

MASTER'S PROJECT PROPOSAL

Soham Sadhu
Department of Computer Science
Rochester Institute of Technology
Rochester, NY 14623
sxs9174@rit.edu

July 14, 2013

Chair: Prof. Stanisław Radziszowski spr@cs.rit.edu

signature

date

Reader: Prof. Alan Kaminsky ark@cs.rit.edu

signature

date

Observer: Prof. Edith Hemaspaandra eh@cs.rit.edu

signature

date

ABSTRACT

Hash functions, have applications in computer security, in fields of authentication and integrity. Due to importance of hash function usage in everyday computing, standards for using hashing algorithm and their bit size have been released by (NIST) which are denoted by nomenclature Standard Hashing Algorithm (SHA).

Due to advances in cryptanalysis of SHA-2, NIST announced a competition in November, 2007 to choose SHA-3. In October, 2012 the winner was selected to be Keccak amongst 64 submissions. All the submissions were open to public scrutiny, and underwent intensive third party cryptanalysis, before the winner was selected. Keccak was chosen for its flexibility, efficient and elegant implementation, and large security margin.

All algorithms submitted to competition have undergone public scrutiny. And other four finalist in the competition were almost equivalent to Keccak, in attributes of security margin and implementation. In this project, I will be comparing Keccak with two other SHA-3 finalists, BLAKE, and Grøstl with respect to their resistance to simulated annealing and tabu search.

Application of tabu search and simulated annealing to hash algorithms will be akin to generic attacks. That is these methods of breaking hash functions are design agnostic or do not depend on the workings of the hash function. Thus ensuring no bias in the experiment. At present, it is computationally infeasible to break the above mentioned hash functions. But the reduced versions of these can be subjected to attacks for near collisions. Thus I will be able to examine and conclude, if reduced instance Keccak has better resistance to generic attacks than reduced instance of BLAKE and Grøstl.

1 Problem Statement

1.1 Hash Functions

A cryptographic hash function, is an algorithm capable of intaking arbitrarily long input string, and output a fixed size string, often as called message digest. The message digest for two strings even differing by a single bit should ideally be completely different, and no two input message should have the same hash value. This

property enables us to finger print a message. Following are the properties of and ideal hash function. [1]

1. Preimage resistance

PREIMAGE

Given: A hash function $h : \mathcal{X} \rightarrow \mathcal{Y}$ and an element $y \in \mathcal{Y}$.

Find: $x \in \mathcal{X}$ such that $h(x) = y$.

If the preimage problem for a hash function cannot be efficiently solved, then it is preimage resistant. That is the hash function is one way, or rather it is difficult to find the input, given the output alone.

2. Second preimage resistance

SECOND PREIMAGE

Given: A hash function $h : \mathcal{X} \rightarrow \mathcal{Y}$ and an element $x \in \mathcal{X}$.

Find: $x' \in \mathcal{X}$ such that $x' \neq x$ and $h(x) = h(x')$.

A hash function for which a different input given another input, that compute to same hash cannot be found easily, is called as having second preimage resistance.

3. Collision resistance

COLLISION

Given: A hash function $h : \mathcal{X} \rightarrow \mathcal{Y}$ **Find:** $x, x' \in \mathcal{X}$ such that $x' \neq x$ and $h(x') = h(x)$.

Collision problem states that, can two different input strings be found, such that they hash to the same value given the same hash function. A hash function is collision resistant, if it is computationally infeasible to find two different values hashing to same value.

2 Background

3 Related Work

4 Methodology

5 Evaluation and expected outcomes

References

- [1] Douglas R. Stinson. *Cryptography Theory and Practice*, chapter 4. Cryptographic Hash Functions. Chapman & Hall/CRC, Boca Raton, FL 33487-2742, USA, third edition, 2006.