Quantum Preimage and Collision Attacks on CubeHash

Gaëtan Leurent

Abstract. In this short note we show a quantum preimage attack on CubeHash-normal-512 with complexity 2^{192}. This kind of attack is expected to cost 2^{256} for a good 512-bit hash function, and we argue that this violates the expected security of CubeHash. The preimage attack can also be used as a collision attack, given that a generic quantum collision attack on a 512-bit hash function require 2^{256} operations, as explained in the CubeHash submission document.
This attack only use very simple techniques: we use the symmetry properties of CubeHash which were already described in the submission document and have been analyzed in detail in [1,5], together with Gover's algorithm which is also discussed in the submission document.

1 Introduction

CubeHash is a hash function designed by Bernstein and submitted to the SHA-3 competition [2]. It has been accepted for the second round of the competition.

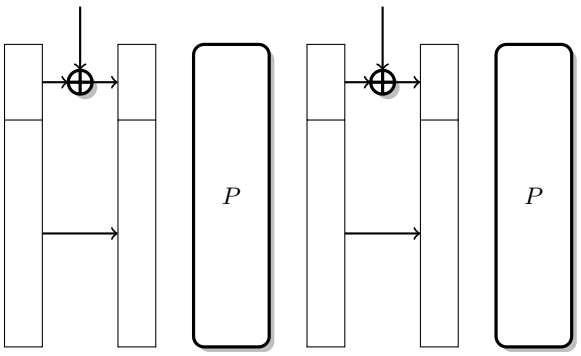

Fig. 1. CubeHash follows the sponge construction

1.1 CubeHash Versions

CubeHash is built following the sponge construction with a 1024-bit permutation. The small size of the state allows for compact hardware implementations, but it is too small to build a fast 512-bit hash function satisfying NIST security requirements. The submission document defines two versions of CubeHash with 512 bits of output: CubeHash-normal (*aka* CubeHash-16/32-512) and CubeHash-formal (*aka* CubeHash-16/1-512), but does not explicitly state which one is to be considered as a SHA-3 candidate. On the one hand, CubeHash-normal reaches a reasonable speed at the cost of security by using a

capacity of 768 bits: this implies that preimage attacks only cost 2^{384}. On the other hand, CubeHash-formal reaches a reasonable security at the cost of speed by using a capacity of 1016 bits, and is much slower than the other second round candidates.

Interestingly, the reference code, the test vectors, and the benchmarks (section 2.B.2) are given for CubeHash-normal[1], but the security claims (section 2.B.4) are made for CubeHash-formal.

In this note we study CubeHash-normal-512, *i.e.* CubeHash-16/32-512.

1.2 Expected Security of CubeHash-normal

Strangely enough, the submission document of CubeHash does not make any formal security claims for CubeHash-normal-512, which is obviously the version that will be targeted by cryptanalysts. Moreover, the expected security of CubeHash-normal does not seem to follow one of the standard security notions.

The submission document only acknowledge that the best known preimage attack against CubeHash-normal-512 has complexity 2^{384}, and argue that it is not sensible to consider attacks with complexity higher than 2^{256}. This led many cryptographers to believe that CubeHash-normal had an expected security against preimage attack of 384 bits, but the designer stated on the NIST mailing list that CubeHash-normal-512 was actually only offering 256 bits of security, and declined to give a more formal statement similar to the claims for CubeHash-formal in the submission document.

In the absence of a clear security claim from the designer, one can either assume that the function does not offer any security, or extrapolate from the pieces of information available. Given that the main motivation for not offering 512 bits of security against *classical* preimage attacks is the availability of a *quantum* preimage attack with complexity 2^{256}, we believe that CubeHash-normal has been designed with quantum attacks in mind. Therefore, in the absence of any specific claim regarding quantum attacks, we assume that it should offer the same level of security as a good hash function against such attacks.

More explicitly, when discussing attacks on CubeHash, the submission document states (in section 2.B.5):

> Of course, these attack strategies need to be compared to attacks that apply to *all* h-bit hash functions:
> - Parallel collision search (1994 van Oorschot–Wiener), finding h-bit collisions in time roughly $2^{h/2}/A$ on circuits of total area A.
> - Parallel quantum preimage search (1996 Grover), finding h-bit preimages in time roughly $2^{h/2}/A^{1/2}$ on quantum circuits of total area A.

Our new observation precisely leads to an attack more efficient than the parallel quantum preimage search of Grover.

[1] Additionally, the title of the tweak document "16 times faster", should actually be "twice as slow" if CubeHash-formal is the main candidate.

When discussing the expected security of CubeHash-normal and CubeHash-formal, Bernstein explained on the hash forum that he was interested in three different security notion [4]:

1. security against all attacks costing below 2^{128},
2. security against all attacks costing below 2^{256}, and
3. security against pre-quantum preimage attacks costing below 2^{512}.

It seems that the three version of CubeHash submitted as second-round candidates target those three security levels (see also [3]): CubeHash-16/32-256 for level (1), CubeHash-16/32-512 for level (2), and CubeHash-16/1-512 for level (3). This would imply that CubeHash-16/32-512 is supposed to resist to quantum attacks up to 2^{256}.

