iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 2 **Issue:** IX **Month of publication:** September 2014

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

An Efficient Approach for Secure Hash Algorithm

K. Ramya [1] K. Suganya [2]

[1]Assistant Professor & Head , Department of Science & Humanities

Kingston Engineering College, Katpadi, Vellore

[2]Associate Professor,Department of Software Engineering & IT[PG]

A.V.C College of Engineering, Mayiladuthurai

Abstract*— Hash functions are the most widespread among all cryptographic primitives, and are currently used in multiple cryptographic schemes and in security protocols. A famous secure hash algorithm given by the National Institute of Standard and Technology (NIST). SHA stands for "secure hash algorithm". The four SHA algorithms are structured differently and are named SHA-0, SHA-1, SHA-2, and SHA-3. SHA-1 is very similar to SHA-0, but corrects an error in the original SHA hash specification that led to significant weaknesses. The SHA-0 algorithm was not adopted by many applications. SHA-2 on the other hand significantly differs from the SHA-1 hash function.SHA-1 is the most widely used of the existing SHA hash functions, and is employed in several widely used applications and protocols.

***Keywords*— NIST, SHA, FIPS, DSS, NSA**

I. INTRODUCTION

The Secure Hash Algorithm is a family of cryptographic hash functions published by the National Institute of Standards and Technology (NIST) as a U.S. Federal Information Processing Standard (FIPS).

The Secure Hash Algorithm (SHA), developed by NIST, along with the NSA, for use with the Digital Signature Standard (DSS) is specified within the Secure Hash Standard (SHS) [National Institute of Standards and Technology (NIST). FIPS Publication 180: Secure Hash Standard (SHS). May 1993.

SHA-1 [National Institute of Standards and Technology (NIST). Announcement of Weakness in the Secure Hash Standard. May 1994.] was a revision to SHA that was published in 1994. The revision corrected an unpublished flaw in SHA.

SHA is a cryptographic message digest algorithm similar to the MD4 family of hash functions developed by Rivest .It differs in that it adds an additional expansion operation, an extra round and the whole transformation was designed to accommodate the DSS block size for efficiency.

The Secure Hash Algorithm takes a message of less than 2^{64} bits in length and produces a 160-bit message digest which is designed so that it should be computationally expensive to find a text which matches a given hash.

II. OVERVIEW OF SHA

At CRYPTO 98, two French researchers, Florent Chabaud and Antoine Joux, presented an attack on SHA-0 (Chabaud and Joux, 1998): collisions can be found with complexity 2^{61}, fewer than the 2^{80} for an ideal hash function of the same size.

In 2004, Biham and Chen found near-collisions for SHA-0—two messages that hash to nearly the same value; in this case, 142 out of the 160 bits are equal. They also found full collisions of SHA-0 reduced to 62 out of its 80 rounds.

On 12 August 2004, a collision for the full SHA-0 algorithm was announced by Joux, Carribault, Lemuet, and Jalby. This was done by using a generalization of the Chabaud and Joux attack. Finding the collision had complexity 2^{51} and took about 80,000 CPU hours on a supercomputer with 256 Itanium 2 processors. (Equivalent to 13 days of full-time use of the computer.)

ISSN: 2321-9653

On 17 August 2004, at the Rump Session of CRYPTO 2004, preliminary results were announced by Wang, Feng, Lai, and Yu, about an attack on MD5, SHA-0 and other hash functions. The complexity of their attack on SHA-0 is 2^{40}, significantly better than the attack by Joux *et al.*

In February 2005,.In light of the results for SHA-0, some experts suggested that plans for the use of SHA-1 in new cryptosystems should be reconsidered.

After the CRYPTO 2004 results were published, NIST announced that they planned to phase out the use of SHA-1 by 2010 in favor of the SHA2 variants.

III. VERSIONS IN SHA

- SHA-0: A retronym applied to the original version of the 160-bit hash function published in 1993 under the name "SHA". It was withdrawn shortly after publication due to an undisclosed "significant flaw" and replaced by the slightly revised version SHA-1.

- SHA-1: A 160-bit hash function which resembles the earlier MD5 algorithm. This was designed by the National Security Agency (NSA) to be part of the Digital Signature Algorithm. Cryptographic weaknesses were discovered in SHA-1, and the standard was no longer approved for most cryptographic uses after 2010.

