An efficient working for secure hash algorithm-256

[1]Govind Singh, [2]Siddharth Choubey
[1, 2] Department of Computer Science & Engineering
[1,2]Shri Shankaracharya College of Engineering & Technology, Bhilai, (C.G.)

[1]singh.govind.ssitm@gmail.com
[2]Siddhartha00@rediffmail.com

***Abstract*— Encryption and Decryption provide secrecy, or confidentially, but not Integrity. One way to preserve the Integrity of a document is Through the use of Fingerprint. The electronic equivalent of the Document and Fingerprint pair is the message and message digest pair. To preserve the integrity of a message, the message is passed through an Algorithm called a hash function. The hash function creates a compressed image of the message that can be used as a Fingerprint. The two pairs document/fingerprint and message/message digest and similar, with some difference. The document and fingerprint are physically linked together; also, neither needs to be kept secret. The message and message digest can be unlinked (or sent) separately and, most importantly, the message digest needs to be kept secret. The message digest is either kept secret in a safe place or encrypted if we need to send it through a communication channel.**

Keywords — Federal Information Processing Standard (FIPS), Secure Hash Algorithm (SHA), Secure Hash Standard (SHS).

I. INTRODUCTION

To be eligible for a hash, a function needs to meet three criteria:-one-wayness, resistance to Weak collision, and resistance to strong collision.SHA-256 is a designed by the National Institute of Standard and Technology (NIST). It was published as a Federal Information Processing This standard specifies four secure hash algorithms, SHA-11, SHA-256, SHA-384, and SHA- 512. All four of the algorithms are iterative, one-way hash functions that can process a message to produce a condensed representation called a message digest. These algorithms enable the determination of a message's integrity: any change to the message will, with a very high probability, result in a different message digest. This property is useful in the generation and verification of digital signatures and message authentication codes, and in the generation of random numbers (bits). Each algorithm can be described in two stages: pre-processing and hash computation. Pre-processing involves padding a message, parsing the padded message into m-bit blocks, and setting initialization values to be used in the hash computation. The hash computation generates a message schedule from the padded message and uses that schedule, along with functions, constants, and word operations to iteratively generate a series of hash values. The final hash value generated by the hash computation is used to determine the message digest. The four algorithms differ most significantly in the number of bits of security that are provided for the data being hashed – this is directly related to the message digest length. When a secure hash algorithm is used in conjunction with another algorithm, there may be requirements specified elsewhere that require the use of a secure hash algorithm with a certain number of bits of security. For example, if a message is being signed with a digital signature algorithm that provides 128 bits of security, then that signature algorithm may require the use of a secure hash algorithm that also provides 128 bits of security (e.g., SHA-256).Additionally, the four algorithms differ in terms of the size of the blocks and words of data that are used during hashing. Figure 1 presents the basic properties of all four secure hash algorithms.

Algorithm	Message Size (Bits)	Block Size (Bits)	Word Size (Bits)	Message Digest Size (Bits)	Security (Bits)
SHA-1	$<2^{64}$	512	32	160	80
SHA-256	$<2^{64}$	512	32	256	128
SHA-384	$<2^{128}$	1024	64	384	192
SHA-512	$<2^{128}$	1024	64	512	256

Figure 1: Secure Hash Algorithm Properties.

II. EVOLUTION

The Secure Hash Algorithm (SHA) was developed by the National Institute of Standards and technology(NIST) And Published as a federal information processing standards(FIPS 180) in 1993, a revised version was issued as FIPS 180-1 in 1995 and is generally referred to as SHA-1. NIST produced a revised version of the standards FIPS 180-2, that defined three new version of SHA, with hash value lengths of 256, 384, 512bits, known as SHA-256,SHA-384 & SHA-512.

The design, implementation and System level performance of an efficient yet compact field programmable gate array(FGPA) based secure hash algorithm(SHA-256) processor is represented.

The four algorithms differ most significantly in the number of bits of security that are provided for the data being hashed – this is directly related to the message digest length. This Standard specifies four secure hash algorithms - SHA-1, SHA-256, SHA-384, and SHA-512 - for computing a condensed representation of electronic data(message). When a message of any length < 2^64 bits (for SHA-1 and SHA-256) or <2^128 bits (for SHA-384 and SHA-512) is input to an algorithm, the result is an output called a message digest. The message digests range in length from 160 to 512 bits, depending on the algorithm. While it is the intent of this standard to specify general security requirements for generating a message digest, conformance to this standard does not assure that a particular implementation is secure.