1.3 CubeHash Symmetries

The design of CubeHash is very symmetric and does not use any constants. Therefore, there exists many symmetry classes for the permutation. This was stated in the submission document, and later work have provided an explicit description of the symmetry classes and analysis of how to use the symmetries classes to attack the hash function [1,5].

The most efficient way to use those symmetries is to use a symmetry class such that the message expansion can produce symmetric messages, following the attack of [1, Section 4.3, variant of the attack], later described in [5]. For instance, we can use the symmetry class called C_2 in [1]. A state is symmetric if:

$$\forall i, j, k, l \quad x_{ijkl0} = x_{ijkl1}$$

When b is 32, as is the case for CubeHash-normal, the message injection gives control over $x_{00klm}, \forall k, l, m$. Therefore, in order to reach a symmetric state, one just has to reach a state satisfying the following 384-bit equation:

$$x_{01kl0} = x_{01kl1} \qquad x_{10kl0} = x_{10kl1} \qquad x_{11kl0} = x_{11kl1} \qquad \forall\, k, l \qquad (1)$$

and the message injection can be used to make the state fully symmetric. This is expected to cost 2^{384} on average.

This can be used to mount a preimage attack with the following steps:

1. Find a message A reaching a symmetric state from the IV.
2. Find a message D reaching a symmetric state backwards from the target value (you should first extend the target value into a full state, and compute the finalisation backwards).
3. Build 2^{192} symmetric messages B_i. Compute the states reached after processing $A\|B_i$.
4. Build 2^{192} symmetric messages C_j. Compute the states reached after processing $C_j\|D$ backwards.
5. With a good probability, there will be a pair of values that match on the last 768 bits (384 of those bits come from free because of the symmetry). Then, use a message bloc X to match the first 256 bits. This yields a preimage $A\|B_{i_0}\|X\|C_{j_0}\|D$

Steps 1 and 2 cost 2^{384}, while step 3 and 4 cost 2^{192}. Note that the meet in the middle technique can actually be done without memory. This attack has essentially the same complexity as a capacity-based attack when b is a power of two, but it becomes more efficient when b is not a power of two[2].

2 New Observation

The most expensive part of the symmetry based attack of [1], recalled in the previous section, is to reach a symmetric state. However, it turns out that it is actually relatively easy to reach a symmetric state using Grover's algorithm on a quantum computer. Indeed, reaching a state satisfying equation (1) is equivalent to finding a preimage of zero for a hash function that would iterate the round function as CubeHash, and whose output would be (without any blank rounds):

$$
\begin{array}{llll}
x_{01000} \oplus x_{01001} & x_{01010} \oplus x_{01011} & x_{01100} \oplus x_{01101} & x_{01110} \oplus x_{01111} \\
x_{10000} \oplus x_{10001} & x_{10010} \oplus x_{10011} & x_{10100} \oplus x_{10101} & x_{10110} \oplus x_{10111} \\
x_{11000} \oplus x_{11001} & x_{11010} \oplus x_{11011} & x_{11100} \oplus x_{11101} & x_{11110} \oplus x_{11111}
\end{array}
$$

This is a 384-bit hash function, therefore Grover's algorithm requires time 2^{192} to find a preimage of zero on a small quantum computer.

Then we can use the same meet-in-the-middle technique as in the previous symmetry based attack, which requires time 2^{192} on a classical computer. This gives a preimage attack on CubeHash-normal with complexity 2^{192}, assuming that quantum computers are available.

2.1 Alternative attack

Instead of using a meet-in-the-middle technique with complexity 2^{192} on a *classical* computer, one can reach any given symmetric state (for instance, the all-zero state) for a cost of 2^{192} on a *quantum* computer using another call to Grover's algorithm.

3 Conclusion

Our work shows that CubeHash-normal can only provide the level of preimage resistance of a 384-bit hash function, even if you believe that *classical* preimage attacks are irrelevant because of more efficient *quantum* preimage attacks. Additionally, we show that the symmetry properties of the round function of CubeHash do actually lead to cryptographic weaknesses of the hash function.

[2]The proposed versions of CubeHash use powers of two for b, but the designer occasionally discussed versions of CubeHash with other values of b

References

1. Aumasson, J.P., Brier, E., Meier, W., Naya-Plasencia, M., Peyrin, T.: Inside the hypercube. In Boyd, C., Nieto, J.M.G., eds.: ACISP. Volume 5594 of Lecture Notes in Computer Science., Springer (2009) 202–213
2. Bernstein, D.J.: Cubehash specification. Submission to NIST (Round 2) (2009)
3. Bernstein, D.J.: Are options prohibited for SHA-3? NIST Hash Forum (23 Aug 2010) Message id: <20100822232203.27042.qmail@cr.yp.to>.
4. Bernstein, D.J.: OFFICIAL COMMENT: CubeHash. NIST Hash Forum (14 Aug 2010) Message id: <20100814021923.7676.qmail@cr.yp.to>.
5. Ferguson, N., Lucks, S., McKay, K.A.: Symmetric states and their structure: Improved analysis of cubehash. Cryptology ePrint Archive, Report 2010/273 (2010) http://eprint.iacr.org/.