Is there Fixed Points exist in Hash Functions??

Presented by,
Rajkumar Pandi

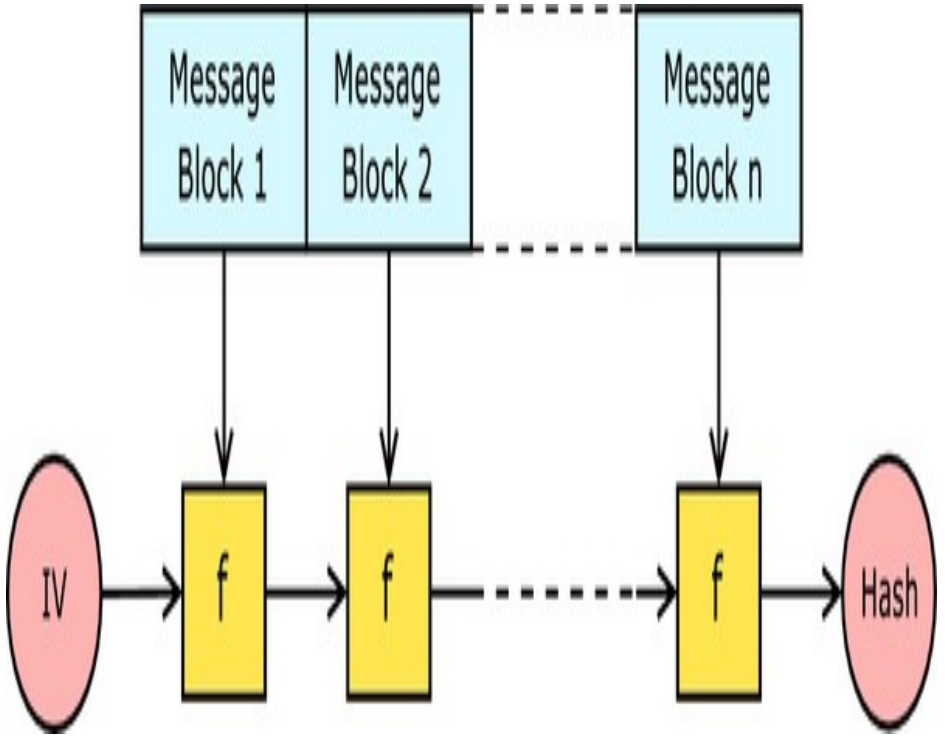
Why its important?

Before using hash function there are important properties that the final digest from the hash function should have:

1)Digest should not have any correlation with the message

2)For distinct messages the digest should be uncorrelated.

Merkley Damgard Iterated compression function



If there exist a fixed point for a given hash function H(x) =x is true then Functional H(f(IV,m)) = f(IV0,m0) is also true where f(IV,m) is the compression function.

If there exist a fixed point in the hash function, then there exist a fixed point in the compression function as well.

Problem Statement

- Can string's of hash function's hash be its own value?

  $$H(x) = x$$

  Taking hash function as functional

  $$H(f(IV, m)) = f(IV0, m0)$$

  where f(IV, m) is a compression function

- In other words, it is possible to detect hash collisions in cryptographic hash functions

Motivation

- Still software developers from famous company are unaware of secure hash function.

  Example: When a .exe file tries to load a library with the api call LoadLibrary , there will be a look up for the checksum at the dll's PE Header which is based on CRC32

  http://www.codeproject.com/Articles/19326/An-Analysis-of-the-Windows-PE-Checksum-Algorithm

  I will show why its really a bad practice in the following slides

Existence of fixed point

Mathematical Proof: $H(x) = x$

W.K.T.

1)f is a cryptographic hash function and the output will always be a string of fixed length except for whirlpool hash function. Therefore we got 2^n candidate for x

2)Hash functions aims to distribute their results over its range as uniformly as possible to minimize hash collisions.

Model f as uniform distribution,

$$P\left(f\left(x\right)=x\right)=\frac{1}{2^n}$$

x is not fixed point of f then,

$$P\left(\forall x: f\left(x\right)\neq x\right)=\left(1-\frac{1}{2^n}\right)^{\left(2^n\right)}$$

W.K.T $\lim_{i}^{\infty}\left(1-\frac{1}{\infty}\right)^{(\infty)}=\frac{1}{e}$

so, $\lim_{n}^{\infty} P\left(\forall\, x: f\left(x\right)\neq x\right)=\frac{1}{e}$

Calculate the opposite that if x is not fixed point of f,

$$\lim_{n}^{\infty} P\left(\exists x : f\left(x\right)=x\right)=1-\frac{1}{e}$$

Therefore the probability that a hash function has fixed point is approximated by

$$1-\frac{1}{e}\approx 0.6321$$

This is a theoretical proof that a fixed point exist in a hash function.

Verification of the proof

- We choose 5 commonly used hash function 1)CRC32,2)MD5 3)RIPEMD1604)DSA & 5)Whirlpool
- Given a starting string S,calculated the hash by using each of the above mentioned hash functions on every possible output of the hash and look for matches
- Description

  Step 1: Given a string S.

  Step 2: Compute H(S) =S1

  .

  .

  .

