J Cryptol (2019) 32:895–940
https://doi.org/10.1007/s00145-018-9299-7

Journal of
CRYPTOLOGY

Beyond Conventional Security in Sponge-Based Authenticated Encryption Modes

Philipp Jovanovic
Decentralized and Distributed Systems Lab, École polytechnique fédérale de Lausanne, Lausanne, Switzerland
philipp.jovanovic@epfl.ch

Atul Luykx
Visa Research, Palo Alto, USA
aluykx@visa.com

Bart Mennink
Digital Security Group, Radboud University, Nijmegen, The Netherlands
b.mennink@cs.ru.nl

Yu Sasaki · Kan Yasuda
NTT Secure Platform Laboratories, Tokyo, Japan
sasaki.yu@lab.ntt.co.jp
yasuda.kan@lab.ntt.co.jp

Communicated by Kaisa Nyberg.

Received 9 August 2016 / Revised 24 May 2018
Online publication 15 June 2018

Abstract. The Sponge function is known to achieve $2^{c/2}$ security, where c is its capacity. This bound was carried over to its keyed variants, such as SpongeWrap, to achieve a $\min\{2^{c/2}, 2^{\kappa}\}$ security bound, with κ the key length. Similarly, many CAESAR competition submissions were designed to comply with the classical $2^{c/2}$ security bound. We show that Sponge-based constructions for authenticated encryption can achieve the significantly higher bound of $\min\{2^{b/2}, 2^{c}, 2^{\kappa}\}$, with $b > c$ the *permutation* size, by proving that the CAESAR submission NORX achieves this bound. The proof relies on rigorous computation of multi-collision probabilities, which may be of independent interest. We additionally derive a generic attack based on multi-collisions that matches the bound. We show how to apply the proof to five other Sponge-based CAESAR submissions: Ascon, CBEAM/STRIBOB, ICEPOLE, Keyak, and two out of the three PRIMATEs. A direct application of the result shows that the parameter choices of some of these submissions are overly conservative. Simple tweaks render the schemes considerably more efficient without sacrificing security. We finally consider the remaining one of the three PRIMATEs, APE, and derive a blockwise adaptive attack in the nonce-respecting setting with complexity $2^{c/2}$, therewith demonstrating that the techniques cannot be applied to APE.

Keywords. Authenticated encryption, CAESAR, Ascon, CBEAM, ICEPOLE, Keyak, NORX, PRIMATEs, STRIBOB, Multi-collisions.

1. Introduction

Authenticated encryption schemes, cryptographic functions that aim to simultaneously provide data privacy and integrity, have gained renewed attention in light of the CAESAR competition [25]. A common approach to building such schemes is to design a block cipher mode of operation, as in CCM [95], OCB1-3 [55,78,79], EAX [14], GCM [57], COPA [5], OTR [63], AEZ [48], and SCT [72]. Nevertheless a significant fraction of the CAESAR competition submissions use modes of operation for *permutations*.

Most of the permutation-based modes follow the basic Sponge design [16]: a state is maintained and regularly updated using a permutation. The state is divided into an outer part of r bits, through which the user enters or extracts data, and an inner part of c bits, which is out of the user's control. The rate r determines how much plaintext can be processed per permutation call, which gives an estimate of the algorithm's performance. Keccak, the eventual winner of the competition and now standardized as SHA-3 [35], internally uses the Sponge construction. The Sponge design also found adoption in the field of lightweight hash functions [24,45].

Security of the Sponge construction as a hash function follows from the fact that the user can only affect the outer state, hence adversaries only succeed with significant probability if they make on the order of $2^{c/2}$ permutation queries, as this many are needed to produce an inner state collision [16]. Bertoni et al. [17] proved tightness of this bound in the indifferentiability framework of Maurer et al. [56]. Keyed versions of the Sponge construction, such as KeyedSponge [20] and SpongeWrap [19], are proven up to a similar bound of 2^{c-a} (pseudorandom function security for the former and privacy and authenticity for the latter), assuming a limit of 2^a on online complexity, but are additionally restricted by the key size κ to 2^κ. The permutation-based CAESAR candidates are no exception and recommend parameters based on either the $2^{c/2}$ or 2^{c-a} bound, as shown in Table 1.

1.1. *Beyond Conventional Security*

Contrary to intuition, a wide range of permutation-based authenticated encryption schemes actually achieve *significantly* higher mode security: the privacy and authenticity bound on the total complexity can be improved from $\min\{2^{c/2}, 2^\kappa\}$ to $\min\{2^{(r+c)/2}, 2^c, 2^\kappa\}$. Intuitively, the improvement demonstrates that, *in the nonce-respecting setting*, inner collisions are not relevant to the adversary; only full state collisions are. We remark that in the nonce-reuse scenario [37,80] the privacy of the scheme can be broken [19], and for authenticity the old bounds hold at best.

The main proof in this work concerns NORX mode v1 and v2 [7,8], but we demonstrate its applicability to the CAESAR submissions Ascon v1 and v1.1 [33,34], CBEAM v1 [83,84],[1] ICEPOLE v1 and v2 [65,66], Keyak v1 [22],[2] two out of three PRIMATEs v1 and v1.02 [2,3], and STRIBOB v1 and v2 [81,85,86].[3] Additionally, we note that it

[1] CBEAM was withdrawn after an attack by Minaud [62], but we focus on modes of operation.

[2] Keyak v2 follows a different design approach.

[3] Both CBEAM and STRIBOB use the BLNK Sponge mode [82].

Table 1. Parameters and the achieved mode security levels of seven CAESAR submissions.

Scheme	Version	b	c	r	κ	τ	Security
Ascon	v1 [33]	320	192	128	96	96	96
		320	256	64	128	128	128
	v1.1 [34]	320	192	128	128	128	128
		320	256	64	128	128	128
CBEAM	v1 [84]	256	190	66	128	64	128
ICEPOLE	v1, v2 [65,66]	1280	254	1026	128	128	128
		1280	318	962	256	128	256
Keyak	v1 [22]	800	252	548	128..224	128	128..224
		1600	252	1348	128..224	128	128..224
NORX	v1 [7]	512	192	320	128	128	128
		1024	384	640	256	256	256
	v2 [8]	512	128	384	128	128	128
		1024	256	768	256	256	256
GIBBON/ HANUMAN	v1, v1.02 [2,3]	200	159	41	80	80	80
		280	239	41	120	120	120
STRIBOB	v1, v2 [85,86]	512	254	258	192	128	192

We remark that ICEPOLE v1,v2 consists of three configurations (two with security level 128 and one with security level 256) and Keyak v1 of four configurations (one with an 800-bit state and three with a 1600-bit state)

directly applies to SpongeWrap [19] and DuplexWrap [22], upon which Keyak v1 is built.

Our results imply that the initial submissions of these CAESAR candidates were overly conservative in choosing their parameters, since reducing c would have lead to the same bound. For instance, Ascon-128 could take $(c, r) = (128, 192)$ instead of $(256, 64)$, NORX64 (the proposed mode with 256-bit security) could increase its rate by 128 bits, and GIBBON-120 and HANUMAN-120 could increase their rate by a factor of 4, without affecting their mode security levels.

These observations only concern the *mode* security, where characteristics of the underlying permutation are set aside. Specifically, the concrete security of the underlying permutations plays a fundamental role in the choice of parameters. For instance, the authors of Ascon [33,34], NORX [7,8], and PRIMATEs [2,3] acknowledge that non-random properties of some of the underlying primitives exist. Furthermore, the authenticity bound degrades as a function of the number of forgery attempts f: $\min\{2^{(r+c)/2}, 2^c/f, 2^\kappa\}$. In practical applications, the amount of forgery attempts may be limited, but if this is not possible, caution must be taken. We refer to [75] for a discussion.

1.2. *Tightness of the Result*

The earlier version of this article by Jovanovic et al. [53] had a security bound of the form $\min\{2^{(r+c)/2}, 2^c/r, 2^\kappa\}$, showing a security loss logarithmic relative to the rate. This loss was, however, not justified by any existing attack; it arose as an artifact of naively bounding the probability of a multi-collision occurring in the outer state, where multiple evaluations of the underlying primitive map to the same outer value.

In this article, we thoroughly analyze multi-collisions and derive bounds on the size of multi-collisions for various possible choices of r and c. Most importantly, we can conclude that if $r \ll c$ or $r \gg c$, multi-collisions have no effect on the security. If $r \approx c$, the security loss approaches $\frac{1.4c}{\log_2 c-2}$, as opposed to the factor r loss from [53]. We refer to Table 2 for a comprehensive description of the bound. Note that for all schemes in Table 1, $r \ll c$ or $r \gg c$.

The rigorous analysis of multi-collisions relies on an application of Stirling's approximation and the Lambert W function. It is not only applicable to Sponge-based modes. For example, there are quite a few cryptographic schemes that have been attacked using multi-collisions, such as block-cipher-based hashing schemes [73], identification schemes [41], JH hash function [58], MDC-2 hash function [54], HMAC and ChopMD MAC [68], the LED block cipher [70], iterated Even-Mansour [32], and strengthened HMAC [88]. Multi-collisions have also influenced various security upper bounds. Typical examples are the indifferentiability proof for the ChopMD construction [27], the collision resistance proof for the Lesamnta-LW hash function [46], and the indistinguishability proof for RMAC [52], where the bound is $\mathcal{O}(2^n/n)$ due to the existence of n-collisions. The compression function proposed by Hirose et al. [47] has a similar type of bound. Finally, the recent line of research on the keyed Sponge and Duplex constructions [6,18,20,26,31,38,60,69] strongly relies on "multiplicities." Some of these security analyses can be improved using our rigorous analysis of multi-collisions.

For $r < c$, the old bound of [53] is dominated by $2^{(r+c)/2}$ and is in fact tight. The new bound improves over the one of [53] for $r \geq c$, and in this work we additionally show that the new bound is tight for all possible choices of (r, c). To this end, we present a multi-collision-based adversary that meets the bound proven in our analysis. The attack is described for a generalized Sponge construction that covers CBEAM, ICEPOLE, Keyak v1, NORX, and STRIBOB. Even for variants with the additional XOR of the secret key at the end, (Ascon, GIBBON, and HANUMAN, see Fig. 4), a similar adversary with slightly higher complexity can meet the bound. A comparison of the earlier bound of [53], the new bound, and the attack complexity for the case of $c = 256$ and $r \geq c$ is given in Fig. 1.

1.3. *APE*

One of the interesting questions triggered by the publication of [53] was regarding APE, the third of the PRIMATEs. In more detail, the schemes listed in Table 1 are proven to achieve a beyond $2^{c/2}$ security level against nonce-respecting adversaries, but the schemes are insecure against nonce-misusing adversaries. In contrast, APE is proven to achieve $2^{c/2}$ security in the nonce-reuse scenario [4], and it is of interest to investigate what security guarantees APE offers against nonce-respecting adversaries. In this work, we include an analysis of APE in this setting and show that there exists a nonce-respecting blockwise adaptive adversary that can break the privacy with a total complexity of about $2^{c/2}$. In other words, while APE is more robust against nonce-misusing adversaries up to common prefix, in the nonce-respecting setting the schemes listed in Table 1 achieve higher security. (We remark that the analysis in this work can be easily extended to the case of blockwise adaptive adversaries.)

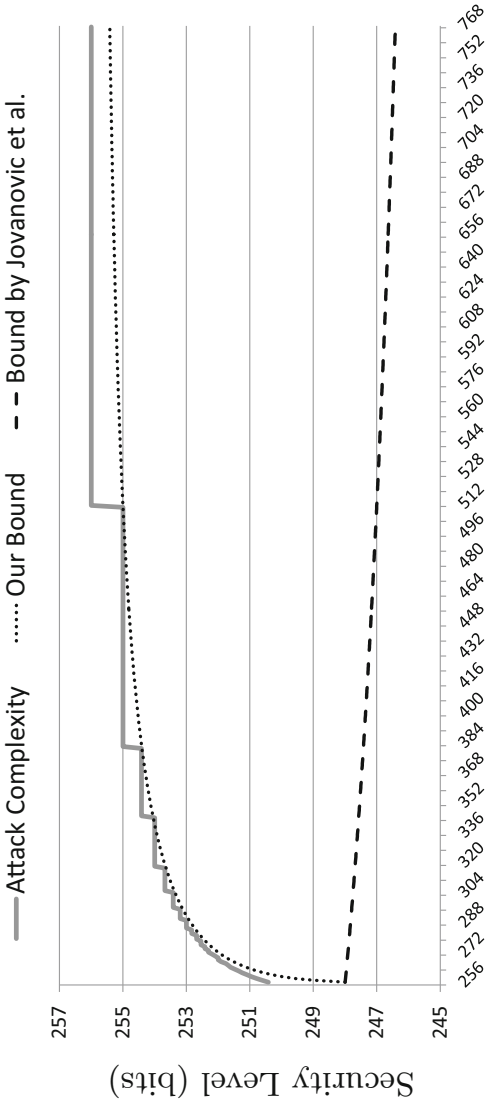

Fig. 1. Comparison of the bound by Jovanovic et al. [53], our improved bound, and the matching attack complexity for the case of $c = 256$ and $r > c$.

1.4. *Publication History and Subsequent Work*

An extended abstract of this article has appeared in the proceedings of ASIACRYPT 2014 [53]. This article is the full version of [53], and additionally includes the proofs that were absent in the proceedings version. New with respect to the full version of [53] are

(i) a more rigorous analysis of multi-collisions and the therewith induced improved security bound (Sect. 3),
(ii) the generic attack on Sponge-based authenticated encryption schemes demonstrating tightness of the bound (Sect. 5),
(iii) a proof that, unlike the schemes of Table 1, APE *does not* achieve beyond $2^{c/2}$ security in the nonce-respecting setting (Sect. 7).

Parts (i) and (ii) are due to Sasaki and Yasuda [90], with whom we have collaborated to combine their ideas for a complete analysis of the Sponge-based modes.

In response to the observations made in [53], the designers of Ascon and NORX have reconsidered their parameter choices. The new parameter choices are also listed in Table 1 and testify of a significant security gain for Ascon v1.1 [34] without sacrificing efficiency, and a significant efficiency gain for NORX v2 [8] without sacrificing security. The adjustments will make the schemes faster and more competitive. Mihajloska et al. [61] recently generalized the analysis of [53] to CAESAR submission π-Cipher [42,43], which is structurally different from NORX in the way it maintains state: a so-called "common internal state" is used throughout the evaluation.

From a more general perspective, the work has triggered analysis in the direction of high-efficiency full-state keyed Duplexes [31,60,89]. The result of Mennink et al. [60] on the full-state keyed Duplex has triggered the designers of Keyak to perform a major revision to their scheme. In more detail, Keyak v2 [23] is built on top of the "Motorist" mode, an alternative to the full-state keyed Duplex that was analyzed by Daemen et al. [31]. We remark that the results on the full-state keyed Sponges and Duplexes are more general than the target design in this work. The most important difference between [31,60] and our work is that we explicitly target nonce-based designs, and this allows for beyond $2^{c/2}$ security. The work has, to certain extent, furthermore triggered the use of permutations for nonce-reuse secure authenticated encryption schemes [29,44,59] beyond APE.

Parallel to the research on keyed Duplexes is the research on the keyed Sponges, i.e., keyed versions of the Sponge that only aim for authenticity. Bertoni et al. [18] introduced the original keyed Sponge. Chang et al. [26] suggested to put the key in the inner part of the Sponge. Andreeva et al. [6] formalized and improved the analysis of the outer- and inner-keyed sponges. The analysis was generalized to the full-state Sponge in [31,38,60,69], following upon ideas that date back to the donkeySponge [21]. Beyond authentication (and encryption), keyed versions of the Sponge have found applications in reseedable pseudorandom sequence generation [18,39].

