Spring 2023 - ICSI 526
Homework 4

Written by Jacob Clouse

May 11, 2023

Contents

1 Question 1

1.1 Hash function A (n = p x q)

Properties:

- One Way Property: This does satisfy the One Way Property even if we know n. We because of the Modulus, we don't know how many times the original function has 'wrapped around' so to speak. We can't predict the output.
- Weak Collision Resistance: This is Weak Collision Resistant. This is because it is computationally difficult to find two distinct inputs x and y such that h1(x) = h1(y). Suppose an attacker has a message x and wants to find another message y such that h1(y) = h1(x). The attacker can try to compute y by solving the equation $y^2 \equiv x^2$ (mod n). Since n = p * q is the product of two large primes, it is computationally infeasible to directly compute the square roots of x^2 (mod n) without knowing the factorization of n.
- Strong Collision Resistance: This is NOT Strong Collision Resistant. The reason is that for any two inputs x1 and x2, the hash function h1(x) produces the same output if x1 = -x2 (mod n). That is, h1(x1) = h1(x2) if x1 ≡ -x2 (mod n). An attacker who knows the primes p and q can easily find two inputs x1 and x2 such that x1 ≡ -x2 (mod n) and then generate a collision by computing h1(x1) and h1(x2).

1.2 Hash function B (h2(m) = M1 $\oplus$ M2 ... $\oplus$ Mn.)

Properties:

- One Way Property: This does satisfy the One Way Property. The hash function h2(x) is one-way because given an input message m, it is computationally infeasible to determine the message m from the hash value h2(m). This is because the hash function uses the bitwise XOR operation to combine the blocks of the message, which is a one-way function.
- Weak Collision Resistance: This is Weak Collision Resistant.The hash function h2(x) is weak-collision resistant because it is computationally infeasible to find two messages m1 and m2 such that h2(m1) = h2(m2). This is because any change to the input message, no matter how small, will result in a completely different hash value due to the bitwise XOR operation.
- Strong Collision Resistance: This is NOT Strong Collision Resistant. This is because it is possible to find two messages m1 and m2 that have the same hash value h2(m1) = h2(m2) with a brute-force attack. For example, if we take the message M1 = M2 = ... = Mn = 0, then h2(m) will always be 0. Which means it is not secure for applications where strong collision resistance is required, such as in digital signatures.

1.3 How to run each Hash Algorithm

Running both hashing programs is very similar, so here is a step by step guide:

- Make sure you have installed python 3 and pip installed the imports listed at the top of each function.
- For Hash A, you will just need to type **python A_strong_col.py** in the terminal and hit run. It may take a while to find any collisions, but this should find them.
- For Hash B, you will just need to type **python B_strong_col.py** in the terminal and hit run. Again, it may take a while to find any collisions, but this should find them.

2 Question 2

2.1 Design and implement a PRNG using AES (OFB mode)

To run the algorithm, follow these steps:

- Make sure you have python 3 installed and pip install the imports designated inside your virtual environment.
- Run the program by using: **python prng.py**
- It should output the random numbers, Fraction of One Bits, and the Fraction of Bits that Match. You are all set!

3 Youtube Link

Here is the Youtube link to my video presentation: **https://youtu.be/QxpQORY8pbg**