Survey of Cryptographic Hashing Algorithms for Message Signing

[1]Ankit Kumar Jain, [2]Rohit Jones, [3]Puru Joshi
[1,2,3]Dept. of Computer Engineering, NMIMS Shirpur, Maharashtra, India

Abstract
In the world of computing, there is always a need for unique values to be generated for numerous applications. Various methods exist with the most common being reliant on an instance of time. But when we need to generate a unique value, based on an input factor such that the exact same value is generated, every time the input factor is the same, cryptographic hashing functions get the job done. Cryptographic hashing functions have many uses like digital signatures, message authentication, checksum generation, digital fingerprinting and securing passwords in a database. In this paper, we review and compare multiple hashing algorithms by different criteria with emphasis on message authentication.

Keywords
Cryptography, Hash, SHA, MD, Keccak, BLAKE, Message Digest, Merkle–Damgård

I. Introduction

Cryptography is often defined as the enciphering and deciphering of messages in secret code or cipher. Usually, cryptography is associated with security but it is just a facet of the overall security of a system. While, Cryptanalysis can be defined as study, development and execution of tools and techniques that are used to undermine or compromise the integrity, security or effectiveness of a cryptographic system. The term Cryptology is associated with the combination of both the above topics. that is, cryptography and cryptanalysis. The field of cryptography has many different applications, most of which can be deemed to be a separate field. Some important and commons uses of cryptography are as follows:

A. Encryption

It is defined as the process of converting some information from one form to another, with the intent of hiding the original message in such a way that, no-one other than the intended recipient of the message can understand the meaning of the encrypted message.

B. Decryption

It can be defined as the reverse process of encryption, that is, the conversion of an encrypted message into its original meaning.

C. Authentication

It can be defined as the process that is used to prove that something is real, authentic, true or genuine. Authentication is used extensively on the internet to grant access to users who rightfully can have access and deny those who don't.

D. Digital Signatures

Just like physically signing a document to prove its authenticity, digital signatures are used to achieve the same concept but for digital goods and services. Just like with physical signatures, it is very difficult to copy or forge digital signatures if not impossible.

E. Hashing

It is used to represent a digital file, message or any entity into a shorter, fixed length and unique string of characters in such a manner that the hash computed for the digital entity will always be the same and it is impossible to retrieve the original digital entity from its hash string.

Cryptographic hash functions map strings (messages) of almost arbitrary length to strings of a fixed, short length, typically somewhere between 128 and 512 bits [1]. Many different terms have been used for the output string.

Among these are the hash, the hash value, and the message digest. A hash function is expected to be very efficient. Different applications expect different properties of the hash function, but some properties are always expected [2].

1. A hash function H is always expected to be one-way. This means that given a randomly chosen image y, it is difficult (i.e., impossible in practice) to find a message x such that F(x) = y. Attacks that attempt to break this property of a hash are termed as pre-image attacks.
2. The hash of a message obtained from a hash function, should be equivalent to a digital fingerprint, such that two different messages also have different hash values. Attacks that attempt to break this property of a hash are termed as hash collision attacks.
3. Theoretically, it is possible for two different messages to generate the same hash, due to the nature of limited output space, but the hash function should compute the hash in such a manner that it is practically infeasible to find such messages [3].

A graphical representation of the above attacks is shown in Figure 1 to help understand the concepts better.

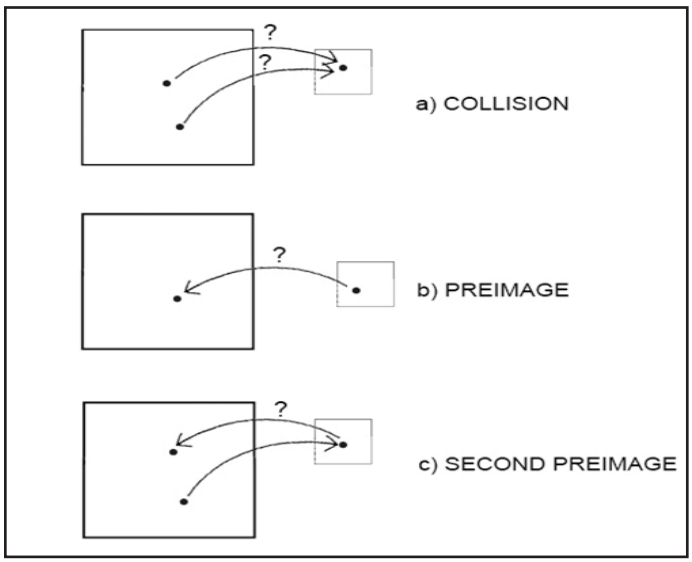

Fig. 1: Different Types of Attacks on Hashing Algorithms

Many popular cryptographic hashing algorithms like MD5, MD6, SHA1, SHA2 and BLAKE2 use the Merkle-Damgård [16] construction to achieve collision resistance.

