ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

*International Journal of Advanced Research in Computer and Communication Engineering*
*Vol. 4, Issue 9, September 2015*

# A Critical Review of Cryptographic Hash Functions

**Smriti Gupta[1], Prof. Sandeep Kumar Yadav[2]**

M. Tech. Student, Dept. of Digital Communication, Govt. Mahila Engineering College, Ajmer, Rajasthan[1]

Asst. Prof, Dept. of Digital Communication, Govt. Mahila Engineering College, Ajmer, Rajasthan[2]

**Abstract:** The cryptographic hash function literature has numerous hash function definitions and hash function requirements, and many of them disagree. This survey talks about the various definitions, and takes steps towards cleaning up the literature by explaining how the field has evolved and accurately depicting the research aims people have today.

**Keywords:** Message Digest (MD'S), Secure Hash Algorithm (SHA), Pseudo Random Function Families (PRFs), Message Authentication Codes (MACs).

## I. INTRODUCTION

The literature on cryptographic hash functions abounds with numerous different definitions and requirements. There is no universal agreement on what a cryptographic hash function is, or what it is supposed to achieve. As an illustration of the problems, we note that most researchers agree that a cryptographic hash function should compress data; in particular, it should map very large domains to fixed size outputs. A hash function is simply a mapping

$$h :\{0, 1\}*\rightarrow \{0, 1\}m$$

Every good hash function has the property that two different inputs are very unlikely to be mapped to the same value. Hash function families were introduced by Damgard [1] in order to make the security requirements very precise in the complexity theory model. He specifically looked at collision resistant hash function families, where one can aim for an algorithm such that no polynomial bounded (in time and size) circuit can find collisions. Nowadays, some people define collision resistant hash functions to be collision resistant [2] while others define them to be both collision resistant and preimage resistant [3]. Despite the fact that Damgard's definition [1] is usually cited, the current definitions do not seem to make a distinction between ordinary hash functions and hash function families. Apparently these definitions are supposed to cover both. Within the literature of ordinary hash functions, we typically find security requirements such as the following:

– Preimage resistance: Given y, it must be computationally infeasible to find x such that h(x) = y.
– Second preimage resistance: Given y and x1 such that h(x1) = y, it must be computationally infeasible to find x2 6= x1 such that h(x2) = y.
– Collision resistance: It must be computationally infeasible to find any x1 andx2 such that h(x1) = h(x2).

In NIST's recent draft call for a new hash standard, one of the security evaluation criterions is
The main body of this paper elaborates on these and other problems with the cryptographic hash functions literature.

The main body of this paper elaborates on these and other problems with the cryptographic hash functions literature. Although we do not have all the answers, we hope that it is a step forward in cleaning up the literature. However, we would like to emphasize that our criticisms of certain definitions and terminology should not be interpreted as a criticism of the people that said them. Cryptographic Hash Functions has been an evolving subject since its origin. Many people had good ideas as a step forward, but the ideas have now become obsolete as further research has been developed.

One of the goals of this paper is to point out idea which are obsolete so that new research can focus with less ambiguity on the current understanding of the way hash functions are intended to be used before we begin, we remark that [4] deals with related issues of hash function terminology. In fact, we will reference this paper many times, since our goals are overlapping with theirs. However, our focus is broader and less technical and is more of a down-to-earth survey rather than original research.

## II. VARIOUS ASPECTSOF HASH FUNCTIONS

Cryptographic hash functions that fulfil certain security properties may be used in cryptographic applications such as digital signatures and pseudo-random number generators. Cryptographic hash functions must not only have good statistical properties. They must also withstand serious attack by malicious and powerful attackers who are trying to invade our privacy. The design of such cryptographic hash functions is an important but extremely difficult task. Many have been proposed, but most of them soon turned out to be too weak to resist attacks. Only two families of hash functions came to be widely used (namely the MD and SHA families, the most well-known members of which are MD5 and SHA-1, respectively). Unfortunately, their security relies on heuristic arguments rather than mathematical proofs. As might be expected, weaknesses have recently been found in both of them and

DOI 10.17148/IJARCCE.2015.49103

as a result, there currently exist no secure and practical cryptographic hash functions. Hence there is little basis for trust in the applications that use them, and a great need for research into good cryptographic hash functions.

These recent developments in cryptanalysis have clearly shown that currently used cryptographic hash functions are not good enough. But research in this area is not only very important because most existing hashes have been broken. The problem is not so much that flaws have been found in current designs, but that their construction often seems ad-hoc and their security cannot be proven. Information security is too important to be left up to assumptions and luck. What we really need are hash functions the security of which can be trusted.

## III. HISTORY AND IMPROVEMENT OF HASH FUNSTIONS

The most popular Hash function of 1980's decade was MD-2 (message digest-2) developed by Ronald Rivest in 1989[5]. The life of this hash function was not enough long just because the speed of the electronics systems was developing rapidly according to Moore's law.

Later in 1990 another digest algorithm was developed by Ronald Rivest himself, called MD-4 [6] which was enough complex to be attacked with that time technology. However Weaknesses in MD4 were demonstrated by Den Boer and Bosselaers in a paper published in 1991 [7]. The first full-round MD4 collision attack was found by Hans Dobbertin in 1995, which took only seconds to carry out at that time.

The next improvement in digest algorithms was launched in 1991 by Ronald Rivest called MD-5 [8]. This algorithm was slightly complex than their predecessors with greater immunity to collision and preimage attacks.

Now-a-days the world of communication is going better, bigger and faster. So, we have to switch to a better and complex signature algorithm like Secure Hash Functions. At the earlier days the SHAs were up to 256 bit long unlike the MDs (SHA supports variable length of output key string), but now they are up to 512 bit long with higher complexity and more immune towards attack.