1.5. *Outline*

We present our security model in Sect. 2. In Sect. 3, we perform an in-depth analysis of multi-collisions with respect to Sponges. A security proof for NORX is derived in Sect. 4. Tightness of the bound is proven in Sect. 5. In Sect. 6, we show that the proof of NORX generalizes to other CAESAR submissions, as well as to SpongeWrap and DuplexWrap. We consider the security of APE against nonce-respecting adversaries in Sect. 7. The work is concluded in Sect. 8, where we also discuss possible generalizations to Artemia [1].

2. Security Model

For $n \in \mathbb{N}$, let $\mathsf{Perm}(n)$ denote the set of all permutations on n bits. When writing $x \xleftarrow{\$} \mathcal{X}$ for some finite set $\mathcal{X}$, we mean that x gets sampled uniformly at random from $\mathcal{X}$. For $x \in \{0,1\}^n$, and $a, b \leq n$, we denote by $[x]^a$ and $[x]_b$ the a leftmost and b rightmost bits of x, respectively. For tuples (j,k), (j',k') we use lexicographical order: $(j,k) > (j',k')$ means that $j > j'$, or $j = j'$ and $k > k'$.

Let Π be an authenticated encryption scheme, with an encryption function $\mathcal{E}$ and a decryption function $\mathcal{D}$, where

$$(C, A) \longleftarrow \mathcal{E}_K(N; H, M, T) \qquad \text{and} \qquad M/\perp \longleftarrow \mathcal{D}_K(N; H, C, T; A).$$

Here, N denotes a nonce value, H a header, M a message, C a ciphertext, T a trailer, and A an authentication tag. The values (H, T) will be referred to as associated data. If verification is successful, then the decryption function $\mathcal{D}_K$ outputs M, and $\perp$ otherwise. The scheme Π is also determined by a set of parameters such as the key size, state size, and block size, but these are left implicit. In addition, we define \$ to be an ideal version of $\mathcal{E}_K$, where \$ returns $(C, A) \xleftarrow{\$} \{0,1\}^{|M|+\tau}$ for every query $(N; H, M, T)$.

We follow the convention in analyzing modes of operation for permutations by modeling the underlying permutations as being drawn uniformly at random from $\mathsf{Perm}(b)$, where b is a parameter determined by the scheme.

An adversary $\mathcal{A}$ is a probabilistic algorithm that has access to one or more oracles $\mathcal{O}$, denoted $\mathcal{A}^{\mathcal{O}}$. By $\mathcal{A}^{\mathcal{O}} = 1$ we denote the event that $\mathcal{A}$, after interacting with $\mathcal{O}$, outputs 1. We consider adversaries $\mathcal{A}$ that have unbounded computational power and whose complexity is solely measured by the number of queries made to their oracles. These adversaries have query access to (i) the underlying idealized permutations, (ii) $\mathcal{E}_K$ or its counterpart \$, and possibly (iii) $\mathcal{D}_K$. The key K is randomly drawn from $\{0,1\}^\kappa$ at the beginning of the security experiment. The security definitions below follow [11,37,51,77,80].

Privacy

Let $\mathbf{p}$ denote a list of idealized permutations, which Π may depend on. We define the advantage of an adversary $\mathcal{A}$ in breaking the privacy of Π as follows:

$$\mathbf{Adv}^{\mathrm{priv}}_{\Pi}(\mathcal{A}) = \left|\mathbf{Pr}_{\mathbf{p},K}\left(\mathcal{A}^{\mathbf{p}^{\pm},\mathcal{E}_K} = 1\right) - \mathbf{Pr}_{\mathbf{p},\$}\left(\mathcal{A}^{\mathbf{p}^{\pm},\$} = 1\right)\right|,$$

where the probabilities are taken over the random choices of $\mathbf{p}$, \$, K, and $\mathcal{A}$, if any. The fact that the adversary has access to both the forward and inverse permutations in $\mathbf{p}$ is denoted by $\mathbf{p}^{\pm}$. We assume that adversary $\mathcal{A}$ is nonce-respecting, which means that it never makes two queries to $\mathcal{E}_K$ or \$ with the same nonce. By $\mathbf{Adv}^{\mathrm{priv}}_{\Pi}(q_p, q_{\mathcal{E}}, \lambda_{\mathcal{E}})$ we denote the maximum advantage taken over all adversaries that query $\mathbf{p}^{\pm}$ at most q_p times, and that make at most $q_{\mathcal{E}}$ queries of total length (over all queries) at most $\lambda_{\mathcal{E}}$ blocks to $\mathcal{E}_K$ or \$. We remark that this privacy notion is also known as the *indistinguishability under chosen plaintext attack* (IND-CPA) security of an (authenticated) encryption scheme.

Integrity

As above, let $\mathbf{p}$ denote the list of underlying idealized permutations of Π. We define the advantage of an adversary $\mathcal{A}$ in breaking the integrity of Π as follows:

$$\mathbf{Adv}^{\mathrm{auth}}_{\Pi}(\mathcal{A}) = \mathbf{Pr}_{\mathbf{p},K}\left(\mathcal{A}^{\mathbf{p}^{\pm},\mathcal{E}_K,\mathcal{D}_K} \text{ forges}\right),$$

where the probability is taken over the random choices of $\mathbf{p}$, K, and $\mathcal{A}$, if any. We say that "$\mathcal{A}$ forges" if $\mathcal{D}_K$ ever returns a message other than $\perp$ on input of $(N; H, C, T; A)$ where (C, A) has never been output by $\mathcal{E}_K$ on input of a query $(N; H, M, T)$ for some M. We assume that adversary $\mathcal{A}$ is nonce-respecting, which means that it never makes two queries to $\mathcal{E}_K$ with the same nonce. Nevertheless, $\mathcal{A}$ is allowed to repeat nonces in decryption queries. By $\mathbf{Adv}^{\mathrm{auth}}_{\Pi}(q_p, q_{\mathcal{E}}, \lambda_{\mathcal{E}}, q_{\mathcal{D}}, \lambda_{\mathcal{D}})$ we denote the maximum advantage taken over all adversaries that query $\mathbf{p}^{\pm}$ at most q_p times, make at most $q_{\mathcal{E}}$ queries of total length (over all queries) at most $\lambda_{\mathcal{E}}$ blocks to $\mathcal{E}_K$, and at most $q_{\mathcal{D}}$ queries of total length at most $\lambda_{\mathcal{D}}$ blocks to $\mathcal{D}_K$.

3. Multi-Collisions

Consider the following game of balls and bins. Let $R \geq 1$ be the number of bins and σ the number of balls. The σ balls are thrown uniformly at random into the R bins. By $\mathsf{multcol}(R, \sigma, \rho)$ we denote a ρ-collision, namely the event that there exists a bin that contains ρ or more balls after all σ balls are thrown.

A folklore result [67, Theorem 3.1], [64, Lemma 5.1] states the following upper bound on the probability of a ρ-collision for $\rho \geq 2$:

$$\mathbf{Pr}\left(\mathsf{multcol}(R, \sigma, \rho)\right) \leq \frac{1}{R^{\rho-1}}\binom{\sigma}{\rho}, \tag{1}$$

where $R \geq 1$ and $\sigma \geq \rho$. Note that σ can be smaller or larger than R.

The bound of (1) involves a binomial coefficient and hence factorials. To evaluate these factorials we rely on Stirling's approximation. Formally, Stirling's approximation can be written as an inequality as [71]

$$x! \geq \sqrt{2\pi x}\left(\frac{x}{e}\right)^x \geq \sqrt{x}\left(\frac{x}{e}\right)^x, \tag{2}$$

where $\pi = 3.14\ldots$ and $e = 2.71\ldots$, which holds for all $x \geq 1$.

For the purpose of the paper we combine inequalities (1) and (2) in the following way. Let S be some positive number limiting the maximum value of σ, i.e., $\sigma \leq S$. From (1) and (2), we get

$$\mathbf{Pr}\left(\mathsf{multcol}(R, \sigma, \rho)\right) \leq \frac{1}{R^{\rho-1}}\frac{\sigma^\rho}{\rho!} \leq \left(\frac{S}{R}\right)^{\rho-1}\frac{\sigma}{\rho!} \tag{3}$$

$$\leq \left(\frac{S}{R}\right)^{\rho-1}\frac{\sigma}{\sqrt{\rho}\,(\rho/e)^\rho} = \left(\frac{eS}{\rho R}\right)^\rho \frac{R}{\sqrt{\rho}}\frac{\sigma}{S}. \tag{4}$$

This derivation is identical to that in [67, Theorem 3.1], [64, Lemma 5.1], be it with a slightly more accurate bound for $x!$. In the remainder of the section, we will introduce the Lambert W function in Sect. 3.1, and derive simplified bounds on $\mathbf{Pr}\left(\mathsf{multcol}(R, \sigma, \rho)\right)$ in Sect. 3.2.

Remark 1. The probability that $\mathsf{multcol}(R, \sigma, \rho)$ occurs can also be bounded using the Chernoff bound [28]. Consider any fixed bin, and for $i = 1, \ldots, \sigma$, denote

$$X_i = \begin{cases} 1 \text{ with probability } 1/R\,, \\ 0 \text{ with probability } 1 - 1/R. \end{cases}$$

Defining $X = \sum_{i=1}^{\sigma} X_i$ as the number of balls in that specific bin, the Chernoff bound states that for any $t > 0$ [64, Section 4.2],

$$\mathbf{Pr}\left(X \geq \rho\right) \leq \mathbf{Pr}\left(e^{tX} \geq e^{t\rho}\right) \leq \frac{\mathbf{Ex}\left(e^{tX}\right)}{e^{t\rho}}.$$

As in our case the events X_i are mutually independent,

$$\mathbf{Ex}\left(e^{tX}\right) = \prod_{i=1}^{\sigma} \mathbf{Ex}\left(e^{tX_i}\right) = \left(1 + \frac{e^t - 1}{R}\right)^\sigma.$$

One therefore finds, for any $t > 0$,

$$\mathbf{Pr}\left(\mathsf{multcol}(R, \sigma, \rho)\right) \leq R \cdot \frac{\left(1 + \frac{e^t - 1}{R}\right)^\sigma}{e^{t\rho}}. \tag{5}$$

Looking ahead, in our applications we will need an upper bound of this term of the form σ/S, where ρ is a function of R and S. The bound of (4) is more suited for that.

Likewise, specific variants of (5) such as

$$\mathbf{Pr}\left(\mathsf{multcol}(R,\sigma,(1+\delta)\sigma/R)\right) \leq R\cdot\left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\sigma/R},$$

obtained from (5) by setting $\rho = (1+\delta)\sigma/R$ and $t = \ln(1+\delta)$ [67, Theorem 4.1], [64, Theorem 4.4], do not directly seem to give an improved result for our specific parameter setting.

An alternative approach to bound the probability that $\mathsf{multcol}(R,\sigma,\rho)$ occurs, is via the first and second moments, as done by Raab and Steger [74]. In detail, Raab and Steger demonstrate that $\mathbf{Pr}\left(\mathsf{multcol}(R,\sigma,\rho(R,\sigma))\right) = o(1)$ for various parameter settings and choices of ρ as a function of R and σ [74, Theorem 1]. This approach, as well as the related approaches in the field of cryptography [10,49], again does not fit our targeted upper bound.

3.1. *Lambert W Function*

Stirling's approximation contains a "self-exponential" function x^x, and we will need to solve equations of the form

$$\xi^{\xi} = d \tag{6}$$

for variable ξ. For this purpose we utilize the Lambert W function [71]. Consider the function $f(w) = we^w$ defined for complex numbers w. Then, the Lambert W function is the inverse relation of f. More precisely, $Z = W(Z)e^{W(Z)}$ is the defining equation for W, and Eq. (6) can be solved, using W, as

$$\xi = e^{W(D)}, \tag{7}$$

where $D := \ln d$ [30].

In this work, we can restrict the domain of W to real numbers $X \geq -1/e$ and the range to real numbers $W(X) \geq -1$, and we focus on the principal branch W_{p}, which is a single-valued function. Hoornar and Hassani [50] derived the following inequality on $W_{\mathrm{p}}(X)$ for any $X \geq e$:

$$W_{\mathrm{p}}(X) \leq \ln X - \ln\ln X + \ln(1+e^{-1}).$$

Back to (6), when ξ is restricted to real numbers, the solution (7) becomes

$$\xi = e^{W_{\mathrm{p}}(D)} \leq e^{\ln D - \ln\ln D + \ln(1+e^{-1})} = \frac{(1+e^{-1})D}{\ln D}. \tag{8}$$

It should be emphasized that this bound is valid only under the condition $D \geq e$, or equivalently, $d \geq e^e$.

3.2. *Bounding Multi-Collision Probability*

We will derive Sponge-oriented bounds for ρ. In more detail, consider parameters b, r, c such that $b = r + c$, write $R = 2^r$, and $S = \min\{2^{b/2}, 2^c\}$. We will derive choices for ρ (depending on r and c), such that the probability of a multi-collision of (1) is bounded by σ/S.

Lemma 1. *Write $b = r + c$, $R = 2^r$, and $S = \min\{2^{b/2}, 2^c\}$. Assume that $c \geq 13$. Then,*

$$\mathbf{Pr}\left(\mathsf{multcol}(R, \sigma, \rho(r, c))\right) \leq \frac{\sigma}{S},$$

where

$$\rho(r,c) := \begin{cases} \left\lceil e2^{(c-r)/2} \right\rceil & \text{if } r \leq c/5 \text{ (case (i))}, \\ \left\lceil 3.4 \cdot 2^{(c-r)/2} \right\rceil & \text{if } c/5 < r \leq c - 2\log_2 c \text{ (case (ii))}, \\ \left\lceil 8.0 \cdot 2^{(c-r)/2} \right\rceil & \text{if } c - 2\log_2 c < r \leq c - 2\log_2 c + 7.2 \text{ (case (iii))}, \\ \left\lceil \frac{0.7(5r-c)}{2\log_2(5r-c) + r - c - 8} \right\rceil & \text{if } c - 2\log_2 c + 7.2 < r < c \text{ (case (iv))}, \\ \left\lceil \frac{1.4r}{\log_2 r + r - c - 2} \right\rceil & \text{if } c \leq r \leq c + e\log_2 c - e\beta \text{ (case (v))}, \\ \left\lceil \frac{r}{r-c} \right\rceil & \text{if } c + e\log_2 c - e\beta < r < 2c \text{ (case (vi))}, \\ 2 & \text{if } 2c \leq r \text{ (case (vii))}, \end{cases}$$

where $\beta := \log_2 e + \log_2 \log_2 e$.

The proof of Lemma 1 is constructive, and the bounds for ρ are derived constructively rather than simply proven to hold. However, the reasoning is structurally different for the cases where $r < c$ (cases (i-iv)) and for the cases where $r \geq c$ (cases (v-vii)).

Proof of Lemma 1(i-iv). For the case $r < c$, our basic strategy is to bound $\mathbf{Pr}\left(\mathsf{multcol}(R, \sigma, \rho)\right)$ by σ/S, where $S = 2^{b/2}$, by means of setting

$$\rho := \left\lceil \theta 2^{(c-r)/2} \right\rceil$$

for sufficiently large parameter θ. Note that, by the generalized pigeonhole principle, $2^{(c-r)/2}$ is the minimum value of ρ when σ reaches $S = 2^{b/2}$.

