Measuring the Utility of a Collision: The Neighborhood of MD4 Collisions

Alexander M. Scheel
Iowa State University
Ames, IA, USA
Email: alexander.m.scheel@gmail.com

Eric W. D. Rozier
Iowa State University
Ames, IA, USA
Email: erozier@iastate.edu

***Abstract*—We propose a series of metrics for evaluating the utility of collision classes and develop a set of techniques for generating new collision classes in hash functions utilizing a logical cryptanalaysis framework. We demonstrate the effectiveness of this framework on the hash function MD4 and show that the existing attacks are of high utility, but do not allow for second preimage attacks under an expansive neighborhood. We partially analyze over 35,000 new collision classes for their utility under these methods.**

I. Introduction

Cryptographic hash functions form the core of many protocols. From file integrity checks to verify long term storage of data, to cache invalidation techniques, and use as a building block in network protocols such as Kerberos and TLS, this class of functions necessarily has some strong guarantees about the properties of these functions. There are three basic properties hash functions must have to be considered cryptographically secure:

- Preimage Resistance: It should be computationally hard to find the inverse of a hash function.
- Second Preimage Resistance: Given an input block, it should be computationally hard to find a second block which hashes to the same value as the first block.
- Collision Resistance: It should be computationally hard to find two blocks which hash to the same value.

Note that a second preimage is necessarily a collision and is also a stronger result.

From an attacker's perspective, finding a preimage or second preimage has been difficult. To the author's knowledge, the current bound for a preimage attack in MD5 is $2^{123.4}$ by Y. Sasaki and K. Aoki [1]. Further, while J. Kelsey and B. Schneier proposed a method for finding second preimages in less than 2^n, we note that this requires rather rather long messages [2]. However, collision attacks are well within reach for adversaries, and well past only theoretical attacks. The work of H. Dobbertin [3], X. Wang [4], M. Schlaffer [5] and others demonstrate the ease with which collisions can be found.

From a cryptographic perspective, the existence of a feasible collision attack breaks the requisite properties, thus deprecating the function's use. In some instances however, there continues to be widespread use of deprecated hash functions. One widespread example is the continued use by Git of SHA-1, despite M. Stevens et al. having published the first known full SHA-1 collision in February of 2017 [6]. However, this attack had a bound of $2^{63.1}$ – faster than the theoretical 2^{80} but still requiring significant investment in compute resources.

We define a series of criteria for analyzing the utility of a collision class. Collision classes of high utility are closer to a second preimage attack in the amount of flexibility they provide to an attacker, whereas classes of low utility may not affect all systems which use a particular hash function. Furthermore, we demonstrate new techniques for working with collisions under a logical cryptanalysis framework. Throughout this paper, we make use of the MD4 hash function due to the ease of generating new classes of collisions.

II. Terminology & Notation

TODO.

III. The Utility of a Collision

We propose the following metrics for evaluating the utility of a collision class:

1) The number of unique differentials a collision class has.
2) The number of unit-step neighbors a collision class has.
3) The maximum count of zeros in a the binary representation of a colliding block (and likewise with ones).
4) Whether there exists a block which collides under multiple initial values.
5) Whether or not zero, one, or both of the blocks in a collision may be of ASCII values under any input block difference.

Note that the first three are quantitative measures providing some measure of flexibility of a collision class, whereas the latter two are merely looking for a single witness for having the property. Depending on the scenario, specific properties of a collision may be of more interest than others.

The first metric evaluates the flexibility of the differential path. A differential path with more flexibility will have more differentials which produce blocks with the given differential path. More differentials implies a greater flexibility in choice of colliding block, and possibly allowing for multiple collisions for a given colliding block. Furthermore, a collision with more differentials is more likely to satisfy the last metric, having a pair of blocks–both ASCII–which produce a collision.

The second metric evaluates the density of the neighborhood of a collision class. If a collision resides in a dense neighborhood, it provides more possible collision classes to search for a second preimage, chosen prefix, or other structure desired in a collision. If however, a collision class has no neighbors, then it cannot be used to find other possible classes for other input blocks.

The maximum quantity of zeros (or ones) in the binary representation of a block serve as a measure of the extremes to which a collision can be pushed. This can additionally be extended to any suitable bit pattern in any base to provide a more relevant metric as desired by the system under study.

The fourth metric evaluates the utility of a collision when the internal state of the hash function is unknown. If a collision occurs under multiple initial values, this could be used to attack some systems where user provided input is appended to unknown data and then hashed. If a block collides under a suitably large number of initial values, the attack becomes highly likely to occur successfully. However, measuring the exact number of initial values a block collides under is beyond the scope of this work.

The last metric is similar to the third in that it looks for specific bit patterns in a collision. ASCII is one example of a widely used constraint system. Further examples, such as JSON, XML, etc., may likewise be supplemented based on the specifics of the system.

If a collision class satisfies many of these properties, then it is more flexible and thus more likely to be used to target deployed systems. If, however, a collision class does not satisfy these properties, it impact is likely severely limited in scope, and may not provide useful information to find other collision classes which have higher utility.

IV. Conclusion

The conclusion goes here.

Acknowledgment

The authors would like to thank the Department of Electrical and Computer Engineering.

References

[1] Y. Sasaki and K. Aoki, *Finding Preimages in Full MD5 Faster Than Exhaustive Search*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 134–152, https://www.iacr.org/archive/eurocrypt2009/54790136/54790136.pdf.

[2] J. Kelsey and B. Schneier, *Second Preimages on n-Bit Hash Functions for Much Less than 2^n Work*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 474–490, https://www.schneier.com/academic/paperfiles/paper-preimages.pdf. [Online]. Available: https://doi.org/10.1007/11426639_28

[3] H. Dobbertin, "Cryptanalysis of md4," *Journal of Cryptology*, vol. 11, no. 4, pp. 253–271, Sep 1998, https://link.springer.com/content/pdf/10.1007/s001459900047.pdf. [Online]. Available: https://doi.org/10.1007/s001459900047

[4] X. Wang, D. Feng, X. Lai, and H. Yu, "Collisions for hash functions md4, md5, haval-128 and ripemd," Cryptology ePrint Archive, Report 2004/199, 2004, http://eprint.iacr.org/2004/199.

[5] M. Schläffer and E. Oswald, *Searching for Differential Paths in MD4*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 242–261, https://link.springer.com/content/pdf/10.1007%2F11799313_16.pdf. [Online]. Available: https://doi.org/10.1007/11799313_16

[6] M. Stevens, E. Bursztein, P. Karpman, A. Albertini, and Y. Markov, "The first collision for full sha-1," Cryptology ePrint Archive, Report 2017/190, 2017, http://eprint.iacr.org/2017/190.