The responsible authority in each agency or department shall assure that an overall implementation provides an acceptable level of security. This standard will be reviewed every five years in order to assess its adequacy.A hash algorithm is used to map a message of arbitrary length to a fixed-length message digest. Federal Information Processing Standard (FIPS) 180-3, the Secure Hash Standard (SHS) [FIPS 180-3], specifies five approved hash algorithms: SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512. Secure hash algorithms are typically used with other cryptographic algorithms. This Recommendation provides security guidelines for achieving the required or desired security strengths of several cryptographic applications that employ the approved cryptographic hash functions specified in Federal Information Processing Standard (FIPS) 180-3 [FIPS 180-3], such as digital signature applications [FIPS 186-3], Keyed-hash Message Authentication Codes (HMACs) [FIPS 198-1] and Hash-based Key Derivation Functions (HKDFs) [SP 800-56A] & [SP 800-56B].

This Recommendation has been developed by the National Institute of Standards and Technology (NIST) in furtherance of its statutory responsibilities under the Federal Information Security Management Act (FISMA) of 2002, Public Law 107-347. NIST is responsible for developing standards and guidelines, including minimum requirements, for providing adequate information security for all agency operations and assets, but such standards and guidelines shall not apply to national security systems. This recommendation is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130, Section 8b(3), Securing Agency Information Systems, as analyzed in A-130, Appendix IV: Analysis of Key Sections. Supplemental information is provided in A-130, Appendix III.

This Recommendation has been prepared for use by Federal agencies. It may be used by non-governmental organizations on a voluntary basis and is not subject to copyright (attribution would be appreciated by NIST). Nothing in this Recommendation should be taken to contradict standards and guidelines made mandatory and binding on Federal agencies by the Secretary of Commerce under statutory authority. Nor should this Recommendation be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other federal official.

Conformance testing for implementations of this Recommendation will be conducted within the framework of the Cryptographic Module Validation Program (CMVP), a joint effort of NIST and the Communications Security Establishment of the Government of Canada.

Beginning January 1, 2011, the Federal government requires the use of SHA-256 in all digital signatures generated by Certification Authorities (CAs) signing Personal Identity Verification (PIV) Cards. In addition, beginning January 1, 2011, the Federal government requires the use of SHA-256 in all digital signatures generated. While some limited use of SHA-1 in a deprecated mode is allowed, use in the PIV environment is not permitted after January 1, 2011. The risk of continued use of SHA-1 is significant, and 80-bit security strength for cryptography does not provide an acceptable level of protection. These risks increased the urgency for transition. Therefore, the FPKI Community transitioned its infrastructures to the stronger SHA-256 algorithm.

The FPKI SHA-256 Working Group was established to support the transition from SHA-1 to SHA-256 within the FPKI Community and provide a forum for inter-agency communication and information sharing. During and after the transition, a number of suggestions for improvement were discussed that could be applied to future transitions.

The security of Dynamic SHA 2 the outcome is that, despite the reliance of Dynamic SHA 2 on data-dependent rotations and even data-dependent functions, their security is subverted by the vast control and knowledge the adversary has while attacking a hash function. We also find out that Dynamic SHA2 is not suitable to be selected as SHA-3, because of there lack of security. Following table gives a clear picture of our results

Hash Function	Attack
Dynamic SHA-256	Collision
Dynamic SHA-512	Collision
Dynamic SHA-256	Second Preimage
Dynamic SHA-512	Second Preimage
Dynamic SHA-256	First Preimage
Dynamic SHA-512	First Preimage
Dynamic SHA2-256	Collision
Dynamic SHA2-512	Collision

III. ADVANTAGES OF HASH FUNCTION

1. Collision resistance: It is computationally infeasible to find two different inputs to the cryptographic hash function that have the same hash value. That is, if hash is a cryptographic hash function, it is computationally

infeasible to find two different inputs *x* and *x'* for which hash(*x*) = hash(x'). Collision resistance is measured by the amount of work that would be needed to find a collision for a cryptographic hash function with high probability. If the amount of work is 2N, then the collision resistance is Nbits. The estimated strength for collision resistance provided by a hash-function is half the length of the hash value, L, produced by a given cryptographic hash function. For example, SHA-256 produces a (full-length) hash value of 256 bits; SHA-256 provides an estimated collision resistance of 128 bits.

2. Preimage resistance2: Given a randomly chosen hash value, hash value, it is computationally infeasible to find an xso that hash(x)= hash value. This property is also called the one-way property. Preimage resistance is measured by the amount of work that would be needed to find a preimage for a cryptographic hash function with high probability. If the amount of work is 2N, then the preimage resistance is Nbits. The estimated strength for preimage resistance provided by a hash-function is the length of the hash value, L, produced by a given cryptographic hash function. For example, SHA-256 produces a (full-length) hash value of 256 bits; SHA-256 provides an estimated preimage resistance of 256 bits.