  .H(S9999)=S10000

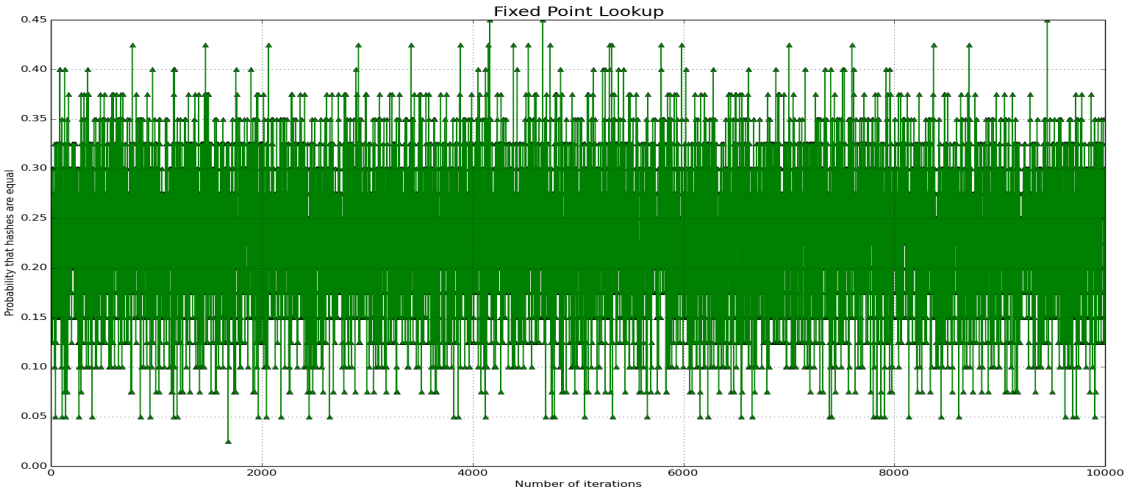
  This goes for 10000 iterations.

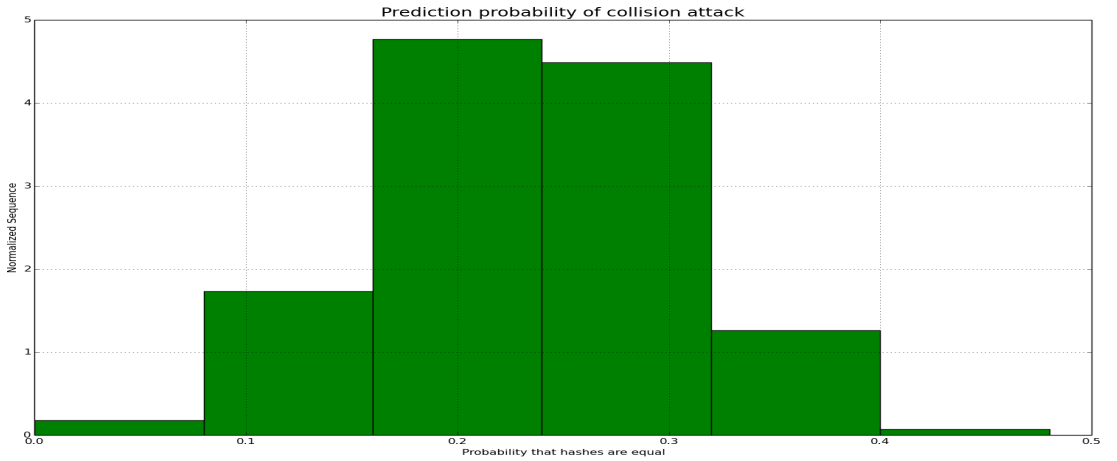
  Step 3: Calculate the probability of the consecutive hashes based on hamming distance between them and plot it with number of iterations on x axis with the probability that two consecutive hashes are equal on y axis.

  Step 4: Compare the probability of the consecutive hashes with the calculated Shanon Entropy of the hashes. All the results are shown in the following slides.