Assume that $\rho \geq eS/R = e2^{b/2}/2^r = e2^{(c-r)/2}$, i.e., $\theta \geq e$. Then, (4) becomes

$$\mathbf{Pr}\left(\mathsf{multcol}(R,\sigma,\rho)\right) \leq \left(\frac{eS}{\rho R}\right)^{\rho} \frac{R}{\sqrt{\rho}}\frac{\sigma}{S} \leq \left(\frac{e2^{b/2}}{\theta 2^{(c-r)/2} 2^r}\right)^{\theta 2^{(c-r)/2}} \frac{2^r}{\sqrt{\theta 2^{(c-r)/2}}}\frac{\sigma}{S}$$
$$= \left(\frac{e}{\theta}\right)^{\theta 2^{(c-r)/2}} \frac{2^{(5r-c)/4}}{\sqrt{\theta}}\frac{\sigma}{S}, \tag{9}$$

and we start from this equation for the cases (i-iv).

Case (i): $\boldsymbol{r \leq c/5}$. Since $r \leq c/5$, we have $2^{(5r-c)/4} \leq 1$. Therefore, the bound of (9) satisfies

$$\left(\frac{e}{\theta}\right)^{\theta 2^{(c-r)/2}} \frac{2^{(5r-c)/4}}{\sqrt{\theta}}\frac{\sigma}{S} \leq \left(\frac{e}{\theta}\right)^{\theta 2^{(c-r)/2}} \frac{1}{\sqrt{e}}\frac{\sigma}{S} \leq \left(\frac{e}{\theta}\right)^{\theta 2^{(c-r)/2}} \frac{\sigma}{S}. \tag{10}$$

We can choose the minimum $\theta := e = 2.71\ldots$ so that $(e/\theta)^{\theta 2^{(c-r)/2}} = 1$, which implies that (10) is upper bounded by σ/S, as desired. The size of a multi-collision is bounded by

$$\rho = \left\lceil e2^{(c-r)/2} \right\rceil.$$

Case (ii): $\boldsymbol{c/5 < r \leq c - 2\log_2 c}$. If $r > c/5$, then the factor $2^{(5r-c)/4}$ in the bound (9) becomes larger than 1, and we need to somehow cancel this factor by increasing the value of θ. The factor $\sqrt{\theta}$ is too small for this purpose, and hence we aim at the factor $(e/\theta)^{\theta 2^{(c-r)/2}}$. The following observation suggests that we need to increase the value of θ by only a small amount, as long as $r \leq c - 2\log_2 c$:

□

Claim. If $r \leq c - 2\log_2 c$, then we have $2^{(c-r)/2} \geq (5r-c)/4$.

Proof of claim. Direct computation yields $2^{(c-r)/2} \geq 2^{\log_2 c} = c \geq (5r-c)/4$. □

Hence, it remains to ensure that $(e/\theta)^{\theta} \leq 1/2$. For this we set $\theta := 3.4$, so that $(\mathrm{e}/\theta)^{\theta} = (2.71\ldots/3.4)^{3.4} = 0.46\ldots \leq 1/2$. Then, the bound of (9) satisfies

$$\left(\frac{e}{\theta}\right)^{\theta 2^{(c-r)/2}} \frac{2^{(5r-c)/4}}{\sqrt{\theta}}\frac{\sigma}{S} \leq \left(\frac{1}{2}\right)^{2^{(c-r)/2}} \frac{2^{(5r-c)/4}}{\sqrt{3.4}}\frac{\sigma}{S} \leq \frac{\sigma}{S}$$

by the above claim. The size of a multi-collision is bounded by

$$\rho = \left\lceil 3.4 \cdot 2^{(c-r)/2} \right\rceil.$$

Case (iii): $\boldsymbol{c - 2\log_2 c < r \leq c - 2\log_2 c + 7.2}$. This is a technical case to bridge a gap between case (ii) and case (iv). The reason behind the constant 7.2 will become clear in the analysis of case (iv).

Set $\delta := r - c + 2\log_2 c$, so that $\delta \in (0,\ 7.2]$. Then we have $2^{(c-r)/2} = 2^{\log_2 c - \delta/2} = 2^{-\delta/2}c \geq 2^{-\delta/2}(5r-c)/4$. Hence, the bound of (9) satisfies

$$\left(\frac{e}{\theta}\right)^{\theta 2^{(c-r)/2}} \frac{2^{(5r-c)/4}}{\sqrt{\theta}}\frac{\sigma}{S} \leq \left(\frac{e}{\theta}\right)^{\theta 2^{-\delta/2}(5r-c)/4} \frac{2^{(5r-c)/4}}{\sqrt{\theta}}\frac{\sigma}{S} = \left(2\left(\frac{e}{\theta}\right)^{\theta 2^{-\delta/2}}\right)^{(5r-c)/4} \frac{1}{\sqrt{\theta}}\frac{\sigma}{S},$$

and we want to ensure that $(e/\theta)^{\theta 2^{-\delta/2}} \leq 1/2$. Given the previous constant 3.4 and the new factor $2^{-\delta/2}$, let us put $\theta := 3.4 \cdot 2^{0.17\delta}$ and define $\varphi(\zeta) := \left(e/(3.4 \cdot 2^{0.17\zeta})\right)^{3.4\cdot 2^{0.17\zeta} 2^{-\zeta/2}} = \left(e/(3.4 \cdot 2^{0.17\zeta})\right)^{3.4\cdot 2^{-0.33\zeta}}$ that is defined for real numbers $\zeta \in [0,\ 7.2]$. It remains to show the following:

Claim. We have $\varphi(\zeta) \leq 0.495 \leq 1/2$.

Proof of claim. The derivative of φ is computed as

$$\varphi'(\zeta) = 3.4 \cdot 2^{-0.33\zeta} \ln 2 \left(\frac{e}{3.4 \cdot 2^{0.17\zeta}}\right)^{3.4\cdot 2^{-0.33\zeta}} \left(0.33 \ln(3.4 \cdot 2^{0.17\zeta}) - 0.5\right),$$

and equation $\varphi'(\zeta) = 0$ has a unique solution $\zeta_0 := \log_2(e^{0.5/0.33}/3.4)/0.17 = 2.47\ldots$. Direct computation shows that the second-order derivative $\varphi''(\zeta)$ is positive for $\zeta \in [0,\ 7.2]$, implying that $\varphi(\zeta)$ is minimum at ζ_0. We already know that $\varphi(0) = (e/3.4)^{3.4} = 0.46\cdots \leq 1/2$, and so we end the proof by computing $\varphi(7.2) = \left(e/(3.4 \cdot 2^{0.17\cdot 7.2})\right)^{3.4\cdot 2^{-0.33\cdot 7.2}} = 0.495\ \ldots \leq 1/2$. □

The value of θ grows as δ increases, from 3.4 to $3.4 \cdot 2^{0.17\cdot 7.2} = 7.94\ldots \leq 8.0$.

Case (iv): $\boldsymbol{c - 2\log_2 c + 7.2 < r < c}$. The value of θ needs to increase as r approaches to c, and in general θ cannot be bounded by a constant but is rather a function of r and c. The Lambert W function can handle such a case, yielding a fairly sharp bound.

Put $\varphi(\zeta) := (e/\zeta)^{\zeta 2^{(c-r)/2}} 2^{(5r-c)/4}$, defined for real numbers $\zeta \geq e$. Then $\varphi(\zeta)$ is strictly decreasing. This leads us to solve equation $\varphi(\zeta) = 1$ to determine the value of θ, as a function of r and c. Let ζ_0 be a solution of this equation. Then the equality $\varphi(\zeta_0) = 1$ becomes $(\zeta_0/e)^{\zeta_0 2^{(c-r)/2}} = 2^{(5r-c)/4}$, which is equivalent to

$$\left(\frac{\zeta_0}{e}\right)^{\zeta_0/e} = 2^{(5r-c)/\left(4e2^{(c-r)/2}\right)}. \tag{11}$$

We can solve Eq. (11) for ζ_0 using formula (7) by the Lambert W function, via setting $\xi = \zeta_0/e$ and $d = 2^{(5r-c)/4e2^{(c-r)/2}}$, as

$$\frac{\zeta_0}{e} = e^{W(E)}, \tag{12}$$

where $E := \ln 2^{(5r-c)/4e2^{(c-r)/2}} = (5r-c)/\big(4e2^{(c-r)/2}\log_2 e\big)$. Now we want to use inequality (8) for the function W_{p} to upper bound ζ_0, but for that purpose we need to make sure that $E \geq e$. It is exactly for this reason that we have chosen the constant 7.2, as shown by the following claim.

Claim. Let $c \geq 13$. The inequality

$$E \geq 2.75 \geq e \tag{13}$$

holds for all $r \in [c - 2\log_2 c + 7.2,\ c\,]$. (The condition $c \geq 13$ is to make the range of r non-empty.)

Proof of claim. We have

$$E = \frac{5r-c}{4e2^{(c-r)/2}\log_2 e} \geq \frac{5(c-2\log_2 c+7.2)-c}{4e2^{\log_2 c-3.6}\log_2 e} = \frac{2^{2.6}(2c-5\log_2 c+18)}{ec\log_2 e},$$

which leads us to study the function $\psi(\omega) := 2^{2.6}(2\omega - 5\log_2\omega + 18)/e\omega\log_2 e$ defined for real numbers $\omega \geq 13$. We compute the derivative of $\psi(\omega)$ as $\psi'(\omega) = 2^{2.6}(5\log_2\omega + 18 - 5\log_2 e)/e\omega^2\log_2 e$, and equation $\psi'(\omega) = 0$ has a unique solution $\omega_0 = 2^{18/5}e = 32.9\ldots$, at which $\psi(\omega_0) = 2.75\ldots$. Since the second-order derivative $\psi''(\omega) = 2^{2.6}(-10\log_2\omega + 36 + 15\log_2 e)/e\omega^3\log_2 e$ is positive for $13 \leq \omega < 2^{18/5}e^{3/2} = 54.3\ldots$, we conclude that $\psi(\omega) \geq 2.75$ for all $\omega \geq 13$, and hence $E \geq 2.75 \geq e$ for all $c \geq 13$ and $r \in [c - \log_2 c + 7.2,\ c\,]$, as desired. □

Now we can apply inequality (8) to our case (12) to get

$$\frac{\zeta_0}{e} \leq \frac{(1+e^{-1})E}{\ln E} = \frac{(1+e^{-1})(5r-c)/\big(2e2^{(c-r)/2}\big)}{2\log_2(5r-c)+r-c-4-2\beta}.$$

We compute $(1+e^{-1})/2 = 0.68\ldots$ and $-4-2\beta = -7.94\ldots$. Set

$$\theta := 0.7(5r-c)/2^{(c-r)/2}/\big(2\log_2(5r-c)+r-c-8\big)$$

so that $\theta \geq \zeta_0$ and $\varphi(\theta) \leq 1$ (recall that $\varphi(\zeta)$ is strictly decreasing). In addition, since $E \geq e$ from condition (13), we have $\theta/e \geq \zeta_0/e = e^{W_{\mathrm{p}}(E)} \geq e$, meaning $\theta \geq e^2 = 7.38\ldots$. Therefore, the bound of (9) satisfies

$$\Big(\frac{e}{\theta}\Big)^{\theta 2^{(c-r)/2}} \frac{2^{(5r-c)/4}}{\sqrt{\theta}}\frac{\sigma}{S} = \frac{\varphi(\theta)}{\sqrt{\theta}}\frac{\sigma}{S} \leq \frac{1}{\sqrt{e^2}}\frac{\sigma}{S} \leq \frac{\sigma}{S}.$$

We thus obtain

$$\rho = \lceil \theta 2^{(c-r)/2} \rceil = \left\lceil \frac{0.7(5r-c)}{2\log_2(5r-c)+r-c-8} \right\rceil.$$

□

Proof of Lemma 1(v-vii). The analysis is different from the cases (i-iv) in the sense that we do not need to rely on the factor $\sqrt{\rho}$ in Stirling's approximation, and the Lambert W function is more easily applicable.

Case (v): $\boldsymbol{c \le r \le c + e\log_2 c - e\beta}$. Consider bound (4). We have $R = 2^r$ and $S = 2^c$, and hence,

$$\mathbf{Pr}\left(\mathsf{multcol}(R,\sigma,\rho)\right) \le \left(\frac{eS}{\rho R}\right)^{\rho} \frac{R}{\sqrt{\rho}} \frac{\sigma}{S} = \left(\frac{e2^c}{\rho 2^r}\right)^{\rho} \frac{2^r}{\sqrt{\rho}} \frac{\sigma}{S} = \left(\frac{e}{\rho 2^{r-c}}\right)^{\rho} \frac{2^r}{\sqrt{\rho}} \frac{\sigma}{S}.$$

Put $\varphi(\zeta) := (e/\zeta 2^{r-c})^{\zeta} 2^r$ that is defined for real numbers $\zeta \ge 2$. We see that $\varphi(\zeta)$ is strictly decreasing, and at $\zeta = 2$ we have $\varphi(2) = (e/2)^2 2^{2c-r}$ which is greater than 1 because $2c \ge r$. So we would like to solve equation $\varphi(\zeta) = 1$. Let ζ_0 be a solution of this equation. This means that $(\zeta_0 2^{r-c}/e)^{\zeta_0} = 2^r$, which is equivalent to

$$\left(\frac{\zeta_0 2^{r-c}}{e}\right)^{\zeta_0 2^{r-c}/e} = 2^{r 2^{r-c}/e}. \tag{14}$$

To apply (7) to solve (14), set $\xi = \zeta_0 2^{r-c}/e$ and $d = 2^{r2^{r-c}/e}$. We obtain

$$\frac{\zeta_0 2^{r-c}}{e} = e^{W(G)},$$

where $G := \ln 2^{r2^{r-c}/e} = r2^{r-c}(\ln 2)/e$. As $r \ge c \ge 13 \ge 11$, we have $G \ge 11 \cdot (\ln 2)/e = 2.80\ldots \ge e$. Using (8),

$$\frac{\zeta_0 2^{r-c}}{e} = e^{W_{\mathrm{p}}(G)} \le \frac{(1+e^{-1})G}{\ln G} = \frac{(1+e^{-1})r2^{r-c}/e}{\log_2 r + r - c - \beta},$$

where $\beta = \log_2 e + \log_2 \log_2 e = 1.97\ldots$. Since $(1+e^{-1}) = 1.36\ldots$, we can set

$$\zeta_0 \le \rho := \left\lceil \frac{1.4r}{\log_2 r + r - c - 2} \right\rceil. \tag{15}$$

Case (vi): $\boldsymbol{c + e\log_2 c - e\beta < r < 2c}$. Technically, the bound of case (v) is valid only for $r \le 2c$. To obtain bounds for $r \ge 2c$ we perform a different kind of

analysis. We do not start with (4) but go back further to (3), and consider a simplified bound

$$\rho := \left\lceil \frac{r}{r-c} \right\rceil. \tag{16}$$

The intuition behind (16) is as follows. The "folklore" approach to obtaining a ρ-collision on r-bit values takes about $2^{(\rho-1)r/\rho}$ trials. Suzuki et al. showed that even under this amount of trials, the probability of finding a ρ-collision is actually quite low, about $1/\rho!$ [91,92]. Inspired by this, we consider equation $2^c = 2^{(\rho-1)r/\rho}$. Solving this equation for variable ρ yields $\rho = r/(r-c)$, as desired.

