Digital Investigation (2005) 2, 36—40

ELSEVIER

Digital Investigation

www.elsevier.com/locate/diin

MD5 collisions and the impact on computer forensics

Eric Thompson

AccessData Corporation, 384 South 400 West, Lindon, UT 84042, United States

Abstract In August 2004 at the annual cryptography conference in Santa Barbara, California a group of cryptographers, Xianyan Wang, Dengguo Feng, Xuejia Lai, Hongbo Yu, made the announcement that they had successfully generated two files with different contents that had the same MD5 hash. This paper reviews the announcement and discusses the impact this discovery may have on the use of MD5 hash functions for evidence authentication in the field of computer forensics.

Hash functions are one of the basic building blocks of modern cryptography. In cryptography hash functions are used for everything from password verification to digital signatures. A hash function has three fundamental properties:

- A hash function must be able to easily convert digital information (i.e. a message) into a fixed length hash value.
- It must be computationally infeasible to derive any information about the input message from just the hash.
- It must be computationally infeasible to find two files that have the same hash. Hash(Message 1) = Hash(Message 2).

In computer forensics hash functions are important because they provide a means of identifying and classifying electronic evidence. Because hash functions play a critical role in evidence authentication it is critical a judge or jury can trust the hash values that uniquely identify electronic evidence.

The third property of a hash function states that it must be computationally infeasible to find two files to have the same hash. The research published by Wang, Feng, Lai and Yu demonstrated that MD5 fails this third requirement since two different messages have been generated that have the same hash. This situation is called a collision.

Birthday paradox

One method of demonstrating that a hash function is insecure is to find ANY two messages that have

E-mail address: eric@accessdata.com

1742-2876/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.diin.2005.01.004

the same hash. The easiest method of accomplishing this is through what is frequently referred to as a birthday attack or birthday paradox. When a person enters a room, how many people need to be in the room before there is greater than 50% probability that one of those people in the room will share the first person's birthday (same day not the same year)? The answer is 183 (365/2). This is because you are attempting to find someone who matches a specific date. However, how many people must be in a room before there is a probably greater than 50% that there exist ANY two pairs in the room that have the same birthday. The number is surprisingly low: 23. When 23 people are in the room there are a total of 253 different pairs of dates. This is called the birthday paradox. For a more detailed description of the birthday paradox see Patterson (1987).

The birthday paradox is a standard statistical problem. For each additional person *n* who enters a room the number of pairs of birthdays increases by $n-1$. As more people enter the room the number of birthday pairs increases rapidly until a matching pair is found.

In the first example, the attempt was to find a person in the room that matched one specific birthday. When matching a specific day, each person has only a 1/365 chance of being born on the specific day in question. In cryptography, the first example is analogous to a brute force or exhaustive key space attack. This is the process used in most password recovery/password guessing attacks. In the second example ANY birthday pair will suffice. This second type of attack is the process used by cryptographers to attack hash functions.

If a hash function has a key space of 64 bits, then an exhaustive key space attack would require a computer test up to 2^{64} combinations. If a single computer could process one million hashes per second and an advisory could use a distributed network attack to harness the CPU power of 10,000 computers it would still take up to 58 years to exhaust the key space. However, if the goal was simply to find ANY hash match a single computer could find that match in slightly more than an hour.

Fortunately MD5 and other common hash functions have substantially larger key length than 64 bits. For MD5 the key length is 128 bits, for SHA-1 the key length is 160 bits, SHA-256 the key length is 256 bits. However, if a cryptographic weakness is discovered in the design of the hash algorithm this weakness can reduce the effective key length of the hash function to be less than the intended design length. In this case the weakness makes possible the potential for a birthday attack to successfully find a hash collision. This weakness induced collision is what occurred with the MD5 algorithm.

MD5

The MD5 hash function was developed in 1994 by the cryptographer Ron Rivest as a stronger alternative to the MD4 algorithm developed in 1992. The algorithm breaks a file into 512 bit input blocks. Each block is run through a series of functions to produce a unique 128 bit hash value for the file. Changing just one bit in any of the input block should have a cascade effect that completely alters the hash results. Furthermore, since the key size of 128 bits has 3.4×10^{38} possible combinations the chance of randomly finding two files that produce the same hash value should be computationally infeasible (Schneier, 1996).