- SHA-2: A family of two similar hash functions, with different block sizes, known as SHA-256 and SHA-512. They differ in the word size; SHA-256 uses 32-bit words where SHA-512 uses 64-bit words. There are also truncated versions of each standard, known as SHA-224 and SHA-384. These were also designed by the NSA.

- SHA-3: A hash function formerly called Keccak, chosen in 2012 after a public competition among non-NSA designers. It supports the same hash lengths as SHA-2, and its internal structure differs significantly from the rest of the SHA family.

The corresponding standards are FIPS PUB 180 (original SHA), FIPS PUB 180-1 (SHA-1), FIPS PUB 180-2 (SHA-1, SHA-256, SHA-384, and SHA-512). NIST has said that FIPS 180-5 will include SHA-3.

	SHA-1	SHA-224	SHA-256	SHA-384	SHA-512
Message Digest Size	160	224	256	384	512
Message Size	$<2^{64}$	$<2^{64}$	$<2^{64}$	$<2^{128}$	$<2^{128}$
Block size	512	512	512	1024	1024
Word size	32	32	32	64	64
Number of Steps	80	64	64	80	80

Comparison of SHA Parameters

IV. STEPS FOR SHA ALGORITHM

The steps of SHA-1 algorithm are

1. Padding

○ Pad the message with a single one followed by zeroes until the final block has 448 bits.

○ Append the size of the original message as an unsigned 64 bit integer.

2 Initialize the 5 hash blocks (h0, h1, h2,h3,h4) to the specific constants defined in the SHA1standard.

3 Hashes (for each 512bitBlock)

○ Allocate an 80 word array for the message schedule

■ set the first 16 words to be the 512bitblock split into 16 words.

■ the rest of the words are generated using the following algorithm

Word [i3]

XOR word [i8]

XOR word [i14]

XOR word [i16]

Then Rotated 1 bit to the left.

○ Loop 80 times doing the following. (Shown in Image1)

■ Calculate SHA function () and the constant K (these are based on the Current round number.

■ e=d

■ d=c

■ c=b (rotated left 30)

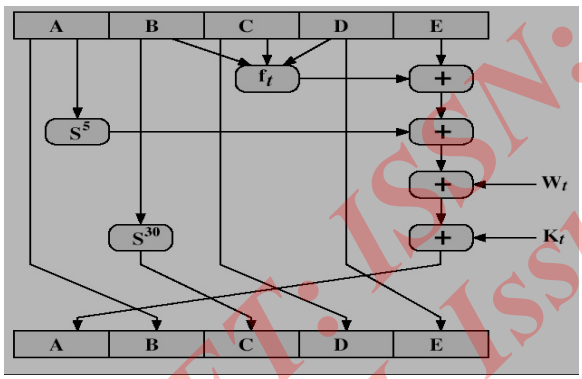
■ b=a

■ a = a (rotated left 5) + SHAfunction() + e + k + word[i]

Add a,b,c,d and e to the hash output.

Output the concatenation (h0,h1,h2,h3,h4) which is the message digest.

Functions in SHA



Example of SHA-1 Algorithm

Input(Text file	Output(SHA1 Hash)
hi	8afg3dh4sfg5adnm3gh2jkm3cbg5jk4nhjkl4bn5
hello	Gh2h4jjkl5lnbh5jkl7mmk8lkhdghj2kklrn4nb5b
This is very good	Ghkl3jbv4bnmj5klg5bm,l5lhgj7bhkl6lop3b5

V. SHA-1 DESIGN AND ANALYSIS

Message compress standard SHA is designed for DSS. The input of SHA is a message which is no longer than

264 bit, and it can generate a 160 bit message abstract[2]. If a message no longer than 264 bit, it needs to be added zeros to make the message become a 264 bit one. And if a message longer than 264 bit, it need to be separated into several groups.

Every group contains 264 bit. Then the message groups will be converted into message abstract groups by SHA

algorithm.[2,3] When message abstract is generated, five 32 bit initial values A, B, C, D, E will be used

A=0x67452301
B=0xefcdab89
C=0x98badcfe
D=0x10325476
E=0xc3d2e1f0

Every time SHA-1 operates, non-linear function Ft, constant Wt and Kt are different if t is different value.

