

MiniSHAKE

Guido Bertoni¹, Joan Daemen², Seth Hoeffert, Silvia Mella²,
Michaël Peeters³, Gilles Van Assche³ and Ronny Van Keer³

¹ Security Pattern, Italy

² Radboud University, The Netherlands

³ STMicroelectronics, Belgium

all@keccak.team

Abstract. The overwhelming majority of the KECCAK instances standardized and used in practice today make use of the KECCAK- $f[1600]$ permutation or reduced-round versions of it. There are a few use cases where using a smaller permutation width would be beneficial, notably when one has to hash many very short messages. To address this, the present note defines instances that use the KECCAK- $p[800, n_r = 12]$ permutation, collectively under the name MiniSHAKE.

Keywords: symmetric cryptography, hashing, Keccak

The performance of hash and extendable output functions is often studied in the scope of long inputs or long outputs, for which optimizing the execution time leads to more substantial benefits than for short inputs and outputs. In some cases, however, one needs to hash many short messages in batch, and the performance must be studied accordingly.

A particular use case lies in the scope of zero-knowledge proofs, and more specifically succinct non-interactive arguments of knowledge (SNARKs), where one proves the knowledge of the input to a hash function without revealing it. While this is traditionally an application where primitives working over large finite fields perform best, recent works have shown that these can also be efficient for more traditional binary-oriented functions evaluated in batches [1, 6, 14].

The cost of these proof systems depends on the Boolean circuit that represents the hash function. In the case of the family KECCAK- $p[b, n_r]$ of permutations, the number of Boolean operations is proportional to the width b and to the number of rounds n_r .

- Let us first discuss the impact of a narrower permutation, and let us illustrate this with a concrete example. Assume we need to evaluate a hash function on an input size of 512 bits and to get an output of 256 bits. SHAKE128 is based on KECCAK[$r = 1344, c = 256$] with a block size of $r = 1344$ bits and processes our input using a single call to KECCAK- $p[1600]$ [12]. Alternatively, KECCAK[$r = 544, c = 256$] can also make our input (and its padding) fit in a single block, hence using a single call to KECCAK- $p[800]$, thereby halving the number of Boolean operations compared to SHAKE128. (For this specific example, and sticking to $c = 256$ for 128-bit security, shrinking further down to KECCAK[$r = 144, c = 256$] is less useful, as processing a 512-bit input would require 4 calls to KECCAK- $p[400]$, but there may be other use cases where this permutation could come in handy.)
- Next, the number of rounds n_r is an important parameter that influences both performance and security, and it needs to be high enough to provide safety margin against possible progresses in attacks. Ideally, this choice is driven by cryptanalysis on round-reduced versions, but cryptanalysis requires hard work by cryptographic

experts. Fortunately, KECCAK has received quite a large amount of cryptanalysis since its publication. In the light of these publications, we feel we can confidently propose to reduce the number of rounds in KECCAK- $p[800]$ from the nominal 22 to 12 without compromising security. This is also the number of rounds chosen when defining TurboSHAKE [3].

1 Specifications of MiniSHAKE

MiniSHAKE is a family of eXtendable Output Functions (XOF) parameterized by their capacity c , where the capacity directly relates to the claimed security level as detailed in Section 2. We restrict the capacity c to multiples of 8 not greater than 512.

A given instance, denoted $\text{MiniSHAKE}[c]$, takes as input:

- a message M , a byte string of variable length, and
- a domain separation parameter D , a byte with a value in the range $[0\text{x}01, \dots, 0\text{x}7\text{F}]$ in hexadecimal.

As a XOF, the output of $\text{MiniSHAKE}[c]$ is unlimited, and the user can request as many output bits as desired. It can be used for traditional hashing simply by generating outputs of the desired digest size.

MiniSHAKE produces unrelated outputs on different tuples (c, M, D) . For a given capacity, the value D is meant to provide domain separation, that is, for two different values $D_1 \neq D_2$, $\text{MiniSHAKE}[c](\cdot, D_1)$ and $\text{MiniSHAKE}[c](\cdot, D_2)$ act as two independent functions of M . We believe the range of D to be sufficient to cover all use cases.

Users that do not require multiple instances can take as default $D = 0\text{x}01$.

Named instances In addition, we define:

- MiniSHAKE128 as $\text{MiniSHAKE}[c = 256]$, and
- MiniSHAKE256 as $\text{MiniSHAKE}[c = 512]$.