Cryptanalysis of MD5

MD5 has been intensely scrutinized by the cryptographic community since its original release. Prior to 2004, most of the research attacks against MD5 demonstrated only minor weaknesses in the MD5 design. However, there are two particularly notable exceptions that discovered more serious design problems.

The first indication that MD5 might have a design flaw was in a paper published by Den Boer and Booselaer in which it was demonstrated that given certain different input conditions it was possible for there to exist identical internal states for some of the MD5 computations. However, Boer and Booselaer were not able to expand upon these internal anomalies to produce duplicate hashes for different input values (Den Boer and Bosselaers, 1994).

The second significant research advancement occurred in 1996 when Dobbertin was able to demonstrate that the MD5 algorithm could produce identical hashes for two different messages if the initialization vector could be chosen (Dobbertin, 1996). The initialization vector is the value to which the MD5 internal variables are initially set before beginning the hashing process. Because MD5, when used in real life, is always set to the same initialization state (IV_0) Dobbertin's result did not present an immediate security concern. However, his work did demonstrate that an eventual MD5 collision would probably be discovered.

Table 1 MD5 collsion

Message 1	1st Block	02DD31D1 C4EEE6C5 069A3D69 5CF9AF98 **87**B5CA2F AB7E4612 3E580440 897FFBB8 0634AD55 02B3F409 8388E483 5A41**71**25 E8255108 9FC9CDF7 **F2**BD1DD9 5B3C3780
	2nd Block	D11D0B96 9C7B41DC F497D8E4 D555655A **C7**9A7335 0CFDEBF0 66F12930 8FB109D1 797F2775 EB5CD530 BAADE822 5C15**CC**79 DDCB74ED 6DD3C55F **D8**0A9BB1 E3A7CC35
Message 2	1st Block	02DD31D1 C4EEE6C5 069A3D69 5CF9AF98 **07**B5CA2F AB7E4612 3E580440 897FFBB8 0634AD55 02B3F409 8388E483 5A41**F1**25 E8255108 9FC9CDF7 **72**BD1DD9 5B3C3780
	2nd Block	D11D0B96 9C7B41DC F497D8E4 D555655A **47**9A7335 0CFDEBF0 66F12930 8FB109D1 797F2775 EB5CD530 BAADE822 5C15**4C**79 DDCB74ED 6DD3C55F **58**0A9BB1 E3A7CC35
MD5 Hash		8D5E7019 6324C015 715D6B58 61804E08

In the summer of 2004 the cryptographers Wang et al. demonstrated their ability to generate MD5 collisions using the standard initialization vector IV_0. This research showed that it is possible to create two related 512 bit input blocks and modify specific bits within these blocks, creating two slightly different messages, that have the same hash value. The amount of time to create an MD5 message pair was on average 1 h (Wang et al., 2004).

Example of an MD5 collision

In their paper Wang et al. provided an example of two MD5 collisions. One of the collisions is as given in Table 1.

Response of the cryptographic community to MD5 collisions

The response of the cryptographic community has been what should be expected. While these results are mathematically significant they do not present an immediate cause for alarm. Creating two messages that have identical MD5 hashes requires very specific circumstances that would have an extremely rare chance of actually existing in the regular world. Additionally this research does not provide a hacker with any new technique to break through a firewall, attack a public key encryption system or fabricate a false digitally signed message. Nevertheless, this research does point out a design weakness in the MD5 algorithm and as a result the cryptographic community needs to increase the diligence in which it searches for a new hash standard.

Bruce Schneier summarized the feelings of many in the cryptographic community with his statement:

''The magnitude of the results depends on who you are. If you're a cryptographer, this is a huge deal. While not revolutionary, these results are substantial advances in the field. The techniques described by the researchers are likely to have other applications, and we'll be better able to design secure systems as a result. ... As a user of cryptographic systems — as I assume most readers are — this news is important, but not particularly worrisome. MD5 and SHA aren't suddenly insecure. No one is going to be breaking digital signatures or reading encrypted messages anytime soon with these techniques. The electronic world is no less secure after these announcements than it was before.'' (Schneier, 2004)

The impact of MD5 collision on the use of MD5 in computer forensics

The recent research on MD5 collision should have little impact on the use of MD5 for evidence authentication in computer forensics. Three reasons for this are:

(1) *MD5 is still secure against a brute force attack* — It is computationally infeasible to

modify the contents of a message such that the hash of the new message matches some predetermined hash value. No one in the cryptographic research community has yet to be able to generate a new file or modify an existing file so that the new file will convey intelligible information and still match a pre-determined MD5 hash from a different file.

