Post-quantum security of the sponge construction

Jan Czajkowski
QuSoft, Univ of Amsterdam

Leon Groot Bruinderink
TU Eindhoven

Andreas Hülsing
TU Eindhoven

Christian Schaffner
QuSoft, Univ of Amsterdam

Dominique Unruh
University of Tartu

Abstract. We investigate the post-quantum security of hash functions based on the sponge construction. A crucial property for hash functions in the post-quantum setting is the collapsing property (a strengthening of collision-resistance). We show that the sponge construction is collapsing (and in consequence quantum collision-resistant) under suitable assumptions about the underlying block function. In particular, if the block function is a random function or a (non-invertible) random permutation, the sponge construction is collapsing.

A full version of this paper can be found here: `http://www.ut.ee/~unruh/sponge.pdf`.

Hashes functions and the sponge construction. Cryptographic hash functions are one of the central primitives in cryptography. They are virtually used everywhere: As cryptographically secure checksums to verify integrity of software or data packages, as building block in security protocols, including TLS, SSH, IPSEC, as part of any efficient variable-input-length signature scheme, to build full-fledged hash-based signature schemes, in transformations for CCA-secure encryption, and many more.

While all widely deployed public-key cryptography is threatened by the rise of quantum computers, hash functions are widely believed to only be mildly effected. The reason for this is twofold. On the one hand, generic quantum attacks achieve at most a square-root speed up compared to their pre-quantum counterparts and can be proven asymptotically optimal [7, 16, 11]. On the other hand, there do not exist any dedicated quantum attacks on any specific hash function (excluding of course those based on number theory like, e.g., VSH [8]) that perform better than the generic quantum attacks.

One of the most important properties of a hash function H is collision-resistance. That is, it is infeasible to find $x \neq x'$ with $H(x) = H(x')$. Intuitively, collision-resistance guarantees some kind of computational injectivity – given $H(x)$, the value x is effectively determined. Of course, information-theoretically, x is not determined, but in many situations, we can treat the preimage x as unique, because we will never see another value with the same hash. For example, collision-resistant hashes can be used to extend the message space of signature schemes (by signing the hash of the message), or to create a commitment schemes (e.g., sending $H(x\|r)$ for random r commits us to x; the sender cannot change his mind about x because he cannot find another preimage).

In the post-quantum setting, however, it was shown by Unruh [14] that collision-resistance is weaker than expected: For example, the commitment scheme sketched in the previous paragraph is not binding: it is possible for an attacker to send a hash h, then to be given a value x, and then to send a random value r such that $h = H(x\|r)$, thus opening the commitment to any desired value – even if H is collision-resistant against quantum adversaries. This contradicts the intuitive requirement that $H(x)$ determines x.

Fortunately, Unruh [14] also presented an strengthened security definition for post-quantum secure hash functions: collapsing hash functions. Roughly speaking, a hash function is collapsing if, given a superposition of values m, measuring $H(m)$ has the same effect as measuring m (at least from the point of view of a computationally limited observer). Collapsing hash functions serve as a drop-in replacement for collision-resistant ones in the post-quantum setting: Unruh showed that several natural classical commitment schemes (namely the scheme sketched above, and the statistically-hiding schemes from [10]) become post-quantum secure when using a collapsing hash function instead of a collision-resistant one. (And the collapsing property also directly implies collision-resistance.)

In light of these results, it is desirable to find hash functions that are collapsing. Unruh [14] showed that the random oracle is collapsing. (That is, a hash function $H(x) := \mathcal{O}(x)$ is collapsing when $\mathcal{O}$ is a random oracle.) However, this has little relevance for real-world hash functions: A practical hash function is typically constructed by iteratively applying some elementary building block (e.g., a "compression function") in order to hash large messages. So even if we are willing to model the elementary building block as a random oracle, the overall hash function construction should arguably not be modeled as a random oracle.[1]

For hash functions based on the Merkle-Damgård (MD) construction (such as SHA2 [12]), Unruh [15] showed: If the compression function is collapsing, so is the hash function resulting from the MD construction. In particular, if we model the compression function as a random oracle (as is commonly done in the analysis of practical hash functions), we have that hash functions based on the MD construction are collapsing (and thus suitable for use in a post-quantum setting).

However, not all hash functions are constructed using MD. Another popular construction is the sponge construction [3], underlying for example the current international hash function standard SHA3 [13], but also other hash functions such as Quark [1], Photon [9], Spongent [5], and Gluon [2]. The sponge construction builds a hash function H from a block function[2] $\mathbf{f}$. In the classical setting, we know that the sponge construction is collision-resistant if the block function $\mathbf{f}$ is modeled as a random oracle, or a random permutation, or an invertible random permutation [4].[3] However, their proof does not carry over to the post-quantum setting: their proof relies on the fact that queries performed by the adversary to the block function are classical (i.e., not in superposition between different values). As first argued in [6], random oracles and related objects should be modeled as functions that can be queried in superposition of different inputs. (Namely, with a real hash function, an adversary can use a quantum circuit implementing SHA3 and can thereby query the function in superposition. The adversary could evaluate the sponge on the uniform superposition over all messages of a certain length, possibly helping him to, e.g., find a collision.) Thus, we do not know whether the sponge construction (and thus hash functions like SHA3) is collapsing (or at least collision-resistant).

