INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

SECURED HASH ALGORITHM-1: Review Paper

CHAITYA B. SHAH[1], DRASHTI R. PANCHAL[2]

[1]Indus Institute of Technology and Engineering, Gujarat Technological University,
B/2, Swastik App., Vasna Barrage Road, Vasna, Ahmedabad - 380007, India,
chaitya1001@gmail.com
[2]Indus Institute of Technology and Engineering, Gujarat Technological University,
Neha Bungalow, Sindur Society, Ishwarbhuvan Road, Navjivan Post, Ahmedabad - 380014, India,
drpanchal93@gmail.com

Abstract: *This paper discusses about the Secure Hash Algorithm. To throw some light on the origins of Secure Hash Algorithm, the SHA was designed by NSA-National Security Agency which is a U.S. Federal Information Processing Standard (FIPS) published by the United States NIST-National Institute of Standards and Technology. Secure Hash Algorithm is abbreviated as SHA. SHA has many versions starting from SHA-0 followed by SHA-1, SHA-2 etc. Here the main discussion is about SHA-1. Before starting with the SHA-1 it is necessary to know about SHA-0 and why was SHA-1 introduced after SHA-0.SHA-1 is the modified version of SHA-0 which was published in 1993. SHA-0 was not adopted by many applications so the NSA proposed SHA-1 after discovering some weaknesses in the SHA-0. SHA-1 was published in 1995 and holds certification of FIPS PUB 180-4 and CRYPTREC [1].*

Keywords: *Secure Hash Algorithm, SHA-1, Cryptography, computer security, message digest, hash function, hash algorithm, Secure Hash Standard.*

1. INTRODUCTION

The SHA i.e. Secure Hash Algorithm is basically based on the concept of hash function. The basic idea of a hash function is that it takes a variable length message as input and produces a fixed length message as output which can also be called as hash or message-digest. The trick behind building a good, secured cryptographic hash function is to devise a good compression function in which each input bit affects as many output bits as possible [2]. It is used with the Digital Signature Standard (DSA) for digital signature so it has a particular importance.

SHA-1 has a set of cryptographic hash functions very similar to the MD family of hash functions. But MD family uses more bits in hash function. That is the main difference between MD and SHA-1. Because of this difference SHA-1 is more secure. SHA-1 differs from SHA-0 only by a single bitwise rotation in the message schedule of its compression function. This was done by the NSA in order to correct the flaw in the original algorithm which reduced its cryptographic security. The NSA did not provide any further explanation or it did not identify the flaw that was corrected. SHA-1 appears to provide greater resistance to attacks. In SHA-1 input data is called message and the hash value is called message digest. Hash function takes a variable length message as an input and as an output produces a fixed length message which can also be called hash or message digests [3]. SHA-1 has a message size of 264 bits and a message digest of 160 bits. SHA-1 is designed so that it is practically infeasible to find output of the two input messages the same. It is also impossible to get back the input message from the obtained message digest.

2. LITERATURE REVIEW

Wikipedia is a free encyclopaedia. This page contains a collection of information about the SHA algorithms, history and future works [1]. Wade Trappe and Lawrence C. Washington explain about Classic Cryptosystem, Basic Number Theory and The Data Encryption Standard in this book. In the part of Classic Cryptosystem they suggest the need of good compression function [2].

This paper is published by Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, in 2012. This paper is about Secure Hash Standard (SHS). This standard specifies hash algorithms that can be used to generate digests of messages. The digests are used to detect whether messages have been changed since the digests were generated [3]. In this book, Atul Kahate has explained in detail about Cryptography and Hash Functions. This book includes many cryptography algorithms and standards [4]. This paper explains about SHA Algorithm, Original Design of the Algorithm and then the revised design [5]. Vincent Rijmen and Elisabeth Oswald explain in this paper about experiments in order to assess the security of SHA-1 against the attack by Chabaud and Joux [6].

