International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 7 Issue: 6 48 - 51

Performance and Limitation Review of Secure Hash Function Algorithm

Iti Malviya
M.Tech scholar
Department of Electronics and Communication
SISTec, Bhopal (M.P.)

Prof. Tejasvini Chetty
Assistant Professor
Department of Electronics and Communication
SISTec, Bhopal (M.P.)

***Abstract*—**A cryptographic hash work is a phenomenal class of hash work that has certain properties which make it fitting for use in cryptography. It is a numerical figuring that maps information of emotional size to a bit string of a settled size (a hash) and is expected to be a confined limit, that is, a limit which is infeasible to adjust. Hash Functions are significant instrument in information security over the web. The hash functions that are utilized in different security related applications are called cryptographic hash functions. This property is additionally valuable in numerous different applications, for example, production of digital signature and arbitrary number age and so on. The vast majority of the hash functions depend on Merkle-Damgard development, for example, MD-2, MD-4, MD-5, SHA-1, SHA-2, SHA-3 and so on, which are not hundred percent safe from assaults. The paper talks about a portion of the secure hash function, that are conceivable on this development, and accordingly on these hash functions additionally face same attacks.

Keywords- *Secure, function, MD-2, MD-4, MD-5, SHA-1, SHA-2, SHA-3.*

*****

I. INTRODUCTION

True A cryptographic hash function is an extraordinary class of hash function that has certain properties which make it reasonable for use in cryptography. It is a scientific algorithm that maps information of self-assertive size to a bit string of a fixed size (a hash) and is intended to be a single direction function, that is, a function which is infeasible to reverse. The best way to reproduce the information from a perfect cryptographic hash function's yield is to endeavor an animal power search of potential contributions to check whether they produce a match, or utilize a rainbow table of coordinated hashes.

The perfect cryptographic hash function has five primary properties:

•It is deterministic so a similar message dependably results in a similar hash

•It rushes to figure the hash an incentive for some random message

•It is infeasible to create a message from its hash an incentive aside from by attempting every single imaginable message

•A little change to a message should change the hash esteem so broadly that the new hash worth seems uncorrelated with the old hash esteem.

•It is infeasible to discover two distinct messages with similar hash esteem.

Cryptographic hash functions have numerous information-security applications, remarkably in digital signatures, message confirmation codes (Macintoshes), and different types of validation. They can likewise be utilized as common hash functions, to list information in hash tables, for fingerprinting, to distinguish copy information or particularly recognize documents, and as checksums to identify unintentional information defilement. For sure, in information-security settings, cryptographic hash esteems are in some cases called (digital) fingerprints, checksums, or simply hash esteems, despite the fact that every one of these terms represent progressively broad functions with rather various properties and purposes.

SHA-3 (Secure Hash Algorithm 3) is the most recent individual from the Secure Hash Algorithm group of models, discharged by NIST on August 5, 2015. Albeit part of a similar arrangement of benchmarks, SHA-3 is inside not the same as the MD5-like structure of SHA-1 and SHA-2. SHA-3 is a subset of the more extensive cryptographic crude family Keccak structured by Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche, expanding upon RadioGatún. Keccak's creators have proposed extra uses for the function, not (yet) institutionalized by NIST, including a stream figure, a confirmed encryption framework, a "tree" hashing plan for quicker hashing on certain architectures,and AEAD figures Keyak and Ketje. NIST does not at present intend to pull back SHA-2 or expel it from the amended Secure Hash Standard. The motivation behind SHA-3 is that it very well may be directly substituted for SHA-2 in current applications if important, and to altogether improve the strength of NIST's general hash algorithm toolbox. SHA-3 utilizes the wipe development, wherein information is "assimilated" into the wipe, at that point the outcome is "squeezed" out.

II. LITERATURE SURVEY

D. K. N et al., [1] This work focuses on the Plan of Parameterizable Execution of SHA-256 algorithm in FPGA giving Blockchain Ideas. SHA-256 is the key guideline used in Blockchain design to grant security and protection into a framework. The proposed technique empowers any piece length information message to get changed over to fixed length message overview known as Hash. The plan for the proposed engineering was reproduced in Modelsim and incorporated in Xilinx Vivado Structure Suite utilizing Artix 7 FPGA.