Our contributions. In the present paper we tackle the question whether the sponge construction is collision-resistant and collapsing. We show:

- If the block function $\mathbf{f}$ is collision-resistant when restricted to the left and right half of its output and it is hard to find a zero-preimage of $\mathbf{f}$ (restricted to the right half of its output), then the sponge construction is collision resistant.
- If the block function $\mathbf{f}$ is collapsing when restricted to the left and right half of its output, respectively, and if it is hard to find a zero-preimage of $\mathbf{f}$ (restricted to the right half of its output), then the sponge construction is collapsing.

[1] For example, hash functions using the Merkle-Damgård construction are not well modeled as a random oracle. If we use $MAC(k, m) := H(k\|m)$ as a message authentication code (MAC) with key k, we have that MAC is secure (unforgeable) when H is a random oracle, but easily broken when H is a hash function built using the Merkle-Damgård construction.

[2] It is not called a compression function, since the domain and range of $\mathbf{f}$ are identical.

[3] [4] shows that the sponge construction is indifferentiable from a random oracle *in the classical setting*. Together with the fact that the random oracle is collision-resistant, collision-resistance of the sponge construction follows.

- If the block function $\mathbf{f}$ is a random oracle or a random permutation, then the sponge construction is collapsing.
- For a random block function $\mathbf{f}$, we give a quantum attack for actually finding collision in the sponge construction where the number of quantum queries to $\mathbf{f}$ matches the above bounds (in the case that the output length of the sponge is one block).

It should be stressed that we *do not* show that the sponge construction is collapsing (or even collision-resistant) if the block function $\mathbf{f}$ is an *efficiently invertible* random permutation. In this case, it is trivial to find zero-preimages by applying the inverse permutation to 0. This means that the present result cannot be directly used to show the security of, say, SHA3, because SHA3 uses an efficiently invertible permutation as block function. Our results apply to hash functions where the block function is not (efficiently) invertible, e.g., Gluon [2]. But we believe that our results are also a first step towards understanding the sponge construction for invertible block functions, and towards showing the post-quantum security of SHA3.

References

[1] Jean-Philippe Aumasson, Luca Henzen, Willi Meier, and María Naya-Plasencia. "Quark: A Lightweight Hash". In: *CHES 2010*. Vol. 6225. LNCS. Springer, 2010, pp. 1–15.

[2] Thierry P. Berger, Joffrey D'Hayer, Kevin Marquet, Marine Minier, and Gaël Thomas. "The GLUON Family: A Lightweight Hash Function Family Based on FCSRs". In: *Africacrypt 2012*. Berlin, Heidelberg: Springer, 2012, pp. 306–323.

[3] Guido Bertoni, J. Daemen, Michaël Peeters, and Gilles van Assche. *Sponge functions*. Ecrypt Hash Workshop, `http://sponge.noekeon.org/SpongeFunctions.pdf`. May 2007.

[4] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles van Assche. "On the Indifferentiability of the Sponge Construction". In: *Eurocrypt 2008*. Vol. 4965. LNCS. Berlin, Heidelberg: Springer, 2008, pp. 181–197.

[5] Andrey Bogdanov, Miroslav Knezevic, Gregor Leander, Deniz Toz, Kerem Varici, and Ingrid Verbauwhede. "SPONGENT: The Design Space of Lightweight Cryptographic Hashing". In: *IEEE Transactions on Computers* 62.10 (2013), pp. 2041–2053.

[6] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. "Random oracles in a quantum world". In: *Asiacrypt 2011*. Seoul, South Korea: Springer, 2011, pp. 41–69.

[7] Gilles Brassard, Peter Hoyer, and Alain Tapp. "Quantum algorithm for the collision problem". In: *arXiv preprint quant-ph/9705002* (1997).

[8] Scott Contini, Arjen K. Lenstra, and Ron Steinfeld. "VSH, an Efficient and Provable Collision-Resistant Hash Function". In: *Eurocrypt 2006*. Springer, 2006, pp. 165–182.

[9] Jian Guo, Thomas Peyrin, and Axel Poschmann. "The PHOTON Family of Lightweight Hash Functions". In: *Crypto 2011*. Springer, 2011, pp. 222–239.

[10] Shai Halevi and Silvio Micali. "Practical and Provably-Secure Commitment Schemes from Collision-Free Hashing". English. In: *Crypto '96*. Vol. 1109. LNCS. Springer, 1996, pp. 201–215.

[11] Andreas Hülsing, Joost Rijneveld, and Fang Song. "Mitigating Multi-target Attacks in Hash-Based Signatures". In: *PKC 2016*. Springer, 2016, pp. 387–416.

[12] National Institute of Standards and Technology (NIST). *Secure Hash Standard (SHS)*. FIPS PUBS 180-4. 2015.

[13] NIST. *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*. Draft FIPS 202. Available at `http://csrc.nist.gov/publications/drafts/fips-202/fips_202_draft.pdf`. 2014.

[14] Dominique Unruh. "Computationally binding quantum commitments". In: *Eurocrypt 2016*. LNCS. Springer, 2016, pp. 497–527.

[15] Dominique Unruh. "Collapse-binding quantum commitments without random oracles". In: *AsiaCrypt 2016*. Vol. 10032. LNCS. Springer, 2016, pp. 166–195.

[16] Mark Zhandry. "A note on the quantum collision and set equality problems". In: *Quantum Information & Computation* 15.7&8 (2015), pp. 557–567. URL: `http://www.rintonpress.com/xxqic15/qic-15-78/0557-0567.pdf`.