This article suggests about an attack on SHA-1, found by the research team of Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu (mostly from Shandong University in China) [7] And explains the attack in more detail further [8]. In this report, Christophe De Cannière and Christian Rechberger describe a method to search for characteristics in an automatic way. This is particularly useful for multi-block attacks, and as a proof of concept, they gave a two-block collision for 64-step SHA-1 based on a new characteristic [9].

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

This paper provides a brief report on the collision search for the reduced SHA-1. With a few improvements to the De Canni\`ere-Rechberger automatic collision search method E.A.Grechnikov managed to construct two new collisions for 72- and 73-step reduced SHA-1 hash function [10]. This news article shows that how the SHA-1 hash function is under pressure and also shiws security threats [11]. Marc Stevens has explained the most efficient attack on SHA-1 in this paper [12]. In an article 'When Will We See Collisions For SHA-1?' by Schneier , he discusses about the estimation of Jesse Walker that how much will it take to find a practical collision on SHA-1 [13]. Amit Keswani and Vaibhav Khadilkar discuss about SHA-1 algorithms and gives some examples of it in this paper [14].

3. ALGORITHM

The SHA-1 is closely modelled after MD5 .The MD5 produces message digest of 128 bits whereas SHA-1 produces message digest of 160 bits i.e. 32 bits more than the message digest produced by MD5. After some initial processing the input text is processed in 512 bit blocks which are further divided into 16 32-bit blocks. The output of SHA-1 algorithm is 5 32-bit blocks making a total of 160 bits message digest.

3.1 PROCEDURE

There are various steps involved in the SHA-1. They are listed as follows [4]:

1. Message Padding
2. Append Length
3. Divide the Input into 512 bit blocks
4. Initialize chaining variables
5. Process Blocks
   5.1 Copy variables to register
   5.2 Divide one 512 bit block into 16 blocks of 32 bit each
   5.3 4 rounds, each round consisting of 20 steps.
   5.4 Diagram + process P + all chaining variables.

Step 1: Padding

Adding padding bits to the original message is the first step of SHA-1 algorithm. The main objective of this step is to make the length of the original message equal to a value which is 64 bits less than an exact multiple of 512.

For example, if the original message is 900 bits, then we add a padding of 60 bits which makes the message length 960 which is hence 64 bits less than 1024 (1024=512 × 2). The padding consists of a single 1 bit followed by as many 0's bits as required. It is mandatory to add padding bits even if the original message length is itself 64 bits less than the multiple of 512.

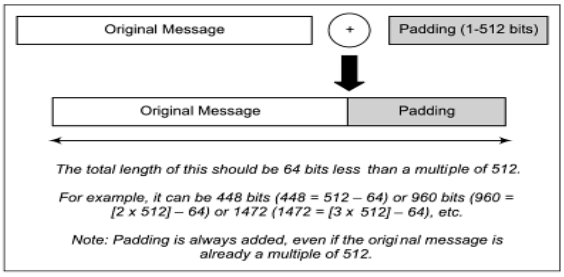

Figure 1 : Padding

Step 2: Append Length

The next step after adding padding bits is to calculate the original length of the message and append it to the end of the message after padding. Now the question is how is it done? The length of the message is calculated excluding the padding bits i.e. the length of the message before the padding bits were added. For example, if the original message was of 900 bits and a padding of 60 bits was added to make the length 64 bits less than the multiple of 512 then here the length is considered 900 bits instead of 960 bits. This length is now expressed as a 64 bit value and appended to the end of the original message + padding. This process is better explained by the figure. Now if the length of the original message exceeds 264 bits then only lower order 64 bits are used here i.e. length mod 264 is calculated in that case.

Hence the length of the message is now an exact multiple of 512. This becomes the message whose message digest will be calculated.