The working of the Merkle-Damgård construction is as follows. The input to the structure is a multiple of a fixed number (example: 512 or 1024) which is achieved by padding the input message, as the construction does not allow an arbitrary sized input. The input message is then spit into multiple message blocks of a fixed size after which they are processed by the compression function in a sequential fashion such that the output of the previous message block's compression is combined with the input of the next message block's compression. As a security measure, the message's original length is hidden by padding the message. The first message block, does not have any previous message block whose compression are to be combined to use as an input, hence, an initialization vector is used in its place. The final stage of the construction, implements a finalization function that is used for many purposes such as compressing a big internal state into a smaller one, guarantying better mixing and obtaining a better avalanche effect.\

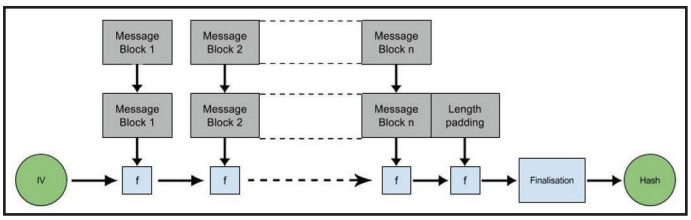

Fig. 2: Standard Merkle-Damgård Construction

II. Study of hashing algorithms

This section gives a brief description of different hashing algorithms and their working.

A. Message Digest 2 (MD2)

MD2 is a cryptographic hashing algorithm that was published in 1989 and is used to generate message digests of 128 bits by using a compression function of 18 rounds [10]. Post 2004, MD2 is known to be susceptible to preimage attacks of time complexity equivalent to 2104 applications of the compression function (Muller, 2004). Hence in the words of the MD2 author, "MD 2 can no longer be considered a secure one-way hash function". Post 2008, MD2 was shown to be more vulnerable and exploitable than it was initially thought out to be with successful preimage attacks being carried out with a time complexity of 273 compression function evaluations making it more feasible to exploit and more dangerous to use. MD2 is also proved to be susceptible to collision attacks, in 2009, of time complexity of 263.3 compression function evaluations.

B. Message Digest 4 (MD4)

MD4 is a cryptographic hashing algorithm that was published in 1990 and is used to generate message digests of 128 bits by using a compression function of 48 rounds and a word size of 32 bits [11]. As with many other hashing algorithms, it follows the small endian notation. In MD4, the message length in bits is extended until the total length is congruent 448 mod 512. For padding, MD4 adds a '1' bit at the end of the message and adds '0' bits until the padding conditions are met. The length before padding (64-bits) is finally appended. MD-4 was proved to be very ineffective hashing algorithm in 2007 when it was found that hash collisions can be found in under 2 hash operations in a collision attack that was subsequently published in the same year. It has also been proved that it is weak to pre-image attacks.

C. Message Digest 5 (MD5)

MD-5 is a cryptographic hashing algorithm that was developed by Ronald Rivest in 1991, to produces a 128 bits fixed length hash value [12]. MD5 is the successor of the flawed MD4 and was designed around the features and performance of 32-bit processors in mind but is in fact slower than MD4 but bears heavy resemblance to MD4. An emphasis on 32-bit processors is done because, the four word buffers (A, B, C, D) that are used to compute the message digest are, each, a 32-bit register. The main process of MD5 is very like MD4 and contains the same phases of appending padding bits followed but appending the length of the original message followed the initialization of the MD buffers followed by processing the message in 16-word blocks which consists of 64 operations, grouped in four rounds of 16 operations followed by the final output. In 2005 Xiaoyun Wang and Hongbo Yu demonstrated that it was possible to perform a modular differential attack and break the collision resistance of MD5. MD-5 is broken regarding collisions, but not in regard of pre-images or second-pre-images.

D. Secure Hash Algorithm 1 (SHA-1)

SHA-1 is a cryptographic hashing algorithm that was developed by the NIST [4] in 1993 to produce a 160-bit message digest. SHA-1 bears a remarkable similarity to the MD5 cryptographic hashing algorithm. At one point of time, it was the most preferred hashing algorithms for integrity checking due to its time efficiency and versatility. Post 2010, It is no longer actively used as it was proved to be venerable to hash collisions by Marc Stevens, with a complexity of 261 operations [5].