J. Haj-Yahya et al., [2] Verifying a huge number of associated, asset obliged processing gadgets is a noteworthy test these days. Adding to the test, outsider specialist co-ops need standard access to the framework. To guarantee the uprightness of the framework and validness of the product merchant,

secure boot is upheld by a few business processors. In this composition, we propose a lightweight equipment based secure boot engineering. The engineering utilizes effective execution of Elliptic Curve Digital Signature Algorithm (ECDSA), Secure Hash Algorithm 3 (SHA3) hashing algorithm and Direct Memory Access (DMA).

A. Sengupta et al.,[3] Digital signal processing (DSP) part based licensed innovation (IP) center structures a fundamental element of shopper hardware gadgets. In this way, security of these IP centers against figuring out assault is significant. Functional confusion fills in as an incredible instrument to counter this equipment danger. The proposed approach utilizing lightweight secure hashing algorithm (SHA-512)-based key encryption custom equipment reconfigures the key-bits (coming about into auxiliary reconfiguration) of the securing rationale a functionally muddled DSP configuration enlarged with the total rationale amalgamation of the structure.

H. Liu et al., [4] Secure hash function assumes a significant job in cryptography. This work builds a hash algorithm utilizing the hyperchaotic Lorenz framework, which fills in as a wipe function to retain information message through various parameters time-fluctuating bother. Initially, the information message is isolated into four 1D clusters, to produce four bother groupings through parameter refreshing standard, the annoyed parameters are still inside their critical interims, to cause the framework to remain a hyperchaotic state. The trial assessment and examination exhibited the hash function's protection from differential assault and second pre-picture assault. The proposed hash function can be connected in the recognizable proof, information trustworthiness, and figure signature.

C. Biswas et al.,[5] To secure information or information has turned into a test in this focused world. There are numerous strategies for verifying information/information, for example, cryptography, steganography and so forth. In this work hybrid cryptography has been connected utilizing AES and RSA. In this hybrid cryptography, the symmetric key utilized for message encryption is likewise encoded, which guarantees a superior security. An extra element of this work is to make a digital signature by scrambling the hash estimation of message. At the accepting side this digital signature is utilized for respectability checking. Here hybrid cryptography gives a superior security, steganography reinforces the security. Message uprightness checking is an extraordinary element of this algorithm. Effective reenactments have been appeared to help the possibility of this algorithm.

III. TYPES OF CRYPTOGRAPHIC HASH ALGORITHMS

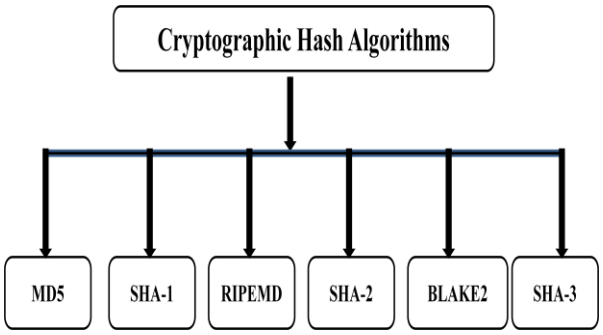

Figure 1: Types of crypto hash function

Hashing methods are categorized into two groups:

A. Data-oriented hashing versus security-oriented hashing

(i) Data-Oriented Hashing Data-oriented hashing refers to methods techniques that expect to utilize hashing to accelerate information recovery or examination, where a hash table is regularly kept up for an inquiry.

(ii)Security-Oriented Hashing Security-oriented hashing refers to methods that utilization hashing for confirmation or approval. For instance, a client may download programming from an open web server yet is stressed whether the product hosts been changed by a third gathering.

B. SHA-3

In SHA-3, the state S comprises of a 5 × 5 exhibit of w-bit words (with w=64), b = 5 × 5 × w = 5 × 5 × 64 = 1600 bits all out. Keccak is likewise characterized for littler intensity of-2 word sizes wdown to 1 bit (complete condition of 25 bits). Little state sizes can be utilized to test cryptanalytic assaults, and middle of the road state sizes (from w = 8, 200 bits, to w = 32, 800 bits) can be utilized in down to earth, lightweight applications.

For SHA-3-224, SHA-3-256, SHA-3-384, and SHA-3-512 occurrences, r is more noteworthy than d, so there is no requirement for extra square stages in the pressing stage; the main dbits of the state are the ideal hash. Be that as it may, SHAKE-128 and SHAKE-256 permit a discretionary yield length, which is valuable in applications, for example, ideal lopsided encryption cushioning.

C. MD5

MD5 was planned by Ronald Rivest in 1991 to supplant a previous hash function MD4, and was determined in 1992 as RFC 1321. Impacts against MD5 can be determined inside seconds which makes the algorithm unsatisfactory for most use situations where a cryptographic hash is required. MD5 produces a review of 128 bits (16 bytes).