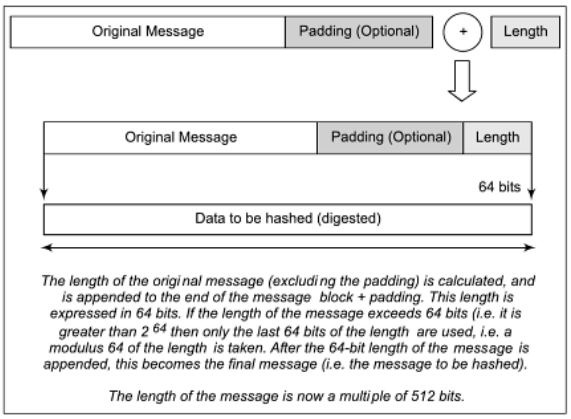

Figure 2 : Append Length

Step 3: Divide the Input into 512 bit blocks

The next step is to divide the input message into blocks, each of length 512 bits. Now these blocks become the input to the message digest processing logic.

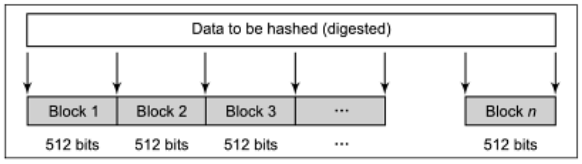

Figure 3 : Divide The Input into 512 bits blocks

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

Step 4: Initialize the chaining variables

There are five chaining variables A through E. These five chaining variables are initialized in this step. MD5 had four chaining variables each of 32 bits (total length will be 4 × 32=128 bits) but in the case of SHA-1 we need a message digest of 160 bits hence there are five chaining variables here making a total of 5 × 32=160 bits. The values for these chaining variables are as shown in the figure.

A	Hex	01	23	45	67
B	Hex	89	AB	CD	EF
C	Hex	FE	DC	BA	98
D	Hex	76	54	32	10
E	Hex	C3	D2	E1	F0

Figure 4 : Initialize the chaining variables

Step 5: Process Blocks

Here the actual algorithm begins.

Step 5.1:

The chaining variables 'A-E' are copied in five registers 'a-e', resulting in a combined register 'abcde' which will be considered as a single register for storing the temporary intermediate as well as the final results.

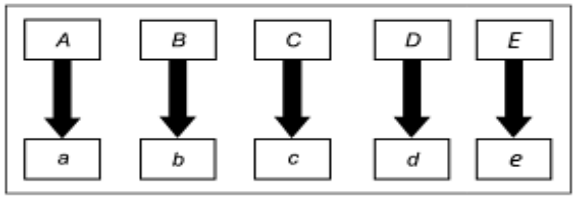

Figure 5 : Registers

Step 5.2:

In this step the current 512 bit block is divided into 16 sub-blocks of 32 bits each.

Step 5.3:

SHA-1 has four rounds each of 20 steps. As inputs to one round are current 512 bit block, the register abcde and a constant K[t] where t = 0 to 79. The contents of register abcde are updated using the SHA-1 algorithm steps. Here there are only four constants defined for K[t], one used in each round. The values of K[t] are shown in figure.

Table 1 : Values of K[t]

Round	Values of t between	K[t] in hexadecimal	K[t] in decimal (only integer portion of the value)
1	1and 19	5A 92 79 99	$2^{30} \times \sqrt{2}$
2	20 and 39	6E D9 EB A1	$2^{30} \times \sqrt{3}$
3	40 and 59	9F 1B BC DC	$2^{30} \times \sqrt{5}$
4	60 and 79	CA 62 C1 D6	$2^{30} \times \sqrt{10}$

Step 5.4:

The SHA-1 consists of four rounds each consisting of 20 iterations which makes a total of 80 iterations. The entire operation of SHA-1 is shown in figure [4].