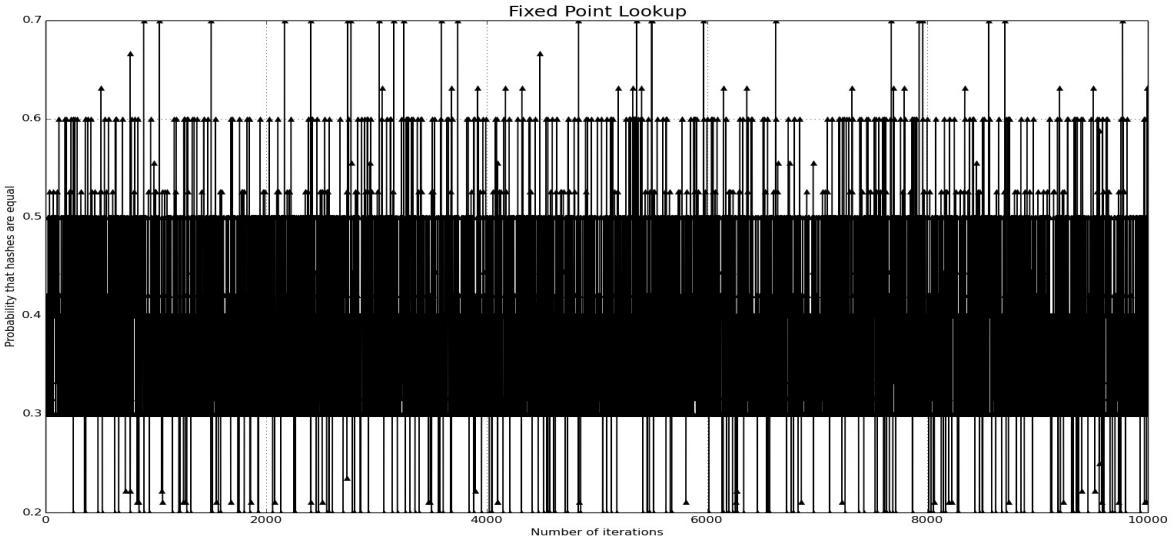
Results

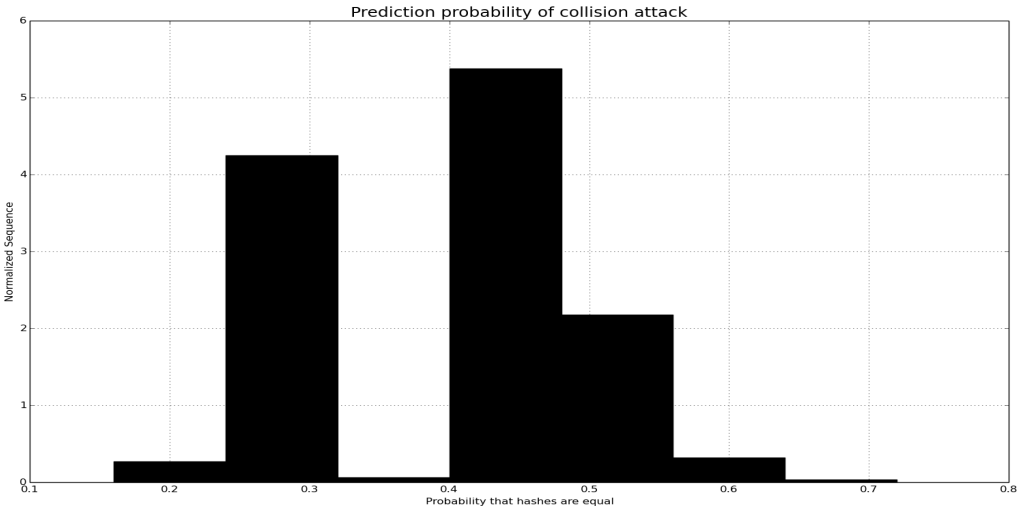
1.RIPEMD160



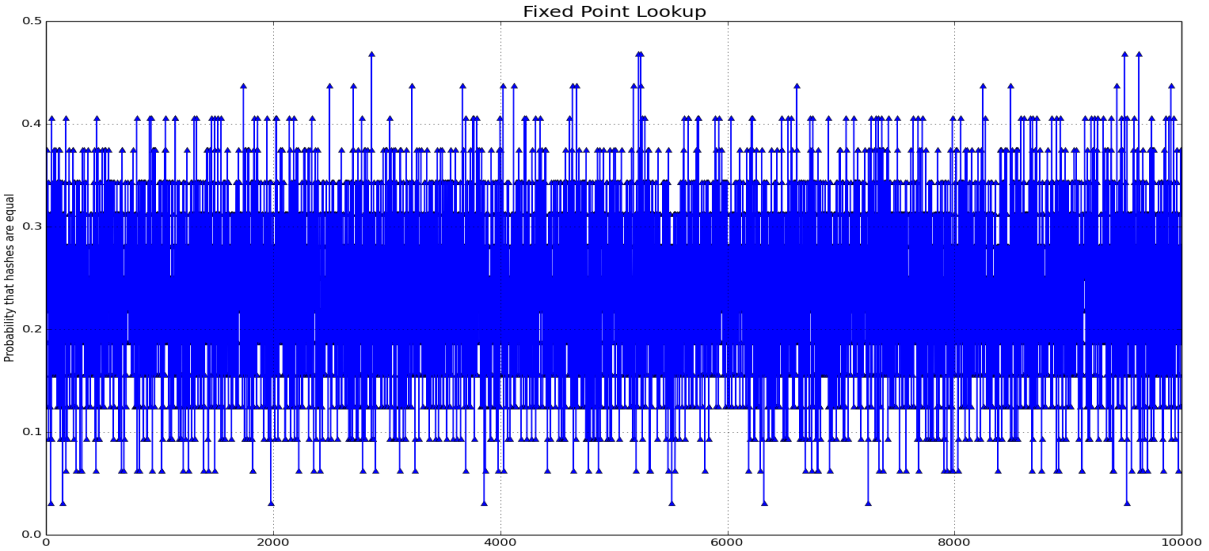


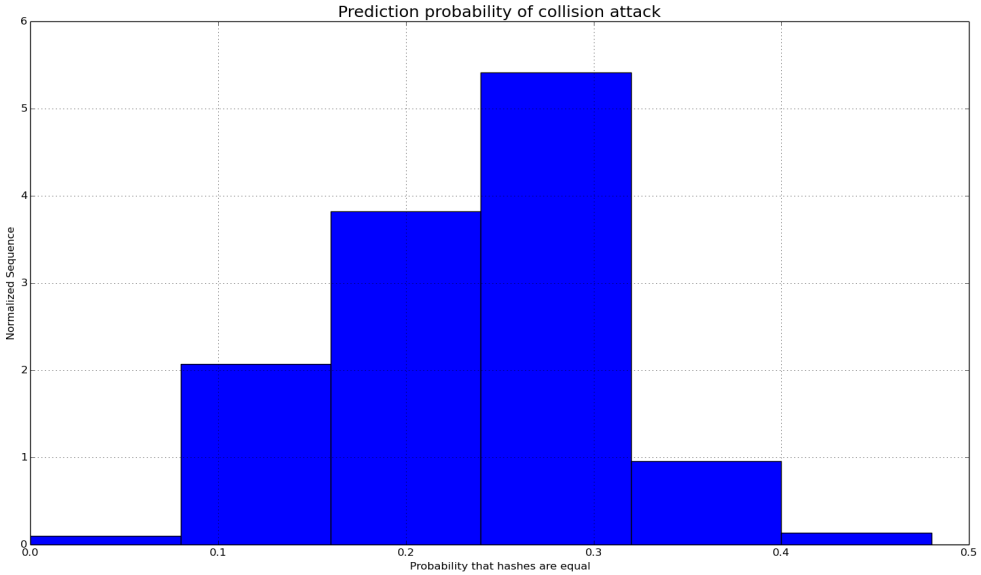
2.CRC32


Fixed Point Lookup
Probability that hashes are equal
Number of iterations
0.7
0.6
0.5
0.4
0.3
0.2
0
2000
4000
6000
8000
10000