E. Secure Hash Algorithm (SHA-2)

SHA-2 is a cryptographic hashing algorithm that was developed by NSA. It has two variations namely SHA-256 and SHA-512 [13]. The primary difference between the two variants are the size of the words used. While SHA-256 uses 32-bit words, SHA-512 uses 64-bit words. Although, neither SHA-256 nor SHA-512 have been proved to be flawed, they are still not preferred for integrity verification as they are not as efficient as SHA-1 in terms of time complexities. Also, as SHA-2 is derived from SHA-1 which in turn is based on the MerkleDamgård structure, that was exploited to break the SHA-1 cryptographic hashing algorithm, thus, theoretically SHA-2 can also be broken.

F. Secure Hash Algorithm 3 (SHA-3)

SHA-3 is a cryptographic hashing algorithm that was chosen by the NSA in 2012 after a public competition among non-NSA designers [14]. The prior name of the SHA-3 hashing algorithm prior to the results of the competition was keccak. When keccak emerged as the winner of the SHA-3 competition, it was renamed to SHA-3. While SHA-3 supports the same hash lengths as SHA-2, the internal structure very different and is invulnerable to attacks like length extension which both the MD5 and SHA-1 were proved to be susceptible to. The main reason for the creation of the SHA-3 algorithm is due to the theoretical attacks that are possible against SHA-2. While there no practical proof has been submitted exposing the flaws of SHA-2, one cannot deny that it is indeed possible.

G. BLAKE2

BLAKE is a cryptographic hashing algorithm that was developed by

Jean-Philippe Aumasson, Luca Henzen, Willi Meier, and Raphael C.-W [15]. Phan as a participant to the SHA-3 competition. Dan Bernstein's ChaCha stream cipher with a few modifications like, but a permuted copy of the input block, XORed with some round constants, is added before each ChaCha round, make core of the BLAKE hashing algorithm. While BLAKE did not win SHA-3 competition, it is still a very capable algorithm that has been further improved in the form of BLAKE2. Some improvements of BLAKE2 over the original BLAKE include higher performance due to factors like reducing the number of rounds of compression from 16 to 12 for BLAKE2b and 14 to 10 for BLAKE2s and reducing the number of initialization words from 24 to 8. Due to the lower number of rounds, the random-access memory requirement of the BLAKE2 algorithm is significantly lower than the original BLAKE by as much as 33%. BLAKE2 implements tree hashing for incremental update or verification of large files. BLAKE2 implements minimal padding for messages and is overall, computationally faster and simpler than BLAKE to implement. Just like the original BLAKE algorithm, there exist two main variants of the BLAKE2 hashing algorithm based on different word sizes namely BLAKE2s the 32-bit variant that is used to obtain hashes that are 256 bits long and the 64-bit variant that is used to obtain hashes that are 512 bits long. BLAKE2s is optimized for small architectures while BALKE2b is optimized for 64-bit architectures. There also exist parallelized versions of both BLAKE2 algorithms called BLAKE2sp and BLAKE2bp that are up to 8 and 4 times faster, by using multiple cores and SMID. On 64-bit platforms, BLAKE2 is often significantly faster than MD5, yet provides security like that of SHA-3 like up to 256-bit collision resistance, immunity to length extension, indifferentiability from a random oracle, etc.

III. Comparative Analysis

After The section provides a comparison between the different hashing algorithms. Some of the algorithms under scrutiny are evinced to be weak and breakable. Subsequent improvements have been made to the newer ones to ensure higher security. Most of the common aspects of a hashing algorithm are put into focus to help understand the advantages and disadvantages of a hashing algorithm in comparison to another hashing algorithm.

Table 1: Comparison of Multiple Hashing Algorithms

Hashing Algorithm	Properties of Algorithm		
	DigestLength (in bits)	Number of Rounds	Collision Status
MD2	128	18	YES
MD4	128	3	YES
MD5	128	60	YES
MD6	<=512	Max(80,40+[d/4])	NO
SHA-1	160	80	YES
SHA-2	256/512	60/80	THEORITICAL
SHA-3	256/512	24	NO
BLAKE-2	256/512	10/12	NO

From Table 1, we see that both SHA-3 and BLAKE 2 have not been proved to be susceptible to hash collision and are thus contenders for ideal hashing algorithm for message singing. Further, SHA-3 and BLAKE2 have no known security issues. SHA-1, MD5, SHA-256, and SHA-512 are susceptible to length-extension. SHA-1 and MD5 are vulnerable to collisions. MD5 is vulnerable to collisions. SHA-2, SHA-3, MD6 and BLAKE2 have some of the highest digest lengths. A detailed comparison between the two most popular hashing algorithms in active use (MD-5 and SHA-1) and a strong contender for our hashing purposes, BLAKE2 is shown in Table 2 to help to further understand these hashing algorithms and their features.