As we will show, the bound (16) "works" not only for $r \geq 2c$ but for all $r > c$. Moreover, it turns out that (16) is actually better than (15) for a large part of $r \in (c,\ 2c\,]$, except where $r \approx c$. □

Claim. Let $r > c$. For ρ of (16), we have $\mathbf{Pr}\left(\mathsf{multcol}(R, \sigma, \rho)\right) \leq \sigma/S$.

Proof of claim. We go back to (3). Set $R = 2^r$ and $S = 2^c$. We have

$$\begin{aligned}
\mathbf{Pr}\left(\mathsf{multcol}(R, \sigma, \rho)\right) &\leq \left(\frac{S}{R}\right)^{\rho-1} \frac{\sigma}{\rho!} = \left(\frac{2^c}{2^r}\right)^{\lceil r/(r-c)\rceil - 1} \frac{\sigma}{\rho!} \\
&\leq \left(\frac{1}{2^{r-c}}\right)^{r/(r-c)-1} \frac{\sigma}{\rho!} \\
&= \left(\frac{1}{2^{r-c}}\right)^{c/(r-c)} \frac{\sigma}{\rho!} = \left(\frac{1}{2^c}\right)\frac{\sigma}{\rho!} = \frac{1}{\rho!}\frac{\sigma}{S} \leq \frac{\sigma}{S},
\end{aligned}$$

as desired. □

Claim. Let $r > c \geq 11$. Then, (16) is better (smaller) than (15) if $r > c + e\log_2 r - e\beta$.

Proof of claim. Define the function

$$\Delta_c(u) := \frac{(1+e^{-1})u}{u - c + \log_2 u - \beta} - \frac{u}{u-c}$$

whose domain is the real numbers $u \in (c,\ 2c\,]$ with $c \geq 11$ and $\beta = \log_2 e + \log_2 \log_2 e = 1.97\ldots$. Then equation $\Delta_c(u) = 0$ becomes $u = c + e\log_2 u - e\beta$, whose solution we denote by u_0. We differentiate Δ_c with respect to u as

$$\frac{\partial \Delta_c}{\partial u} = \frac{(1+e^{-1})(-c + \log_2 u - \beta - \log_2 e)}{(u - c + \log_2 u - \beta)^2} + \frac{c}{(u-c)^2},$$

and at $u = u_0$ we have

$$\left.\frac{\partial \Delta_c}{\partial u}\right|_{u=u_0} = \frac{u_0 - e\log_2 e}{(1+e)(u_0 - c)^2}$$

using $u_0 = c + e\log_2 u_0 - e\beta$. We see that $u_0 \geq e\log_2 e = 3.92\ldots$ because $r > c \geq 11$. □

Note that $c + e\log_2 r - e\beta > c + e\log_2 c - e\beta$, making the distinction between this case (vi) and the previous case (v) clear.

Case (vii): $\mathbf{2}c \leq r$. In this case, we can use the reasoning of case (vi), with $\rho = 2$ by (16). □

4. NORX

We introduce NORX at a level required for the understanding of the security proof and refer to Aumasson et al. [7,8] for the formal specification. Let p be a permutation on b bits. All b-bit state values are split into an outer part of r bits and an inner part of c bits. We denote the key size of NORX by κ bits, the nonce size by ν bits, and the tag size by τ bits. The header, message, and trailer can be of arbitrary length and are padded using 10^*1-padding to a length of a multiple of r bits. Throughout, we denote the r-bit header blocks by $H_1, \ldots, H_u$, message blocks by $M_1, \ldots, M_v$, ciphertext blocks by $C_1, \ldots, C_v$, and trailer blocks by $T_1, \ldots, T_w$.

Unlike other permutation-based schemes, NORX allows for parallelism in the encryption part, which is described using a parameter $D \in \{0, \ldots, 255\}$ corresponding to the number of parallel chains. Specifically, if $D \in \{1, \ldots, 255\}$ NORX has D parallel chains, and if $D = 0$ it has v parallel chains, where v is the block length of M or C.

NORX consists of five proposed parameter configurations: NORX W-R-D for $(W, R, D) \in \{(64, 4, 1), (32, 4, 1), (64, 6, 1), (32, 6, 1), (64, 4, 4)\}$. The parameter R denotes the number of rounds of the underlying permutation p, and W denotes the word size which we use to set $r = 10W$ and $c = 6W$. The default key and tag size are $\kappa = \nu = 4W$. The corresponding parameters for the two different choices of W, 64 and 32, are given in Table 1.

Although NORX starts with an initialization function init which requires the parameters (D, R, τ) as input, as soon as our security experiment starts, we consider (D, R, τ) fixed and constant. Hence, we can view init as a function that maps (K, N) to $(K \| N \| 0^{b-\kappa-\nu}) \oplus \mathsf{const}$, where const is irrelevant to the mode security analysis of NORX, and will be ignored in the remaining analysis.

After init is called, the header H is compressed into the rate, then the state is branched into D states (if necessary), the message blocks are encrypted in a streaming way, the D states are merged into one state (if necessary), the trailer is compressed, and finally the tag A is computed. All rounds are preceded with a domain separation constant XORed into the capacity: `01` for header compression, `02` for message encryption, `04` for trailer compression, and `08` for tag generation. If $D \neq 1$, domain separators `10` and

20 are used for branching and merging, along with pairwise distinct lane indices id_k for $k = 1, \ldots, D$ (if $D = 1$ we write $id_1 = 0$). In Fig. 2 we depict NORX for $D = 1$ and $D = 2$.

The privacy of NORX is proven in Sect. 4.1 and the integrity in Sect. 4.2. In both proofs we consider an adversary that makes q_p permutation queries and $q_{\mathcal{E}}$ encryption queries of total length $\lambda_{\mathcal{E}}$. In the proof of integrity, the adversary can additionally make $q_{\mathcal{D}}$ decryption queries of total length $\lambda_{\mathcal{D}}$. To aid the analysis, we compute the number of permutation calls made via the $q_{\mathcal{E}}$ encryption queries. The exact same computation holds for decryption queries with the parameters defined analogously.

Consider a query to $\mathcal{E}_K$, consisting of u header blocks, v message blocks, and w trailer blocks. We denote its corresponding state values by

$$\left(s^{\text{init}};\ s_0^H, \ldots, s_u^H;\ \begin{bmatrix} s_{1,0}^M, \ldots, s_{1,v_1}^M \\ \vdots \qquad\quad \vdots \\ s_{D,0}^M, \ldots, s_{D,v_D}^M \end{bmatrix};\ s_0^T, \ldots, s_w^T;\ s^{\text{tag}} \right), \tag{17}$$

as outlined in Fig. 2. Here, $\sum_{k=1}^{D} v_k = v$. If there are no branching and merging phases, i.e., $D = 1$, then the state values corresponding to the branching and merging, $\{s_{1,0}^M, \ldots, s_{D,0}^M\}$ and s_0^T, are left out of the tuple. Note that the length of this tuple equals the number of primitive calls made in this encryption query, as every state value corresponds to the input of exactly one primitive call. A simple calculation shows that if the jth$\mathcal{E}_K$ query is of length $u + v + w$ blocks, it results in $u + v + w + 3$ state values if $D = 1$, in $u + v + w + D + 4$ state values if $D > 1$, and in $u + 2v + w + 4$ state values if $D = 0$.[4] We denote the number of state values by $\sigma_{\mathcal{E},j}$, where the dependence on D is suppressed as D does not change during the security game. In other words, $\sigma_{\mathcal{E},j}$ denotes the number of primitive calls in the jth query to $\mathcal{E}_K$. Furthermore, we define $\sigma_{\mathcal{E}}$ to be the total number of primitive evaluations via the encryption queries, and find that

$$\sigma_{\mathcal{E}} := \sum_{j=1}^{q_{\mathcal{E}}} \sigma_{\mathcal{E},j} \leq \begin{cases} 2\lambda_{\mathcal{E}} + 4q_{\mathcal{E}}, \text{ if } D = 0\,, \\ \lambda_{\mathcal{E}} + 3q_{\mathcal{E}}, \text{ if } D = 1\,, \\ \lambda_{\mathcal{E}} + (D+4)q_{\mathcal{E}}, \text{ if } D > 1. \end{cases} \tag{18}$$

This bound is rather tight. Particularly, for $D = 0$ an adversary can meet this bound by only making queries without header and trailer. For queries to $\mathcal{D}_K$ we define $\sigma_{\mathcal{D},j}$ and $\sigma_{\mathcal{D}}$ analogously.

[4] For $D = 0$, the original specification dictates an additional $10^{b-2}1$-padding for every complete message block. This means that lanes $1, \ldots, v-1$ consist of two rounds. We do not take this padding into account, noting that it is unnecessary for the security analysis.

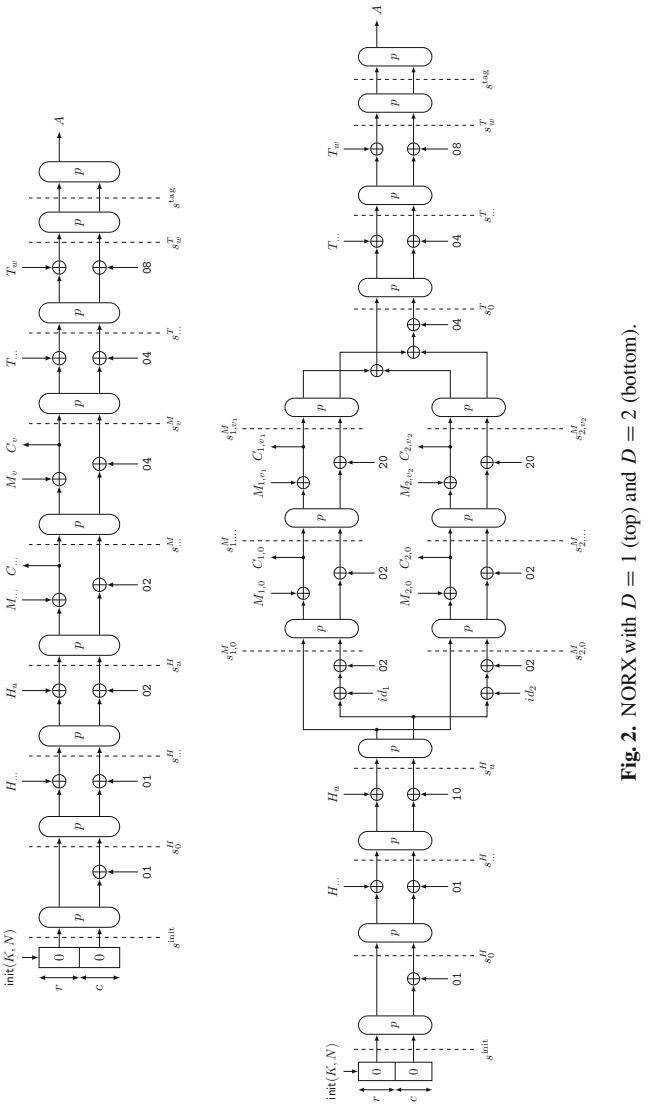

Fig. 2. NORX with $D = 1$ (top) and $D = 2$ (bottom).

4.1. *Privacy of NORX*

Theorem 1. *Let $\Pi = (\mathcal{E}, \mathcal{D})$ be NORX based on an ideal underlying primitive p. Then,*

$$\mathbf{Adv}^{\mathrm{priv}}_{\Pi}(q_p, q_{\mathcal{E}}, \lambda_{\mathcal{E}}) \leq \frac{3(q_p + \sigma_{\mathcal{E}})^2}{2^{b+1}} + \frac{\sigma_{\mathcal{E}}}{\min\{2^{b/2}, 2^c\}} + \frac{2\rho q_p}{2^c} + \frac{q_p + \sigma_{\mathcal{E}}}{2^\kappa},$$

where $\sigma_{\mathcal{E}}$ is defined in (18), and where $\rho = \rho(r, c)$ is the function defined in Lemma 1.

Theorem 1 can be interpreted as implying that NORX provides privacy security as long as the total complexity $q_p + \sigma_{\mathcal{E}}$ does not exceed $\min\{2^{b/2}, 2^\kappa\}$ and the total number of primitive queries q_p, also known as the offline complexity, does not exceed $2^c/\rho$. The presence of the term ρ makes the bound a bit unclear; in Table 2 we give the main implication of this bound for the various possible values of r and c as outlined in Lemma 1. See Table 1 for the security level of the various parameter choices of NORX: for NORX v1 [7], we are concerned with case (vi), where $\rho = \lceil 2.5 \rceil = 3$ for both $b \in \{512, 1024\}$; for NORX v2 [8], we are in case (vii), where $\rho = 2$.

The proof is based on the observation that NORX is indistinguishable from a random scheme as long as there are no collisions among the (direct and indirect) evaluations of p. Due to uniqueness of the nonce, state values from evaluations of $\mathcal{E}_K$ collide with probability approximately $1/2^b$. Regarding collisions between direct calls to p and calls via $\mathcal{E}_K$: while these may happen with probability about $1/2^c$, they turn out not to significantly influence the bound. The latter is demonstrated in part using the principle of multiplicities [18]: roughly stated, the maximum number of state values with the same outer part. We use Lemma 1 to bound multiplicities. The formal security proof is more detailed. Furthermore, we remark that, at the cost of readability and simplicity of the proof, the bound could be improved by a constant factor.

Proof. Consider any adversary $\mathcal{A}$ with access to either $(p^{\pm}, \mathcal{E}_K)$ or $(p^{\pm}, \$)$ and whose goal is to distinguish these two worlds. For brevity, we write

$$\mathbf{Adv}^{\mathrm{priv}}_{\Pi}(\mathcal{A}) = \Delta_{\mathcal{A}}(p^{\pm}, \mathcal{E}_K; p^{\pm}, \$). \tag{19}$$

We start by replacing $p^{\pm}$ by a random function to simplify analysis. This is done with a "URP-URF" switch [13], in which we make a transition from $p^{\pm}$ to a primitive $f^{\pm}$ defined as follows (as done by Andreeva et al. [4]).

The primitive $f^{\pm}$ maintains an initially empty list $\mathcal{F}$ of query/response tuples (x, y) where the set of domain and range values are denoted by $\mathrm{dom}(\mathcal{F})$ and $\mathrm{rng}(\mathcal{F})$, respectively. For a forward query $f(x)$ with $x \in \mathrm{dom}(\mathcal{F})$, the value in $\{y \mid (x, y) \in \mathcal{F}\}$ which occurs lexicographically first is returned. For a new forward query $f(x)$, the response y is randomly drawn from $\{0, 1\}^b$, then the tuple (x, y) is added to $\mathcal{F}$. The description for f^{-1} is similar. We let abort denote the event that a new query $f(x)$ results in a value y where y is already in $\mathrm{rng}(\mathcal{F})$, or a new query $f^{-1}(y)$ results in a value x where x is already in $\mathrm{dom}(\mathcal{F})$.

Table 2. High-level security bounds of Theorem 1.