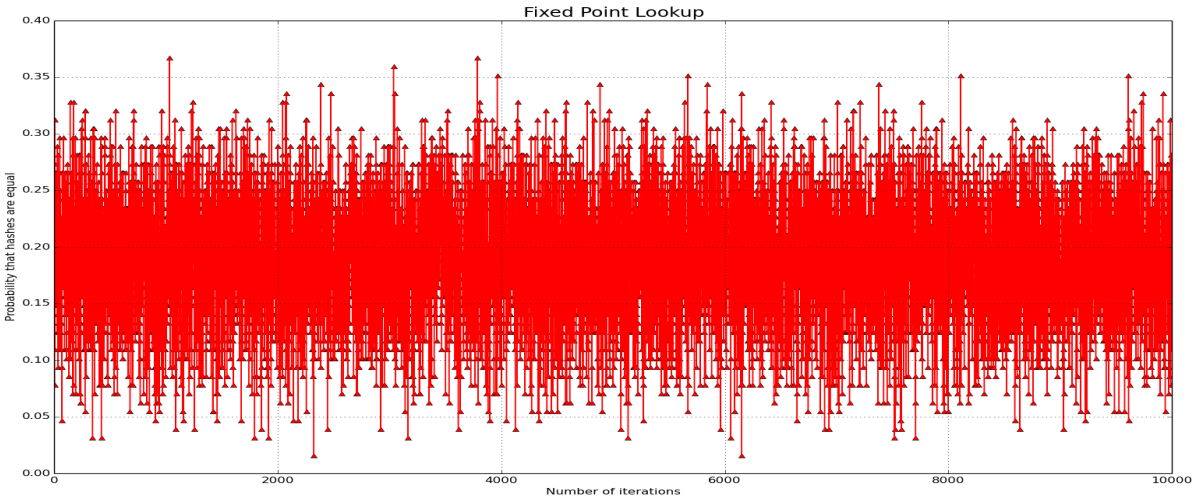

Prediction probability of collision attack
Normalized Sequence
Probability that hashes are equal
6
5
4
3
2
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

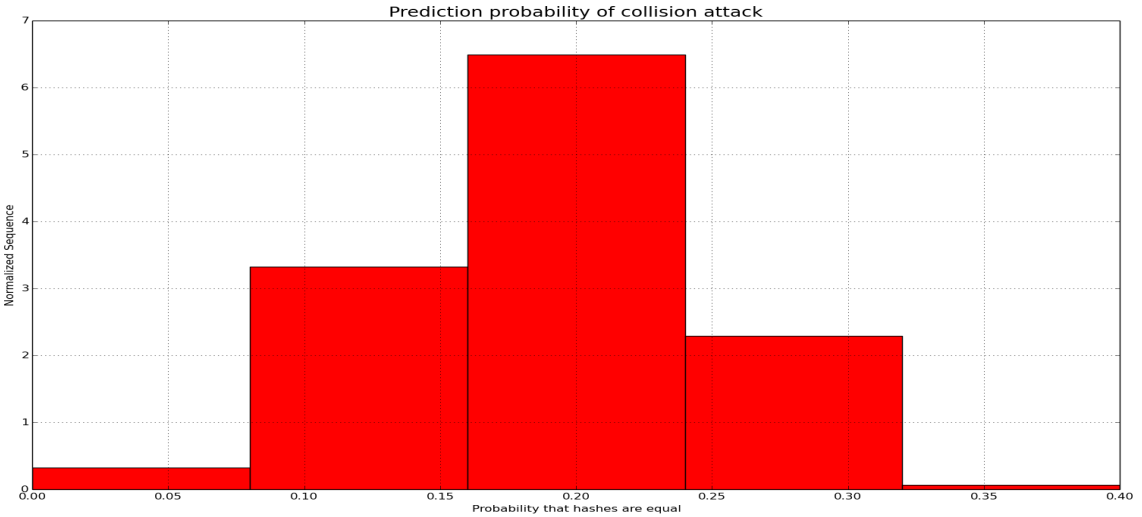
3. md5


Fixed Point Lookup
Probability that hashes are equal
0.0
0.1
0.2
0.3
0.4
0.5
0
2000
4000
6000
8000
10000

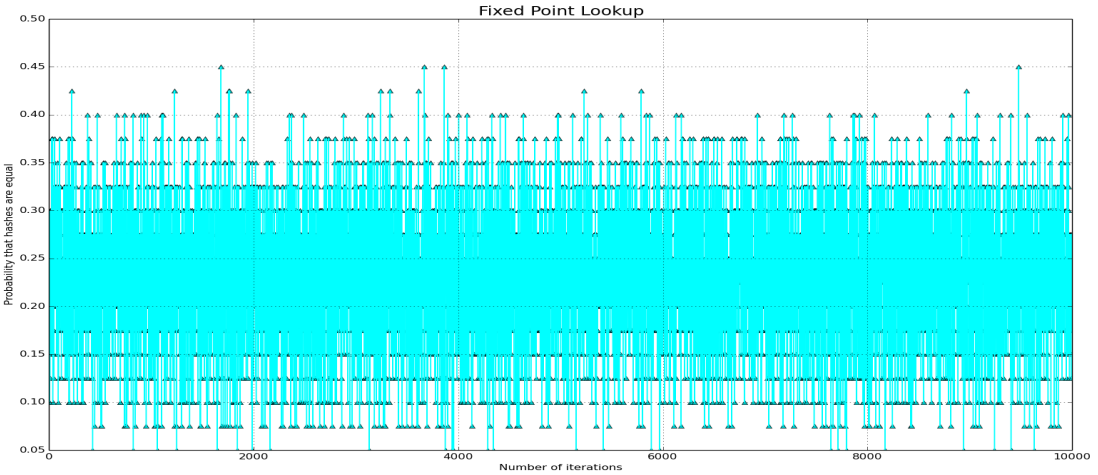

Prediction probability of collision attack
Normalized Sequence
0
1
2
3
4
5
6
0.0
0.1
0.2
0.3
0.4
0.5
Probability that hashes are equal