Procedure To compute $\text{MiniSHAKE}[c](M, D)$, proceed as follows. Let $R = 100 - c/8$ be the rate in bytes and f the KECCAK- $p[800, n_r = 12]$ permutation [12].

1. Input preparation
 - (a) Append to M the byte D , followed by the minimum number of bytes $0\text{x}00$ (possibly none) until $M' = M || D || 0\text{x}00^*$ has a length that is multiple of R bytes.
 - (b) Bitwise add (XOR) the byte $0\text{x}80$ into the last byte of M' .
 - (c) Cut M' into m blocks of R bytes each, i.e., $M' = M_1 || \dots || M_m$.
2. Absorbing phase
 - (a) Let $S = 0\text{x}00^{100}$.
 - (b) For each block M_i for $i = 1$ to m :
 - i. Let $S \leftarrow f(S \oplus (M_i || 0\text{x}00^{100-R}))$.
3. Squeezing phase
 - (a) Repeat as long as necessary:
 - i. Output the first R bytes of S .
 - ii. Let $S \leftarrow f(S)$.
 - (b) Truncate the output if longer than needed.

2 Security claim

We make a flat sponge claim [4] with c bits of claimed capacity in Claim 1. Informally, it means that MiniSHAKE shall offer the same security strength as a random oracle whenever that offers a strength below $c/2$ bits and a strength of $c/2$ bits in all other cases.

Claim 1 (Flat sponge claim [4]). *The success probability of any attack on MiniSHAKE[c] shall not be higher than the sum of that for a random oracle and $1 - e^{-\frac{N^2}{2^{c+1}}}$, with N the attack complexity in calls to KECCAK- $p[800, n_r = 12]$ or its inverse. We exclude from the claim weaknesses due to the mere fact that the function can be described compactly and can be efficiently executed, e.g., the so-called random oracle implementation impossibility [10], as well as properties that cannot be modeled as a single-stage game [13].*

The flat sponge claim covers all attacks against MiniSHAKE[c] up to a given security strength of $c/2$ bits. Informally, saying that a cryptographic function has a security strength of s bits means that no attacks exist with complexity N and success probability p such that $N/p < 2^s$ [11]. For more details on the implications of the claim, we refer to [5, Section 4.1].

3 Rationale

In this section, we exhibit the equivalence with KECCAK reduced to 12 rounds and motivate our security claim.

Equivalence Consider the sponge function on top of the KECCAK- $p[800, n_r = 12]$ permutation, with multi-rate padding pad10^*1 , capacity c and rate $r = 800 - c$, as defined in the FIPS 202 standard [12], and let us call it $\mathcal{MS}_c$ for short, i.e.,

$$\mathcal{MS}_c \triangleq \text{SPONGE}[\text{KECCAK-}p[800, n_r = 12], \text{pad10}^*1, r = 800 - c] .$$

Then, MiniSHAKE[c](M, D) is equivalent to $\mathcal{MS}_c(M || \text{unpad}(D))$, where

- D and each byte of M is interpreted as a string of 8 bits, from the least to the most significant bit of the byte;
- $\text{unpad}(D)$ removes the trailing bits ‘0’ of D , if any, then the last bit ‘1’ (e.g., $\text{unpad}(0x01)$ is the empty string, $\text{unpad}(0x0B) = \text{‘110’}$).

The reasoning is the same as for TurboSHAKE [3].

Security Changing the number of rounds in the underlying permutation from 22 as specified in FIPS 202 to 12 in MiniSHAKE implies a drastic reduction in safety margin. Still, MiniSHAKE is a reduced-round version of KECCAK and thereby directly benefits from all the cryptanalysis on the latter.

The cryptanalysis on KECCAK is a living activity, with new papers published every year. To keep track of these, we maintain a list of cryptanalysis results on our ciphers [2]. At the time of this writing, the best collision attack applicable to MiniSHAKE or to any KECCAK instance works only when the permutation is reduced to 6 rounds [15, 7, 18], and preimage attacks reach only 5 rounds [8, 19, 9]. Hence our proposal has a safety margin of 6 out of 12 rounds for collision and (second) preimage resistance.

Currently, the structural distinguisher that reaches the highest number of rounds is called SymSum and works on KECCAK reduced to 9 rounds [17] or even 10 rounds [16] although with a complexity that does not contradict MiniSHAKE’s security claim. This distinguisher considers self-symmetric strings of bits, that is, strings of the form $X || X || Y || Y ||$

... $||Z||Z$, where X , Y and Z are 32-bit strings. The SymSum distinguisher produces a set of self-symmetric strings such that the set of corresponding outputs through the function sums to a self-symmetric string.

