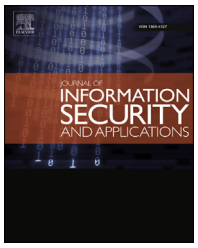
Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/jisa



ASH-160: A novel algorithm for secure hashing using geometric concepts☆

Venkateswara Rao Pallipamu [a,*], K. Thammi Reddy [b], P. Suresh Varma [a]

[a] Department of Computer Science, Adikavi Nannaya University, Rajahmundry 533105, Andhra Pradesh, India
[b] Department of Computer Science and Engineering, GITAM University, Visakhapatnam 530045, Andhra Pradesh, India

ARTICLE INFO

Article history:
Available online 17 June 2014

Keywords:
Cryptography
Hash function
Message digest
Authentication
Geometric concepts

ABSTRACT

Cryptographic hash function plays a pivotal role in many parts of cryptographic algorithms and protocols, especially in authentication, non-repudiation and data integrity services. A cryptographic hash function takes an input of arbitrary large size message and produces a fixed small size hash code as an output. In the proposed hash algorithm ASH-160 (Algorithm for Secure Hashing-160), each 512-bit block of a message is first reduced to 480-bit block and then divided into ten equal blocks and each one is further divided into three sub-blocks of 16-bits each. These three sub-blocks act as three points of a triangle, which are used in Area calculation. The calculated area values are in turn processed to generate message digest. ASH-160 is simple to construct, easy to implement and exhibits strong avalanche effect, when compared to SHA1 and RIPEMD160.

1. Introduction

Authentication is an important concept in network security, which can be achieved by using hash function. A hash function is a public function that maps a message of any length into a fixed-length value called a hash code or message digest, which serves as an authenticator. Message digest is also referred to as a summary of the message or finger print of the message. One way of hash function is a variant of the message authentication code. Hash code is a function of all the bits of the message which provides an error detection capability. Any change made in the message results a major change in the hash code. Message authentication is said to protect the integrity of a message, ensuring that each message that it is received and deemed to be acceptable is arriving in the same condition that it was sent out-with no bits inserted, missing or modified. Digital signatures can be used to authenticate the source of messages. When ownership of a digital signature secret key is bound to a specific user, a valid signature shows that the message was sent by that user. Importance of high confidence in sender authenticity is especially obvious in a financial context. This paper is organized as follows. Section 2 gives an overview of cryptographic hash functions. In Section 3, we propose a new secure hash function ASH-160, designed based on geometric concepts. Section 4 presents results and discussions. The analysis of ASH-160 is discussed in Section 5. Section 6 concludes with a note on future enhancements.

2. Overview of cryptographic hash functions

Hash functions are currently fascinating topic of research. Particularly the area of information security welcomes new

☆ This article belongs to the special issue Security, Privacy and Trust in Future Networks and Mobile Computing.

* *Corresponding author*. Tel.: +91 9441447037.
E-mail address: venkat.aknu@gmail.com (V.R. Pallipamu).

http://dx.doi.org/10.1016/j.jisa.2014.05.001
2214-2126/

approaches to design the secure hash functions. There are innumerable hash functions that have been developed (Tiwari and asawa, 2010; Mathew and Jacob, 2010; Status Report on the First Round of the SHA-3 Cryptographic Hash Algorithm Cometition, September 2009; Federal Information Processing Standards Publication and 8900 October, 2008; Knudsenl et al., 2007; Mironov, 2005). Some of the famous hash functions (Stallings, 2003; NIST, 2002; Gauravaram et al., 2004; Preenel, 1994) are discussed below:

a. *The MDx Family*

R. Rivest of RSA DataSecurity Inc., has designed a series of hash functions, named as MD (Message Digest) followed by a number. The MD1 is a proprietary algorithm, MD2 (Rivest, 1992c) was suggested in 1990 and was recommended to replace BMAC. MD3 was never published, and it seems to have been abandoned by its designer. MD4 (Rivest, 1992a) and MD5 (Kahate, 2006; Rivest, 1992b) algorithms were both designed by Rivest and MD4 was a novel design, first proposed in 1990. It has shown very good performance in software implementations on 32-bit architecture, which was the most attractive feature. However, it was soon discovered that its security level was much lower than expected, as demonstrated by some attacks on reduced versions of the algorithm. This prompted the design of MD5 (1991), intended as an improved variant of MD4. The questions regarding the security of MD4 (Wang et al., 2004) were confirmed when Dobbertin, in 1996, demonstrated a very practical collision-finding attack on the full MD4 algorithm (Gauravaram et al., 2006). Furthermore, some serious weaknesses have been shown in MD5 (Wang and Yu, 2005; Wang et al., 2004; Wang1 et al., 2004) as well.