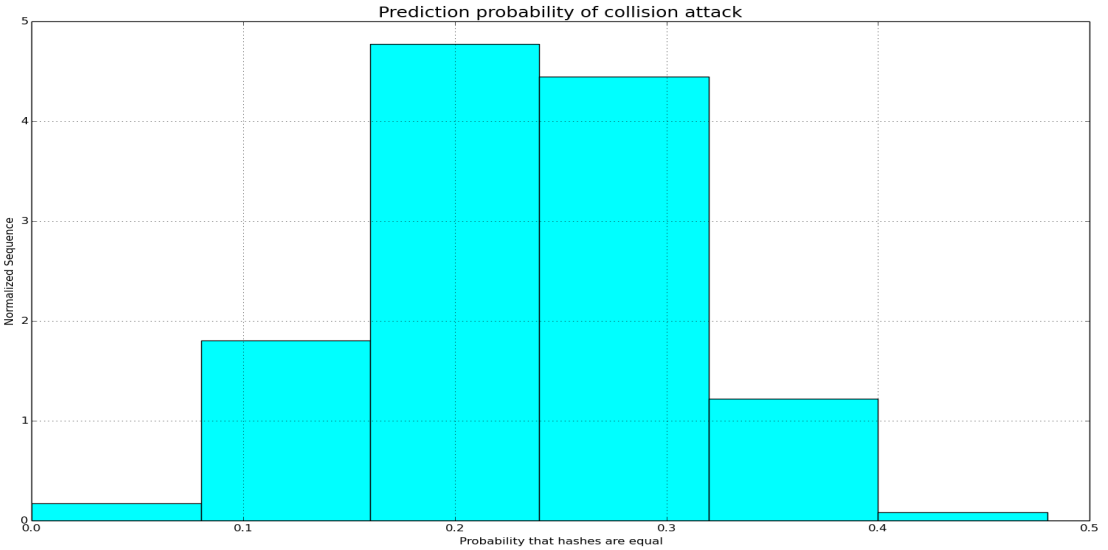
4.Whirlpool


Fixed Point Lookup
Probability that hashes are equal
Number of iterations
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0
2000
4000
6000
8000
10000


Prediction probability of collision attack
Normalized Sequence
Probability that hashes are equal
0
1
2
3
4
5
6
7
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40

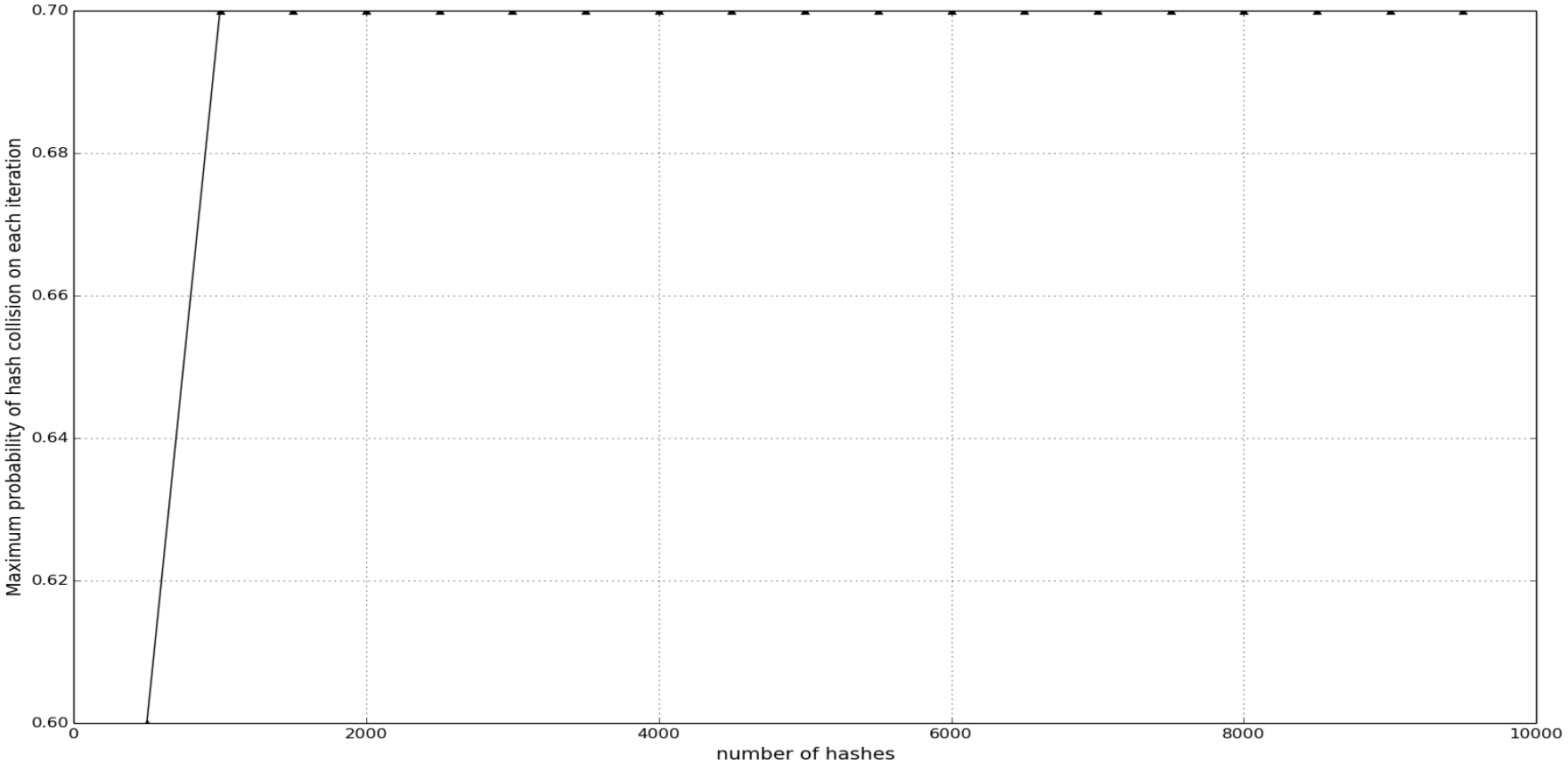
5.DSA


Fixed Point Lookup
Probability that hashes are equal
Number of iterations
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0
2000
4000
6000
8000
10000