3. Second preimage resistance: It is computationally infeasible to find a second input that has the same hash value as any other specified input. That is, given an input x, it is computationally infeasible to find a second input x*'* that is different from *x*, such that hash(*x*) = hash(x*'*). Second preimage resistance is measured by the amount of work that would be needed to find a second preimage for a cryptographic hash function with high probability; more detail can be found in the Appendix A. If the amount of work is 2N, then the second preimage resistance is Nbits. The estimated strength for second preimage resistance provided by a hash-function is the length of the hash value, L, produced by a given cryptographic hash function. For example, SHA-256 produces a (full-length) hash value of 256 bits; SHA-256 provides an estimated second preimage resistance of 256 bits.

4. The security strength of a cryptographic hash function is determined by either: its collision resistance strength, preimage resistance strength or second preimage resistance strength, depending on the property(ies) that the cryptographic application needs from the cryptographic hash function.If an application requires more than one property from the cryptographic hash function, then the weakest property is the security strength of the cryptographic hash function for the application. For instance, the security strength of a cryptographic hash function for digital signatures is defined as its collision resistance strength, because digital signatures require collision resistance and second preimage resistance from the cryptographic hash function, and the collision resistance strength of the cryptographic hash function (L/2) is less than its second preimage resistance strength (i.e., L).

5. A cryptographic hash function that is not suitable for one application might be suitable for other cryptographic applications that do not require the same security properties. For example, SHA-1 is not suitable for digital signature applications (as specified in [FIPS 186-3]) that require 112 bits of security unless randomized hashing is used as discussed. However, SHA-1 can be used to provide 112 bits of security for HMAC applications (as specified in [FIPS 198-1]). In the case of digital signatures, SHA-1 does not provide 112 bits of collision resistance needed to achieve the security strength. On the other hand, SHA-1 does provide 112 bits of preimage resistance that is needed to achieve.

IV. STRENGTHS OF THE APPROVED HASH ALGORITHMS

Table 1 provides a summary of strengths of the security properties

	SHA-1	SHA-256	SHA-384	SHA-512
Collision Resistance strength in bits	$<80^3$	128	192	256
Preimage Resistance Strength in bits	160	256	384	512
Second Preimage Resistance Strength in bits	105-160	201-256	384	394-512

Table 1: Strengths of the Security Properties of Approved Hash Algorithms

TRUNCATED MESSAGE DIGEST

Some applications may require a message digest that is shorter than the (full-length) message digest provided by an approved cryptographic hash function specified in [FIPS 180-3]. In such cases, it may be appropriate to use a subset of the bits produced by the cryptographic hash function as the (shortened) message digest. For application interoperability, a standard method for truncating cryptographic hash function outputs (i.e., message digests) is provided strictly as a convenience for implementers and application developers. The proper use of a truncated message digest is an application-level issue.

Let the shortened message digest be called a truncated message digest, and let λ be its desired length in bits. A truncated message digest may be used if the following requirements are met:

1. If collision resistance is required, λ**shall** be at least twice the required collision resistance strength s(in bits) for the truncated message digest (i.e., $\lambda \geq 2s$).

2. The length of the output block of the approved cryptographic hash function to be used **shall** be greater than λ(i.e., L > λ).

3. The λ left-most bits of the full-length message digest **shall** be selected as the truncated message digest. For example, if a truncated message digest of 96 bits is desired, the SHA-256 cryptographic hash function could be used (e.g., because it is available to the application, and provides an output larger than 96 bits). The leftmost 96 bits of the 256-bit message digest generated by the SHA-256 cryptographic hash function are selected as the truncated message digest, and the rightmost 160 bits of the message digest are discarded. Truncating the message digest can impact the security of an application. By truncating a message digest, the estimated collision resistance strength is reduced from L/2 to λ/2 (in bits). For the example in item 3 above, even though SHA-256 provides 128 bits of collision resistance, the collision resistance provided by the 96-bit truncated message digest is half the length of the truncated message digest, which is 48 bits, in this case.

The truncated message digest of λbits provides an estimated preimage resistance of λbits, not Lbits, regardless of the cryptographic hash function used. The estimated second preimage resistance strength of a message digest truncated to λbits is determined as specified in the Appendix A. For example, a 130-bit truncated message digest generated using SHA-256 has an estimated second preimage strength of 130 bits, rather than a value in the range specified in Table 1 above for SHA-256.

