Scribe: Cryptography and Network Security

Ashish Kumar Singh

13-Nov-2020

1 Introduction

Second preimage resistance is the property of a hash function that it is computationally infeasible to find any second distinct input that has the same output as a given input.
in other words, second-preimage resistance is computationally infeasible to find any second input which has the same output as that of a specified input; i.e., given x, it is difficult to find a second preimage $x' \neq$ x such that h(x) = h(x')

2 Algorithm to Find Second Preimage

Algorithm

FIND-SECOND-PREIMAGE(h,x,Q)

- $y <-$ h(x)
- Choose $X_o \subseteq$X\{x}, $|X_o|$=Q $-$ 1
- for each $x_o \in X_o$, do
  - if h(x_o)=y, then return (x_o)
- return (failure)

The analysis of FIND-SECOND-PREIMAGE algorithm is similar to the FIND-PREIMAGE algorithm. The only difference is that we require an *extra* application of h to compute y=h(x) for the input value x.

Theorem : For any $X_o \subseteq$X\{x} with $|X_o|$=Q $-$ 1 , the success probability of FIND-SECOND-PREIMAGE algorithm is ϵ=$1-(1-\frac{1}{M})^{Q-1}$

3 Algorithm FindCollision

Algorithm

FIND-Collision(h,Q)

- Choose $X_o \subseteq X$, $|X_o| = Q$
- **for each** $x \in X_o$
  - **do** $y_x < -h(x)$
- **if** $y_x = y_{x'}$ for some $x' \neq x$
  - **then return** (x,x')
- **else return** (failure)

Theorem : For any $X_o \subseteq X$ with $|X_o| = Q$, the success probability of FIND-COLLISION algorithm is :

$$\epsilon = 1 - (\tfrac{M-1}{M})(\tfrac{M-2}{M})(\tfrac{M-3}{M}) \ldots\ldots\ldots\ldots (\tfrac{M-Q+1}{M})$$

Relating Q and ϵ

$$Q \approx \sqrt{2 * M * ln\tfrac{1}{1-\epsilon}}$$

If we take ϵ=0.5 then Q $\approx$ 1.17 $\sqrt{M}$

- So, if we hash around $\sqrt{M}$ values, we have a 50% chance of collision.
- Thus, algorithm is ($\frac{1}{2}$,O($\sqrt{M}$) algo.
- Collision solving is better than solving preimage or second preimage, as it is easier than this.
- $Collision_{Hardness} << Preimage_{Hardness}$
- Resistance against Collision => Preimage Resistance

4 First Reduction

Algorithm

COLLISION-TO-SECOND-PREIMAGE(h)

- **external** ORACLE-2ND-PREIMAGE
- choose x $\in X$ uniformly at random

- **if** ORACLE-2ND-PREIMAGE(h,x)=x'
  - **then return** (x,x')
- **else return**(failure)

ORACLE-2ND-PREIMAGE is an (ϵ,q) algorithm. If it gives answer then it will be correct as it is a Las-Vegas algorithm. So, x≠x' and h(x)=h(x').
Thus, the collision is also found.
COLLISION-TO-SECOND-PREIMAGE is also an (ϵ,q) Las-Vegas algo.

5 Second Reduction

Algorithm

COLLISION-TO-PREIMAGE(h)

- **external** ORACLE-PREIMAGE
- choose x $\in X$ uniformly at random
- $y <-$ h(x)
- **if** (ORACLE-PREIMAGE(h,x)=x') and (x'≠x)
  - **then return** (x,x')
- **else return**(failure)

ORACLE-PREIMAGE is a (1,Q) las Vegas Algo.

Theorem:Suppose h:$X->Y$ is a hash function where $|X|$ and $|Y|$ are finite and $|X| \geq 2|Y|$. Suppose ORACLE-PREIMAGE is (1,Q)-algo for Preimage, for the fixed hash function h. Then COLLISION-TO-PREIMAGE is a ($\frac{1}{2}$,Q+1)-algo for collision, for the fixed hash function.
Proof: The Probability of success of the algorithm COLLISION-TO-PREIMAGE is computed by averaging over all possible choice for x:

Pr[success]=Pr[x≠x']=$\frac{1}{|X|}\sum_{x\in X}\frac{|[x]|-1}{|[x]|}$

$=\frac{1}{|X|}\sum_{C\in Y}\sum_{x\in C}\frac{|C|-1}{|C|}$

$=\frac{1}{|X|}\sum_{C\in Y}(|C|-1)$

$=\frac{1}{|X|}(\sum_{C\in Y}|C|-\sum_{C\in Y}1)$

$=\frac{|X|-|Y|}{|X|}$

$\geq\frac{|X|-\frac{|X|}{2}}{|X|}$

$\geq\frac{1}{2}$

6 Conclusion

We have a COLLISION-TO-PREIMAGE algo of Las-Vegas type , which has a average case success probability of atleast $\frac{1}{2}$