

# Analysis of NIST Cryptographic Hash Function Competition Candidates

Master's Thesis Proposal

Joel Lathrop  
Department of Computer Science  
Rochester Institute of Technology  
Rochester, NY 14623 USA  
jal6806@cs.rit.edu

Chair: Stanisław Radziszowski spr@cs.rit.edu  
Reader: Christopher Homan cmh@cs.rit.edu  
Observer: Edith Hemaspaandra eh@cs.rit.edu

September 2008

# Contents

<b>1</b>	<b>Summary</b>	<b>3</b>
<b>2</b>	<b>Background</b>	<b>3</b>
2.1	Hash Functions . . . . .	3
2.2	Hash function security problems . . . . .	4
2.3	Merkle-Damgård chaining scheme . . . . .	4
2.4	Design of MD5 . . . . .	5
2.5	Design of SHA-1 . . . . .	6
2.6	Further Details . . . . .	6
<b>3</b>	<b>Problem</b>	<b>6</b>
3.1	The breaking of MD5 . . . . .	6
3.2	Attacks against SHA-1 . . . . .	6
3.3	Implications of these attacks . . . . .	7
<b>4</b>	<b>Current research toward a solution</b>	<b>7</b>
4.1	NIST cryptographic hash function competition . . . . .	7
<b>5</b>	<b>Proposed thesis work</b>	<b>8</b>
5.1	Goal and plan to obtain it . . . . .	8
5.1.1	Preliminary research of analysis methods . . . . .	8
5.1.2	Selection of promising candidates and security analysis . . . . .	9
5.1.3	Benchmarking of selected candidates . . . . .	9
5.2	Deliverables . . . . .	9
5.3	Timeline (tentative) . . . . .	10
<b>6</b>	<b>Conclusion</b>	<b>10</b>

# 1 Summary

Cryptographic hash functions are a vital part of our current computer systems. They are a core component of digital signatures, message authentication codes, file checksums, and many other protocols and security schemes. Because of this, their cryptographic strength is paramount to the continuance of computer security as we know it.

Unfortunately, recently that strength has been weakened. In 2005 Wang, et. al. released details on how to break the widely used MD5 hash function by finding colliding messages [14]. This was followed by an attack against the SHA-1 hash function, which while not producing a practical break did significantly reduce the complexity of finding one [13]. With both MD5 and SHA-1 – the two most widely used hash functions – weakened by successful attacks, it became vital to provide more secure hash functions in order to protect the continuation of effective cryptography in computer security.

To this end, the National Institute of Standards and Technology announced a cryptographic hash algorithm competition to produce a new hash family which they would standardize as SHA-3 [7]. The purpose of this competition would be to provide a hash function which had undergone extensive public review to verify its strength and effectiveness. The competition was announced in November 2, 2007 [8] and is scheduled to choose a final winner in 2012 [6].

The selection of a winner will involve extensive public review of the candidates. While NIST will be forming an internal committee to review the hash functions, detailed cryptanalysis and comments from the public will be a vital part of their review. To this end, we propose a Master’s thesis which details cryptanalysis of several of the promising hash function candidates. This analysis will assist the cryptographic community in determining the value of these hash functions as well as providing a stepping stone for further analysis should any of these functions enter the next rounds.

## 2 Background

### 2.1 Hash Functions

A hash function<sup>1</sup> is a cryptographic primitive that compresses an arbitrary length input into a fixed length output called a message digest. It does this in such a way that the output is effectively unique<sup>2</sup> with regard to the input, and the process cannot be reversed to yield the input from the output.

In more specific mathematical terms, a hash function can be defined as:

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

---

<sup>1</sup>For the purposes of this proposal, when we say “hash function” we will be referring to an unkeyed, one-way hash function. Discussion of other less common forms of hash functions is beyond the scope of this proposal.

<sup>2</sup>By “effectively unique” we mean that while other inputs which would produce the same output may exist, the likelihood of finding such an input is so negligible that it is not pragmatically worth considering.

where  $m$  is the fixed length of the hash function  $h$  in bits.