	Case	Range	Security	Note
Essentially ideal	(i)	$r \leq c/5$	$\min\{2^{b/2}/e, 2^c, 2^\kappa\}$	
Slightly worse	(ii)	$c/5 < r \leq c - 2\log_2 c$	$\min\{2^{b/2}/3.4, 2^c, 2^\kappa\}$	
	(iii)	$c - 2\log_2 c < r \leq c - 2\log_2 c + 7.2$	$\min\{2^{b/2}/8.0, 2^c, 2^\kappa\}$	
Reaching $\frac{1.4c}{\log_2 c - 2}$	(iv)	$c - 2\log_2 c + 7.2 < r < c$	$\min\{2^{b/2}, 2^c/\alpha, 2^\kappa\}$	$\alpha = \frac{0.7(5r-c)}{2\log_2(5r-c)+r-c-8}$
	(v)	$c \leq r \leq c + e\log_2 c - e\beta$	$\min\{2^{b/2}, 2^c/\alpha, 2^\kappa\}$	$\alpha = \frac{1.4r}{\log_2 r+r-c-2}$
Gradually recovering	(vi)	$c + e\log_2 c - e\beta < r < 2c$	$\min\{2^{b/2}, 2^c/\alpha, 2^\kappa\}$	$\alpha = \frac{r}{r-c}$
Optimal	(vii)	$2c \leq r$	$\min\{2^{b/2}, 2^c/2, 2^\kappa\}$	

Here, $\beta = \log_2 e + \log_2 \log_2 e$

By applying the triangle inequality, we have

$$\begin{aligned}\Delta_{\mathcal{A}}(p^{\pm}, \mathcal{E}_K; p^{\pm}, \$) \leq \Delta_{\mathcal{A}}(f^{\pm}, \mathcal{E}_K; f^{\pm}, \$) + \Delta_{\mathcal{A}}(p^{\pm}, \mathcal{E}_K; f^{\pm}, \mathcal{E}_K) \\ + \Delta_{\mathcal{A}}(p^{\pm}, \$; f^{\pm}, \$). \end{aligned} \tag{20}$$

The two rightmost terms are bounded above by the maximum advantage of any adversary distinguishing $p^{\pm}$ and $f^{\pm}$ in at most $q_p + \sigma_{\mathcal{E}}$ queries. Since $p^{\pm}$ and $f^{\pm}$ are identical until abort, by the Fundamental Lemma of Game Playing [12,13] we have that the two rightmost terms are in turn bounded by $\binom{q_p+\sigma_{\mathcal{E}}}{2}/2^b \leq (q_p + \sigma_{\mathcal{E}})^2/2^{b+1}$, hence

$$\Delta_{\mathcal{A}}(p^{\pm}, \mathcal{E}_K; p^{\pm}, \$) \leq \Delta_{\mathcal{A}}(f^{\pm}, \mathcal{E}_K; f^{\pm}, \$) + \frac{(q_p + \sigma_{\mathcal{E}})^2}{2^b}. \tag{21}$$

We restrict our attention to $\mathcal{A}$ with oracle access to $(f^{\pm}, F)$, where $F \in \{\mathcal{E}_K, \$\}$. Without loss of generality, we can assume that the adversary only queries full blocks and that no padding rules are involved since the padding rules are injective, allowing the proof to carry over to the case of fractional blocks with 10^*1-padding.

We introduce some terminology. Queries to $f^{\pm}$ are denoted (x_i, y_i) for $i = 1, \ldots, q_p$, while queries to F are written as elements $(N_j; H_j, M_j, T_j; C_j, A_j)$ for $j = 1, \ldots, q_{\mathcal{E}}$. If $F = \mathcal{E}_K$, the state values are denoted as in (17), subscripted with a j:

$$\left(s_j^{\text{init}};\ s_{j,0}^H, \ldots, s_{j,u}^H;\ \begin{bmatrix} s_{j,1,0}^M, \ldots, s_{j,1,v_1}^M \\ \vdots \quad\quad \vdots \\ s_{j,D,0}^M, \ldots, s_{j,D,v_D}^M \end{bmatrix};\ s_{j,0}^T, \ldots, s_{j,w}^T;\ s_j^{\text{tag}} \right). \tag{22}$$

If the structure of (22) is irrelevant we refer to the tuple as $(s_{j,1}, \ldots, s_{j,\sigma_{\mathcal{E},j}})$, where we use the convention to list the elements of the matrix column-wise. In this case, we write $\text{parent}(s_{j,k})$ to denote the state value that lead to $s_{j,k}$, with $\text{parent}(s_{j,1}) := \varnothing$ and $\text{parent}(s_{j,0}^T) := (s_{j,1,v_1}^M, \ldots, s_{j,D,v_D}^M)$. We remark that the characteristic structure of NORX, with the D parallel states, only becomes relevant in the two technical lemmas that will be used at the end of the proof. We point out that $s_{j,1}$ corresponds to the initial state value of the evaluation, which requires special attention throughout the remainder of the proof.

We define two collision events, guess and hit. Let $i \in \{1, \ldots, q_p\}$, $j, j' \in \{1, \ldots, q_{\mathcal{E}}\}$, $k \in \{1, \ldots, \sigma_{\mathcal{E},j}\}$, and $k' \in \{1, \ldots, \sigma_{\mathcal{E},j'}\}$:

$$\begin{aligned} \mathsf{guess}(i; j, k) &\equiv x_i = s_{j,k}\,, \\ \mathsf{hit}(j, k; j', k') &\equiv \text{parent}(s_{j,k}) \neq \text{parent}(s_{j',k'}) \ \wedge\ s_{j,k} = s_{j',k'}. \end{aligned}$$

Event $\mathsf{guess}(i; j, k)$ corresponds to a primitive call in an encryption query hitting a direct primitive query, or vice versa, while $\mathsf{hit}(j, k; j', k')$ corresponds to non-trivial primitive calls colliding in encryption queries. We write $\mathsf{guess} = \vee_{i;j,k} \mathsf{guess}(i; j, k)$, $\mathsf{hit} = \vee_{j,k;j',k'} \mathsf{hit}(j, k; j', k')$, and set $\mathsf{event} = \mathsf{guess} \vee \mathsf{hit}$.

The remainder of the proof is divided as follows. In Lemma 2 we prove that $(f^{\pm}, \mathcal{E}_K)$ and $(f^{\pm}, \$)$ are identical until event occurs. In other words, by applying the Fundamental

Lemma of Game Playing [12,13],

$$\Delta_{\mathcal{A}}(f^{\pm}, \mathcal{E}_K; f^{\pm}, \$) \leq \mathbf{Pr}\left(\mathcal{A}^{f^{\pm}, \mathcal{E}_K} \text{ sets event}\right). \tag{23}$$

Then, in Lemma 3 we bound this term by

$$\frac{q_p\sigma_{\mathcal{E}} + \sigma_{\mathcal{E}}^2/2}{2^b} + \frac{\sigma_{\mathcal{E}}}{\min\{2^{b/2}, 2^c\}} + \frac{2\rho q_p}{2^c} + \frac{q_p + \sigma_{\mathcal{E}}}{2^{\kappa}}, \tag{24}$$

where $\rho = \rho(r, c)$ is the function defined in Lemma 1. Noting that $\frac{q_p\sigma_{\mathcal{E}} + \sigma_{\mathcal{E}}^2/2}{2^b} \leq \frac{(q_p + \sigma_{\mathcal{E}})^2}{2^{b+1}}$, this completes the proof via equations (19, 21, 23). □

Lemma 2. *The outputs of* $(f^{\pm}, \mathcal{E}_K)$ *and* $(f^{\pm}, \$)$ *are identically distributed until* event *occurs.*

Proof. The outputs of $f^{\pm}$ are sampled independently and uniformly at random in $(f^{\pm}, \$)$. This holds in the real world as well, unless a query to $f^{\pm}$ collides with an $f^{\pm}$ query made via $\mathcal{E}_K$. Therefore, until guess occurs, the outputs of $f^{\pm}$ are distributed identically in both worlds. Furthermore, $f^{\pm}$'s outputs are independent of the distinguisher's query history, hence, assuming all past queries were identically distributed across worlds, a query to $f^{\pm}$ will not change the fact that both worlds are identically distributed, until guess occurs.

Let N_j be a new nonce used in the F-query $(N_j; H_j, M_j, T_j)$, with corresponding ciphertext and authentication tag (C_j, A_j). Denote the query's state values as in (22). Let u, v, and w denote the number of padded header blocks, padded message blocks, and padded trailer blocks, respectively.

Consider the jth query. By the definition of \$, in the ideal world we have $(C_j, A_j) \xleftarrow{\$} \{0,1\}^{|M_j|+\tau}$. We will prove that (C_j, A_j) is identically distributed in the real world, under the assumption that guess $\vee$ hit has not yet occurred. Denote the message blocks of M_j by $M_{j,k,\ell}$ for $k = 1, \ldots, D$ and $\ell = 1, \ldots, v_k$.

We know that $s^H_{j,u}$ is new and that $f(s^H_{j,u})$ does not collide with any other f-query because otherwise event would have occurred. Since $s^M_{j,k,0} = f(s^H_{j,u}) \oplus id_k$ we conclude that $s^M_{j,k,0}$ is new for $k = 1, \ldots, D$, as, again, event would be set otherwise. Similarly, $s^M_{j,k,\ell}$ is new for all $\ell > 0$. The ciphertext blocks $C_{j,k,\ell}$ are computed as

$$C_{j,k,\ell} = M_{j,k,\ell} \oplus [f(s^M_{j,k,\ell-1})]^r.$$

As the state value $s^M_{j,k,\ell-1}$ has not been evaluated by f before (neither directly nor indirectly via an encryption query), $f(s^M_{j,k,\ell-1})$ outputs a uniformly random value from $\{0,1\}^b$, hence $C_{j,k,\ell} \xleftarrow{\$} \{0,1\}^r$. We remark that similar reasoning shows that a ciphertext block corresponding to a truncated message block is uniformly randomly drawn as well,

yet from a smaller set. The fact that $A_j \xleftarrow{\$} \{0,1\}^\tau$ follows the same reasoning, using that s_j^{tag} is a new input to f. Thus, $A_j = [f(s_j^{\text{tag}})]^\tau \xleftarrow{\$} \{0,1\}^\tau$. □

Looking at the reasoning of the proof of Lemma 2 above, we notice that if event has not yet occurred, then each state value in an F-query is sampled independently and uniformly at random. In particular, once the adversary fixes the inputs to an F-query, each state value in that F-query is independent of the adversary's input, and independent of each other. Furthermore, the inner part of those state values are never released to the adversary, hence the adversary's future queries are independent of the inner parts of the state values. Hence, we have the following result:

Corollary 1. *Until* event *occurs, the state values in an $\mathcal{E}_K$ query are distributed independently and uniformly from each other and from the adversary's input to that $\mathcal{E}_K$ query. Furthermore, the inner parts of the state values in all $\mathcal{E}_K$ queries are distributed independently and uniformly from each other and from all of the adversary's oracle-inputs, until* event *occurs.*

Lemma 3. $\mathbf{Pr}\left(\mathcal{A}^{f^{\pm},\mathcal{E}_K} \text{ sets event}\right) \leq \frac{q_p\sigma_{\mathcal{E}} + \sigma_{\mathcal{E}}^2/2}{2^b} + \frac{\sigma_{\mathcal{E}}}{\min\{2^{b/2}, 2^c\}} + \frac{2\rho q_p}{2^c} + \frac{q_p + \sigma_{\mathcal{E}}}{2^\kappa}$, *where $\rho = \rho(r,c)$ is the function defined in Lemma 1.*

Proof. Consider the adversary interacting with $(f^{\pm}, \mathcal{E}_K)$, and let $\mathbf{Pr}\,(\mathsf{guess} \vee \mathsf{hit})$ denote the probability we aim to bound. For $i \in \{1, \ldots, q_p\}$, define

$$\mathsf{key}(i) \quad \equiv \quad [x_i]^\kappa = K,$$

and $\mathsf{key} = \vee_i\, \mathsf{key}(i)$, which corresponds to a primitive query hitting the key. Let $j \in \{1, \ldots, q_{\mathcal{E}}\}$ and $k \in \{1, \ldots, \sigma_{\mathcal{E},j}\}$, and consider any threshold $\rho \geq 1$, then define

$$\mathsf{multi}(j,k) \quad \equiv \quad \Big[\max_{\alpha \in \{0,1\}^r} \left|\left\{j' \leq j, 1 < k' \leq k \;:\; \alpha \in \{[s_{j',k'}]^r, [f(s_{j',k'})]^r\}\right\}\right| \Big] > \rho.$$

Event $\mathsf{multi}(j,k)$ is used to bound the number of states that collide in the outer part. Note that state values $s_{j',1}$ are not considered here as they will be covered by key. We define $\mathsf{multi} = \mathsf{multi}(q_{\mathcal{E}}, \sigma_{\mathcal{E},q_{\mathcal{E}}})$, which is a monotone event. By basic probability theory,

$$\mathbf{Pr}\,(\mathsf{guess} \vee \mathsf{hit}) \leq \mathbf{Pr}\,(\mathsf{guess} \vee \mathsf{hit} \mid \neg(\mathsf{key} \vee \mathsf{multi})) + \mathbf{Pr}\,(\mathsf{key} \vee \mathsf{multi})\,. \quad (25)$$

In the remainder of the proof, we bound these probabilities as follows (a formal explanation of the proof technique is given in "Appendix"): we consider the ith forward or inverse primitive query (for $i \in \{1, \ldots, q_p\}$) or the kth state of the jth construction query (for $j \in \{1, \ldots, q_{\mathcal{E}}\}$ and $k \in \{1, \ldots, \sigma_{\mathcal{E},j}\}$), and bound the probability that this evaluation makes $\mathsf{guess} \vee \mathsf{hit}$ satisfied, under the assumption that this query does not

set $\mathsf{key} \vee \mathsf{multi}$ and also that $\mathsf{guess} \vee \mathsf{hit} \vee \mathsf{key} \vee \mathsf{multi}$ has not been set before. For the analysis of $\mathbf{Pr}\,(\mathsf{key} \vee \mathsf{multi})$ a similar technique is employed.

Event guess. This event can be set in the ith primitive query (for $i = 1, \ldots, q_p$) or in any state evaluation of the jth construction query (for $j = 1, \ldots, q_{\mathcal{E}}$). Denote the state values of the jth construction query as in (22). Consider any evaluation, assume this query does not set $\mathsf{key} \vee \mathsf{multi}$ and assume that $\mathsf{guess} \vee \mathsf{hit} \vee \mathsf{key} \vee \mathsf{multi}$ has not been set before. Firstly, note that $x_i = s_j^{\text{init}}$ for some i, j would imply $\mathsf{key}(i)$ and hence invalidate our assumption. Therefore, we can exclude s_j^{init} from further analysis on guess. For $i = 1, \ldots, q_p$, let $j_i \in \{1, \ldots, q_{\mathcal{E}}\}$ be the number of encryption queries made before the ith primitive query. Similarly, for $j = 1, \ldots, q_{\mathcal{E}}$, denote by $i_j \in \{1, \ldots, q_p\}$ the number of primitive queries made before the jth encryption query.