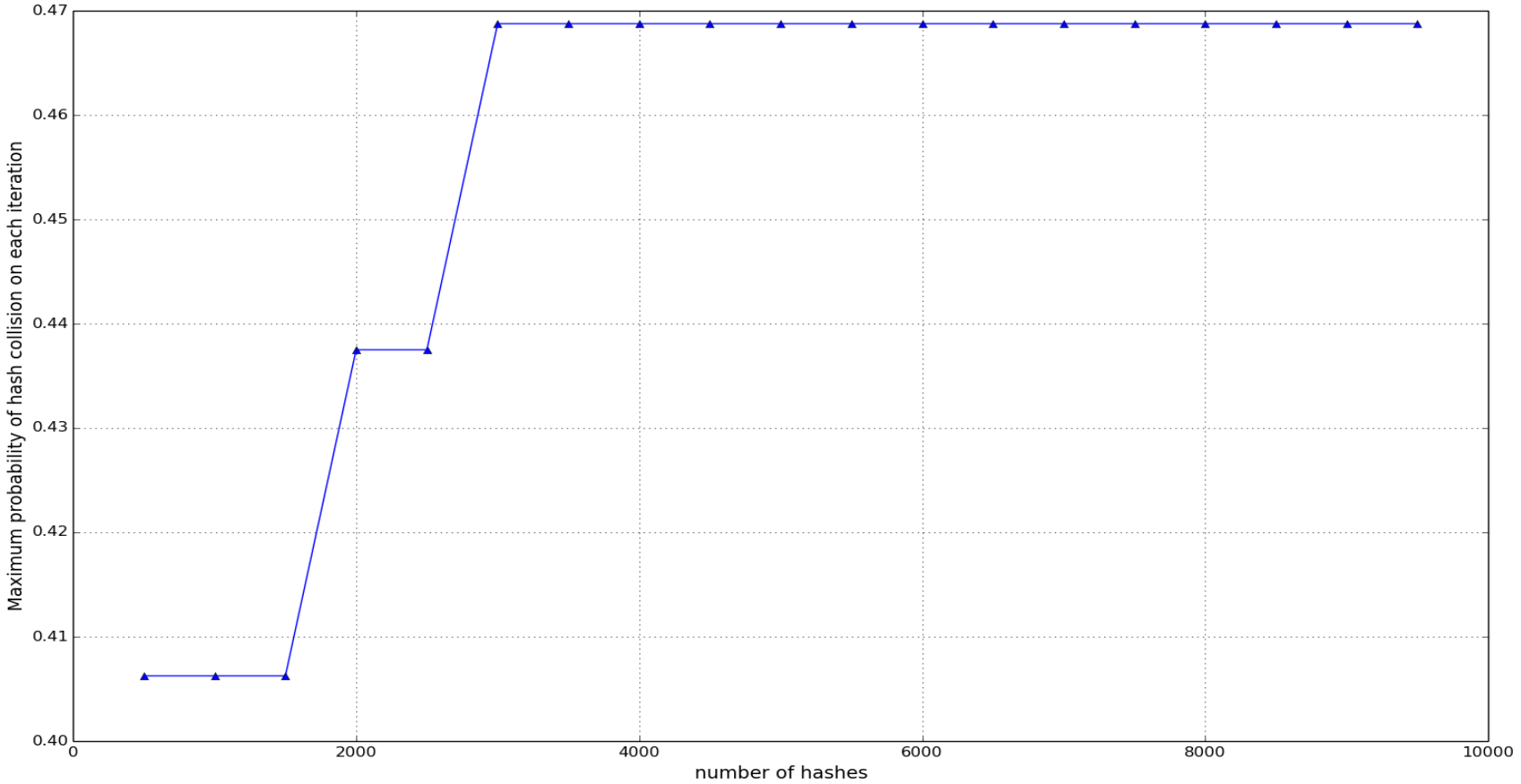

Prediction probability of collision attack
Normalized Sequence
Probability that hashes are equal
5
4
3
2
1
0
0.0
0.1
0.2
0.3
0.4
0.5

