CALIFORNIA STATE SCIENCE FAIR
2016 PROJECT SUMMARY

Name(s)

Kevin K. Pho

Project Number

S1418

Project Title

Digital Fingerprints: Constructing One-Way Hash Functions

Abstract

Objectives/Goals

The objective of this experiment is to test numerous designs (based on mathematical problems and basic operations) to aid in the construction of faster and more secure hash algorithms. The designs were evaluated by speed and the lack of collisions, which occur when two inputs share the same output. The construction of more efficient hash algorithms provide security of passwords, integrity of files, et cetera by preventing inversion of the function and the discovery of a collision.

Methods/Materials

A laptop computer with the Python interpreter was used to design and test the algorithms. The computer and interpreter provided the timer for testing speed. The algorithms were tested on a sample word list of common passwords (to demonstrate applicability). Some of the algorithms were derived from source code or programming libraries, such as hashlib (Python).

Results

The data indicates that primitive operations performed quickly, clocking in at under a second. They provided many collisions (most yielded more than 500). The fastest operations included addition, XOR, and bit-shifting while the most secure included bit-shifting, addition, modulus, and subtracting. On the other hand, the investigated mathematical constructs that were the fastest included SHA-3, MD5, the position-based design, the addition construct, and the linear congruential generator while those that provided the most resistance to collisions included SHA-3, MD5, the sum of three cubes design, the linear shift feedback register, and the discrete logarithm problem design.

Conclusions/Discussion

The speed of primitive operations allows hash algorithms to quickly perform operations on the bits in order to "mix" them. Although the quantity of these operations affects the speed of the algorithm, the security of these do not create most of the security of the algorithm; the mathematical constructs do. The constructs are the steps that properly transform/diffuse the bits, so that the output provides a seemingly random output. Furthermore, if the constructs are based on computational hardness assumptions, they can be further proven to be secure as an input must be impossible to retrieve from its output. This information encourages the construction of hash algorithms with NP problems (problems known to be hard to solve in polynomial time), and it can aid in the construction of future hash algorithms to further enhance cryptographic security.

Summary Statement

A hash function is most effective (measured by speed and lack of collisions) when simple mathematical problems, such as the discrete logaritihm problem, are used in junction with primitive operations, such as addition and XOR.

Help Received

I designed and programmed the hash algorithm and evaluating tests and analyzed the data myself. Some of the algorithm designs were based on pre-existing ones, such as MD5, SHA-3, and Adi Shamir's Discrete Logarithm Hash Function.