Truncating the message digest can have other impacts, as well. For example, applications that use a truncated message digest risk attacks based on confusion between different parties about the specific amount of truncation used, as well as the specific cryptographic hash function that was used to produce the truncated message digest. Any application using a truncated message digest is responsible for ensuring that the truncation amount and the cryptographic hash function used are known to all parties, with no chance of ambiguity. It is also important to note that there is no guarantee that truncation will not make any truncated message digest weaker than its expected security strength.

SHA-256 FUNCTIONS

SHA-256 uses six logical functions, where each function operates on 32-bit words, which areRepresented as x, y, and z. The result of each function is a new 32-bit word.

$$\mathrm{Ch}(x,y,z) = (x \wedge y) \oplus (-x, \wedge z)$$

$$\mathrm{Maj}(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)$$

$$\sum_0^{\{256\}}(x) = \mathrm{ROTR}^2(x) \oplus \mathrm{ROTR}^{13}(x) \oplus \mathrm{ROTR}^{22}(x)$$

$$\sum_1^{\{256\}}(x) = \mathrm{ROTR}^6(x) \oplus \mathrm{ROTR}^{11}(x) \oplus \mathrm{ROTR}^{25}(x)$$

$$\sigma_0^{\{256\}}(x) = \mathrm{ROTR}^7(x) \oplus \mathrm{ROTR}^{18}(x) \oplus \mathrm{SHR}^3(x)$$

$$\sigma_1^{\{256\}}(x) = \mathrm{ROTR}^{17}(x) \oplus \mathrm{ROTR}^{19}(x) \oplus \mathrm{SHR}^{10}(x)$$

SHA-256

The following operations are applied to w-bit words in all secure hash algorithms. and SHA-256 operate on 32-bit words (w = 32).

1. Bitwise logical word operations: ∨, ⌐,∧ , and +.

2. Addition modulo 2w The operation x + y is defined as follows. The words x and y represent integers X and Y, where 0 <= X < 2w and 0 <= Y < 2w. For positive integers U and V, let U modV be The remainder upon dividing U by V. Compute Z = (X + Y) mod 2w. Then 0 <= Z < 2w. Convert the integer Z to a word, z, and define z = x + y.

3. The right shift operation SHR n(x), where x is a w-bit word and n is an integer with 0 <=n<w, is defined by SHR n(x) = x >> n.

4. The rotate right (circular right shift) operation ROTR n(x), where x is a w-bit word And n is an integer with 0 <= n < w, is defined by ROTRn(x) = (x >> n) ∨ (x << w - n). Thus, ROTR n(x) is equivalent to a circular shift (rotation) of x by n positions to the Right.

5. Note the following equivalence relationships, where w is fixed in each relationship:

ROTL n(x)ʃ ROTR w-n(x)

ROTR n(x)ʃ ROTL w-n(x).

V. CONCLUSION AND FURTHER DEVELOPMENT

The SHA-256 model represents a great security of computing system. Because the digest of msg is provide the best security of messages, it is in the service user's best interest to maximize utilization while still providing a high quality of service to the customer hence the ability are essential to achieving high rates of utilization and reduces the cost and provide flexibility because the communication can quickly provided to thousands of servers to make resources available as they're needed. In summary, I have studied a SHA-256 scenario to communicate between two users.

REFERENCES

1. [1] I Canadian Journal on Network and Information Security Vol. 1, No. 1, April2010:- Cryptanalysis of Dynamic SHA 2 AsimShahzad, SajjadHussain, SajjadAnjum.

2. NIST Special Publication 800-107 Recommendation for Applications Using Approved Hash Algorithms :-Quynh Dang,Computer Security Division , Information Technology Laboratory COMPUTER SECURITY , February 2009February 2009.

3. Federal PKI (FPKI) Community Transition to SHA-256 :-Version 1.0 January 18, 2011 .

4. An FGPA based SHA-256 Processor:-Kurt.K.Thing,C.L.Yuen& Philip H.W.Leong.Dept.of.Computer Science & Engineering,

The Chinese University of Hong Kong, New territories, Hong Kong.

5 Federal Public Key Infrastructure Policy Authority SHA-256 Transition Lessons Learned ,Version 1.0 ,May 21, 2011 .

6 Federal Information Processing Standards Publication 180-22002 :-August 1 Announcing the SECURE HASH STANDARD.

Govind Singh , B.E., M.E. Scholar in Computer Technology & Application from Shri Shankaracharya College of Engineering & Technology, Bhilai, India. Research areas are Computer Networks, wireless network & its enhancement.

Siddharth Choubey, Associate Professor in Dept. of Computer Science & Engineering at Shri Shankaracharya College of Engineering & Technology, Bhilai. India**.** Having Wide experience in the field of teaching. Research areas are Computer Networks, Wireless Network, its Enhancements, and His research work has been published in many national and international journals.