- Consider a primitive query (x_i, y_i) for $i \in \{1, \ldots, q_p\}$, which may be a forward or an inverse query, and assume it has not been queried to $f^{\pm}$ before. If it is a forward query x_i, by $\neg\mathsf{multi}$ there are at most ρ state values s with $[x_i]^r = [s]^r$, and thus $x_i = s$ with probability at most $\rho/2^c$. Here, we remark that the inner part of s is unknown to the adversary and it guesses it with probability at most $1/2^c$, as established by Corollary 1. A slightly more complicated reasoning applies for inverse queries. Denote the query by y_i. By $\neg\mathsf{multi}$ there are at most ρ state values s with $[y_i]^r = [f(s)]^r$, hence, using Corollary 1 again, $y_i = f(s)$ with probability at most $\rho/2^c$. If y_i equals $f(s)$ for any of these states, then $x_i = s$, otherwise $x_i = s$ with probability at most $\sum_{j=1}^{j_i} \sigma_{\mathcal{E},j}/2^b$. Therefore, the probability that guess is set via a direct query is at most $\frac{q_p \rho}{2^c} + \sum_{i=1}^{q_p} \sum_{j=1}^{j_i} \frac{\sigma_{\mathcal{E},j}}{2^b}$;
- Next, consider the probability that the jth construction query sets guess, for $j \in \{1, \ldots, q_{\mathcal{E}}\}$. For simplicity, first consider $D = 1$, hence the message is processed in one lane and we can use state labeling $(s_{j,1}, \ldots, s_{j,\sigma_{\mathcal{E},j}})$. We range from $s_{j,2}$ to $s_{j,\sigma_{\mathcal{E},j}}$ (recall that $s_{j,1} = s_j^{\text{init}}$ can be excluded) and consider the probability that this state sets guess assuming it has not been set before. Let $k \in \{2, \ldots, \sigma_{\mathcal{E},j}\}$. The state value $s_{j,k}$ equals $f(s_{j,k-1}) \oplus v$, where v is some value determined by the adversarial input prior to the evaluation of $f(s_{j,k-1})$, including input from (H_j, M_j, T_j) and constants serving as domain separators. By assumption, $\mathsf{guess} \vee \mathsf{hit}$ has not been set before, and $f(s_{j,k-1})$ is thus randomly drawn from $\{0,1\}^b$. It hits any x_i ($i \in \{1, \ldots, i_j\}$) with probability at most $i_j/2^b$. Next, consider the general case $D > 1$. We return to the labeling of (22). A complication occurs for the branching states $s_{j,1,0}^M, \ldots, s_{j,D,0}^M$ and the merging state $s_{j,0}^T$. Starting with the branching states, these are computed from $s_{j,u}^H$ as

$$\begin{pmatrix} s_{j,1,0}^M \\ \vdots \\ s_{j,D,0}^M \end{pmatrix} = f(s_{j,u}^H) \oplus \begin{pmatrix} v_1 \\ \vdots \\ v_D \end{pmatrix},$$

where $v_1, \ldots, v_D$ are some distinct values determined by the adversarial input prior to the evaluation of the jth construction query. These are distinct by the XOR of the lane numbers $id_1, \ldots, id_D$. Any of these nodes equals x_i for $i \in \{1, \ldots, q_p\}$ with

probability at most $i_j D/2^b$. Finally, for the merging node $s^T_{j,0}$ we can apply the same analysis, noting that it is derived from a sum of D new f-evaluations. Concluding, the jth construction query sets guess with probability at most $i_j \sigma_{\mathcal{E},j}/2^b$ (we always have in total at most $\sigma_{\mathcal{E},j}$ new state values). Summing over all $q_{\mathcal{E}}$ construction queries, we get $\sum_{j=1}^{q_{\mathcal{E}}} i_j \sigma_{\mathcal{E},j}/2^b$.

Concluding,

$$\mathbf{Pr}\left(\mathsf{guess} \mid \neg(\mathsf{key} \vee \mathsf{multi})\right) \leq \frac{q_p \rho}{2^c} + \sum_{i=1}^{q_p} \sum_{j=1}^{j_i} \frac{\sigma_{\mathcal{E},j}}{2^b} + \sum_{j=1}^{q_{\mathcal{E}}} \frac{i_j \sigma_{\mathcal{E},j}}{2^b} = \frac{q_p \rho}{2^c} + \frac{q_p \sigma_{\mathcal{E}}}{2^b}.$$

Here we use that $\sum_{i=1}^{q_p} \sum_{j=1}^{j_i} \sigma_{\mathcal{E},j} + \sum_{j=1}^{q_{\mathcal{E}}} \sum_{k=1}^{\sigma_{\mathcal{E},j}} i_j \sigma_{\mathcal{E},j} = q_p \sigma_{\mathcal{E}}$, which follows from a simple counting argument.

Event hit. We again employ ideas of guess, and particularly that as long as $\mathsf{guess} \vee \mathsf{hit}$ is not set, we can consider all new state values (except for the initial states) to be randomly drawn from a set of size 2^b. Particularly, we can refrain from explicitly discussing the branching and merging nodes (the detailed analysis of guess applies) and label the states as $(s_{j,1}, \ldots, s_{j,\sigma_{\mathcal{E},j}})$. Clearly, $s_{j,1} \neq s_{j',1}$ for all j, j' by uniqueness of the nonce. Any state value $s_{j,k}$ for $k > 1$ (at most $\sigma_{\mathcal{E}} - q_{\mathcal{E}}$ in total) hits an initial state value $s_{j',1}$ only if $[s_{j,k}]^\kappa = K$, which happens with probability at most $\sigma_{\mathcal{E}}/2^\kappa$, assuming $s_{j,k}$ is generated randomly. Finally, any two other states $s_{j,k}, s_{j',k'}$ for $k, k' > 1$ collide with probability at most $\binom{\sigma_{\mathcal{E}} - q_{\mathcal{E}}}{2}/2^b$. Concluding, $\mathbf{Pr}\left(\mathsf{hit} \mid \neg(\mathsf{key} \vee \mathsf{multi})\right) \leq \binom{\sigma_{\mathcal{E}}}{2}/2^b + \sigma_{\mathcal{E}}/2^\kappa$.

Event key. For $i \in \{1, \ldots, q_p\}$, the query sets $\mathsf{key}(i)$ if $[x_i]^\kappa = K$, which happens with probability $1/2^\kappa$ (assuming it did not happen in queries $1, \ldots, i-1$). The adversary makes q_p attempts, and hence $\mathbf{Pr}\left(\mathsf{key}\right) \leq q_p/2^\kappa$.

Event multi. Event multi can be related to $\mathsf{multcol}$ of Sect. 3, in the following way. Consider any new state value $s_{j,k-1}$; then it contributes to the bin α if $[f(s_{j,k-1})]^r = \alpha$ or $[s_{j,k}]^r = [f(s_{j,k-1}) \oplus v]^r = \alpha$. If a threshold ρ needs to be exceeded for some α, at least $\rho/2$ of them are *either* of the first kind *or* of the second kind. The event multi can henceforth be seen as a balls and bins game with 2^r bins, $\sigma_{\mathcal{E}}$ balls, and threshold $\rho' = \rho/2$:

$$\mathbf{Pr}\left(\mathsf{multi}\right) \leq \mathbf{Pr}\left(\mathsf{multcol}(2^r, \sigma_{\mathcal{E}}, \rho')\right).$$

By Lemma 1, we know that $\mathbf{Pr}\left(\mathsf{multcol}(2^r, \sigma_{\mathcal{E}}, \rho')\right) \leq \frac{\sigma_{\mathcal{E}}}{\min\{2^{b/2}, 2^c\}}$, where ρ' is the function described in Lemma 1 (parameters r, c are implicit). Note that we put $\rho = 2\rho'$. Addition of the four bounds via (25) gives

$$\mathbf{Pr}\left(\mathsf{guess} \vee \mathsf{hit}\right) \leq \frac{q_p \sigma_{\mathcal{E}} + \sigma_{\mathcal{E}}^2/2}{2^b} + \frac{\sigma_{\mathcal{E}}}{\min\{2^{b/2}, 2^c\}} + \frac{2\rho' q_p}{2^c} + \frac{q_p + \sigma_{\mathcal{E}}}{2^\kappa}.$$

where $\rho' = \rho(r, c)$ is the function defined in Lemma 1. □

4.2. *Authenticity of NORX*

Theorem 2. *Let $\Pi = (\mathcal{E}, \mathcal{D})$ be NORX based on an ideal underlying primitive p. Then,*

$$\mathbf{Adv}^{\mathrm{auth}}_{\Pi}(q_p, q_{\mathcal{E}}, \lambda_{\mathcal{E}}, q_{\mathcal{D}}, \lambda_{\mathcal{D}}) \leq \frac{(q_p + \sigma_{\mathcal{E}} + \sigma_{\mathcal{D}})^2}{2^b} + \frac{\sigma_{\mathcal{E}}}{\min\{2^{b/2}, 2^c\}} + \frac{2\rho q_p}{2^c} + \frac{q_p + \sigma_{\mathcal{E}} + \sigma_{\mathcal{D}}}{2^\kappa} + \frac{(q_p + \sigma_{\mathcal{E}} + \sigma_{\mathcal{D}})\sigma_{\mathcal{D}}}{2^c} + \frac{q_{\mathcal{D}}}{2^\tau},$$

where $\sigma_{\mathcal{E}}, \sigma_{\mathcal{D}}$ are defined in (18), and where $\rho = \rho(r, c)$ is the function defined in Lemma 1.

The bound is more complex than the one of Theorem 1, but intuitively implies that NORX offers integrity as long as it offers privacy and the number of forgery attempts $\sigma_{\mathcal{D}}$ is limited, where the total complexity $q_p + \sigma_{\mathcal{E}} + \sigma_{\mathcal{D}}$ should not exceed $2^c/\sigma_{\mathcal{D}}$. See Table 1 for the security level for the various parameter choices of NORX. Needless to say, the exact bound is more fine-grained.

Proof. We consider any adversary $\mathcal{A}$ that has access to $(p^{\pm}, \mathcal{E}_K, \mathcal{D}_K)$ and attempts to make $\mathcal{D}_K$ output a non-$\perp$ value. As in the proof of Theorem 1, we apply a URP-URF switch to find

$$\mathbf{Adv}^{\mathrm{auth}}_{\Pi}(\mathcal{A}) = \mathbf{Pr}\left(\mathcal{A}^{p^{\pm}, \mathcal{E}_K, \mathcal{D}_K} \text{ forges}\right) \leq \mathbf{Pr}\left(\mathcal{A}^{f^{\pm}, \mathcal{E}_K, \mathcal{D}_K} \text{ forges}\right) + \frac{(q_p + \sigma_{\mathcal{E}} + \sigma_{\mathcal{D}})^2}{2^{b+1}}. \tag{26}$$

Then we focus on $\mathcal{A}$ having oracle access to $(f^{\pm}, \mathcal{E}_K, \mathcal{D}_K)$. As before, we assume without loss of generality that the adversary only makes full-block queries.

We inherit terminology from Theorem 1. The state values corresponding to encryption and decryption queries will both be labeled (j, k), where j indicates the query and k the state value within the jth query. If needed we will add another parameter $\delta \in \{\mathcal{D}, \mathcal{E}\}$ to indicate that a state value $s_{\delta,j,k}$ is in the jth query to oracle δ, for $\delta \in \{\mathcal{D}, \mathcal{E}\}$ and $j \in \{1, \ldots, q_\delta\}$. Particularly, this means we will either label the state values as in (22) with a δ appended to the subscript, or simply as $(s_{\delta,j,1}, \ldots, s_{\delta,j,\sigma_{\delta,j}})$.

As before, we employ the collision events guess and hit, but expanded to the new notation with $\delta = \mathcal{E}$. Next, we define two $\mathcal{D}$-related collision events $\mathcal{D}$guess and $\mathcal{D}$hit. Let $i \in \{1, \ldots, q_p\}$, $(\mathcal{D}, j, k)$ be a decryption query index, and (δ', j', k') be an encryption or decryption query index:

$$\begin{aligned} \mathcal{D}\mathsf{guess}(i; j, k) &\equiv x_i = s_{\mathcal{D},j,k}, \\ \mathcal{D}\mathsf{hit}(j, k; \delta', j', k') &\equiv \mathrm{parent}(s_{\mathcal{D},j,k}) \neq \mathrm{parent}(s_{\delta',j',k'}) \wedge s_{\mathcal{D},j,k} = s_{\delta',j',k'}, \end{aligned}$$

We write $\mathcal{D}\mathsf{guess} = \vee_{i;j,k} \mathcal{D}\mathsf{guess}(i; j, k)$ and $\mathcal{D}\mathsf{hit} = \vee_{j,k;\delta',j',k'} \mathcal{D}\mathsf{hit}(j, k; \delta', j', k')$, and define $\mathsf{event} = \mathsf{guess} \vee \mathsf{hit} \vee \mathcal{D}\mathsf{guess} \vee \mathcal{D}\mathsf{hit}$.

Observe that from (26) we get

$$\mathbf{Pr}\left(\mathcal{A}^{f^{\pm},\mathcal{E}_K,\mathcal{D}_K} \text{ forges}\right) \leq \mathbf{Pr}\left(\mathcal{A}^{f^{\pm},\mathcal{E}_K,\mathcal{D}_K} \text{ forges} \mid \neg\mathsf{event}\right) + \mathbf{Pr}\left(\mathcal{A}^{f^{\pm},\mathcal{E}_K,\mathcal{D}_K} \text{ sets } \mathsf{event}\right). \tag{27}$$

A bound on the probability that $\mathcal{A}$ sets event is derived in Lemma 4.

The remainder of this proof centers on the probability that $\mathcal{A}$ forges given that event does not happen. Such a forgery requires that $[f(s^{\mathrm{tag}}_{\mathcal{D},j})]^{\tau} = A_j$ for some decryption query j. By ¬event, we know that $s^{\mathrm{tag}}_{\mathcal{D},j}$ is a new state value for all $j \in \{1,\ldots,q_{\mathcal{D}}\}$, hence f's output under $s^{\mathrm{tag}}_{\mathcal{D},j}$ is independent of all other values and uniformly distributed for all j. As a result, we know that the jth forgery attempt is successful with probability at most $1/2^{\tau}$. Summing over all $q_{\mathcal{D}}$ queries, we get

$$\mathbf{Pr}\left(\mathcal{A}^{f^{\pm},\mathcal{E}_K,\mathcal{D}_K} \text{ forges} \mid \neg\mathsf{event}\right) \leq \frac{q_{\mathcal{D}}}{2^{\tau}},$$

and the proof is completed via (26, 27) and the bound of Lemma 4, where we again use that $\dfrac{q_p\sigma_{\mathcal{E}} + \sigma_{\mathcal{E}}^2/2}{2^b} \leq \dfrac{(q_p + \sigma_{\mathcal{E}} + \sigma_{\mathcal{D}})^2}{2^{b+1}}$. □

Lemma 4. $\mathbf{Pr}\left(\mathcal{A}^{f^{\pm},\mathcal{E}_K,\mathcal{D}_K} \textit{ sets } \mathsf{event}\right) \leq \dfrac{q_p\sigma_{\mathcal{E}} + \sigma_{\mathcal{E}}^2/2}{2^b} + \dfrac{\sigma_{\mathcal{E}}}{\min\{2^{b/2}, 2^c\}} + \dfrac{2\rho q_p}{2^c} + \dfrac{q_p + \sigma_{\mathcal{E}} + \sigma_{\mathcal{D}}}{2^{\kappa}} + \dfrac{(q_p + \sigma_{\mathcal{E}})\sigma_{\mathcal{D}} + \sigma_{\mathcal{D}}^2/2}{2^c}$, *where* $\rho = \rho(r,c)$ *is the function defined in Lemma 1.*

Proof. Recall that $\mathsf{event} = \mathsf{guess} \vee \mathsf{hit} \vee \mathcal{D}\mathsf{guess} \vee \mathcal{D}\mathsf{hit}$. Employing events key and multi from Lemma 3, we find:

$$\begin{aligned}&\mathbf{Pr}\left(\mathsf{guess} \vee \mathsf{hit} \vee \mathcal{D}\mathsf{guess} \vee \mathcal{D}\mathsf{hit}\right)\\ &\quad\leq \mathbf{Pr}\left(\mathsf{guess} \vee \mathsf{hit} \vee \mathcal{D}\mathsf{guess} \vee \mathcal{D}\mathsf{hit} \mid \neg(\mathsf{key} \vee \mathsf{multi})\right)\\ &\qquad+ \mathbf{Pr}\left(\mathsf{key} \vee \mathsf{multi}\right).\end{aligned} \tag{28}$$

The proof builds upon Lemma 3, and in particular we will use the same proof technique of running over all queries and computing the probability that a query sets event, assuming event has not been set before. The bounds on $\mathbf{Pr}\left(\mathsf{guess} \vee \mathsf{hit} \mid \neg(\mathsf{key} \vee \mathsf{multi})\right)$ and $\mathbf{Pr}\left(\mathsf{key} \vee \mathsf{multi}\right)$ carry over from Lemma 3 verbatim, where we additionally note that for a given query, the previous decryption queries are of no influence as by hypothesis $\mathcal{D}\mathsf{guess} \vee \mathcal{D}\mathsf{hit}$ was not set before the query in question. We continue with the analysis of $\mathcal{D}$guess and $\mathcal{D}$hit.