Fixed Point Analysis

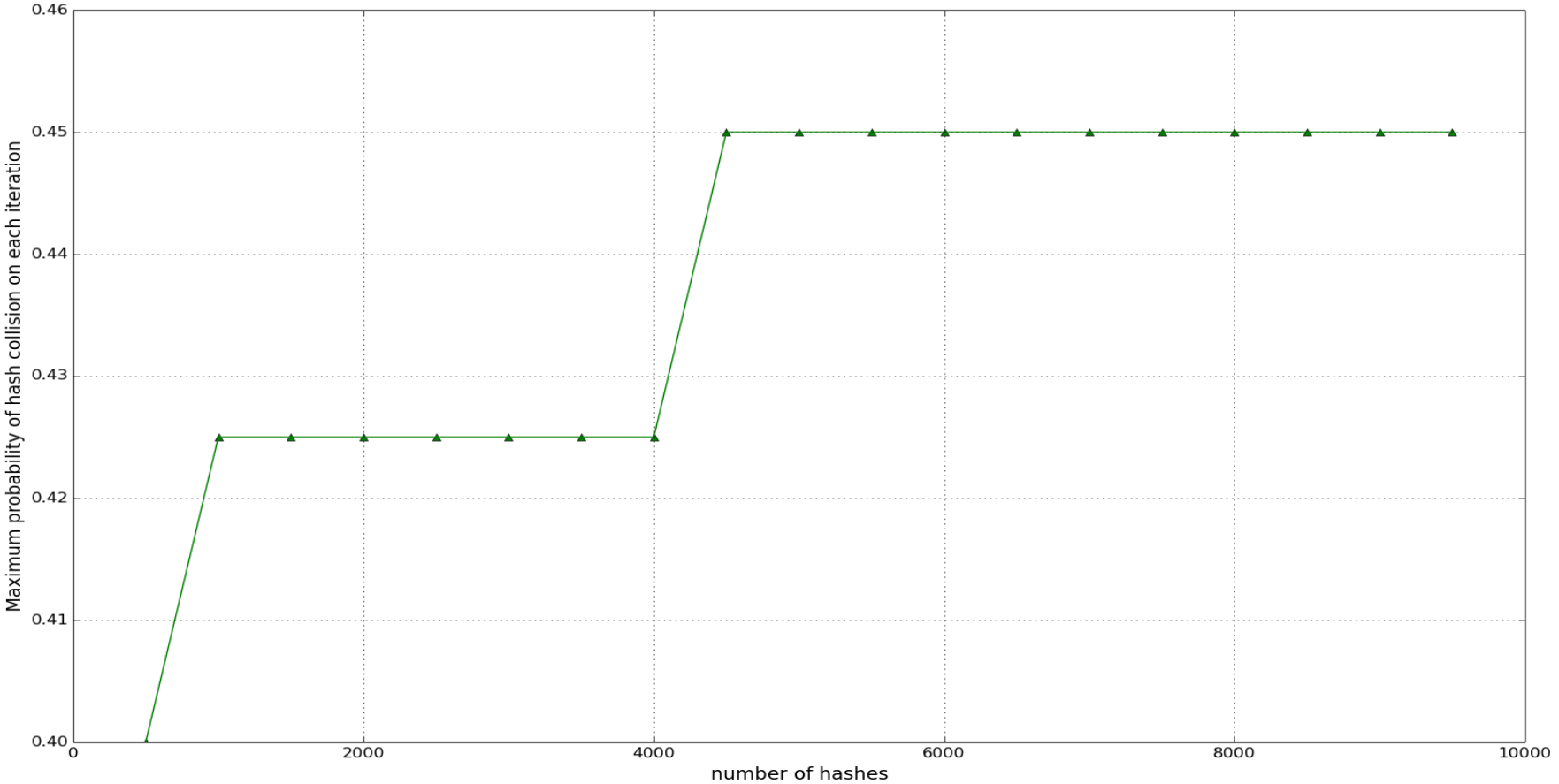
- CRC 32



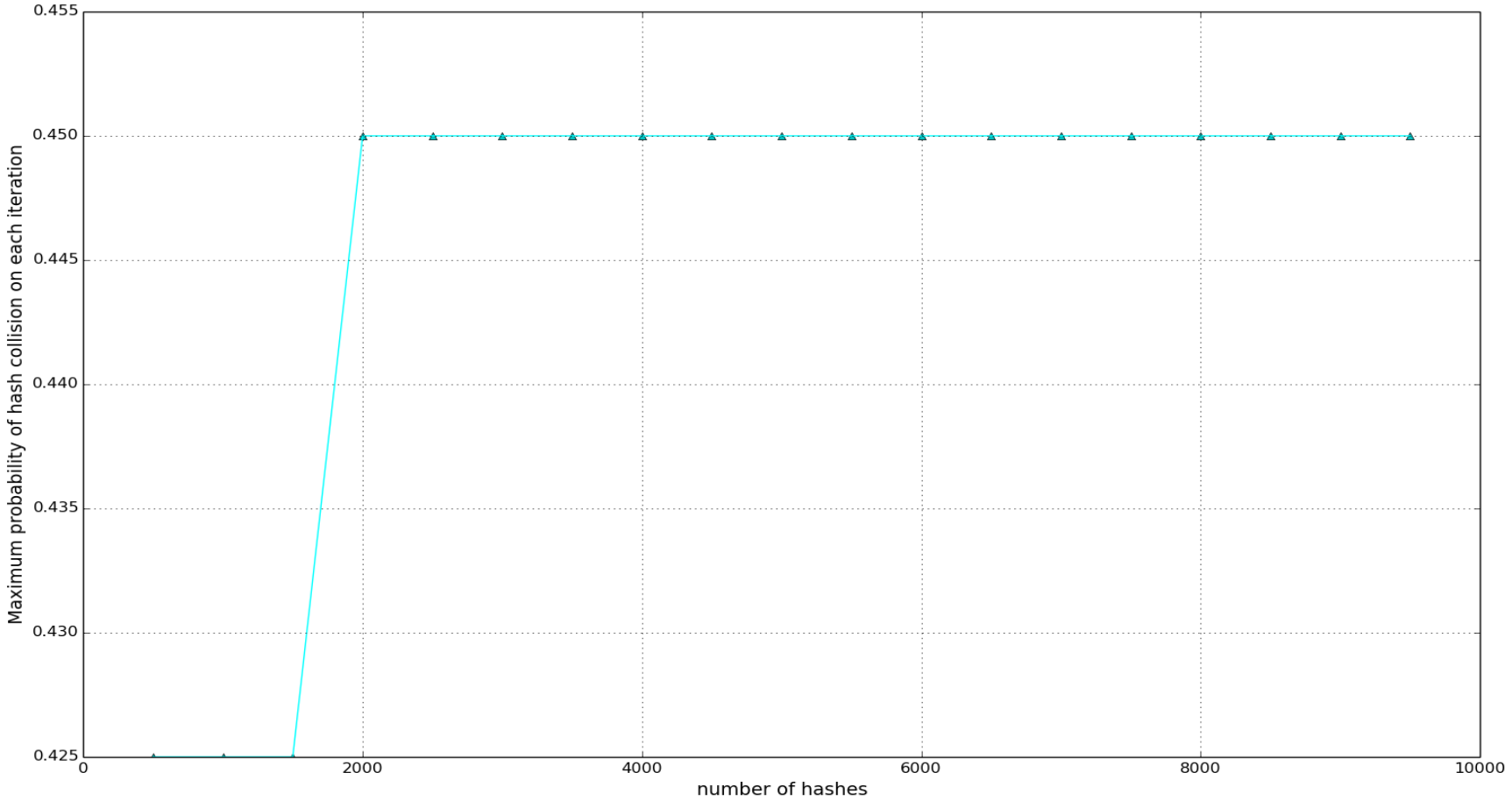
- md5


0.47
0.46
0.45
0.44
0.43
0.42
0.41
0.40
Maximum probability of hash collision on each iteration
0
2000
4000
6000
8000
10000
number of hashes

