Finding valid Roots

The following details how to find a set of valid roots for a given set of hash functions

Algorithm

This algorithms finds a CN0 given a set of hash functions and in turn is able to find a valid root value.

Overview

- Given a set of hash functions
- Generate a set of hash functions (UHF, LHF, PHF)
- Calculate the combined hash function CHF by adding the set of hash functions (UHF + LHF + PHF)
- Calculate a possible root by checking all possible combinations for:
  - First 2 values (AB)
  - First 3 values (AC, BC)
  - First 4 values (AD, BD, CD)
- Generate a CN0 by plugging in suitable values generated for ABCD
- Once CN0 is obtained use the inverse of UHF to find the actual root.

Key

X[n]->{Y} Set of valid numbers for Y if X == n

Variables

Root - ABCD Combined Hash - (CHF) { x ∈ R | X>=0 ^ X<10}

Example

Valid Set of Hashes

UHF -10,-2,-7,-8 LHF +4,-8,+5,-10 PHF +3,+2,-9,+8 CHF =[7,2,9,0]

Find the possible values for AB - [A,B,-,-]

AC - [A,-,C,-] BC - [-,B,C,-]

AD - [A,-,-,D] BD - [-,B,-,D] CD - [-,-,C,D]

AB

CHF = [7,2,9,0]

- Find the difference between A and B A - B = 7 - 2 = 5;
- From 0, create a set of numbers where each number is incremented by the difference (5), and do this one for each lock. So 5 locks = 5 times.

{0,5,10,15,25}

- wrap values around 10 and take out duplicate values

{5,0}

- Calculate the inverse, and this is the set of numbers that can be used if A == 0

A[0]->{B} - {1,2,3,4,6,7,8,9}

AC

CHF = [7,2,9,0] diff = 7-9 = 2

({0,-8,-6,-4,-2} inverse) == ({0, 2,4, 6,8} inverse) ({2,4,6,8,0} inverse) A[0]->C - {1,3,5,7,9}

BC

CHF = [7,2,9,0] diff = 2 - 9 = -7

({0,-7,-14,-21,-28} inverse) == ({0,3,6,9,2} inverse) B[0]->C - {1,4,5,7,8}

AD

CHF = [7,2,9,0] diff = 7 - 0 = 7

({0,7,14,21,28} inverse) == ({0,7,4,1,8} inverse) A[0]->D - {2,3,5,6,9}

BD

CHF = [7,2,9,0] diff = 2 - 0 = 2

({0,2,4,6,8} inverse) B[0]->D - {1,3,5,7,9}

CD

CHF = [7,2,9,0] diff = 9 - 0 = 9

({0,9,8,7,6}) C[0]->D - {1,2,3,4,5}

Calculating CN0

CN0 =

Finding AB

- Plug a value into A (Can start with 0) and generate the values that B give the functions produced above

A[0]->{B} = {1,2,3,4,6,7,8,9} -- Add 1 to both sides -- A[1]->{B} = {2,3,4,5,7,8,9,0}

So if A = 0, B can be any of the values in the set. Lets choose 2 B = 2

CN0 = 02__

Finding C

- To find C, we need to find a value where the numbers in the set AC and BC both work. To do this we find the intersection of the two sets. Since we've chosen 0 and 2 as out A and B we have to plug those values into the function

C = A[0]->{C} ∩ B[2]->{C} = {1,3,5,7,9} ∩ {3,6,7,9,0} = {3,7,9}

- So C can be anything from {3,7,9}. Lets pick 3 for this example

CN = 023_

Finding D

- Now we repeat the process for D, plugging in the values and intersecting the sets that are produced

D = A[0]->D ∩ B[2]->{D} ∩ C[3]->{D} = {2,3,5,6,9} ∩ {3,5,7,9,1} ∩ {4,5,6,7,8} = {5}

- There is only one valid option for D if ABC = 023, and that is 5, so D = 5

CN0 = 0235

Finding Root

Given that CN0 = 0235 and UHF = -10,-2,-7,-8, we can simply do the inverse of UHF to find the original root

UHF = [-10, -2, -7, -8] UHF' = [0, 2, 7, 8]

CN0 = [0, 2, 3, 5]

ROOT = [0, 4, 0, 3]

Checking Root and Functions

- To test if the result is correct we take the root and apply UHF to it to get the first CN
- Then we can produce CN1, CN2, CN3, CN4 using CN0 and the combined hash function

CHF = [7,2,9,0]

CN0 - [0, 2, 3, 5] CN1 - [7, 4, 2, 5] CN2 - [4, 6, 1, 5] CN3 - [1, 8, 0, 5] CN4 - [8, 0, 9, 5]

There are no repeats in CN0 - CN4 so the root is valid!