Event $\mathcal{D}$guess. Note that the adversary may freely choose the outer part in decryption queries and primitive queries. Indeed, the ciphertext values that $\mathcal{A}$ chooses in decryption queries define the outer parts of the state values. Consequently, $\mathcal{D}$guess gets set as

soon as there is a primitive state and a decryption state whose capacities are equal. This happens with probability at most $\mathbf{Pr}\left(\mathcal{D}\mathsf{guess} \mid \neg(\mathsf{key} \vee \mathsf{multi})\right) \leq q_p \sigma_{\mathcal{D}}/2^c$.

Event $\mathcal{D}$hit. A technicality occurs in that the adversary can reuse nonces in decryption. To increase readability, we first state that any decryption state s satisfies $[s]^\kappa = K$ only with probability at most $\sigma_{\mathcal{D}}/2^\kappa$, and in the remainder we can exclude this case. Next, we define an event $\mathsf{innerhit}$. Let (δ, j, k) and (δ', j', k') be two decryption query indices, and let $\mathsf{const} \in \{0, 01 \oplus 02, 01 \oplus 04, 01 \oplus 08, 01 \oplus 10, 02 \oplus 04, 02 \oplus 08, 02 \oplus 20, 02 \oplus 20 \oplus id_i, 04 \oplus 08\}$:

$$\mathsf{innerhit}(\delta, j, k; \delta', j', k'; \mathsf{const}) \equiv \mathrm{parent}(s_{\delta,j,k}) \neq \mathrm{parent}(s_{\delta',j',k'}) \wedge [s_{\delta,j,k}]_c = [s_{\delta',j',k'}]_c \oplus \mathsf{const}.$$

Note that for any choice of indices and const, we have $\mathbf{Pr}(\mathsf{innerhit}(\delta, j, k; \delta', j', k'; \mathsf{const})) \leq 1/2^c$.

We consider the general case $D \neq 1$. Consider the $\bar{j}$th decryption query $(N; H, C, T; A)$. Say it consists of u header blocks $H_1 \ldots H_u$, v ciphertext blocks $C_1 \ldots C_v$, and w trailer blocks $T_1 \ldots T_w$, and write its state values as in (17). Let $(N_{\delta,j}; H_{\delta,j}, C_{\delta,j}, T_{\delta,j}; A_{\delta,j})$ be an older ciphertext tuple that shares the longest common blockwise prefix with $(N; H, C, T; A)$. Note that this tuple may not be unique (for instance if N is new), and that it may come from an encryption or decryption query. Say that this query consists of $u_{\delta,j}$ header blocks, $v_{\delta,j}$ ciphertext blocks, and $w_{\delta,j}$ trailer blocks, and write its state values as in (22). We proceed with a case distinction.

(1) $(\boldsymbol{N}; \boldsymbol{H}, \boldsymbol{C}, \boldsymbol{T}) = (\boldsymbol{N_{\delta j}}; \boldsymbol{H_{\delta j}}, \boldsymbol{C_{\delta j}}, \boldsymbol{T_{\delta j}})$ **but** $\boldsymbol{A \neq A_{\delta j}}$. In this case the query renders no new states and $\mathcal{D}$hit cannot be set by definition;

(2) $(\boldsymbol{N}; \boldsymbol{H}, \boldsymbol{C}) = (\boldsymbol{N_{\delta j}}; \boldsymbol{H_{\delta j}}, \boldsymbol{C_{\delta j}})$ **but** $\boldsymbol{T \neq T_{\delta j}}$. Let $\ell \in \{1, \ldots, \min\{w, w_{\delta,j}\}, \infty\}$ be minimal such that $T_\ell \neq T_{\delta,j,\ell}$, where $\ell = \infty$ means that T is a substring of $T_{\delta,j}$ (if $w < w_{\delta,j}$) or vice versa (if $w > w_{\delta,j}$). We make a further distinction between $\ell = \infty$ and $\ell < \infty$.

(a) $\boldsymbol{\ell = \infty}$. Note that $s^T_{\min\{w,w_{\delta,j}\}} = s^T_{\delta,j,\min\{w,w_{\delta,j}\}} \oplus 04 \oplus 08$. If this input to f is old, it implies $\mathsf{innerhit}(\delta, j, \min\{w, w_{\delta,j}\}; \delta', j', k'; 04 \oplus 08)$ for some (δ', j', k') older than the current query $(\mathcal{D}, \bar{j}, \min\{w, w_{\delta,j}\})$, which is the case with probability at most $1/2^c$ (for all possible index tuples). Otherwise, f generates a new value and new state value s (s^T_{w+1} if $w > w_{\delta,j}$ or s^{tag} if $w < w_{\delta,j}$), which sets $\mathcal{D}$hit if it sets $\mathsf{innerhit}$ with an older state $s_{\delta',j',k'}$ under $\mathsf{const} = 0$. This also happens with probability at most $1/2^c$ for any (δ', j', k'). This procedure propagates to s^{tag}. In total, the $\bar{j}$th decryption query sets $\mathcal{D}$hit with probability at most $\sum_{k=1}^{\sigma_{\mathcal{D},\bar{j}}} \frac{\sigma_{\mathcal{E}}+\sigma_{\mathcal{D},1}+\cdots+\sigma_{\mathcal{D},\bar{j}-1}+(k-1)}{2^c}$;

(b) $\boldsymbol{\ell < \infty}$. In this case $s^T_{\ell-1} = s^T_{\delta,j,\ell-1}$ and $s^T_\ell = s^T_{\delta,j,\ell} \oplus (T_\ell \| 0^c) \oplus (T_{\delta,j,\ell} \| 0^c) \neq s^T_{\delta,j,\ell}$.[5] As before, s^T_ℓ is a new input to f, except if $\mathsf{innerhit}(\delta, j, \ell; \delta', j', k'; 0)$ for some (δ', j', k') older than the current query $(\mathcal{D}, \bar{j}, \ell)$. This is the case

[5] Note that if (δ, j) were not unique, then we similarly have $s^T_{\ell-1} = s^T_{\delta',j',\ell-1}$ and $s^T_\ell = s^T_{\delta',j',\ell} \oplus (T_\ell \| 0^c) \oplus (T_{\delta',j',\ell} \| 0^c) \neq s^T_{\delta',j',\ell}$ for all other queries (δ', j') with the same prefix (possibly XORed with $04 \oplus 08$).

with probability at most $1/2^c$ for all possible older queries. The procedure propagates to s^{tag} as before, and the same bound holds;

(3) $(N; H) = (N_{\delta,j}; H_{\delta,j})$ **but** $C \neq C_{\delta,j}$. The analysis is similar but a special treatment is required to deal with the merging phase. Consider the ciphertext C to be divided into blocks $C_{k,\ell}$ for $k = 1, \ldots, D$ and $\ell = 1, \ldots, v_k$. Similarly for $C_{\delta,j}$. For $k = 1, \ldots, D$, let $\ell_k \in \{1, \ldots, \min\{v_k, v_{\delta,j,k}\}, \infty\}$ be minimal such that $C_{k,\ell_k} \neq C_{\delta,j,k,\ell_k}$. Again, $\ell_k = \infty$ means that C_k is a substring of $C_{\delta,j,k}$ (if $v_k \leq v_{\delta,j,k}$) or vice versa (if $v_k \geq v_{\delta,j,k}$). We make a further distinction between whether or not $(\ell_1, \ldots, \ell_D) = (\infty, \ldots, \infty)$.

(a) $(\ell_1, \ldots, \ell_D) = (\infty, \ldots, \infty)$. As $C \neq C_{\delta,j}$, there must be a k such that $v_k \neq v_{\delta,j,k}$ and thus that C_k is a strictly smaller substring of $C_{\delta,j,k}$ or vice versa. Consequently, $s^C_{k,v_k} = s^C_{\delta,j,k,v_k} \oplus \texttt{02} \oplus \texttt{20} \oplus id_k[\min\{v_k, v_{\delta,j,k}\} = 1]$ (or $\oplus$ `02` $\oplus$ `04` if $D = 1$ and there is no merging phase, or $\oplus$ `02` $\oplus$ `08` if there is furthermore no trailer). Then, this state is new to f except if $\mathsf{innerhit}(\delta, j, k, v_k; \delta', j', k'; \mathsf{const})$ is set for the const described above. (We slightly misuse notation here in that v_k is input to innerhit.) This means that also s^T_0 will be new except if it hits a certain older state, which happens with probability $1/2^c$. The reasoning propagates up to s^{tag} as before, and the same bound holds;

(b) $(\ell_1, \ldots, \ell_D) < (\infty, \ldots, \infty)$. Let k be such that $\ell_k < \infty$. Then, $s^C_{k,\ell_k-1} = s^C_{\delta,j,k,\ell_k-1}$ and $s^C_{k,\ell_k} = C_{k,\ell_k} \| [s^C_{\delta,j,k,\ell_k}]_c \neq s^C_{\delta,j,k,\ell_k}$. The reasoning of case (2b) carries over for all future state values;

(4) $N = N_{\delta,j}$ **but** $H \neq H_{\delta,j}$. The analysis follows fairly the same principles, albeit using const $\in \{\texttt{0}, \texttt{01} \oplus \texttt{02}, \texttt{01} \oplus \texttt{04}, \texttt{01} \oplus \texttt{08}, \texttt{01} \oplus \texttt{10}\}$;

(5) $N \neq N_{\delta,j}$. The nonce N is new (hence the query shares no prefix with any older query). There has not been an earlier state s that satisfies $[s]^\kappa = K$ (by virtue of the analysis in hit and key, and the first step of this event $\mathcal{D}$hit). Therefore, s^{init} is new by construction and a simplification of above analysis applies.

Summing over all queries:

$$\begin{aligned}\mathbf{Pr}\left(\mathcal{D}\mathsf{hit} \mid \neg(\mathsf{key} \vee \mathsf{multi})\right) &\leq \sum_{\tilde{j}=1}^{q_{\mathcal{D}}} \sum_{k=1}^{\sigma_{\mathcal{D},\tilde{j}}} \frac{\sigma_{\mathcal{E}} + \sigma_{\mathcal{D},1} + \cdots + \sigma_{\mathcal{D},\tilde{j}-1} + (k-1)}{2^c} + \frac{\sigma_{\mathcal{D}}}{2^\kappa} \\ &\leq \frac{\sigma_{\mathcal{E}}\sigma_{\mathcal{D}} + \binom{\sigma_{\mathcal{D}}}{2}}{2^c} + \frac{\sigma_{\mathcal{D}}}{2^\kappa},\end{aligned}$$

where the last term comes from the exclusion of the event that any decryption state satisfies $[s]^\kappa = K$.
Together with the bound of Lemma 3 we find via (28),

$$\begin{aligned}\mathbf{Pr}\left(\mathsf{event}\right) \leq &\frac{q_p\sigma_{\mathcal{E}} + \sigma_{\mathcal{E}}^2/2}{2^b} + \frac{\sigma_{\mathcal{E}}}{\min\{2^{b/2}, 2^c\}} + \frac{2\rho' q_p}{2^c} + \frac{q_p + \sigma_{\mathcal{E}} + \sigma_{\mathcal{D}}}{2^\kappa} \\ &+ \frac{(q_p + \sigma_{\mathcal{E}})\sigma_{\mathcal{D}} + \sigma_{\mathcal{D}}^2/2}{2^c},\end{aligned}$$

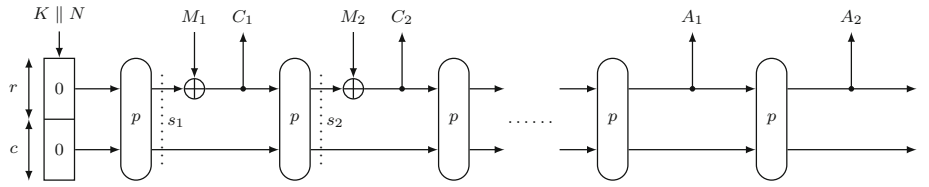

Fig. 3. Target structure in key recovery attack.

where $\rho' = \rho(r, c)$ is the function defined in Lemma 1. □

5. Tightness of the Bound

We derive a generic attack on Sponge-based authenticated encryption schemes. The attack exploits multi-collisions on the outer part of the internal state. Using the multi-collision bounds of Suzuki et al. [91,92], we demonstrate that the attack actually matches the proven security bound, meaning that the bounds of Sect. 4 are tight. Therefore, we first describe our simplified target structure in Sect. 5.1. The attack is described in Sect. 5.2 and evaluated in Sect. 5.3.

5.1. *Target Structure*

We consider the simplified structure of Fig. 3. Without loss of generality, we consider a key $K \in \{0,1\}^\kappa$, nonce $N \in \{0,1\}^{b-\kappa}$ (hence $\nu = b - \kappa$), and we assume that init initializes the state as $(K, N) \mapsto K \| N$. (The attack can be generalized to the setting where the key is absorbed in multiple evaluations of p, or where the key is XORed into the state before outputting A. See also Sect. 5.4.) We consider no associated data, or in terminology of Sect. 2, we put $H, T \leftarrow \texttt{Null}$. The message size must be at least one complete block. Note that, in many schemes, the message of one complete block will expand to two blocks by a padding procedure. We consider a general setting where the τ-bit authentication tag A may be generated in multiple extraction rounds (two in Fig. 3), and we assume that $\tau \geq c$. We ignore minor issues irrelevant to our attack, such as padding, frame bits, domain separation for message processing and tag generation parts, and truncation of the tag.

As shown in Fig. 3, the b-bit state after the first permutation call is denoted s_1. Its outer and inner part are denoted $[s_1]^r$ and $[s_1]_c$, respectively. Then, an r-bit message block M_1 is XORed into $[s_1]^r$ and the first ciphertext block $C_1 = [s_1]^r \oplus M_1$ is output. The state is evaluated using the permutation, and the resulting state is s_2. Note that the values M_i and C_i reveal the outer part of state s_i as $[s_i]^r = M_i \oplus C_i$.

5.2. *Distinguishing Attacks via Key Recovery*

Let $\rho \geq 2$. If $2^\kappa \leq 2^c/\rho$ a naive key recovery attack can be performed in complexity 2^κ, and we assume that $2^\kappa > 2^c/\rho$.

We first give an overview of the attack. Once a b-bit state in the structure of Fig. 3 is recovered, the secret key K can be recovered immediately by computing the inverse of the permutation. Our attack aims to recover the internal state s_1 after the first permutation call. It consists of an online phase followed by an offline phase.

In the online phase, the adversary searches for a ρ-collision on the r-bit value C_1. It makes a certain amount of encryption oracle queries for different N and possibly different M_1. Let q denote the total number of encryption queries needed. The online phase results in ρ pairs of (N, M_1) which produce the same C_1 but different $[s_1]_c$. The adversary also stores the tag A for each pair.

