Quantum Preimage and Collision Attacks on CubeHash

Gaëtan Leurent

University of Luxembourg, Gaetan.Leurent@uni.lu

Abstract. In this paper we show a quantum preimage attack on CubeHash-512-normal 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 uses very simple techniques, most of which are borrowed from previous analysis of CubeHash: we just combine symmetry based attacks [1,8] with Grover's algorithm. However, it is arguably the first attack on a second-round SHA-3 candidate that is more efficient than the attacks considered by the designer.

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.

In this paper we show a quantum preimage attack on CubeHash-512-normal with complexity 2^{192}. We show that this attack should be considered as better that the attacks considered by the designer, and we explain what were our expectations of the security of CubeHash.

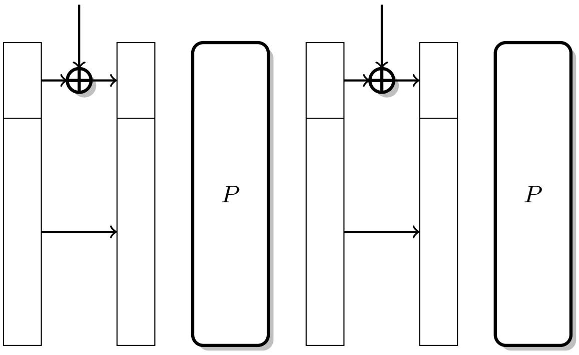

Fig. 1. CubeHash follows the sponge construction

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 updated submission document for the second round of the SHA-3 competition 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 paper we study CubeHash-512-normal, *i.e.* CubeHash-16/32-512.

1.2 Expected Security of CubeHash-normal

Strangely enough, the submission document of CubeHash for the second round does not make any formal security claims for CubeHash-512-normal, 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-512-normal has complexity 2^{384}, and argue that it is not sensible to consider attacks with complexity higher than 2^{256}.

However, a few sections discuss comparisons between attacks. In section 2.B.5 the submission document states:

> 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.

And in file Round2Mods.pdf there are some extra comments regarding quantum attacks:

> CubeHash expected strength (2.B.4) (strength.pdf) has been modified to note the expected impact of quantum computers. Grover's algorithm will find (e.g.) 224-bit preimages for any of the SHA–3 candidates in only about 2^{112} quantum operations. This quantum computer
> - has a much higher success chance than a conventional computer performing 2^{200} operations and
> - is much more likely to be available to future attackers than a conventional computer performing 2^{200} operations,

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

> so considering the conventional threat while ignoring the quantum threat makes no sense from a risk-analysis perspective.

The first quote justifies that a classical preimage attack with complexity 2^{384} on a 512-bit hash function is uninteresting because it is less practical than Grover's attack, which is the best generic preimage attack on CubeHash-512-normal with a quantum complexity of 2^{256}. The second quote clearly implies that a quantum attack of complexity 2^{192} is better that a classical attack of complexity 2^{384}. From those quotes it seems relatively clear that a quantum preimage attack with complexity 2^{192} is significantly better than all known attacks on CubeHash-512-normal. Since no such attack is discussed in the submission document, we can only assume that it was not anticipated by the designer.

We discuss the expected security of CubeHash-512-normal in more details in Section 3, including the recent security claims for CubeHash-512-normal published as part of a proposed tweak for the third round [4].

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 [1,8].

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 [8]. For instance, we can use the symmetry class called C_2 in [1]. For the symmetry class, 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:

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

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 which satisfy

$$b_{01kl0} = c_{01kl0} \qquad b_{10kl0} = c_{10kl0} \qquad b_{11kl0} = c_{11kl0} \qquad \forall\, k, l$$

and the symmetry (1) implies:

$$b_{01kl1} = c_{01kl1} \qquad b_{10kl1} = c_{10kl1} \qquad b_{11kl1} = c_{11kl1} \qquad \forall\, k, l$$

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

The second part of the attack uses a meet-in-the-middle technique with complexity 2^{192} on a *classical* computer. Alternatively 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 instance of Grover's algorithm.

[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

3 Security Claims for CubeHash-normal

In Section 1.2, we explained why the submission document of CubeHash implied that this observation led to a better attack that all previous attacks. In this section, we look at more recent statements of the designer, and we try to understand the expected security of CubeHash-512-normal.

Since the announcement of this attack, there has been some discussion on the NIST Hash Forum over whether this observation should be considered as an attack[3]. One of the consequences of this debate has been the inclusion of precise security claims for CubeHash-512-normal in a new document proposing a tweak for CubeHash-512-normal for the third round of the competition [4]. This document, published after the announcement of our attack, makes the following claims:

Function	Preimage	Collision	Post-quantum Preimage	Post-quantum Collision
CubeHash512	$2^{384+\epsilon}$ bit ops	$2^{256+\epsilon}$ bit ops	$2^{192+\epsilon}$ qubit ops	$2^{192+\epsilon}$ qubit ops

Obviously, this claim is compatible with all the known attacks, but it does not describe what we had understood of the security of CubeHash-512-normal when we made our observation. In the remaining of this section, we try to explain why we had a different idea of the security of CubeHash-512-normal.