Table 2: Feature Comparison of Hashing Algorithms

FEATURES	Hashing Algorithms		
	MD-5	SHA-1	BLAKE-2
Security	Less secure than SHA-1	More secure	Secure as SHA-3
Length of message digest	128 bits	160 bits	256 bits or 512 bits
No. of attacks needed to find original message	$2^{123.4}$ bit operations required [8]	$2^{151.1}$ bit operations required [9]	2^{256} or 2^{512} (Exhaustive search)
Attacks to try and find two message producing the same MD	$2^{49.8}$ bit operations required [6]	Between $2^{60.3}$ and $2^{65.3}$ bit operations [7]	2^{256} or 2^{256} (Exhaustive search)
Speed	Faster, 60 iterations	Slower, 80 iterations	Faster than SHA and MD
Successful attacks reported	YES	YES	NO

Although all cryptographic hashing algorithms strive to obtain and prefect the core principles of cryptographic the process they follow are very different from each other. Thus, while all cryptographic hashing algorithms are using to generate a hash, each algorithm generates a different hash, compared to each other, for the same input. For example, the message digests for the string "The quick brown fox jumps over the lazy dog" produced by the hashing algorithms are shown in Table 3.

Table 3. Hash of "The Quick Brown Fox Jumps Over the Lazy Dog"

HASHING ALGORITHM	OUTPUT DIGEST
SHA-1	2fd4e1c67a2d28fced849ee1bb76e7391b93eb12
SHA-2	d7a8fbb307d7809469ca9abcb0082e4f8d5651e46 d3cdb762d02d0bf37c9e592
SHA-3	0624a09a3d9e4261b140d8b71ffb559c6df6c3c98 46e033d34a645c4888498fc9100fa74549cf0744 e3e29dd539a80ad6f3e9c6b19e0f5a93de752 f05913c0ac
MD-2	03d85a0d629d2c442e987525319fc471
MD-4	1bee69a46ba811185c194762abaeae90
MD-5	9e107d9d372bb6826bd81d3542a419d6
MD-6	dcba0c6593fbd83a0f5f148588baa79530579c1f5 e7f19d500fe282d137bff465106f25c9f0619b4082 a730683d5f58311c0c1913068e91b0ebdf 9ace3ff5b9e
BLAKE-2	91b27f225ee86f26ef2103de210fd19e7e9f6cb3d 10f204a6ad359d90abbd5f06425dc9dc801a035d8 6d6dff977a69b5922a2d22a143ed8d63f026 bb875009ec

All algorithms are judged by how well they perform and hashing algorithms are no exception. Speed comparison of various popular hash functions, taken from eBACS's "hydra7" measurements is shown in the Table 4 and Fig. 2 [15].

Table 4: Performance Analysis of Hashing Algorithms

HASHING ALGORITHMS	PROCESSING IN MEBIBYTES PER SECOND
BLAKE2b	890
BLAKE2s	554
MD-5	550
SHA-1	571
SHA-256	169
SHA-512	266
SHA3-256	271
SHA3-512	144

From Fig. 2, we see that, on an Intel Sandy Bridge, BLAKE2b is 3.35 times as fast as SHA-512, BLAKE2b is 6.18 times as fast as SHA3-512, BLAKE2s is 3.28 times as fast as SHA-256, and BLAKE2s is 2.01 times as fast as SHA3-256. BLAKE2b reaches 4.76 cycles per byte, or approximately 890 mebibytes per second, against 266 for SHA-512 and 144 for SHA3-512, on a CPU clocked at 3.5GHz. Similarly, we see that BLAKE2s reaches approximately 5.2 cycles per byte, or approximately 554 mebibytes per second, against 196 for SHA-256 and 271 for SHA3-256.

IV. Conclusion

After extensive reviewing of multiple algorithms such as MD and SHA family, we have found that BLAKE2 does indeed live up to its claims of being faster and stronger than the competition and would hence be a very viable option to make use of its hashing capabilities for signing messages. BLAKE2 has a very flexible offering in the form of multiple implementations for different hardware and software use-cases such as BLAKE2 as BLAKE2b, BLAKE2s, BLAKE2bp and BLAKE2sp which makes it an ideal choice considering the versatility of hardware available in numerous forms in this modern age.