SHA family was developed by NSA (National Security Agency). The first related paper came into the market in 1993 which was withdrawn in 1995. Later SHA-1 came in light in 1995 followed by other hash algorithms up to 2007.

Various analyses are going on about security and complexity of SHA-512 launched on Jan 1, 2007.

## IV. ADDITIONAL REQUIREMENTSOF HASH FUNCTIONS

Nowadays, hash functions are used in many different ways in practice. One of the most common ways is with Message Authentication Codes (MACs) which are a means that two users with a shared secret key can authenticate between each other. The widely used HMAC standard comes with a security proof [9]– it is secure provided that the underlying keyed compression function of the hash is a PRF. Hash functions are also widely used for pseudo random number generation. In one instance, security can be proved provided that the hash function behaves as a PRF where the secret involves adding a key to the message space [10].One of the reasons hash functions have taken on such a diverse role is because they have not been subject to export regulations, contrary to other cryptographic primitives. Additionally, they offered speed advantages and could be used without a license (unlike the IDEA block cipher). Today, export regulations are less strict and there are plenty of efficient alternative public domain cryptographic primitives available, so the mentioned benefits have disappeared. If we are going to continue to apply hashing to the various scenarios that we are using today, then we require a single function that satisfies all of these security requirements. An alternative is to cut back in the way we are using these functions.

## V. CRITICAL REVIEW

We have so far eluded the question of how these hash functions should be defined. Given that so many people have cluttered the literature with different definitions, it is against our judgment to offer new definitions that are attempts at overriding others. However, we do recommend rejecting certain definitions. Generally, we would like to avoid definitions that do not accurately reflect the security properties that they are intended to have. Examples include Merkle's weak and strong one-way hash functions, Winternitz's version of a one-way hash function, and Preneel's version of a collision-resistant hash function (since the name does not specify preimage resistance also). We would also like to avoid definitions that are not main stream, such as definitions that do not involve compression. Many hash function designers today simply call their design a "hash function "without adjectives such as "one-way" or "collision resistant." An interpretation of this vernacular is a compressing and easy to compute function that has additional security properties. We do not oppose such a definition (although it is informal), but we do strongly recommend that researchers say exactly what those additional properties are for their design. In particular, if researchers are proposing a single solution to be used everywhere like the way we are usingSHA-1 today, then they should include PRF and random oracle emulation in their stated security goals, and at least include a reference to how to interpret such definitions formally. Moreover, when researchers develop a hash function family; do not omit the word "family."

We recommend that standards bodies involved in selecting new standards for cryptographic hash functions, in collaboration with the cryptographic community, should aim to specify a set of well-defined security requirements for cryptographic hash functions. This set of security requirements would then allow an objective assessment of the security of candidate functions submitted for standardization. Such requirements are defined by specifying an interactive computational game between an adversary and a challenger, and defining a condition on the outcome of the game which defines success of the adversary in 'winning' the game, and a quantity called the

DOI 10.17148/IJARCCE.2015.49103

advantage of the adversary (which is determined by its success probability). The security requirement can then be quantified by the maximal advantage of the adversary in the game given bounds on its computational resources (run-time/program, size, number of oracle queries, etc). In the standard complexity-theory model, the maximal adversary advantage is taken over all adversaries with the given resource bound.

The security requirements needed from cryptographic hash functions are ultimately determined by their applications. Therefore, as a step towards the above stated goal, we present below a list of the main current practical applications of cryptographic hash functions. For each such application, we cite known well-defined security requirements on the underlying hash function which guarantee the security of the application (if such requirements are known). We also list other requirements from the hash function (whether a function family is needed, whether the family key is secret or public, the function input/output domain) and the relevant references to the literature where the security requirement was defined and shown sufficient for the application.

## VI. CONCLUSION

The field of cryptographic hash functions has been evolving since its origin approximately 30 years ago, and will continue to do so for quite some time. The informality of hash function terminology has resulted in cluttered literature, lacking a clean list of goals summarizing our security requirements. Consequently, there is no objective way of evaluating the security of new hash function proposals, except designs that are very obviously broken. This survey has emphasized the importance of formal terminology and a clear set of objectives. The hope is that researchers will take our view into consideration as a first step of trying to clean up the cryptographic hash in literature.

## REFERENCES

1. B. Damgard. Collision free hash functions and public key signature schemes. In Advances in Cryptology – EUROCRYPT'87, volume 304 of LNCS, pages 203–216.Springer, 1987.
2. A.J. Menezes, P. C. van Oorschot, and S. A. Vanstone. Handbook of Applied Cryptography. CRC Press, 1996.
3. B. Preneel. Analysis and design of cryptographic hash functions. PhD thesis, Katholieke Universities Leuven, 1993.
4. P. Rogaway. Formalizing human ignorance. In Progress in Cryptology - VI-ETCRYPT'06, volume 4341, pages 211–228. Springer, 2006.
5. http://en.wikipedia.org/wiki/MD2_(cryptography).
6. http://en.wikipedia.org/wiki/MD4.
7. Boer B., Bosselaers A., "An Attack on the Last Two Rounds of MD4," ACM digital library, 1991.
8. http://en.wikipedia.org/wiki/MD5.
9. M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. Lecture Notes in Computer Science, 1109, 1996.
10. Desai, A. Hevia, and Y. Yin. A practice-oriented treatment of pseudo random number generators. In EUROCRYPT 2002, pages 368–383. Springer, 2002.

DOI 10.17148/IJARCCE.2015.49103