Evaluation of hill climbing, simulated annealing, tabu search and random selection: search algorithms on cryptographic hash functions BLAKE, Grøstl and Keccak

Soham Sadhu

May 23, 2014

Abstract

- In October 2012, Keccak was chosen as the winner of SHA-3 competition amongst 64 candidates, including the finalists BLAKE and Grøstl.
- I have attempted to find near collisions in reduced versions of BLAKE, Grøstl and Keccak; using hill climbing, random selection, simulated annealing and tabu search.

Table of contents

1. Introduction
   - Hash function
   - Property of hash function
   - Security model
   - Application
   - Standards and SHA-3 competition
2. SHA-3 finalists
   - BLAKE

Hash function

A *hash family* is a four-tuple $(\mathcal{X}, \mathcal{Y}, \mathcal{K}, \mathcal{H})$, satisfying the following conditions.[1]

- $\mathcal{X}$ is a set of possible messages
- $\mathcal{Y}$ is a finite set of hash function output
- $\mathcal{K}$, the *keyspace*, is a finite set of possible keys
- For each $K \in \mathcal{K}$, there is a hash function $h_k \in \mathcal{H}$. Each $h_k : \mathcal{X} \to \mathcal{Y}$

[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.

Property of Hash function[2]

1. **Preimage resistance**
   Given: A hash function $h : \mathcal{X} \to \mathcal{Y}$ and an element $y \in \mathcal{Y}$.
   Find: $x \in \mathcal{X}$ such that $h(x) = y$.
2. **Second preimage**
   Given: A hash function $h : \mathcal{X} \to \mathcal{Y}$ and an element $x \in \mathcal{X}$.
   Find: $x' \in \mathcal{X}$ such that $x' \neq x$ and $h(x) = h(x')$.
3. **Collision resistance**
   Given: A hash function $h : \mathcal{X} \to \mathcal{Y}$
   Find: $x, x' \in \mathcal{X}$ such that $x' \neq x$ and $h(x') = h(x)$.

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

Security model

- **Random Oracle** model, proposed by Bellare and Rogaway. Algorithm is secure, modulo the way it creates the random outputs.[3]
- **Birthday paradox:** In a sample size of M, minimum N number of attempts to find, two elements with same value is given by equation $N \approx 1.17\sqrt{M}$.

[3] Gerrit Bleumer. Random oracle model. In HenkC.A. van Tilborg and Sushil Jajodia, editors, Encyclopedia of Cryptography and Security, pages 10271028. Springer US, 2011.

Application of hash functions

1. **Digital forensics:** take a hash value of evidence, to later prove that it has not been tampered. [4]
2. **Password stored:** is salted and hased, before inserting to database.
3. **File integrity:** take hash value of files between time intervals, to make sure; they have not been tampered.
4. **Pseudo random:** generator, based on a seed value.

[4] Richard P. Salgado. Fourth Amendment Search And The Power Of The Hash, volume 119 of 6, pages 38 46. Harvard Law Review Forum, 2006.

Standards and SHA-3 competition

1. SHA-0 proposed by NSA in 1993, standardised by NIST. In 1995, SHA-0 replaced by SHA-1, designed by NSA. [5]
2. SHA-1 had block size of 512 bits, size of 160 bits; and additional circular shift operation, to rectify weakness from SHA-0.
3. SHA-2 designed by NSA, released by NIST in 2001. Family of functions of SHA-224, SHA-256, SHA-384, SHA-512.
4. SHA-3 competition announced on November, 2007. Submissions accepted till October, 2008. From 64 submissions, that included 5 finalist, Keccak announced as winner on October 2, 2012 by NIST.

[5] James Joshi. Network Security: Know It All: Know It All. Newnes Know It All. Elsevier Science, 2008.

BLAKE construction

BLAKE is built on HAIFA (HAsh Iterative FrAmework) structure [6]which is an improved version of Merkle-Damgård function.

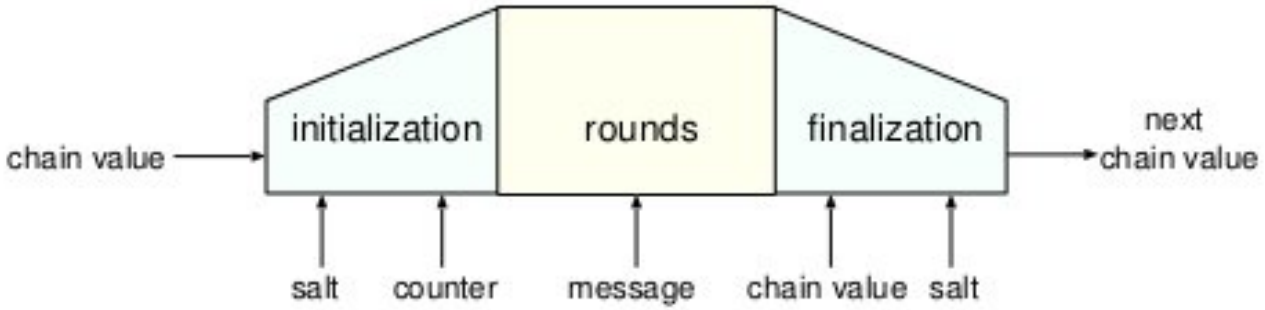

Figure: Local wide construction of BLAKE's compression function[7]

[6]Eli Biham and Orr Dunkelman. A framework for iterative hash functions - haifa. Cryptology ePrint Archive, Report 2007/278, 2007.

[7]Jean-Philippe Aumasson, Luca Henzen, Willi Meier, and Raphael C.-W. Phan. Blake. http://www.131002.net/blake/blake.pdf, April 2012.

Compression algorithm

Algorithm 1 BLAKE Compression procedure[8]

1: $h^0 \leftarrow IV$
2: **for** $i = 0, \ldots, N-1$ **do**
3: $\quad h^{i+1} \leftarrow compress(h^i, m^i, s, l^i)$
4: **end for**
5: **return** h^N

[8] Jean-Philippe Aumasson, Luca Henzen, Willi Meier, and Raphael C.-W. Phan. Blake. http://www.131002.net/blake/blake.pdf, April 2012.