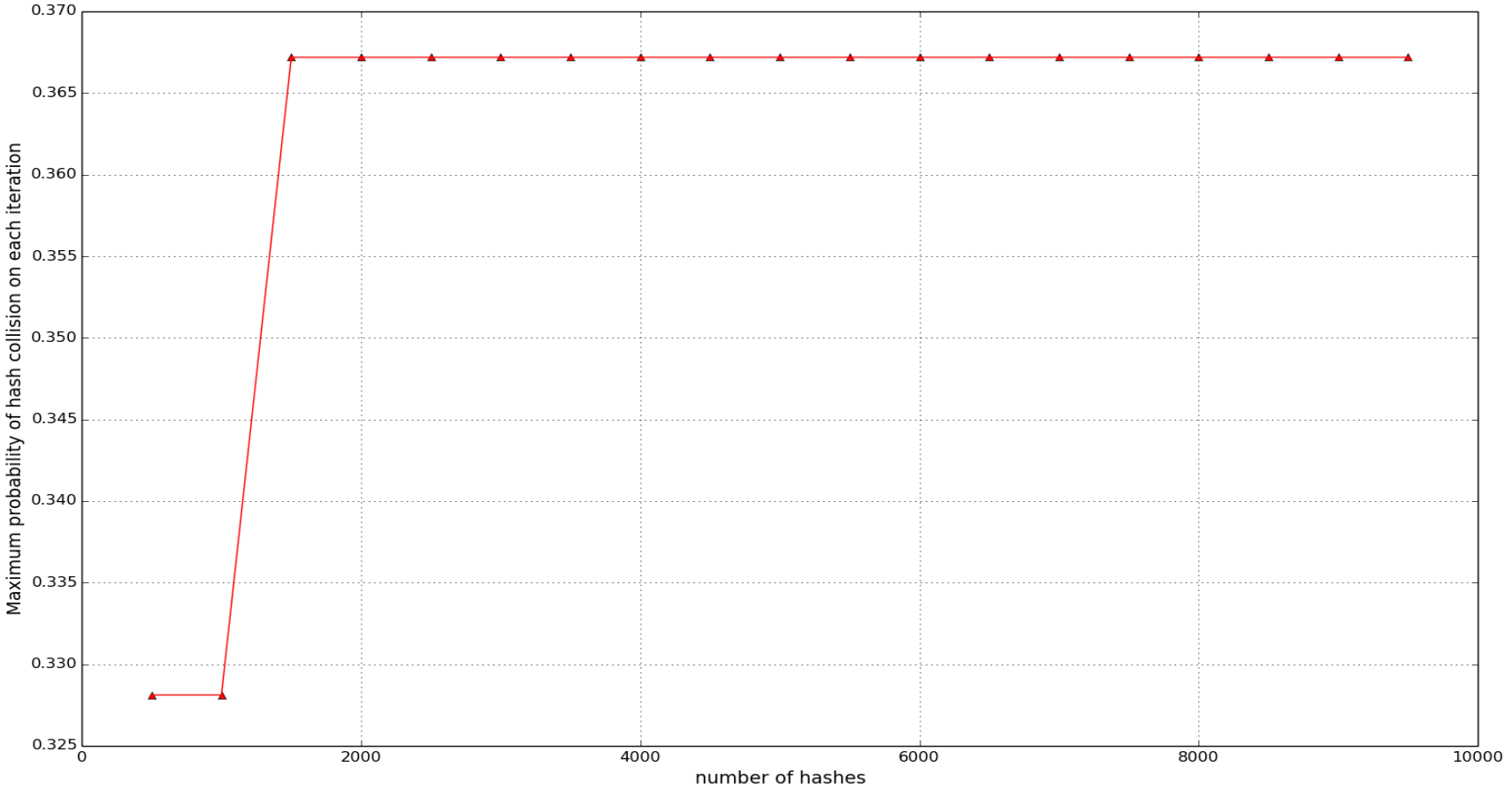
- RIPEMD160


0.46
0.45
0.44
0.43
0.42
0.41
0.40
Maximum probability of hash collision on each iteration
0
2000
4000
6000
8000
10000
number of hashes

- DSA


0.455
0.450
0.445
0.440
0.435
0.430
0.425
Maximum probability of hash collision on each iteration
0
2000
4000
6000
8000
10000
number of hashes

- Whirlpool


0.370
0.365
0.360
0.355
0.350
0.345
0.340
0.335
0.330
0.325
Maximum probability of hash collision on each iteration
0
2000
4000
6000
8000
10000
number of hashes

Recall.....

Its theoretically proven that the probability that a hash function of sufficient length has a fixed point is approximated by $1 - \frac{1}{e} \simeq 0.6321$

Now lets see the table that shows the probability of the five different hash function has a fixed point based on our experiments in the next slide.....

CRC32	MD5	RIPEMD160	DSA	WHIRLPOOL
0.70	0.468	0.45	0.45	0.368

- CRC32 has the highest probability, and that clearly explains why crc32 is vulnerable to hash collision attack ??

  http://www.cosc.canterbury.ac.nz/greg.ewing/essays/CRC-Reverse-Engineering.html

- Second comes, MD5 which also vulnerable to hash collision attack.

  http://www.mathstat.dal.ca/~selinger/md5collision/

Conclusions

- The above method not only helps to detect hash collision attack but its also helpful to find how similar two functions are.
- By observing the distribution of RIPEMD160 and DSA, it can be inferred that both follow the same distribution.
- This work also brings to light the best hash function to be used.

References

- https://davidlyness.com/post/fixed-points-of-hash-functions
- Pershing.com/20110504/hash-collision-probabilities
- Damgʃard, A Design Principle for Hash Functions,"advances in Cryptology{Crypto 89 Proceedings Springer-Verlag, 1989.
- Lucks, \Design Principles for Iterated Hash Functions," IACR preprintarchive, http://eprint.iacr.org/2004/253.pdf, 2004.
- Joux, \Multicollisions in Iterated Hash Functions. Applications to Cascaded Constructions,"Advances in Cryptology{Crypto 2004 Proceedings Springer-Verlag, 2004
- Bruce Schneier and John Kelsey - Second Preimages on n bit Hash Functions

Thank you