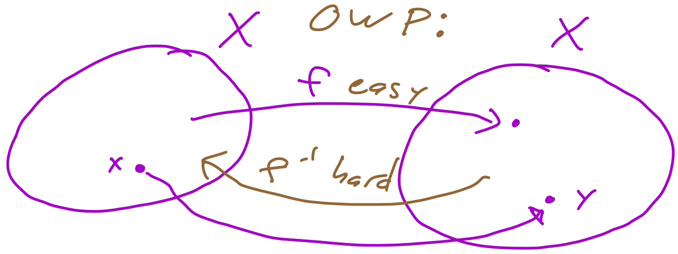
Recall setup:

Def: A permutation of finite set X:

bijection $X \longrightarrow X$

one-to-one + onto

OWP:



$\forall y \in X$, $\exists$ unique x s.t. $f(x) = y$

i.e. $f^{-1}: X \to X$ is well-defined

(Informal) "one-way permutation" (OWP) on $X = \{0,1\}^n$

a perm. $f: \{0,1\}^n \to \{0,1\}^n$ s.t.

- there's a poly(n) time alg. to compute f, but
- any poly(n) time alg. can't compute f correctly even on a $\frac{1}{\text{poly}(n)}$ frac. of inputs.

(Informal) Trapdoor OWP (TOWP):

a OWP, but one s.t. if have secret "trapdoor" info, then it's poss. to compute f^{-1} in poly(n) time.

Ex: trapdoor info is prime factorization p, q of $N = p \cdot q$. ; if knew p, q, could invert f, but without knowing p, q, no poly(n)-time alg. can invert f.

Public-Key Crypto: PKC

way for Bob to send msg securely to Alice, even in presence of Eve (sdropper) who hears his whole message.

Setup: (picks p, q)

- Alice creates TOWP f ("encryption fn"), she knows how to compute f^{-1} (knows p, q trapdoor info)
- Alice publishes alg. for computing f → (publishes $N = pq$). Doesn't publish secret trapdoor info.

To communicate:

- To send msg y to Alice, Bob computes $f(y)$ & sends it to Alice.
- Alice applies f^{-1} to $f(y)$, decrypts to get y.
- Eve? Sees: $f(y)$

  alg to compute f

  but lacks trapdoor info so can't invert f to get y.

Key for us: Given TOWP/PKC,

decryption fn f^{-1} must be hard to learn, why?

under unif. dist. on domain X

Suppose have learning alg A.

Eavesdropping world

Eve sees $f(y)$, wants to compute $f^{-1}(f(y)) = y$

Key

Eve can construct pairs for herself!!!

Learning world:

Learner wants to learn f^{-1}, gets pairs

$(x, f^{-1}(x))$

$x \sim$ unif dist over X.

A uses, comes up w/ hyp h highly acc. for f^{-1}.

Eve picks unif. $z \sim X$, computes $f(z)$:

$(f(z), z)$ distributed just like $(x, f^{-1}(x))$

z unif, so $f(z)$ uniform, b/c f is a permutation

x uniform

So if a poly(n) time learning alg existed, E could use to get high-acc. hyp. for f^{-1}, + eavesdrop/break TOWP.

Instantion of above:

a specific hard-to-learn $\mathcal{C}$, based on (presumed) hardness of "discrete cube roots"

Let $N = p \cdot q$, p, q both $\frac{n}{2}$-bit primes,
both $\equiv 2 \mod 3$.

(Ex: $p = 17$, $q = 5$ $N = 85$)

Def: $\mathbb{Z}_N = \{1, \ldots, N\}$ "mod N"

$70 + 69 = 139 \equiv 54 \mod 85$

Def: $\mathbb{Z}_N^* = \{ j \in \mathbb{Z}_N : \gcd(j, pq) = 1 \}$
↳ values rel. prime to N.

$\mathbb{Z}_N^* = \{1, 2, 3, 4, 6, 7, 8, 9, 11, 12, 13, 14, 16, 18, \ldots 84\}$.

$|\mathbb{Z}_N^*| = ?$ It's $\varphi(N) = (p-1) \cdot (q-1)$
$= pq - q - p + 1$.

$p - 1 = 16$
$q - 1 = 4$ $\varphi(85) = 64$

Fact: $\mathbb{Z}_N^*$ is a group under $\times$ (mod N):

- if, $a, b \in \mathbb{Z}_N^*$, then $ab \pmod N$ is $\in \mathbb{Z}_N^*$;

• for any $a \in \mathbb{Z}_N^*$, there's a unique elt $a^{-1} \in \mathbb{Z}_N^*$ st $a \cdot a^{-1} = 1 \pmod N$

$$a = 2 : \quad a^{-1} = 43$$

$$a \cdot a^{-1} = 86 \equiv 1 \bmod 85.$$

<u>Fact:</u> for any finite group G + any $g \in G$,

$$g^{|G|} = 1.$$

$$|\mathbb{Z}_N^*| = \varphi(N) = (p-1)(q-1)$$

☆ So for any $a \in \mathbb{Z}_N^*$, $\boxed{a^{\varphi(N)} \equiv 1 \bmod N.}$ ☆

$4^{64} \equiv 3^{64} = 2^{64} \equiv 1 \bmod 85$

etc.