Finally, as for TurboSHAKE, we limit the supported capacity to at most 512 bits, as we do not think that it makes much sense to claim more than 256 bits of security.

Acknowledgments

We wish to thank all the researchers who analyzed KECCAK and contributed to its understanding. This is what allowed us to confidently propose schemes based on round-reduced KECCAK.

References

- [1] Binius authors, *Binius: Binary proofs, blazing fast*, <https://www.binius.xyz/>.
- [2] Guido Bertoni, Joan Daemen, Seth Hoffert, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer, *Third-party cryptanalysis*, 2026, https://keccak.team/third_party.html.
- [3] Guido Bertoni, Joan Daemen, Seth Hoffert, Michaël Peeters, Gilles Van Assche, Ronny Van Keer, and Benoît Viguier, *TurboSHAKE*, Cryptology ePrint Archive, Paper 2023/342, 2023.
- [4] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche, *Cryptographic sponge functions*, SHA-3 competition (round 3) (2011).
- [5] Guido Bertoni, Joan Daemen, Michaël Peeters, Gilles Van Assche, Ronny Van Keer, and Benoît Viguier, *KangarooTwelve: Fast Hashing Based on Keccak-p*, ACNS, 2018, pp. 400–418.
- [6] Benedikt Bünz, Ron D. Rothblum, and William Wang, *Flock: Fast proving for batch boolean computations*, 2026.
- [7] Jian Guo, Guozhen Liu, Ling Song, and Yi Tu, *Exploring SAT for Cryptanalysis: (Quantum) Collision Attacks Against 6-Round SHA-3*, Asiacrypt, 2022, pp. 645–674.
- [8] Xiaoen Lin, Le He, Zhengrong Lu, Yantian Shen, Chongxu Ren, and Hongbo Yu, *Internal differential structure: preimage attacks on up to 5-round Keccak*, Des. Codes Cryptogr. **93** (2025), no. 9, 3761–3808.
- [9] Xiaoen Lin, Zhengrong Lu, Yantian Shen, Chongxu Ren, Hongbo Yu, Enming Dong, and Wenhao Wu, *Practical Preimage Attack on 5-Round Keccak[r=640, c=160]*, IACR Trans. Symmetric Cryptol. **2026** (2026), no. 1, 189–211.
- [10] Ueli M. Maurer, Renato Renner, and Clemens Holenstein, *Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology*, TCC, 2004, pp. 21–39.
- [11] Daniele Micciancio and Michael Walter, *On the Bit Security of Cryptographic Primitives*, Eurocrypt, 2018, pp. 3–28.
- [12] NIST, *Federal information processing standard 202, SHA-3 standard: Permutation-based hash and extendable-output functions*, August 2015, <http://dx.doi.org/10.6028/NIST.FIPS.202>.

- [13] Thomas Ristenpart, Hovav Shacham, and Thomas Shrimpton, *Careful with Composition: Limitations of the Indifferentiability Framework*, Eurocrypt, 2011, pp. 487–506.
- [14] Srinath Setty, *BinarySpartan: Spartan over binary fields*, Cryptology ePrint Archive, Paper 2026/1656, 2026.
- [15] Ling Song, Guohong Liao, and Jian Guo, *Non-Full Sbox Linearization: Applications to Collision Attacks on Round-Reduced Keccak*, Crypto, 2017, pp. 428–451.
- [16] Sahiba Suryawanshi, Shibam Ghosh, Dhiman Saha, and Prathamesh Ram, *Simple vs. vectorial: exploiting structural symmetry to beat the ZeroSum distinguisher*, Des. Codes Cryptogr. **93** (2025), no. 1, 133–174.
- [17] Sahiba Suryawanshi, Dhiman Saha, and Satyam Sachan, *New Results on the SymSum Distinguisher on Round-Reduced SHA3*, Africacrypt, 2020, pp. 132–151.
- [18] Zhongyi Zhang, Chengan Hou, and Meicheng Liu, *Probabilistic Linearization: Internal Differential Collisions in up to 6 Rounds of SHA-3*, Crypto, 2024, pp. 241–272.
- [19] ———, *Preimage Attacks on up to 5 Rounds of SHA-3 Using Internal Differentials*, Eurocrypt, 2025, pp. 333–363.