D. SHA-1

SHA-1 was created as a major aspect of the U.S. Government's Capstone venture. The first determination - presently normally called SHA-0 - of the algorithm was distributed in 1993 under the title Secure Hash Standard, FIPS Bar 180, by U.S. government benchmarks organization NIST (National Foundation of Norms and Innovation). It was pulled back by the NSA not long after production and was supplanted by the changed form, distributed in 1995 in FIPS Bar 180-1 and usually assigned SHA-1. Impacts against the full SHA-1 algorithm can be delivered utilizing the shattered assault and the hash function ought to be viewed as broken. SHA-1 creates a hash condensation of 160 bits (20 bytes).

E. RIPEMD-160

RIPEMD (RACE Uprightness Natives Assessment Message Overview) is a group of cryptographic hash functions created in Leuven, Belgium, by Hans Dobbertin, Antoon Bosselaers and Bart Preneel at the COSIC research bunch at the Katholieke Universiteit Leuven, and first distributed in 1996. RIPEMD depended on the plan standards utilized in MD4, and is comparable in execution to the more famous SHA-1.

RIPEMD-160 has anyway not been broken. As the name suggests, RIPEMD-160 produces a hash summary of 160 bits (20 bytes).

F. SHA-2

SHA-2 (Secure Hash Algorithm 2) is a lot of cryptographic hash functions planned by the US National Security Office (NSA), first distributed in 2001.They are manufactured utilizing the Merkle–Damgård structure, from a single direction pressure function itself fabricated utilizing the Davies–Meyer structure from a (characterized) specific square figure. SHA-2 essentially comprises of two hash algorithms: SHA-256 and SHA-512. SHA-224 is a variation of SHA-256 with various beginning qualities and truncated yield. SHA-384 and the lesser known SHA-512/224 and SHA-512/256 are on the whole variations of SHA-512. SHA-512 is more secure than SHA-256 and is ordinarily quicker than SHA-256 on 64 bit machines, for example, AMD64. The yield estimate in bits is given by the expansion to the "SHA" name, so SHA-224 has a yield size of 224 bits (28 bytes), SHA-256 produces 32 bytes, SHA-384 produces 48 bytes lastly SHA-512 produces 64 bytes.

G. BLAKE2

An improved variant of BLAKE called BLAKE2 was declared in December 21, 2012. It was made by Jean-Philippe Aumasson, Samuel Neves, Zooko Wilcox-O'Hearn, and Christian Winnerlein with the objective to supplant generally utilized, yet broken MD5 and SHA-1 algorithms. At the point when kept running on 64-bit x64 and ARM models, BLAKE2b is quicker than SHA-3, SHA-2, SHA-1, and MD5. In spite of the fact that BLAKE nor BLAKE2 have not been institutionalized as SHA-3 it has been utilized in numerous conventions including the Argon2 secret key hash for the high effectiveness that it offers on current CPUs. As BLAKE was a contender for SHA-3, BLAKE and BLAKE2 both offer a similar yield sizes as SHA-3 - including a configurable yield measure.

H. HASH vs AES

SHA represents Secure Hash Algorithm while AES represents Propelled Encryption Standard. So SHA is a suite of hashing algorithms. AES then again is a figure which is utilized to scramble. SHA algorithms (SHA-1, SHA-256 etc...) will take an info and produce a summary (hash), this is regularly utilized in a digital marking process (produce a hash of certain bytes and sign with a private key). SHA is a hash function and AES is an encryption standard. Given an information you can utilize SHA to deliver a yield which is in all respects probably not going to be created from some other information.

SHA and AES fill various needs. SHA is utilized to produce a hash of information and AES is utilized to scramble information.

Here's a case of when a SHA hash is valuable to you. Let's assume you needed to download a DVD ISO picture of some Linux distro. This is a huge document and now and again things turn out badly - so you need to approve that what you downloaded is right. What you would do is go to a confided in source, (for example, the offical distro download point) and they ordinarily have the SHA hash for the ISO picture accessible. SHA has was utilized to approve information that was not ruined. AES, then again, is utilized to scramble information, or keep individuals from survey that information with knowing some mystery. AES utilizes a shared key which implies that a similar key (or a related key) is utilized to encoded the information as is utilized to unscramble the information. For instance in the event that I scrambled an email utilizing AES and I sent that email to you then you and I would both need to realize the shared key used to encode and decode the email.

