6.857 Recitation 04 Feb 26

Improved Generic Algorithm for 3-Collisions (Joux, Lucks)

Pset 2, Problem 2 (857coin).

Given D = (previous hash, hash of group member names, timestam

Find distinct nounces n_1, n_2, n_3 s.t.

$SHA256(D \| n_1) = SHA256(D \| n_2) = SHA256(D \| n_3) \pmod{2^d}$

Goal:

Given <u>random map</u> $H: [0, N-1] \to [0, \cdots, N-1]$.

find distinct x_1, x_2, x_3 s.t.

$H(x_1) = H(x_2) = H(x_3)$

Remarks:

(1) <u>random map</u>: random oracle, truly random function from $[0, \cdots, N-1] \to [0, \cdots, N-1]$.

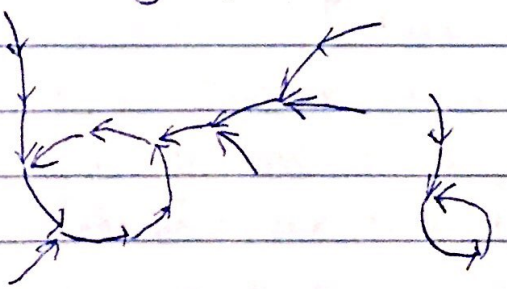
(2) $2^d \ll$ space of nounces, should increase our chance.

(3) expected fractions of points with exactly k distinct preimages is $e^{-1}/k!$. So $\approx 8\%$ points has at least 3 distinct preimages. But we need to find the preimages.

(4) <u>functional graph</u> G_H.

① nodes are all values in $[0, \cdots, N-1]$.

② direct edge from node x to node y iff $H(x) = y$



Components: $\frac{1}{2}\log N$. Among them, there is one giant component. In addition, this giant component

Contains a giant tree.

giant component size: $2N/3$

giant tree size: $N/3$.

So Floyd's 2-Finger algorithm outputs the root of the giant tree w.p. $1/3$.

(5) Find r-collision, for small r, needs to evaluate the map $\Omega(N^{(r-1)/r})$.

And you can easily achieve this by evaluating and storing $O(N^{(r-1)/r})$ random points, sorting and sequential scan. use space $O(N^{(r-1)/r})$

For 2 collision, time $O(N^{1/2})$, space $O(1)$.

Space & Parallelizable !!

(6) Motivations: cryptanalysis of SHA-3 candidates.

Alg 1:

Step 1: store N^{α} random $(x, H(x))$ pairs

Img	$H(x)$	. . .
Pr1	x	- - -
Pr2		

Step 2: evaluate N^{β} random $(y, H(y))$ pairs.

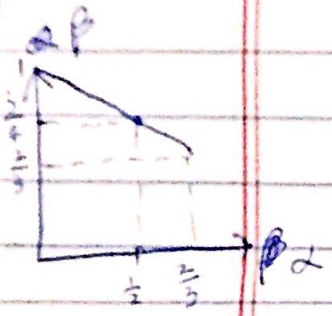
If $H(y)$ appears in Img.

Put y to Pr2 if Pr2 is empty

Otherwise we find a 3-collision.

Analysis: (1) $\alpha \leq \beta$ (space $\leq$ time)

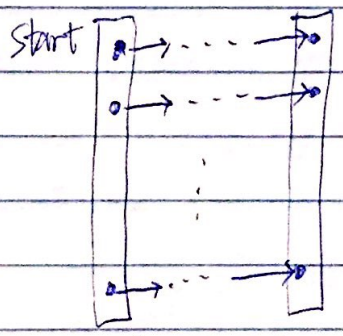
(2) in second step, each $H(y)$ hits Img w.p. N^{α}/N. So $N^{\alpha+\beta-1}$ collisions in Img



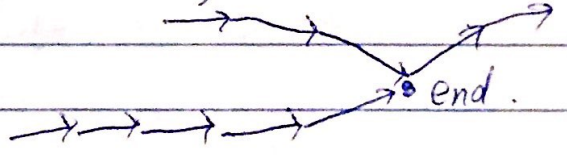
By Birthday paradox after $N^{\alpha/2}$ hits, we expect a double hit.
$\alpha+\beta-1 = \alpha/2 \Rightarrow \alpha+2\beta=2$.

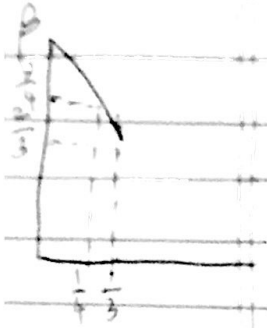
Alg 3:
Step 1: Find N^α collisions.
Step 2: Evaluate N^β more points and compare to those from Step 1. If hit, we have a 3-collision.
Analysis: $\alpha+\beta=1$

Step 1: (In Alg 2, call F2FA on random reshuffle of G_H
(rain bow table) ① Compute N^α chain, each of length $N^{1-\alpha}$



② Find N^α 2-collision. (length)
(1) pick random start. and build $N^{1-\alpha}$ chain
(2) If any part of the chain collide end array we have a 2-collision.



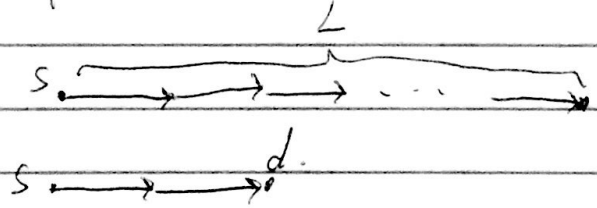


space N^α.
time in step 1: $N^{\alpha+r} \le N^\beta$
expected number of collisions: $N^{2\alpha+2r-1}$.
Let $r = (1-\alpha)/2$.
We have $\alpha+\beta=1$ for any $\alpha \le \frac{1}{3}$.

What if I have GPU? Parallelizable.
$N^{1/3}$ processor. each $O(1)$ space. $\alpha N^{1/3}$ time.
N^θ processor, each $O(N^{1/3-\theta})$ space $O(N^{2/3-\theta})$ time.

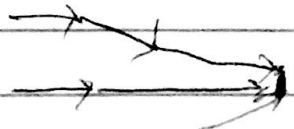
Idea (from 2-collision) distinguished points:
set of points which one can easily test membership.
e.g $[0, \cdots, M-1]$.

Step 1: each processor starts from a random point s, iteratively apply H until a distinguished point d.



transmit (s, d, L) to processor $d \pmod{N_p}$.

Step 2: If d appears at least 3 times.
recompute the chain and find collisions.



$N^{1/3}$ processors. $\Rightarrow$ $N^{1/3}$ chain, each of length $N^{1/3}$.
$M = N^{2/3}$

$H(a)$.
$H(b)$.
$H(c)$.

We compute about $N^{2/3}$ points.
It's likely that we have a 3-collis
x_1, x_2, x_3