The output of  $h$  is to be effectively unique. This means that an effective computation that produces an  $x$  and  $x'$  such that  $x \neq x'$  and  $h(x) = h(x')$  must take essentially about  $2^{m/2}$  hash operations, which is the number of hash operations necessary to achieve a better than even chance of finding  $x$  and  $x'$  by random search alone<sup>3</sup> [9].

The hash function  $h$  must also be irreversible (also known as being “one-way”). This means that given a message digest  $y$  such that  $y = h(x)$ , computing  $x$  from  $y$  should not be possible.

## 2.2 Hash function security problems

In order to maintain these properties and be considered cryptographically secure, a hash function must not be susceptible to the following attacks:

**Preimage** Given a hash function  $h$  and a hash value  $y$ , find message value  $x$  such that  $h(x) = y$ .

**Second-Preimage** Given a hash function  $h$  and a message value  $x$ , find another message value  $x'$  such that  $x \neq x'$  and  $h(x) = h(x')$ .

**Collision** Given a hash function  $h$ , find two message values  $x$  and  $x'$  such that  $x \neq x'$  but  $h(x) = h(x')$ .

**Length Extension** Given a hash function  $h$ , a hash value  $h(x)$ , and the message length  $|x|$ , find an  $x'$  such that  $h(x||x')$  can be calculated, where  $||$  represents concatenation.

## 2.3 Merkle-Damgård chaining scheme

Many hash functions – including MD5 and SHA-1 – are constructed using the Merkle-Damgård chaining scheme. In this scheme, a hash function  $h$  is built up from a compression function  $f$  such that if  $f$  is collision resistant, then  $h$  is as well [11].

This is done as follows. Given a compression function  $f : \{0, 1\}^m \times \{0, 1\}^t \rightarrow \{0, 1\}^m$  and an input  $x$  of length  $n$ , split  $x$  into  $t$  size blocks such that  $x = x_1||x_2||x_3||\dots||x_k$ . If the last block  $x_k$  is less than  $t$  bits in length, right pad it with 0's. Then append a final block  $x_{k+1}$  which contains the right-justified binary representation of  $n$ . The compression function  $f$  is then used to chain the input blocks together by computing  $H_i = f(H_{i-1}, x_i)$ ,  $1 \leq i \leq k+1$ . The initial value  $H_0$  is generally a special constant which is part of the hash function's definition. The hash function result for input  $x$  can now be iteratively defined

---

<sup>3</sup>This known as a “birthday attack”. For more information on the math that results in these probabilities, see [http://en.wikipedia.org/wiki/Birthday\\_attack](http://en.wikipedia.org/wiki/Birthday_attack) or section 7.4 of [9].

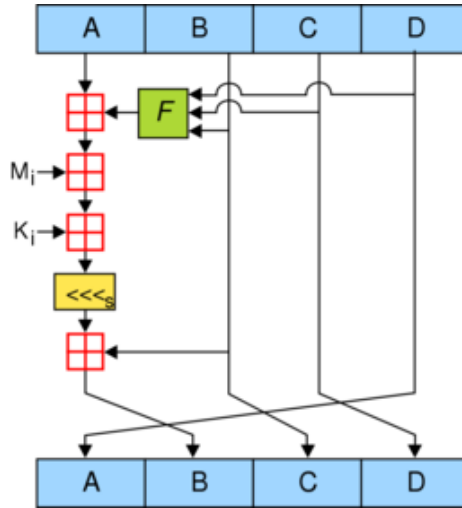


Figure 1: MD5 compression round function (Wikipedia [1])

as  $h(x) = H_{k+1} = f(H_k, x_{k+1})$ . In this way, the hash function  $h$  has been built from the compression function  $f$  via the Merkle-Damgård chaining scheme [5].

## 2.4 Design of MD5

MD5 is a Merkle-Damgård hash function with a 128-bit message digest. Its compression function takes a 128-bit intermediate hash value and a 512-bit message block and processes each block using 64 rounds. A graphical view of the compression round function is given in Figure 1.