.

Table 1: Algorithms and Limitations

Sr No	Hashing Algorithms	Limitations
1	SHA-1	This requires a lot of computing power and resources
2	SHA-2	Increased resistance to collision means SHA256 and SHA512 produce longer outputs (256b and 512b respectively) than SHA1 (160b). Those defending use of SHA2 cite this increased output size as reason behind attack resistance
3	SHA-3	SHA-3 is designed to be a good hash-function, not a good password-hashing-scheme (PHS).
4	MD5	Using salted md5 for passwords is a bad idea. Not because of MD5's cryptographic weaknesses, but because it's fast.

IV. CONCLUSION

In this survey talk about various secure cryptographic algorithm. Here included MD, SHA-1, SHA-2, SHA-3 and so on and discover SHA-3 is most recent planned algorithm which is increasingly reasonable and helpful for secure message in web applications. Various researchers have proposed their own algorithms anyway none of them are time gainful as SHA-3 and besides there are chances of upgrading the internal nature of these algorithms.

REFERENCE

[1]. D. K. N and R. Bhakthavatchalu, "Parameterizable FPGA Implementation of SHA-256 using Blockchain Concept," 2019 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2019, pp. 0370-0374.

[2]. J. Haj-Yahya, M. M. Wong, V. Pudi, S. Bhasin and A. Chattopadhyay, "Lightweight Secure-Boot Architecture for RISC-V System-on-Chip," *20th International Symposium on Quality Electronic Design (ISQED)*, Santa Clara, CA, USA, 2019, pp. 216-223.

[3]. A. Sengupta and M. Rathor, "Security of Functionally Obfuscated DSP Core Against Removal Attack Using SHA-512

Based Key Encryption Hardware," in *IEEE Access*, vol. 7, pp. 4598-4610, 2019.

[4]. H. Liu, A. Kadir and J. Liu, "Keyed Hash Function Using Hyper Chaotic System With Time-Varying Parameters Perturbation," in *IEEE Access*, vol. 7, pp. 37211-37219, 2019.

[5]. C. Biswas, U. D. Gupta and M. M. Haque, "An Efficient Algorithm for Confidentiality, Integrity and Authentication Using Hybrid Cryptography and Steganography," *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, Cox'sBazar, Bangladesh, 2019, pp. 1-5.

[6]. I. Alfiansyah and R. W. Wardhani, "Implementation of Secure Hash Algorithm – 3 for Biometric Fingerprint Access Control Based on Arduino Mega 2560," *2018 International Conference on Applied Information Technology and Innovation (ICAITI)*, Padang, Indonesia, 2018, pp. 31-35.

[7]. F. E. De Guzman, B. D. Gerardo and R. P. Medina, "Enhanced Secure Hash Algorithm-512 based on Quadratic Function," *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM)*, Baguio City, Philippines, 2018, pp. 1-6.

[8]. S. Yakut and A. B. Özer, "Secure and Efficient Hash Based Finishing Algorithm for Real Random Numbers," *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, Malatya, Turkey, 2018, pp. 1-5.

[9]. A. Alzahrani and F. Gebali, "Multi-Core Dataflow Design and Implementation of Secure Hash Algorithm-3," in *IEEE Access*, vol. 6, pp. 6092-6102, 2018.

[10]. J. Holmgren and A. Lombardi, "Cryptographic Hashing from Strong One-Way Functions (Or: One-Way Product Functions and Their Applications)," *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, Paris, 2018, pp. 850-858.

[11]. N. Mouha, M. S. Raunak, D. R. Kuhn and R. Kacker, "Finding Bugs in Cryptographic Hash Function Implementations," in *IEEE Transactions on Reliability*, vol. 67, no. 3, pp. 870-884, Sept. 2018.

[12]. D. Wang, Y. Jiang, H. Song, F. He, M. Gu and J. Sun, "Verification of Implementations of Cryptographic Hash Functions," in *IEEE Access*, vol. 5, pp. 7816-7825, 2017.

[13]. S. Chu, Y. Huang and W. Lin, "Authentication Protocol Design and Low-Cost Key Encryption Function Implementation for Wireless Sensor Networks," in *IEEE Systems Journal*, vol. 11, no. 4, pp. 2718-2725, Dec. 2017.

[14]. S. Koranne, "DÉJÀ VU: An Entropy Reduced Hash Function for VLSI Layout Databases," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 11, pp. 1798-1807, Nov. 2015.