→ means $a^{\varphi(N)-1}$ is a^{-1} $\forall a \in \mathbb{Z}_N^*$

<u>Technical Claim:</u> For $N = p \cdot q$ as above, have $\underline{(p, q \equiv 2 \bmod 3)}$

$$2\varphi(N) + 1 \equiv 0 \bmod 3, \text{ i.e.}$$

$\boxed{2\varphi(N) + 1 = 3d}$ some integer $d = \dfrac{2\varphi(N)+1}{3}$

$$d = \frac{2(p-1)(q-1)+1}{3}.$$

<u>Pf</u>: $\varphi(N) = \underbrace{(p-1)}_{1 \bmod 3}\underbrace{(q-1)}_{1 \bmod 3}$ so $2\varphi(N)$ is $2 \bmod 3$, $2\varphi(N)+1$ is $0 \bmod 3$. ∎

Interesting

Claim: For $N = pq$ as above, the function

$$f_N(x) = x^3 \bmod N$$

is a permutation of $\mathbb{Z}_N^*$.

Pf: Here's the inverse f_N^{-1}: it's

$$\boxed{f_N^{-1}(y) = y^d \bmod N.}$$

Check: for any $x \in \mathbb{Z}_N^*$,

$$f_N(x)^d \bmod N =$$

$$(x^3 \bmod N)^d \bmod N =$$

$$(x^{3d} \bmod N) \bmod N = x^{3d} \bmod N$$

$$= x^{2\varphi(N)+1} \bmod N$$

$$= (x^{2\varphi(N)} \bmod N) \cdot (x \bmod N)$$

$$= ((x^2 \bmod N)^{\varphi(N)} \bmod N) \cdot (x \bmod N)$$

$$= (1 \bmod N) \cdot x \bmod N = x.$$

So y^d is indeed $f_N^{-1}(y)$, so

$f_N(x) = x^3$ is a permutation of $\mathbb{Z}_N^*$ (it maps $\mathbb{Z}_N^*$ to itself + has a well-defined inverse). $\blacksquare$

So f_N is a permutation of $\mathbb{Z}_N^*$.

p, q is trapdoor info:

given p, q, easy to compute

$$d = \frac{2(p-1)(q-1)+1}{3}$$

Have to make hardness assumption to get OW-ness: here it is:

<u>Discrete Cube Root Problem:</u>

<u>Input:</u> 2 #s N, y, where

- $N = p \cdot q$, p, q two primes both $\equiv 2 \bmod 3$
- $y \in \mathbb{Z}_N^*$ (so $y \equiv x^3 \bmod N$ for some unique $x \in \mathbb{Z}_N^*$)

<u>Output</u>: x, i.e. $f_N^{-1}(y)$, i.e. value s.t.

DCRHA

Discrete Cube Root Hardness Assumption:

(classical)

For any poly $p(n)$, there's <u>no</u> $p(n)$-time alg. A s.t. if A given N, y where

- $N = p \cdot q$ for p, q unif. random $\frac{n}{2}$-bit primes $\equiv 2 \bmod 3$,
- y unif rand. element of $\mathbb{Z}_N^*$ (so $y = x^3 \bmod N$ unique x)

then A outputs x w. prob. $\geq \frac{1}{p(n)}$.

$\boxed{\text{if could factor } N, \text{ this is easy! } d = \frac{2(p-1)(q-1)+1}{3}.}$

Let $\mathcal{C} = \{ f_N^{-1} \}$ N ranges over $p \cdot q$ as in DCRHA

maps $\{0,1\}^n \to \{0,1\}^n$

For each N, let $\mathcal{D}_N$ = unif dist over $\mathbb{Z}_N^*$.

Is there a poly-time alg A which "PAC learns" $\mathcal{C}$? Suppose yes.

Then for any c in $\mathcal{C}$ + any $\mathcal{D}$, it works;

succeed if target is f_N^{-1} + $\mathcal{D}$ is $\mathcal{D}_N$.

Alg A, given N + access to rand

$\underline{\underline{(x, f_N^{-1}(x))}}$ ex. $x \sim \mathcal{D}_N$,

with prob. $\geq 1-\delta$ gives ε-acc. h for f_N^{-1}.

But such an A contradicts DCRHA:

given N, y (inputs for DCR problem),

run A using $\mathcal{D}_N$ as dist.: draw unif. $z \sim \mathcal{D}_N^*$,

compute $f_N(z) = z^3 \bmod N$, + use

$(f_N(z), z)$ as our desired

$(x, f_N^{-1}(x))$ pair.

Get an h (w.p. $\geq 1-\delta$) s.t.

$h(y) = f_N^{-1}(y)$ with overall prob. $\geq \underline{1}-\underline{\delta}-\underline{\varepsilon}$.

Can "Booleanize" by considering a bona fide $\hat{\mathcal{C}}$ concept class by looking at i^{th} bit of c for each $i = 1, \ldots, n$

(next time)