V. Future work

A new algorithm BLAKE2x is currently under development and is aimed to be an improvement to the existing BLAKE2 algorithm. In its current state, it is not stable enough to be considered for our purposes. In the future, when the algorithm specification is finalized, it could be used as a potential substitute to the BLAKE2 algorithm with the expected performance advantages of higher computation at pre-cycle. The development team behind BLAKE2x algorithm are open towards any improvement that the public are willing to suggest.

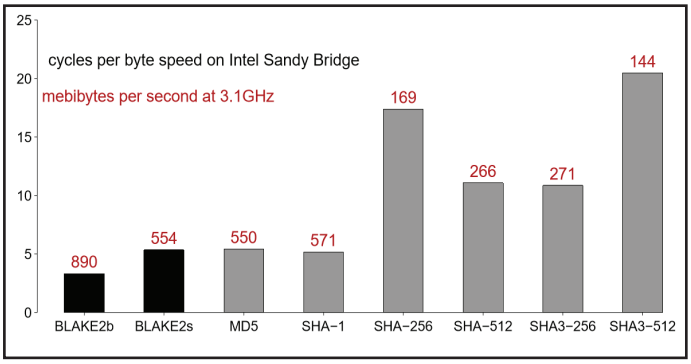

Fig. 3: Performance Analysis of Hashing Algorithms

VI. Acknowledgement

We would like to acknowledge the support of the "Computer Engineering" department of Mukesh Patel School of Technology Management and Engineering for their guidance and support they have provided especially Dr. Nitin SurajkishorChoubey, Head of Department, Computer Engineering, Mukesh Patel School of Technology Management.

We also grateful to Mr. Ojas Shirekar and Mr. Arpit Agrawal without whose insight and support this survey would not have been possible.

References

[1] Preneel, Bart.,"Cryptographic hash functions", European Transactions on Telecommunications 5.4, pp. 431-448, 1994.

[2] Wang, Xiaoyun, et al.,"Cryptanalysis of the Hash Functions MD4 and RIPEMD", Annual International Conference on the Theory and Applications of Cryptographic Techniques", Springer Berlin Heidelberg, 2005.

[3] Wang, Xiaoyun, et al.,"Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD", IACR Cryptology ePrint Archive 2004 (2004): 199.

[4] Eastlake 3rd, D., Paul Jones,"US secure hash algorithm 1 (SHA1)", No. RFC 3174.

[5] Stevens, Marc.,"New collision attacks on SHA-1 based on optimal joint local-collision analysis", Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer Berlin Heidelberg, 2013

[6] Stevens, Marc.,"Fast Collision Attack on MD5", IACR Cryptology ePrint Archive 2006: 104.

[7] Stevens, Marc Martinus Jacobus,"Attacks on hash functions and applications", Mathematical Institute, Faculty of Science, Leiden University, 2012.

[8] Sasaki, Yu, Kazumaro Aoki,"Finding preimages in full MD5 faster than exhaustive search", Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer Berlin Heidelberg, 2009.

[9] Knellwolf, Simon, Dmitry Khovratovich,"New preimage attacks against reduced SHA-1," Advances in Cryptology–Crypto 2012. Springer Berlin Heidelberg, 2012. pp. 367-383.

[10] Kaliski, Burton,"The MD2 message-digest algorithm", No. RFC 1319. 1992.

[11] Rivest, Ronald,"The MD4 message-digest algorithm", 1992.

[12] Rivest, Ronald,"The MD5 message-digest algorithm", 1992.

[13] NIST. Secure Hash Standard, FIPS PUB 180-2, 2002.

[14] Paar, Christof, Jan Pelzl.,"SHA-3 and The Hash Function Keccak", 2010.

[15] Aumasson, Jean-Philippe, et al.,"BLAKE2: Simpler, smaller, fast as MD5", International Conference on Applied Cryptography and Network Security. Springer Berlin Heidelberg, 2013.

[16] Merkle, Ralph Charles, Ralph Charles,"Secrecy, authentication and public key systems", 1979.

ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)

Mr. Ankit Jain has Completed his M.Tech from NIT Rourkela and B.E. from RGPVBhopal. He has more than 5 years of experience in teaching, and iscurrently working as Assistant Professor in Department of Computer Engineering, NMIMS Shirpur. His area of research is Network Security.

Mr. Rohit Jones is a 4th year student pursuing his Bachelors of Technology degree in the field of Computer Engineering from the institute Mukesh Patel School of Technology ManagementEngineering, NMIMS. At present his research interest lies in artificial intelligence and machine learning and their applications into the field of computer security.

Mr. Puru Joshi is a 4th year student pursuing his Bachelors of technology degree in the field of Computer Engineering from the institute Mukesh Patel School of Technology Management Engineering, NMIMS. At present his research interest lies in image processing, machine learning and artificial engineering.