Internal coherence. The security level against quantum preimage attacks is justified by the following:

> Quantum preimage attacks: Half the preimage-attack exponent (as in the second-round CubeHash submission), following the attacker's most optimistic view of what Grover's quantum algorithm can achieve.

This justification does not really make sense, given that Grover's algorithm is an output based attack (its complexity depend on the output size, 512 in the case of CubeHash-512-normal), while the best classical preimage attack — a meet-in-the-middle attack — is capacity based (its complexity depend on the capacity of the sponge construction, 384 in the case of CubeHash-512-normal)[4]. We stress that our quantum attack with complexity 2^{192} is *not a generic attack*, and does not follow only from Grover's algorithm. More explicitly, we are not aware of any quantum attack with complexity 2^{144} against $\textsc{Keccak}[]\rfloor_{512}$, even though it has a capacity of 576 bits.

Given the emphasis on quantum attack in the submission document (shown by the quotes in Section 1.2) it is quite surprising that the security claim against quantum attack is less than the complexity of generic attacks.

[3] We note that would not have been necessary if a clear claim had been available

[4] Using the same reasoning one could say that the security against classical collision attack should be half of the preimage-attack exponent, following an optimistic view of what the birthday attack might achieve.

Comparison with previous statements. The designer of CubeHash wrote the following on the NIST Hash Forum, when discussing the security of CubeHash-512-normal [5]:

> The "Symmetric states" paper takes the 384-bit example completely out of context, and makes the reader falsely believe that (e.g.) 2^{320} operations or 2^{256} operations would qualify, when in fact the CubeHash submission document says that they *can't* qualify.

This states very explicitly that the preimage-attack security of CubeHash-512-normal was *not* 2^{384}, but 2^{256}.

Later, he gave a more precise description of the security of CubeHash, targeting three different security levels [6]:

(1) Many users want 128-bit security, *i.e.*, protection against all attacks performing fewer than 2^{128} simple operations. For these users, the official recommendation is CubeHash16/32–256: *i.e.*, 16 rounds after every 32-byte block, with 256 bits of output.

(2) Some users are concerned that 128-bit security won't be adequate for the long term, and instead want 256-bit security. For these users, the official recommendation is CubeHash16/32–512: *i.e.*, 16 rounds after every 32-byte block, with 512 bits of output. [. . .]

(3) There is a wacky notion that 256-bit security isn't enough, and that hash functions should provide 512-bit security—but that it's okay for this extra security to be only for preimages, not for collisions, and that it's okay for this to be completely broken by quantum computers, as if performing 2^{256} operations were easier than building a quantum computer.
For anyone who subscribes to this wacky notion, the official recommendation is CubeHash16/1–512 [. . .]

The wording of the description of level (3) strongly implies that the previous levels include both classical and quantum attacks. This is made more clear in a later description of those security levels [7]:

(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}.

Since "all attacks" is opposed to "pre-quantum", it is natural to understand "all attacks" as including post-quantum attacks. This was repeated later [3]:

> CubeHash16/32–512 is my main proposal, providing 2^{256} security against all attacks.

In this context, we could hardly expect that the security of CubeHash-512-normal was as described in [4], and we had to assume some other security claim. Our understanding was that CubeHash-512-normal was supposed to be as good as any 512-bit hash function in a post-quantum world, which was very coherent with the statements of the designer. Therefore, our expectations of the security of CubeHash when we did this research was the following:

Function	Preimage	Collision	Post-quantum Preimage	Post-quantum Collision
CubeHash512	$2^{256+\epsilon}$ bit ops	$2^{256+\epsilon}$ bit ops	$2^{256+\epsilon}$ qubit ops	$2^{256+\epsilon}$ qubit ops

The case of CubeHash-384. In the submission document for the second round, there was a proposal to use CubeHash-16/32-384 as CubeHash-384-normal and CubeHash-16/1-384 as CubeHash-384-formal. We note that the new security claims given in [4] imply that the both offer the same security and CubeHash-16/1-384 was removed from the list of official options. However, the fact that both versions were in the second round led us to believe that they should offer a different security level, and that made us believe that the security claim given in [4] was not what the designer indented.

4 Conclusion

Our work shows that CubeHash-normal can only provide the level of preimage resistance of a 384-bit hash function, even if you consider 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: we are not aware of any quantum attack with complexity less than 2^{256} if the symmetry properties of CubeHash are tweaked out.

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 (22 Aug 2010) Message id: `<20100822232203.27042.qmail@cr.yp.to>`.
4. Bernstein, D.J.: CubeHash parameter tweak: 10× smaller MAC overhead. NIST Hash Forum (1 Nov 2010) Available online: `http://cubehash.cr.yp.to/submission2/tweak2.pdf`.
5. Bernstein, D.J.: OFFICIAL COMMENT: CubeHash. NIST Hash Forum (25 Jul 2010) Message id: `<20100725232008.7268.qmail@cr.yp.to>`.
6. Bernstein, D.J.: OFFICIAL COMMENT: CubeHash. NIST Hash Forum (11 Aug 2010) Message id: `<20100811202934.8189.qmail@cr.yp.to>`.
7. Bernstein, D.J.: OFFICIAL COMMENT: CubeHash. NIST Hash Forum (14 Aug 2010) Message id: `<20100814021923.7676.qmail@cr.yp.to>`.
8. 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/`.