SIDE

Preprocessing attacks

- Defn
- Hellman tables
- Rainbow tables
- More apps

* Project presentations are ≈2 weeks away!
* Proj report due 4 weeks

(side below) SIDE

Motivation (Hellman 1980)

There are a small # of crypto tools used everywhere

↳ AES128, SHA256, ECDSA (P-256), DH (P-256)

IDEA: Breaking AES128

Offline [one time]: cook up data structure in advance — Time $\approx 2^{128}$

Online [many times]: Decrypt cts faster later using struct (e.g. AES-GCM) $\approx 2^{64}$

⇒ Preprocessing cost is obscene ... still ∃ applications other than AES

NOTES

Why study?

- Crypto attack with nice theory
- Works in pract
- Solves problem people care about
  - ↳ Widely used "Rainbow tables"
  - ↳ Used to break DES today

Function-inversion problem

Given: * A function $f: [N] \to [N]$ (oracle access only)

(Here $[N] = \{1, \ldots, N\}$)

* A value $y \in [N]$

Find: A value $x \in [N]$ s.t. $f(x) = y$, if one exists

[There are preproc attacks for CRHF, Dlog, Factoring, ...]

Examples: $f_{AES}(k) := AES(k, \text{"00000"})$ $N = 2^{128}$

PRF on zero string

$f_{SHA256}(msg \in \mathcal{M}) := SHA256(msg)$ $N = |\mathcal{M}|$

Used for "cracking" unsalted pw hash; $\mathcal{M}$ = {popular passwords}

Preprocess f_{AES} once, break many keys at reduced cost

Again, in reality preprocessing matters

SIDE

Preproc attack on Fn inversion

In preproc attack, adv is a pair (A_0, A_1)

Offline $A_0^f() \to st_f$

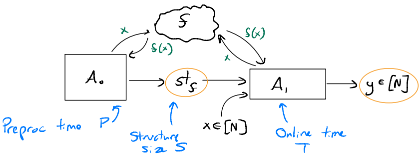
Online $A_1^f(x \in [N]) \to y$

$$\Pr\left[y = f(x) \;:\; \begin{array}{l} f \xleftarrow{R} \mathrm{Funs}[N,N] \\ st_f \leftarrow A_0^f() \\ x \xleftarrow{R} [N] \\ y \leftarrow A_1^f(x) \end{array} \right] \geq \frac{1}{2}$$

Want to minimize: Space $S = |st_f|$

Time T = # of queries A_1 makes

SIDE



Measure running time as # queries to S.

SIDE

What we know:

[I'll ignore $\log N$ factors.]

Brute-force search: $S = O$; $T = N$

Store all inverses: $S = N$; $T = O$

Hellman: for random fn $S = T = N^{2/3}$ All we need for most crypt. apps

Fiat-Naor: for all fn $S = T = N^{3/4}$ non-crypto apps

For $N = 2^{128}$

2^{128} time

$\geq 2^{128}$ storage

$S = T = 2^{86}$

$S = T = 2^{96}$

Consequence

* For dictionary of 2^{40} p.w. $\Rightarrow S = T \simeq 2^{27}$ p.w. $\}$ Big savings $\simeq 8000\times$
* For DES cipher $N = 2^{56} \Rightarrow S = T = 2^{40}$ $\simeq 64{,}000\times$ speedup

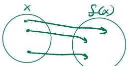
Can we do better?

Yao: $ST \geq N$ $\Rightarrow$ "$S \leq N^{1/2}$; $T \geq N^{1/2}$" $S = 2^{64}$; $T \geq 2^{64}$

Main Q: Is there a fn-inv alg using $S = T = N^{1/2}$ (random fns or all) $S = T = 2^{64}$?

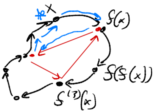
Hellman Tables

Warm-up: Inverting f when f is "one to one" (f has "no collisions")



Preproc $A_0^f()$:

* View f as a graph
* Will be a union of cycles



* Store "Backpointers" every T steps, output as advice
  ↳ N/T pointers; space $\simeq N/T$

Online $A_1^f(x)$:
* Apply f until hitting back-ptr, follow it [Takes T calls to f]
* Once you reach x, element before it is inv

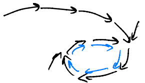
Take $T = \sqrt{N} \Rightarrow \text{Space} = \frac{N}{T} = \sqrt{N}$

SIDE

Random Functions

- PRF, Hash, etc. behave like random fns. — NOT one-to-one
  ($\exists$ many k_1, k_2 s.t. $AES(k_1, 000) = AES(k_2, 000)$)
- Cycle strategy breaks

Graph of f



"Usually" $\nexists$ set of $\sqrt{N}$ "chains" of len $\sqrt{N}$ that cover all points.

↳ Can only invert points "covered by" a chain

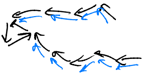
General functions

* There will likely be $N^{1/3}$ chains of len $N^{1/3}$ that cover $N^{2/3}$ points total
  ↳ Only a $\frac{N^{2/3}}{N} = \frac{1}{N^{1/3}}$ fraction of points! Succeed $1/N^{1/3}$ of the time. ☹
* Hellman's cleverness: "Rerandomize f" $N^{1/3}$ times → Expect to succeed w.p. $\approx 1/2$.

Define $N^{1/3}$ flavors of f: $f_1, f_2, \dots, f_{N^{1/3}}$

$$f_i(x) := g_i(f(x))$$

Invertible 1-to-1



Graph of f_1

Analysis

If you can invert f_i can invert f: $f_i^{-1}(x) = g_i^{-1}(f^{-1}(x))$

$$g_i(f_i^{-1}(x)) = f^{-1}(x).$$

Alg: Use cycle alg $N^{1/3}$ times, once per flavor of f. One should work.

Space $= N^{1/3}$ ptrs $\cdot$ $N^{1/3}$ flavors $= N^{2/3}$

Time $= N^{1/3}$ steps $\cdot$ " " $= N^{2/3}$

Rainbow tables - Used in practice [Oechslin'03] SIDE

- Same time & space as Hellman
- Innovation is to reduce # of disk accesses
  - ↳ Hellman needs one per step

$$\bullet \xrightarrow{f_1} \xrightarrow{f_2} \xrightarrow{f_3} \xrightarrow{f_4} \longrightarrow \cdots \xrightarrow{f_{N^{1/3}}} \bullet$$

Run $N^{1/3}$ steps and then follow back-pointer.

More applications

Substring search find query string $\sigma \in \{0,1\}^{\ell}$ in big str $D \in \{0,1\}^N$

$$f_D(i) = D[i : (i+L)]$$

Normally takes N time or N space

Applying fn inversion: $N^{3/4}$ probes to D, $N^{3/4}$ space

Compression [HIW'23]

MIDDLE

Given: Long string $D \in \{0,1\}^n$
Is there a length-ℓ string σ s.t. $Eval(\sigma) = D$?

Say that D is the truth table of a ckt.
$Eval()$ takes a ckt as input and evals it everywhere.

$$f_{Eval}(\alpha) := D$$

With no preproc: 2^n time; w/ fn inversion: $S = T = 2^{\frac{3n}{4}}$