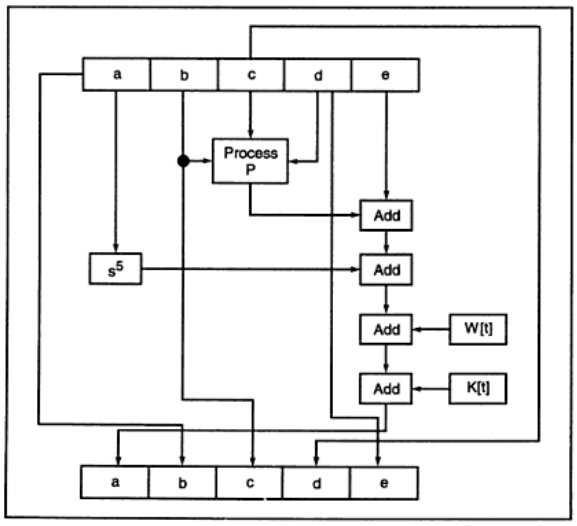

Figure 6: Algorithm Operations

To represent mathematically the operations in one iteration,

$a = (e + \text{Process P} + s^5(a) + W[t] + K[t])$
$b = a$
$c = s^{30}(b)$
$d = c$
$e = d$

Combining all of these results in [4],
Abcde = (e + Process P + s^5(a) + W[t] + K[t]), a, s^{30}(b), c, d
Where,
abcde = the register made of 5 chaining variables
Process P = logical operation.

Table 2 : Process P

Round	Process P
1	(b AND c) OR ((NOT b) AND (d))
2	B XOR c XOR d
3	(b AND c) OR (b and D) OR (c AND d)
4	B XOR c XOR d

s^t = circular left shift of the 32 bits sub-block by t bits
W[t] = a 32 bit derived from the current 32 bit sub-block calculated as follows,

1. For the first 16 words of W (i.e. t = 0 to 15), the contents of the input message sub-block M[t] become the contents of W[t] directly.
2. For remaining 64 values of W are derived using the equation:
   W[t] = s^1 (W[t – 16] XOR W[t – 14] XOR W[t – 8] XOR W[t – 3])
   s^1 = Circular-left shift by 1 bit position

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

3.2 APPLICATION

Secured Hash Algorithm is widely being used in TLS, SSL, SSH, PGP, S/MIME and IPsec [5][1].
SHA-1 hashing is also used in distributed revision control systems like Git, Mercurial, and Monotone to identify revisions, and to detect data corruption or tampering. The algorithm has also been used on Nintendo's Wii gaming console for signature verification when booting [1]. Actually Git uses it for the integrity not for cryptography.

3.3 ATTACKS

First of all in early 2005, Rijmen and Oswald discovered an attack on a reduced version of SHA-1 — 53 out of 80 rounds — which finds collisions with a computational effort of fewer than 280 operations [6]. In February 2005, an attack by Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu was announced [7]. The attacks can find collisions in the full version of SHA-1, requiring fewer than 269 operations [1].
On 17 August 2005, an improvement on the SHA-1 attack was announced on behalf of Xiaoyun Wang, Andrew Yao and Frances Yao at the CRYPTO 2005 rump session, lowering the complexity required for finding a collision in SHA-1 to 263 [8]. Christophe De Cannière and Christian Rechberger further improved the attack on SHA-1 in "Finding SHA-1 Characteristics: General Results and Applications." [9] A two-block collision for 64-round SHA-1 was presented, found using unoptimized methods with 235 compression function evaluations. Their attack was extended further to 73 rounds (of 80) in 2010 by Grechnikov [10].
At the Rump Session of CRYPTO 2006, Christian Rechberger and Christophe De Cannière claimed to have discovered a collision attack on SHA-1 that would allow an attacker to select at least parts of the message [11].
As of 2012, the most efficient attack against SHA-1 is considered to be the one by Marc Stevens [12] with an estimated cost of $2.77M to break a single hash value by renting CPU power from cloud servers [13].

4. NEW PROPOSED SCHEME

After some improvements in SHA-1, NSA designed a set of hash functions named SHA-2. SHA-2 was first published in 2001 by NIST. And in August 2002, it became the new Secure Hash Standard. It provides better security than SHA-1 because it has bigger message digest size. So, because of the bigger message digest, it is harder to break. The SHA-2 family consists of six hash functions with digests (hash values) that are 224, 256, 384 or 512 bits: SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224 and SHA-512/256 [1].
Newest proposed scheme for hash functions is SHA-3. In May 2014 NIST announced the SHA-3, subset of the cryptographic primitive family Keccak and a cryptographic hash function designed by Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche, building upon RadioGatún. SHA-3 is not meant to replace SHA-2, as no significant attack on SHA-2 has been demonstrated. Because of the successful attacks on MD5 and SHA-0 and theoretical attacks on SHA-1 and SHA-2, NIST perceived a need for an alternative, dissimilar cryptographic hash, which became SHA-3 [1].

