Master Thesis Proposal
Continuation of the Security Proofs of the MD6 Hash Function Mode of Operation

By

Ahmed Mohamed Ezzat

American University in Cairo
# Contents

CHAPTER 1 ................................................................................................................................. 3
  INTRODUCTION .................................................................................................................. 3
    Background ....................................................................................................................... 3
    Hash function properties ............................................................................................... 4
    Mode of Operation ......................................................................................................... 5
    Iterative Modes of Operation ........................................................................................ 5
    NIST SHA-3 Competition ............................................................................................. 7

CHAPTER 2 ............................................................................................................................... 9
  PRELIMINARIES ................................................................................................................ 9
    The MD6 Cryptographic Hash Function ........................................................................ 9
    MD6 Mode of Operation ............................................................................................... 10

CHAPTER 3 ............................................................................................................................. 11
  RESEARCH PROBLEM .................................................................................................... 11
    RESEARCH OBJECTIVE .............................................................................................. 11

CHAPTER 4 ............................................................................................................................. 12
  APPROACH ....................................................................................................................... 12

CHAPTER 5 ............................................................................................................................. 13
  REFERENCES ....................................................................................................................... 13
Chapter 1

Introduction

Background

Hash functions are an essential part of the field of cryptography, with their uses ranging across a large spectrum of applications including: message integrity and authentication, digital signatures, secure time-stamping, etc… More specifically, the security of these applications depends considerably on that of the underlying hash function. As such it is considered important to prove the properties of a hash function, including its reliability against well known attacks.

A hash function $H$ is an efficiently-computable algorithm that takes as input an arbitrary-length message $M$ and potentially a fixed-length key $K$ (if we are considering a keyed hash function), and produces a fixed-length output $D$ called the message digest as follows:

$$H(K, M) = D$$

In 2008, the Cryptography and Information Security Group in MIT proposed the MD6 algorithm as a response to the NIST SHA-3 competition. A number of security properties of the MD6 algorithm’s mode of operation have been proven in a thesis by Crutchfield. However, a number of such properties still remain, and it’s our aim in this thesis to continue the proofs for the rest of the required security properties.
Hash function properties

The main properties a hash function should have are:

- **Collision resistance**: An adversary should not be able to find two distinct messages $M$ and $M'$ such that $H(M) = H(M')$ (a collision). As shown in the above hash-and-sign example, the security of the signature scheme using $H$ depends strongly on the collision resistance of $H$.

- **First preimage resistance**: An adversary given a target image $D$ should not be able to find a preimage $M$ such that $H(M) = D$. One reason (among many) why this property is important is that on most computer systems user passwords are stored as the cryptographic hash of the password instead of just the plaintext password. Thus an adversary who gains access to the password file cannot use it to then gain access to the system, unless it is able to invert target message digests of the hash function.

- **Second preimage resistance**: An adversary given a message $M$ should not be able to find another message $M'$ such that $M \neq M'$ and $H(M') = H(M)$. This property is implied by collision resistance.

- **Pseudo-randomness**: (For keyed hash functions) an adversary should not be able to distinguish the outputs of $H(K, \cdot)$ from a truly random function. Note that pseudo-randomness necessarily implies unpredictability, meaning a pseudo-random function (PRF) naturally is a message authentication code (MAC). However, the converse is not necessarily true, and indeed PRF is a much stronger condition than unpredictability.

- **Unpredictability**: (For keyed hash functions) an adversary given oracle access to $H(K, \cdot)$ should not be able to forge the output of a message it did not query. That is it should not be able to produce a hash pair $(M, D)$ where $H(K,M) = D$ without having already queried $M$. We say that if a function is unpredictable, then it is a message authentication code (MAC).
Mode of Operation

A mode of operation $M$ is an algorithm that, given a fixed-length compression function or block cipher $f$, describes how to apply $f$ repeatedly on fixed-length chunks of the arbitrarily-sized input in order to produce a fixed-length output for the whole. In this way, one can construct Variable Input Length (VIL) cryptographic primitives from Fixed Input Length (FIL) cryptographic primitives. This action is commonly referred to as domain extension [2].

Iterative Modes of Operation

Many common hash functions in use today — such as MD5 or SHA-1 — are based on an iterative chaining mode of operation frequently referred to as the Merkle-Damgard construction. The Merkle-Damgard construction typically makes use of a compression function $f : \{0, 1\}^{n+\ell} \rightarrow \{0, 1\}^n$, or a block cipher $E$ made to behave as a compression function via the Davies-Meyer transform: $f(x, y) = E_x(y) \oplus y$.

If $f$ is a compression function as defined above, then the plain Merkle-Damgard construction that uses this compression function, $MD^f$, begins first by padding the input message $m$ to have a length that is an integer multiple of $\ell$, and picking some fixed $n$-bit initial vector $IV$. It then proceeds sequentially through the $\ell$-bit message chunks, starting from the first chunk, and ending after processing the last one. A pseudo-code of an iterative mode of operation is as follows:

Algorithm $MD^f$

Input: $m = m_1 || m_2 || \cdots || m_t$, where $|m_i| = \ell$ for all $i$.

Output: The message digest, $D$.

1. $y_0 \leftarrow IV$
2. for $i \leftarrow 1$ to $t$
3. $y_i \leftarrow f(y_{i-1}, m_i)$
4. $D \leftarrow y_t$
The Merkle-Damgard construction has been well-studied in the literature, and variations on this mode of operation (e.g. strengthened Merkle-Damgard and others) have been shown to be domain extenders for various cryptographic properties: collision-resistance, pseudo-randomness, un-forgeability (MAC), indifferentiability from a random oracle, and several others.
NIST SHA-3 Competition

Although the SHA-2 (secure hash family) family of hash functions has not yet succumbed to the kind of collision-finding attacks that have plagued MD5 and SHA-1 (among others) in recent years, the U.S. National Institute of Standards and Technology put out a call in 2007 for candidate algorithms for a new cryptographic hash function family, called SHA-3. As stated in the call for submissions, “a successful collision attack on an algorithm in the SHA-2 family could have catastrophic effects for digital signatures”, thus necessitating the design of an even more resilient cryptographic hash function family. Although SHA-3 candidates will not differ from SHA-2 in the size of the message digest (which may vary from 224, 256, 384, and 512 bits) or the size of other input parameters such as the key, NIST expects that candidate proposals will improve upon the SHA-2 designs by allowing for randomized (salted) hashing, being inherently parallelizable to take advantage of today’s multi-core processor design, and being resilient to length extension attacks that many Merkle-Damgard based hash functions succumb to. On the last two points, “NIST is open to, and encourages, submissions of hash functions that differ from the traditional Merkle-Damgard model, using other structures, chaining modes, and possibly additional inputs.”

In terms of the security requirements for a proposed SHA-3 candidate, the call for submissions specifically states the following conditions. For a message digest of d-bits, candidate hash functions must have:

- Collision resistance of approximately \( \frac{d}{2} \) bits.
- First-preimage resistance of approximately \( d \) bits.
- Second-preimage resistance of approximately \( d - k \) bits for any message shorter than \( 2k \) bits.

For the keyed variant of the candidate hash function proposal, NIST requires that the hash functions supports HMAC (keyed hash function message authentication codes), PRF (pseudorandom function), as well as randomized hashing. An additional security requirement for these modes (for a message digest of d bits) is:
When using HMAC to construct a PRF, the PRF should not be distinguishable from a truly random function with significantly less than $2^{d/2}$ queries to the hashing oracle and computation significantly less than a preimage attack.
Chapter 2

Preliminaries

The MD6 Cryptographic Hash Function

The MD6 hash function is comprised of two main components: the MD6 compression function and the MD6 mode of operation. The MD6 compression function maps 89 64-bit words of input (64 words of data B, 8 words for the key K, 15 fixed words Q, and 2 auxiliary information words) down to 16 64-bit words of output. Therefore in practice it is a function \( f: \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^c \) with \( k = 8w \), \( n = 66w \), and \( c = 16w \) (where \( w = 64 \))

Figure 2-1: The compression function input contains 89 64-bit words: a 15-word constant vector Q, an 8-word key K, a one-word unique node ID U, a one-word control variable V, and a 64-word data block B. The first four items form the auxiliary (or header) information, shown in grey.

Note that although it takes in 89 words of input, 15 words are fixed for the constant Q, hence in practice it is only a function on 74w total words of input (8w of which are assigned for the key). In addition, since the data portion of its input is of length 64w and its output is c = 16w, the MD6 compression function represents a four-fold.
**MD6 Mode of Operation**

MD6 makes use of a substantially different tree-based mode of operation that allows for greater parallelism. Whereas the Merkle-Damgard construction, when viewed as a graph, is essentially a long chain, MD6 may be viewed as a tree-like construction, with a 4-to-1 compression function reducing the overall length of the message at each level.

![Figure 2-2: The MD6 mode of operation. The computation begins from the bottom and works its way to the top; the root node represents the final compression function which outputs the message digest.](image)

What makes this particular mode of operation different from other tree-based hashing and message authentication code (MAC) schemes in the literature is that each node in the tree is labeled with some auxiliary information that also feeds into the compression function. In particular, each node is given a unique identifier (effectively changing the characteristic of the compression function at each node in the tree) and the root node is “flagged” with a bit $z$ that identifies that it is the final compression function used. This auxiliary information encoded into the input of the each compression function prevents the type of hash function attacks whereby an adversary can produce a cleverly-constructed message query that corresponds to some substructure of another query (for example, preventing length-extension attacks).

A more detailed description of the MD6 algorithm and the MD6 MOO can be found at [3].
Chapter 3

Research problem

As aforementioned, a number of properties still remain unproven. These properties are:

- Second pre-image resistance
- Unpredictability
- Maurer’s indifferentiability from a random oracle. (optional)

Research Objective

The continuation of the security proofs for the MD6 hash function mode of operation.
Chapter 4

Approach

Up-to this point, no obvious approach has appeared on the horizon. This is partly due to the fact that in our case, finding an approach essentially means solving the problem. Regardless, the main facet of any approach one shall undertake is that in this thesis we aim to prove that the MD6 mode of operation has a property if the underlying MD6 hash function has it. Also, one might resort to assuming that the MOO does not hold a property, and from there find the range of input that violates the property. Should that range of input be improbable, then for all intents and purposes, the property would \textit{practically} hold.

In summary, we have two main possible approach categories:

- **Mathematical:** Aiming for a positive result, I will attempt to show that the remaining properties fully hold. This would show that the MD6 MOO is invincible to any possible attack.

- **Empirical:** Assuming the negative case where a property does not hold, I will attempt to find the range of input that violates the property. Hopefully, the improbability of that range will prove the MD6 MOO reliable enough for practical purposes.
Chapter 5

References

[1] Cryptographic hash function. Available at
   http://en.wikipedia.org/wiki/Cryptographic_hash_function

[2] Christopher Yale Crutchfield. Security Proofs for the MD6 Hash Function of
   Operation. Available at http://groups.csail.mit.edu/cis/theses/crutchfield-masters-
   thesis.pdf

   http://groups.csail.mit.edu/cis/md6/submitted-2008-10-
   27/Supporting_Documentation/md6_report.pdf