b. *The HAVAL Algorithm*

In 1992, Y. Zheng et al. proposed the HAVAL algorithm (Mathew and Jacob, 2010; Status Report on the First Round of the SHA-3 Cryptographic Hash Algorithm Cometition, September 2009). It has a structure that is quite similar to the MD4 and MD5 algorithms. In contrast to MD4 and MD5, HAVAL allows the computation of hashes of variable length. It allows a tradeoff between efficiency and security by means of a parameter, the number of rounds, which can be chosen equal to 3, 4 or 5.

c. *The RIPEMD Family*

In 1992, the RIPEMD hash function was designed in the framework of the European RIPE project (Mathew and Jacob, 2010; Kahate, 2006; Wang1 et al., 2004). The design of RIPEMD is based on MD4; its compression function consists essentially of two parallel versions of the MD4 compression function. In 1996, a collision attack was found by Dobbertin on versions of RIPEMD and it was reduced to two rounds out of three. This provoked redesign in Hash functions RIPEMD-128 and RIPEMD-160 (Dobbertin et al., 1996) (proposed by Dobbertin et al.). The RIPEMD-160 algorithm has the advantage of longer hash values (160 bits instead of 128).

d. *The SHA Family*

In 1993, the SHA algorithm (Mathew and Jacob, 2010; Kahate, 2006; Stallings, 2003) was designed by NSA and published by NIST as federal standard FIPS 180. SHA is another Hash function inspired by MD4. The design principle of SHA did not disclose, however in 1994 NIST announced that a technical flaw had been found in SHA which made the algorithm less secure than originally thought. But a small modification was made to the algorithm resulting in the hash function SHA-1, and the corresponding standard FIPS 180-1. In 1998 F. Chabaud and A. Joux found a theoretical collision attack for the original version of SHA (this algorithm is often called SHA-0 now). Their analysis support the change that was made for the SHA-1 (Wang et al., 2005) hash function. In 2002, NIST has published(FIPS 180-2) (Federal Information Processing Standards Publication, March 2012; Third-Round Report of the SHA-3 Cryptographic Hash Algorithm Competitio, November 2012; Status Report on the Second Round of the SHA-3 Cryptographic HasAlgorithm Competition, February 2011; NIST, 2002) new hash functions, SHA-256, SHA-384 and SHA-512. These new hash functions are named as SHA-2. In 2004, another two new hash functions are added to the SHA-2 family. Sofar there were no known successful attacks on the newer versions of SHA.

3. Algorithm for secure hashing (ash-160)

The proposed algorithm is named as ASH-160 'ASH' means the powdery residue of matter that remains after burning which is irreversible; Similarly this algorithm ASH-160 (Algorithm for Secure Hashing) also has the same property. It takes a message as input with a maximum length of less than 2^{64} bits and produces a 160-bit message digest as output. The input is reduced from 512-bit blocks to 480-bit blocks. The compression function accepts two parameters which are 512 bit-block of the message and the chaining variable (160-bits). The process consists of the following steps:

a. *Append padding bits*

The message is padded so that its length is congruent to 448 modulo 512 (length = 448 mod 512). Padding is always done, even if the message is of desired length. Thus, the number of padding bits is in the range of 1–448. The padding consists of a single 1 followed by the necessary number of 0's.

b. *Append length*

A block of 64 bits which contains the length of the message (before padding) is appended to the message. This block is treated as an unsigned 64-bit integer (most significant byte first).

c. *Initialize MD buffer*

A 160-bit buffer is used to hold intermediate and final results of the hash function. The buffer can be represented as