According to parameter t, the non-linear function Ft is

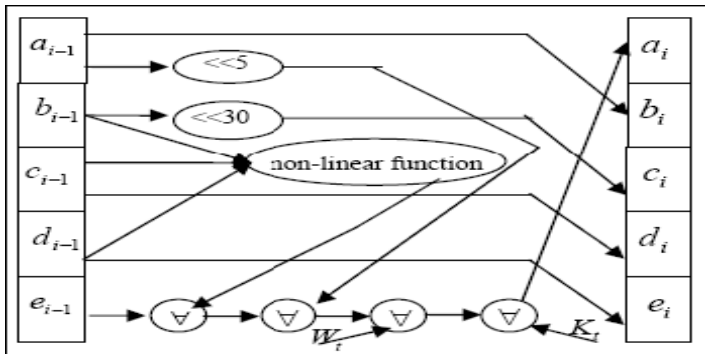
$$\begin{aligned} F_t(x,y,z) &= (x \wedge y) \vee (\overline{x} \wedge z) && (t=0\sim19)\\ F_t(x,y,z) &= x \forall y \forall z && (t=20\sim39)\\ F_t(x,y,z) &= (x \wedge y) \vee (x \wedge z) \vee (y \wedge z) && (t=40\sim59)\\ F_t(x,y,z) &= x \forall y \forall z && (t=60\sim79) \end{aligned}$$

The symbol ‘ л ’ means and logic. ‘ v ’ means or logic. means the opposite number of x. Constant Kt is different according to parameter t.

$$\begin{aligned} K_t &= 0x5a82799 && (t=0\sim19)\\ K_t &= 0x6ed9eba1 && (t=20\sim39)\\ K_t &= 0x8f1bbcdc && (t=40\sim59) \end{aligned}$$

Show SHA-1 Operated Flowchart in One Time.

ISSN: 2321-9653



SHA-1 Operated Flowchart in One Time

The SHA-1 message abstract module can be designed according to the algorithm. The SHA-1 message abstract H(m) is a 160 bit abstract[3] combined with an, bn, cn, dn, en.

SHA-1 arithmetic:

Input: message m (m is the message no longer than 2^{64} bit)
Output: SHA-1 message abstract H(m)
S1. According to the rule, we can find the right W_t, K_t, K_1, K_2, K_3, when t are different.
S2. assign the initial value, a_0=A, b_0=B, c_0=C, d_0=D, e_0=E,
S3. While t from 1 to 79, loop the follow step S3.1
S3.1. $e = F_t(x,y,z) + W_t + K_t + e_{t-1} + (a_{t-1}<<5)$
$a_i = e$, $b_i = a_{i-1}$, $c_i = (b_{i-1}<<30)$, $d_i = c_{i-1}$, $e_i = d_{i-1}$
Return: SHA-1 message abstract H(m)

VI. APPLICATIONS OF SECURE HASH ALGORITHM

SHA-1 can be used in a variety of applications:

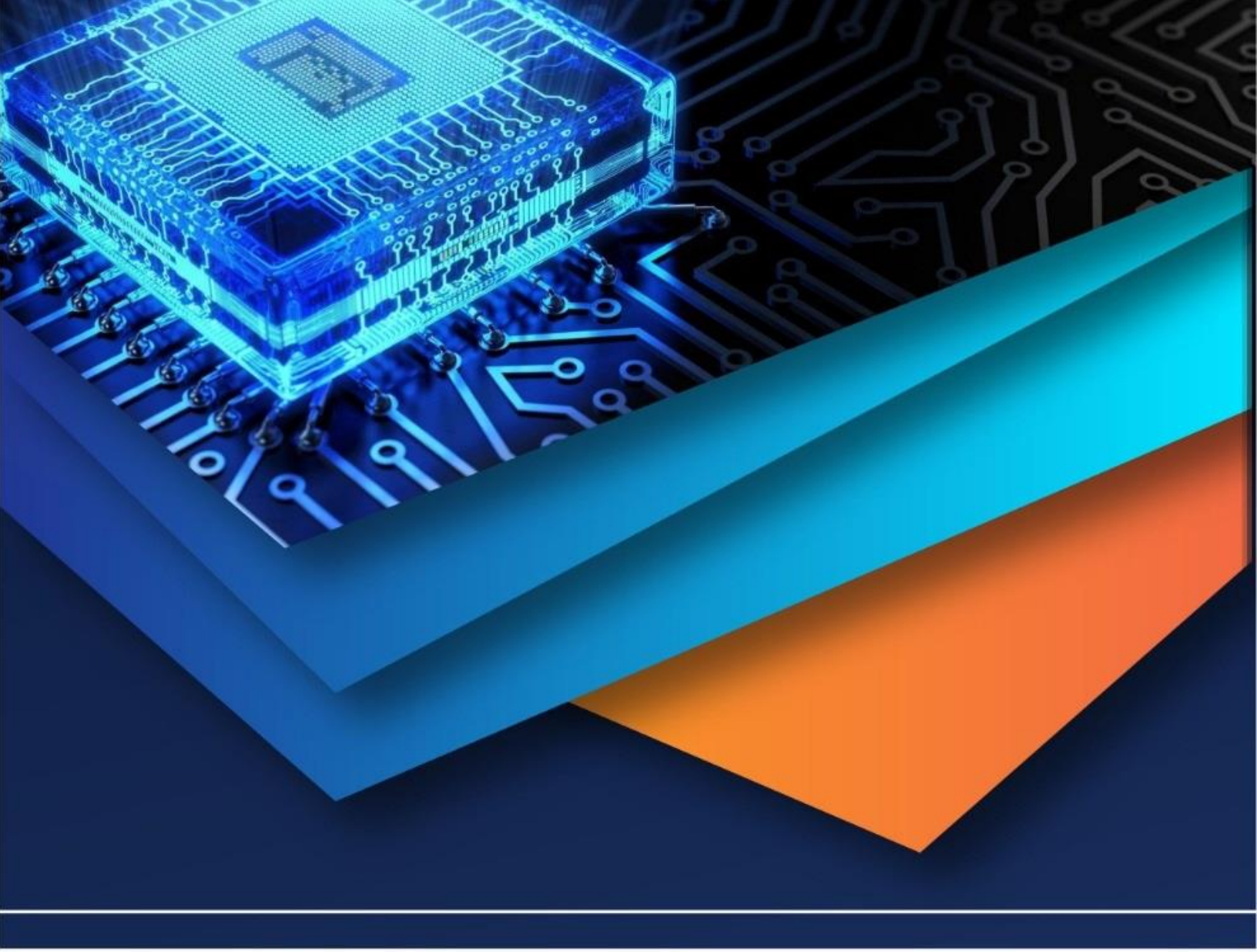
1. Security applications that require authentication
2. E-mail
3. Electronic funds transfer
4. Software distribution
5. Data storage

VII. CONCLUSION

The use of cryptographic hash functions like MD5 or SHA for message authentication has become a standard approach in many Internet applications and protocols. Though very easy to implement, these mechanisms are usually based on adhoc techniques that lack a sound security analysis.

REFERENCES

[1] William Stallings, "Cryptography and Network Security, Principles and Practices" Fourth Edit io n, 2005.
[2] NIST "SECURE HASH STANDARD", Federal Information Processing Standards Publication 180-1, May 1993.
[3] NIST "SECURE HASH STANDARD", Federal Information Processing Standards Publication 180-2, August 2002.
[4] Ilya Mironov, "Hash functions: Theory, a t t a c k s and applications",Microsoft Research, Silicon Valley Campus, November 14, 2005.
[5] K. Jarvinen, "Design and Implementation of a SHA-1 Hash Module on FPGAs", Technical Report, Otakaari 5A, Espoo, Finland, November, 2004.
[6] Dai Zibin and Zho u Ning, "FPGA Implementation of SHA-1 Algorithm", IEEE Proceedings, 5th International Conference on ASIC,2003.
[7] A.P. Kakarountas, G. Theodoridis, T. Laopoulos a nd C.E.Goutis, "High-Speed FPGA Implementation of the SHA-1 Hash Function", IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems:Technology and Applications, Sofia, Bulgaria, 2005.
[8] Guopyin Wang, "An Efficient Implementation of SHA-1 Hash Function",IEEE International Conference on Electro or Information Technology, pp.575-579, 2006.
[9] Cheng Xiao-hui and Deng Jian-zhi, "Design of SHA-1 Algorithm based on FPGA", IEEE Second International Conference on Networks Security, Wireless Communications and Trusted Computing, (NSWCTC), Vol-1, pp. 532-534, 2010.
[10] Zhou Hua and Liu Qiao, "Hardware Design for SHA-1 Based FPGA",IEEE International Conference Publications on Electronics, Communications and Control (ICECC), pp.2076-2078, 2011.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY