lassically: $T \cdot S \geq N$

uantum (lower bound): $T^2 \cdot S \geq N$

n we give a nstruction for this? Tighter lower ound?

Time / Space Tradeoffs for Symmetric Cryptanalysis

DES block cipher – developed by IBM (initally key was 64 bits)

↳ NSA try to convince IBM to reduce key size to 48 bits to enable brute force

↳ Eventually compromised on 56-bit design

Question: what is the cost of breaking DES? Let N denote number of keys.

Exhaustive search: given several message-ciphertext pairs, try all of the keys

time complexity: $O(N)$ space complexity: $O(1)$

Table lookup: suppose we have known message-ciphertext pair and precomputed table of all keys (→ e.g. CPA attack)

time complexity: $O(\log N)$ space complexity: $O(N)$

More general problem: inverting a one-way function (permutation)

↳ Hellman introduced notion of time-memory tradeoff with Hellman tables (CS255)

random starting points

$$\left\{\begin{array}{l} \boxed{x_1} \longrightarrow f(x_1) \longrightarrow f(f(x_1)) \longrightarrow \cdots \longrightarrow \boxed{f^{(t)}(x_1)} \\ \boxed{x_2} \longrightarrow f(x_2) \longrightarrow f(f(x_2)) \longrightarrow \cdots \longrightarrow \boxed{f^{(t)}(x_2)} \\ \vdots \\ \boxed{x_m} \longrightarrow f(x_m) \longrightarrow f(f(x_m)) \longrightarrow \cdots \longrightarrow \boxed{f^{(t)}(x_m)} \end{array}\right.$$

↳ values in Hellman table

Key observation: suppose all elements in table are distinct

↳ success probability of inverting OWF on random input is mt/N

↳ compare with exhaustive search: t/N and table lookup: m/N

↳ constant fraction overlaps ⇒ success prob. reduced by same constant factor

How much overlap should we expect in the table entries? Suppose f is modeled as a random function. How much of the domain can we expect to cover?

Let X_{ij} denote the $(i,j)^{th}$ entry in the table. Let A be the set of values in the table. Then:

$$|A| = \sum_{i=1}^{m}\sum_{j=1}^{t} I\{X_{ij} \text{ is new}\}$$

$$\Pr_{x}[x \in A] = \frac{E[|A|]}{N} \approx \frac{1}{N}\sum_{i=1}^{m}\sum_{j=1}^{t} e^{-ijt/N}$$

Now, $E[|A|] = \sum_{i=1}^{m}\sum_{j=1}^{t} \Pr[x_{ij} \text{ is new}]$ (linearity of expectation, expectation of indicator)

Then,

$$\Pr[x_{ij} \text{ is new}] \geq \Pr[x_{i1}, \dots, x_{ij} \text{ are all new}]$$ (all elements in row are new)

$$= \Pr[x_{i1} \text{ is new}]\,\Pr[x_{i2} \text{ is new} \mid x_{i1} \text{ is new}] \cdots \Pr[x_{ij} \text{ is new} \mid x_{i1} \cdots x_{ij-1} \text{ is new}]$$

$$= \frac{N-|A_i|}{N} \times \frac{N-|A_i|-1}{N} \times \cdots \times \frac{N-|A_i|-j+1}{N}$$

$|A_i|$ is new elements in first i rows

$$\geq \left(\frac{N-it}{N}\right)^j$$

$$\therefore \Pr[x \in A] \geq \frac{1}{N}\sum_{i=1}^{m}\sum_{j=1}^{t}\left(\frac{N-it}{N}\right)^j = \frac{1}{N}\sum_{i=1}^{m}\sum_{j=1}^{t}\left(1-\frac{it}{N}\right)^{N\cdot\frac{j}{N}} \approx \frac{1}{N}\sum_{i=1}^{m}\sum_{j=1}^{t} e^{-ijt/N}$$

Not much gained when $mt^2 \gg N$ (contribution very small) $\rightarrow$ setting $m = t = \sqrt{N}$ <u>not</u> sufficient (too many collisions)

When $mt^2 \ll N \Rightarrow \Pr[x \in A] \approx mt/N$

$\Rightarrow$ set $m = t = N^{1/3}$ $\Rightarrow$ succeed w.p. $\approx N^{-1/3}$

↳ construct $N^{1/3}$ such tables to succeed with constant prob.

$\Rightarrow$ time $N^{2/3}$, space $N^{2/3}$ – 56-bit DES $\Rightarrow$ 38-bits effective security (\$10 cost in 1978)

precomputation is $O(N)$ time and $O(N^{2/3}$ space)

What if we could use $O(N)$ space during precomputation?

↳ construct table of size $O(\sqrt{N})$ to allow online inversion in time $O(\sqrt{N})$

What if f is a permutation (with a single cycle)? – discrete log

↳ again admits a $\sqrt{N}, \sqrt{N}$ time-memory trade-off

Double DES and Meet-in-the Middle Attacks

Double DES: $DES_2((k_1, k_2), x) : DES(k_2, DES(k_1, x))$ looks like 112-bit keys

time-memory tradeoff (with known plaintext m)

- (pre)-compute table $DES(k_1, m)$ for all keys k_1 (size 2^{56})
- given ciphertext c, evaluate $DES^{-1}(k_2, c)$ for all k_2

$$c = DES(k_2, DES(k_1, x))$$

$$DES^{-1}(k_2, c) = DES(k_1, x)$$

(time 2^{56})

$\Downarrow$

2^{57}-cost attack

What if we have only $w < 2^{56}$ memory

↳ then partition the space into blocks of size w and repeat attack for each block

↳ requires $\left(\frac{N}{w}\right)(w + N) = N + \frac{N^2}{w}$ time and w space

Another approach: reduce meet-in-the middle to a collision search problem (reduce space requirements)

Meet in the middle attack: find (k_1, k_2) such that

$$\underbrace{DES^{-1}(k_2, c)}_{f_2(k_2)} = \underbrace{DES(k_1, m)}_{f_1(k_1)}$$

Define a function $f(\underset{\text{key}}{k}, \underset{\text{index}}{i}) = f_i(k)$

observe that

$f(k_2, 2) = f(k_1, 1)$ which is a collision for f

↳ goal: build collision-finding algorithm (also independently important)

Abstract goal: suppose we have a function $f: S \to S$ and we want to find a collision ($|S| = N$)

- Naive strategy: compute $f(x_1), \ldots, f(x_m)$ for random $x_1, \ldots, x_m$ until collision is found
  - ↳ birthday bound: time $O(\sqrt{N})$ and space $O(\sqrt{N})$
- Using a cycle finding algorithm: "rho method"

  start with random x and compute

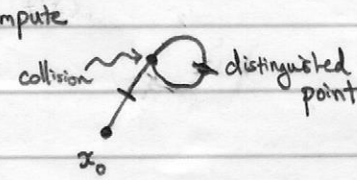
  $$x,\ f(x),\ f(f(x)),\ \ldots,\ f^{(m)}(x)$$

  if f looks like a random function, then after $\sqrt{N}$ applications, will have a collision (e.g., cycle)
- Cycle detection via fast pointer / slow pointer [Floyd]

  choose random $x_0 = x'_0$ and compute

  $x_i \leftarrow f(x_{i-1})$

  $x'_i \leftarrow f(f(x'_{i-1}))$



- How to go from distinguished point to collision:
  1. Compute length of cycle $O(\sqrt{N})$ time
  2. Advance further pointer until it is equidistant from the distinguished point
  3. Advance both pointers in sync – will collide at some position

Rho algorithm: $O(\sqrt{N})$ time, $O(1)$ space for finding collisions

- Naive parallel extension to rho algorithm does not provide compelling speed-up
- Suppose we have m processors each running independent execution of rho algorithm
  - After each processor has evaluated f a total of k times, probability there is no collision

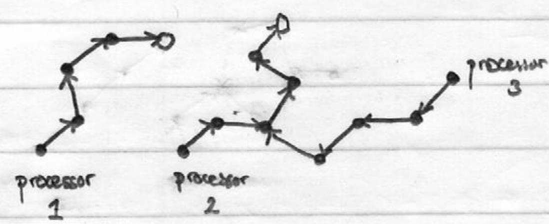
$$\left[\left(1-\frac{1}{N}\right)\left(1-\frac{2}{N}\right)\cdots\left(1-\frac{k-1}{N}\right)\right]^m \lesssim \left(1-\frac{k}{N}\right)^{mk} \approx e^{-k^2 m/N}$$

  ↳ expression is for single processor finding collision over domain of size N/m

  - Collision after $k = \sqrt{\frac{N}{m}}$ steps $\Rightarrow$ only $\sqrt{m}$ speed-up despite m processors

Parallel collision search: getting <u>linear</u> speedup from multiple processors [van Oorschot and Wiener]

- each processor chooses a random point and evaluates f until hitting a "distinguished" point



(restart if trail too long)

- observation: after $O(\sqrt{N})$ total points, there will be a collision

  $\Rightarrow$ after each processor has taken $O(\sqrt{N}/m)$ steps

- choose distinguished points so trails expected to be long - will not require too much space
- gives collision-finding algorithm with small space!