5. CONCLUSION AND FUTURE WORK

The Secure Hash Algorithm (SHA-1) is used for computing a compressed representation of a message. If we give an input message of arbitrary length < 264 bits, it produces a 160-bit output called the message digest. The SHA-1 algorithm is claimed to be secure because it is practically infeasible to compute the message corresponding to a given message digest. Also it is extremely improbable to detect two messages hashing to the same value [14].
So, these days most people still use SHA1 or even MD5, broken or not. Because the current state of the art in hashing is that we have some functions that we know have theoretical vulnerabilities but no real practical breaks, and some unproven functions that we know very little about at all. Even if there has never been a successful complete collision with SHA1, the evolution of our computers' calculation capacities will soon make it possible. So, giant companies like Google, Microsoft, etc planning to kill SHA-1in near future to make web more secure.

References

[1] SHA hash functions - Wikipedia, the free encyclopaedia [online], Available: http:// en.wikipedia.org/wiki/SHA1, http://en.wiki pedia.org/wiki/SHA2, http://en.wiki pedia. org/ wiki/ SHA3, [Accessed : Oct. 7, 2014]

[2] Wade Trappe, Lawrence C. Washington. Introduction to Cryptography with Coding Theory. New Jersey: Pearson Prentice Hall , 2006.

[3] Secure Hash Standard(SHS), FIPS PUB 180-4, FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION, March 2012, http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf

[4] Atul Kahate, Cryptography and Network Security, 3rd Edition , McGraw Hill Education (India) , 2013.

[5] SHA1 Description, B Thomas Golisano College,http://www.cs.rit.edu/~ bcw5910/482 TeamFlux.pdf

[6] Vincent Rijmen and Elisabeth Oswald, Cryptology ePrint Archive: Report 2005/010, Update on SHA-1, http://eprint. iacr.org/2005/010

[7] Schneier on Security: SHA-1 Broken, Feb. 15, 2005, https://www.schneier.com/blog/

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

archives/2005/02/sha1_broken.html [Accessed : Oct. 7, 2014]

[8] Schneier on Security: New Cryptanalytic Results Against SHA-1, Aug. 17, 2005 , https://www.schneier.com/blog/archives/2005/08/new_cryptanalyt.html [Accessed : Oct. 7, 2014]

[9] Christophe De Cannière, Christian Rechberger (2006-11-15). Finding SHA-1 Characteristics: General Results and Applications, 2006, http://link.springer.com/chapter/10.1007%2F11935230_1

[10] "Collisions for 72-step and 73-step SHA-1: Improvements in the Method of Characteristics", Cryptology ePrint Archive: Report 2010/413, Jul. 23, 2010, http://eprint.iacr.org/2010/413

[11] SHA-1 hash function under pressure – heise Security : http://www.heise.de [Accessed : Oct. 7, 2014]

[12] Marc Stevens, CWI, Amsterdam, Cryptanalysis of MD5 & SHA-1 : http://2012.sharcs.org/slides/stevens.pdf [Accessed : Oct. 7, 2014]

[13] When Will We See Collisions for SHA-1? ‘, Oct. 5, 2012, https://www.schneier.com /blog/archives/2012/10/when_will_we_se.html [Accessed : Oct. 7, 2014]

[14] Amit Keswani and Vaibhav Khadilkar, THE SHA-1 ALGORITHM, Lamar University Computer Science Department, Beaumont, TX 77710, USA: http://cs.lamar.edu/faculty/osborne/5340_01/summer_06/project/SHA/Project_Paper.