In the figure,  $M_i$  represents a 32-bit sub-block of the current 512-bit message block. (Precisely which sub-block is a function of the current round.)  $K_i$  is a constant which varies by round. The  $\boxplus$  symbols signify addition modulo  $2^{32}$ . The  $\lll_s$  denotes a left shift by  $s$  bits, where  $s$  also varies based on the current round.

Before the round function is applied, the compression function derives the round function's inputs by splitting  $H_i$  into four 32-bit pieces such that  $A||B||C||D = H_i$ . After all 64 rounds are computed, the resulting values are recombined as  $H_{i+1} = A||B||C||D$  to produce the result of the compression function.

Iterative applications of the MD5 compression function produce the MD5 hash function, as per the Merkle-Damgård chaining scheme.

For a complete definition of MD5, we recommend section 3 of [10] which has a complete yet very concise definition.

## 2.5 Design of SHA-1

SHA-1 is based off of SHA-0 which was effectively based off of MD5. The message digest size is extended to 160 bits, and there are some small changes to the round function and constants, but the majority of the hash remains similar.

See [9] or [11] for a complete definition of SHA-1.

## 2.6 Further Details

For more information on hash functions, we recommend chapter 4 of Douglas R. Stinson's *Cryptography: Theory and Practice*, third edition [11].

# 3 Problem

Unfortunately, the finite nature of a hash function's compression function means that these attacks are possible for any given hash function. It is simply a matter of how difficult it is to find them.

## 3.1 The breaking of MD5

In 2005, Wang, et. al. published an attack which produced collisions in full-round MD5 [14]. This was a serious blow to modern cryptography, because MD5 was one of the most widely used hash functions in other computer security algorithms and computer security software and hardware.

Once the break was published, a number of cryptographers picked up the method used by Wang, et. al. and worked at enhancing it. Wang's original attack took "about an hour" on an IBM P690 supercomputer [14]. This was reduced to the point where collisions could be generated in roughly eight hours on an ordinary notebook computer [4]. Finally, Stevens in his Master's thesis reduced the attack time to approximately 6 seconds on desktop hardware [10]. With it now possible to generate collisions in real-time on desktop hardware, MD5 is utterly unacceptable for use as a secure hash function.

## 3.2 Attacks against SHA-1

With MD5 sufficiently compromised, cryptographers turned their attention to SHA-1. Later in 2005, after releasing their paper on breaking MD5, Wang, et. al. released a paper detailing an attack against full-round SHA-1 which could find a collision in  $2^{69}$  hash operations, the first successful attack against the full SHA-1 which was less than the birthday attack complexity of  $2^{80}$  hash operations [13].

Shortly after the release of this paper, a presentation was given by Wang and others which showed new research that reduced the complexity of a collision attack against full-round SHA-1 to  $2^{63}$  hash operations. [12]

### 3.3 Implications of these attacks

With MD5 hopelessly broken and SHA-1 treacherously close to the upper limit of computational power that is economically feasible for a large entity, neither hash function could any longer be safely considered for use in a secure system. It was clear that they had to be replaced by a stronger hash function. Because MD5 and SHA-1 were found in the majority of security products that use cryptography, this replacement would be an extensive and expensive procedure. Therefore, it was vital that the replacement hash function be durable enough to withstand scrutiny for some years to come.

## 4 Current research toward a solution

While there are a number of more recently developed hash functions which could be used to replace MD5 and SHA-1, they each have their strengths and weaknesses. Since it is valuable for the sake of standardization to have a single hash function which is known to be secure and widely accepted, and to determine which of these hash functions is superior to the others, a hash function competition was called for.

### 4.1 NIST cryptographic hash function competition

In response to the attack against SHA-1, the National Institute of Standards and Technology held a conference to assess the remaining strength of its approved hash functions [6]. At this conference, it was concluded that while the approved hash functions were still secure for the moment, it was necessary to choose a new hash function which would provide a long-term replacement for SHA-1. To that end, NIST announced a competition to pick this hash function [8].

The formal announcement listed a set of submission criteria for candidate hash functions which included the following [8]:

1. The hash function should be publicly disclosed and free of royalties and intellectual property encumbrances.
2. The hash function should be implementable on a wide range of hardware and software platforms.
3. The hash function should be able to produce digest sizes of 224, 256, 384, and 512 bits with a maximum message length of  $2^{64} - 1$  bits.

Each accepted hash function will then be evaluated on the following criteria [8]:

**Security** It should be resistant to specific known attacks (e.g. differential cryptanalysis) and resistant to solutions for the security problems mentioned in section 2.2 of this paper. Additionally, it should have special security properties specific to certain applications such as HMACs and PRNGs.

**Cost** It should be computationally efficient and should consume a minimal amount of memory.

**Algorithm and Implementation Characteristics** It should be flexible and simple.

The submission deadline for hash functions is October 31, 2008. After this deadline, NIST will begin to review the hash functions for adherence to the submission requirements and publish them on their website. NIST will then hold the First SHA-3 Candidate Conference at which the hash function candidates which met the submission criteria will be formally announced and the public will have opportunities to question the hash function creators regarding their submissions [8].

## 5 Proposed thesis work

While a private NIST committee will be choosing the winning hash function, public review of the hash function candidates will factor heavily into their evaluation. In response to this need for public analysis of the hash function candidates, we propose a Master's thesis providing such analysis.

### 5.1 Goal and plan to obtain it

The specific goal of the thesis would be to provide detailed analysis of a few of the most promising looking candidates. By providing this analysis we will be giving useful input to the NIST evaluation committee as well as providing a building block for researchers to do further analysis on the hash functions we selected.

#### 5.1.1 Preliminary research of analysis methods

As the submission deadline is not until October 31, we would begin by doing more in-depth study on modern methods used to evaluate hash functions, with special focus given to gaining comprehension of the recent attacks used against MD5 and SHA-1.<sup>4</sup> In particular, time will be spent trying to gain a better understand of *how* the Wang attack was developed and the means by which the differential paths were created. Significant advancements in complexity reduction (such as Klíma's tunnels) will be studied in depth in an attempt to understand how they were formulated as well as determining how they might be applied to other hash functions.

---

<sup>4</sup>We have actually already been studying these attacks during the past year, and there is still more to be done in order for us to understand them. The Wang attacks were so complex that well known cryptography experts such as Bruce Schneier have stated that even after reading them they still don't understand them.

This research will be included in the thesis as a section detailing the major attacks against MD5 and SHA-1 which led up to the NIST competition. Also included will be older work in hash analysis which predates the Wang attack against MD5.

### 5.1.2 Selection of promising candidates and security analysis

Once NIST begins publishing submissions to their website, we will begin a general review of the candidates looking for those few which seem to offer the most promise, selecting them for further analysis. Of particular interest will be candidates which are modifications on an already successful hash function as well as candidates which do not use the Merkle-Damgård chaining scheme. (An example of a potential candidate which has both of these properties is Maelstrom-0 which uses the 3CM chaining scheme and is based off of Whirlpool [3].)

We will then begin in-depth analysis of each of these selected hash functions based on the evaluation criteria set by NIST. Special focus will be given to the cryptographic security of the hash function, with its strength against differential and linear cryptanalysis being examined. Additionally, we will attempt to use the recent attacks by Wang as well as the enhancements upon them as a basis for attacking the hash function. While we highly doubt that full-round attacks will be successful, reduced round attacks may be possible.

Ultimately, there is only so much we can plan prior to seeing the hash candidates themselves. Some possible candidates have such unusual designs (such as VSH [2] which claims collision resistance based on a compression function devised using number theory) that the ordinary patterns for attack may not prove useful and a new approach must be devised.

### 5.1.3 Benchmarking of selected candidates

We will also test the hash functions' computational efficiency, producing a benchmark of their optimized reference implementations. To accomplish this we will build a pluggable test harness which will test the hash functions plugged into it for various performance characteristics.

## 5.2 Deliverables

The deliverables of this research will be a written thesis detailing our results. It will contain a listing of the NIST candidates we selected, detailed analysis of their security against modern attacks, as well as a benchmark of their performance against each other and current widely used hash functions. The thesis will also contain a review of the major research in hash function cryptanalysis which led up to the NIST competition announcement as well as a brief explanation of the NIST competition history and process.

### 5.3 Timeline (tentative)

The writing of the thesis will be done in parallel with the research. Due to the fact that the NIST competition candidates are not yet published and exact dates for their publication are not known, this timeline should be considered extremely tentative and dependent on when information will be published by NIST or the candidate submitters.

Date	Task
September 15, 2008	Proposal review completed and submitted to the graduate program coordinator.
September 16, 2008	Begin review of hash evaluation methods and modern attacks.
Early November 2008	Begin selection and analysis of promising candidates.
May 15, 2009	Final defense.

Please note that while the defense date is at the end of the school year, the defense may be held much sooner than that. For various reasons, Mr. Lathrop will be in Rochester for the entire school year regardless of when he graduates. The late deadline is given so that if we finish the required work for the thesis but feel that additional research and improvements could be made that would be valuable to the cryptographic community we have the option to do so.

## 6 Conclusion

Cryptographic hash functions are vital to computer security, but the current widely used algorithms have either been broken or weakened. The NIST Advanced Hash Standard competition aims to correct this by providing a long-term replacement hash function. We have proposed a thesis aimed at providing the kind of analysis necessary for this competition to be successful, and in so doing aiding the process of increasing computer and network security through the development of stronger hash functions.

## References

- [1] MD5. Wikipedia, September 8, 2008. <http://en.wikipedia.org/wiki/MD5>.
- [2] Scott Contini, Arjen K. Lenstra, and Ron Steinfeld. VSH, an efficient and provable collision resistant hash function. In *Advances in Cryptology - EUROCRYPT 2006*, Lecture Notes in Computer Science, 2006.
- [3] Décio Luiz Gazzoni Filho, Paulo Barreto, and Vincent Rijmen. The Maelstrom-0 hash function. In *6th Brazilian Symposium on Information and Computer System Security*, 2006.

- [4] Vlastimil Klíma. Finding MD5 collisions on a notebook PC using multi-message modifications. Cryptology ePrint Archive, Report 2005/102, 2005. <http://eprint.iacr.org/2005/102>.
- [5] Alfred J. Menezes, Paul C. Van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.
- [6] National Institute of Standards and Technology. Tentative timeline of the development of new hash functions. <http://csrc.nist.gov/groups/ST/hash/timeline.html>, November 2007.
- [7] National Institute of Standards and Technology. Cryptographic hash algorithm competition. <http://csrc.nist.gov/groups/ST/hash/sha-3/index.html>, January 2008.
- [8] Department of Commerce: National Institute of Standards and Technology. Announcing request for candidate algorithm nominations for a new cryptographic hash algorithm (SHA-3) family. *Federal Register*, 72(212):62212 – 62220, November 2007. Docket No.: 070911510-7512-01.
- [9] Bruce Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C, Second Edition*. Wiley, October 1995.
- [10] M.M.J. Stevens. On collisions for MD5. Master’s thesis, Eindhoven University of Technology, 2007.
- [11] Douglas R. Stinson. *Cryptography Theory and Practice*. Discrete Mathematics and its Applications. Chapman & Hall/CRC, third edition, 2006.
- [12] Xiaoyun Wang, Andrew Yao, and Frances Yao. Cryptanalysis on SHA-1. *Cryptographic Hash Workshop hosted by NIST*, October 2005. [http://csrc.nist.gov/groups/ST/hash/documents/Wang\\_SHA1-New-Result.pdf](http://csrc.nist.gov/groups/ST/hash/documents/Wang_SHA1-New-Result.pdf).
- [13] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding collisions in the full SHA-1. *Lecture Notes in Computer Science*, 3621, 2005.
- [14] Xiaoyun Wang and Hongbo Yu. How to break MD5 and other hash functions. *Lecture Notes in Computer Science*, 3494, 2005.