(2) *Changing one bit in the evidence will still cause a cascade effect that dramatically changes the MD5 hash result* — A collision similar to that one demonstrated by Wang et al. can only be produced using very specific input blocks. There is no reason for these types of input blocks to occur in the real world. Therefore, there is no reason to believe the internal state of the MD5 engine that allowed for the collision would naturally occur. The MD5 engine does a remarkably good job of generating a cascade effect on all the bits in the hash value even when just a single bit in the input file is changed. MD5 can still be relied upon by the forensics community to do an excellent job at identifying even the smallest change in electronic data.

(3) *The chance of a birthday collision from files that are part of the NIST data set or hash keeper project are very remote* — The birthday collision that was produced by these cryptographers required a very special set of circumstances to occur within the internal variables of the MD5 engine. It is unrealistic to believe that this kind of state would occur naturally when analyzing files that would normally be found on a computer, PDA or similar electronic device. In the real world the number of files required for there to be a 50% probability for an MD5 collision to exist is still 2^{64} or 1.8×10^{19}. The chance of an MD5 hash collision to exist in a computer case with 10 million files is still astronomically low.

For those who wish to be overly cautious, it is always possible to hash electronic evidence using both MD5 and another hash function such as SHA-1 or SHA-256. Since these hash functions are linearly independent of each other, the resulting uniqueness of having both these hash values would be the sum of the bits from each individual hash. For example, a file that has been hashed with both MD5 (128 bits) and SHA-1 (160 bits) would have an effective uniqueness of 288 bits or $1{:}10^{86}$. Even if a weakness could be found that reduces the effective key size of one of these hash functions it is still computationally unrealistic that in our life time, there will be two different data streams that would have the same MD5 and SHA-1 hash.

Conclusion

The struggle to make a perfect cryptosystem has long eluded cryptographers. New cryptographic codes are created and broken every day. It is through this challenge that the cryptographic technology advances forward. Cryptographers have new information about how to design hash functions that Ron Rivest did not know back in 1994 when he published his work on the MD5 algorithm. This new announcement does not present a current security threat nor does it make the use of MD5 for evidence authentication any less trustworthy. Instead, this research gives mathematicians information about how to design hash functions so the next generation's codes can be better and stronger.

As a result of these developments, in the next several years a new set of hash algorithms will most certainly emerge. These new algorithms will be resistant to the weakness discovered by Wang et al. One of these new algorithms will rise to the top and for a period of time, serve as the worlds next hash function standard. Several years afterwards, a brilliant mathematician will discover a weakness in this new algorithm, publish their results, and the process of finding another hash standard will start all over again.

The computer forensics community will want to embrace the new hash technology once it has been thoroughly tested by the cryptographic community. Until then, computer forensics examiners should feel comfortable in their continued, all be it short term use of MD5. When possible, hashing electronic evidence with both MD5 and a second hash function such as SHA-1 or SHA-256 is always a good idea, however, the forensics software needs to support multiple hash functions in order for this to be possible. Unless new information emerges showing a further weakness in the MD5 hash algorithm, there should not be an immediate requirement to discontinue the use of MD5. Rather, forensics examiners should work with the manufacturers of forensics software so that new releases of the forensics software, when possible, will start implementing stronger hash functions such as SHA-1 or SHA-256 into the forensics process.

References

Den Boer B, Bosselaers A. Collisions for the compression function of MD5, Advances in Cryptology — EUROCRYPT'93. LNCS 765; 1994. p. 293—304.

Dobbertin Hans. Cryptanalysis of MD5 compress. German Information Security Agency; May 1996.

Patterson Wayne. Mathematical cryptology for computer scientists and mathematicians. Rowman & Littlefield, Publishers; 1987. p. 156—8.

Schneier Bruce. Applied cryptography, second edition protocols, algorithms and source code in C. John Wiley & Sons, Inc.; 1996. p. 436—41.

Schneier Bruce. Opinion: cryptanalysis of MD5 and SHA: time for a new standard. Computerworld; April 19, 2004.

Wang Xianyan, Feng Dengguo, Lai Xuejia, Yu Hongbo. Collisions for hash functions MD4, MD5 Haval-128 and RIPEMD. CRYPTO'04; Revised August 17, 2004.

Available online at www.sciencedirect.com

SCIENCE DIRECT®