In the offline phase, the adversary recovers an inner part $[s_1]_c$. Using the value C_1, the same for all tuples, the value $[s_1]_c$ is exhaustively guessed. In a bit more detail, the adversary computes the authentication tag A from $C_1 \| [s_1]_c$ offline, and checks if there is a match with any stored tag. Because ρ tags are stored, the attack cost is about $2^c/\rho$. Once $[s_1]_c$ is recovered, the adversary can compute $p^{-1}((M_1 \oplus C_1) \| [s_1]_c)$ and recover K.

The formal description of the attack is given below. Here, we denote the data D for the kth block in the jth query by $D_{j,k}$. We omit the second subscript for the data where the block length is always 1, e.g., nonce N_j.

Online Phase

1. Choose q different pairs $(N_q, M_{q,1})$ for $i = 1, 2, \ldots, q$;
2. Query $(N_i, M_{i,1})$ for $i = 1, 2, \ldots, q$ and receive $(C_{i,1}, A_{i,1} \| A_{i,2} \| \ldots)$;
3. Find a ρ-collision on $C_{.,1}$;
4. Store ρ triplets of $(N_i, M_{i,1}, A_{i,1} \| A_{i,2} \| \ldots)$ contributing to the ρ-collision. We denote the colliding value of $C_{.,1}$ by $\overline{C}$, which is also stored.

Offline Phase

1. Re-define the outer part of the state after the computation of $[s_{.,1}]^r \oplus M_{.,1}$ by $\overline{C}$;
2. Make $2^c/\rho$ guesses for $[s_{.,1}]_c$, denoted by $[s_{j,1}]_c$ for $j = 1, 2, \ldots, 2^c/\rho$;
3. For each j, generate the tag $A_{j,1} \| A_{j,2} \| \ldots$ with the state $\overline{C} \| [s_{j,1}]_c$;
4. Check if $A_{j,1} \| A_{j,2} \| \ldots$ matches one of the ρ values $A_{i,1} \| A_{i,2} \| \ldots$ stored in the online phase. If so, assume that $[s_{j,1}]_c$ is the right value. Let i' and j' be matching indices;
5. Compute $p^{-1}\big((M_{i',1} \oplus \overline{C}) \| [s_{j',1}]_c\big)$. If the resulting value matches nonce $N_{i'}$, output the first κ bits of the state as the recovered key K.

5.3. *Attack Evaluation*

In the online phase, the adversary does not strictly need to choose N and M_1, a *given* list of q different tuples suffices. Thus, the attack is a known plaintext attack. The data complexity is q one-block messages and the memory to store q triples $(N_i, M_{i,1}, A_{i,1} \| A_{i,2} \| \ldots)$ for $i = 1, \ldots, q$ is required. The time complexity of at least q memory access is also required. Intuitively, all the complexities in the online phase are q.

In the offline phase, because ρ candidates are stored in the online phase and $2^c/\rho$ guesses are examined, one match is expected. If the internal state values match, the corresponding tag values also match. Thus, the right guess is identified. Due to the

Table 3. Comparison of attack complexity and security bound.

Parameters	Attack complexity			Security bound
	ρ	q	$2^r/\rho$	$2^c/\alpha,\ \alpha = \frac{1.4r}{\log_2 r+r-c-2}$
$c = r = 128$	18	$2^{123.806}$	$2^{123.830}$	$2^{122.837}$
$c = r = 256$	30	$2^{251.057}$	$2^{251.093}$	$2^{250.100}$
$c = r = 512$	51	$2^{506.272}$	$2^{506.327}$	$2^{505.322}$

assumption that the tag size is at least c bits, the match likely only suggests the right guess. In addition, we can further filter out the false positive by r bits with the match of N in the last step. Thus, with a very high probability the key is successfully recovered. For the complexity, the only important factor is the time complexity of $2^c/\rho$ tag generation functions.

What remains is to appropriately choose parameters for q and ρ so that the total complexity $\max\{q, 2^c/\rho\}$ is minimized. Suzuki et al. [91,92] showed that, when $c \leq r$, the complexity q to find a ρ-collision with probability about 0.5 is given by

$$q = (\rho!)^{\frac{1}{\rho}} \cdot 2^{\frac{\rho-1}{\rho}r} + \rho - 1. \tag{29}$$

$\boldsymbol{c = r}$. We demonstrate tightness of the bound for the cases $c = r = 128$, $c = r = 256$, and $c = r = 512$. Note that, provided κ is large enough, the bound of Theorem 1 is dominated by $2^c/\alpha$ with $\alpha = \frac{1.4r}{\log_2 r+r-c-2}$ (cf., Table 2). In Table 3 we evaluate the attack complexity so that $\max\{q, 2^r/\rho\}$ is minimized. This complexity is always bigger but very close to the proven bound, which shows tightness of security bound.

$\boldsymbol{c < r}$. It is common practice to enlarge the rate of Sponge-based authenticated encryption so that more data can be processed per permutation call. We demonstrate tightness of our attack for the case of $c = 256$ and $r \in [257, 768]$. Figure 1 depicts the evaluated attack complexity and our security bound for $c = 256$. For the sake of completeness, it also includes the $2^c/r$ bound of the original ASIACRYPT 2014 article [53], which decreases by approximately a logarithmic factor $\log_2 r$.

Note that the adversary needs to find a multi-collision on r bits with only 2^c trials. When the rate increases, and particularly when $r > 2c$, the adversary cannot even find an ordinary collision within 2^c trials. In this case, the multi-collision-based attack will not be influential. Due to this, our bound is getting close to 2^c when r becomes large. The advantage of the attack comes from the number of generated multi-collisions. Considering that the number of multi-collisions can only take discrete values while our bound can take sequential values, our bound is strictly tight.

$\boldsymbol{c > r}$. Note that, for $c > r$, the security bound of Theorem 1 is not dominated by $2^c/\alpha$ but rather by $2^{b/2}$, omitting constants (cf., Table 2). Tightness of the bound follows by a naive attack that aims to find collisions on the b-bit state.

5.4. *Distinguishing Attacks Without Key Recovery*

As later explained in Fig. 4, several practical designs use key K for the initialization as well as for the tag generation. Those schemes cannot be distinguished with a straightforward application of the above generic procedure, yet it is still possible to distinguish them by increasing the attack complexity only by 1 bit or so.

We focus on Ascon, GIBBON and HANUMAN, in which K in the tag computation prevents the adversary from computing tag A offline. This can be solved by extending the number of message blocks in each query. Instead of the tag $A_{i,1} \| A_{i,2} \| \ldots$, outer parts of the subsequent blocks $[s_{i,2}]^r \| [s_{i,3}]^r \| \ldots$ take a role of filter to identify the correct guess. If the number of filtered bits is much bigger than c, a match suggests the correct guess with very high probability. Owing to the additional message blocks, the attack complexity increases by 1 bit or so, depending how many message blocks are added.

In HANUMAN, K can be recovered from the internal state by inverting the permutation to the initial value. Meanwhile in Ascon and GIBBON, K cannot be recovered and the adversary only can mount distinguishing attacks.

6. Other CAESAR Submissions

In this section we discuss how the mode security proof of NORX generalizes to the CAESAR submissions Ascon, the BLNK mode underlying CBEAM/STRIBOB, ICEPOLE, Keyak (v1 only), and two out of the three PRIMATEs. Before doing so, we make a number of observations and note how the proof can accommodate small design differences.

- NORX uses domain separation constants at all rounds, but this is not strictly necessary and other solutions exist. In the privacy and integrity proofs of NORX, and more specifically at the analysis of state collisions caused by a decryption query in Lemma 4, the domain separations are only needed at the transitions between variable-length inputs, such as header to message data or message to trailer data. This means that the proofs would equally hold if there were simpler transitions at these positions, such as in Ascon. Alternatively, the domain separation can be done by using a different primitive, as in GIBBON and HANUMAN, or a slightly more elaborated padding, as in BLNK, ICEPOLE, and Keyak;
- The extra permutation evaluations at the initialization and finalization of NORX are not strictly necessary: in the proof we consider the monotone event that no state collides assuming no earlier state collision occurred. For instance, in the analysis of $\mathcal{D}\mathsf{hit}$ in the proof of Lemma 4, we necessarily have a new input to p at some point, and *consequently* all next inputs to p are new (except with some probability);
- NORX starts by initializing the state with $\mathsf{init}(K, N) = (K \| N \| 0^{b-\kappa-\nu}) \oplus \mathsf{const}$ for some constant const and then permuting this value. Placing the key and nonce at different positions of the state does not influence the security analysis. The proof would also work if, for instance, the header is preceded with $K \| N$ or a properly padded version thereof and the starting state is 0^b;
- In a similar fashion, there is no problem in defining the tag to be a different τ bits of the final state; for instance, the rightmost τ bits;

– Key additions into the inner part *after* the first permutation are harmless for the mode security proof. Particularly, as long as these are done at fixed positions, these have the same effect as XORing a domain separation constant.

These five modifications allow one to generalize the proof of NORX to Ascon, CBEAM and STRIBOB, ICEPOLE, Keyak, and two PRIMATEs, GIBBON and HANUMAN. The only major difference lies in the fact none of these designs accommodates a trailer, hence all are functions of the form

$$(C, A) \longleftarrow \mathcal{E}_K(N; H, M) \qquad \text{and} \qquad M/\perp \longleftarrow \mathcal{D}_K(N; H, C; A),$$

except for one instance of ICEPOLE which accommodates a secret message number. Additionally, these designs have $\sigma_\delta \leq \lambda_\delta + q_\delta$ for $\delta \in \{\mathcal{D}, \mathcal{E}\}$ (or $\sigma_\delta \leq \lambda_\delta + 2q_\delta$ for CBEAM/STRIBOB). We always write $H = (H_1, \ldots, H_u)$ and $M = (M_1, \ldots, M_v)$ whenever notation permits. In below sections we elaborate on these designs separately, where we slightly deviate from the alphabetical order to suit the presentation. Diagrams of all modes are given in Fig. 4. The parameters and achieved provable security levels of the schemes are given in Table 1.

We remark that the attack of Sect. 5 carries over to CBEAM and STRIBOB, ICEPOLE and a simplified version of Keyak v1 (with only one round of key absorption). It does not apply to Ascon, GIBBON, and HANUMAN due to the additional XOR of the secret key at the end.

6.1. *Ascon*

Ascon is a submission by Dobraunig et al. [33,34] and is depicted in Fig. 4a. It is originally defined based on two permutations p_1, p_2 that differ in the number of underlying rounds. We discard this difference, considering Ascon with one permutation p.

Ascon initializes its state using init that maps (K, N) to $(0^{b-\kappa-\nu} \| K \| N) \oplus \mathsf{const}$, where const is determined by some design-specific parameters set prior to the security experiment. The header and message can be of arbitrary length and are padded to length a multiple of r bits using 10^*-padding. An XOR with 1 separates header processing from message processing. From the above observations, it is clear that the proofs of NORX directly carry over to Ascon.

6.2. *ICEPOLE*

ICEPOLE is a submission by Morawiecki et al. [65,66] and is depicted in Fig. 4c. It is originally defined based on two permutations, p_1 and p_2, that differ in the number of underlying rounds. We discard this difference, considering ICEPOLE with one permutation p.

ICEPOLE initializes its state as NORX does, be it with a different constant. The header and message can be of arbitrary length and are padded as follows. Every block is first appended with a frame bit: 0 for header blocks $H_1, \ldots, H_{u-1}$ and message block M_v, and 1 for header block H_u and message blocks $M_1, \ldots, M_{v-1}$. Then, the blocks are padded to length a multiple of r bits using 10^*-padding. In other words, every padded block of r bits contains at most $r - 2$ data bits. This form of domain separation using

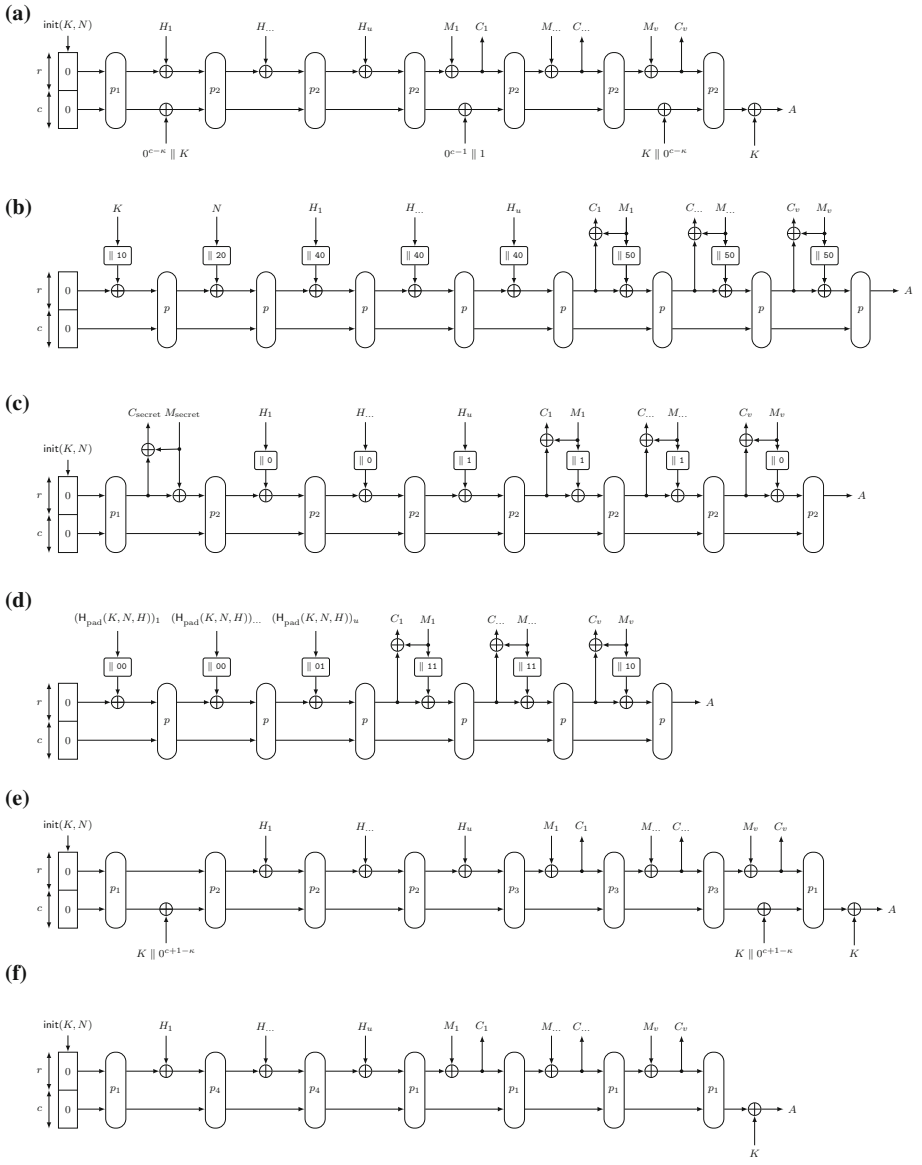

Fig. 4. CAESAR submission modes discussed in Sect. 6, **a** Ascon, **b** BLNK (used in CBEAM and STRIBOB), **c** ICEPOLE, **d** Keyak v1, **e** GIBBON (PRIMATEs) and **f** HANUMAN (PRIMATEs).

frame bits suffices for the proof to go through. One variant of ICEPOLE also allows for a secret message number M_{secret}, which consists of one block and is encrypted prior to the processing of the header, similar to the message. As this secret message number is of fixed length, no domain separation is required and the proof can easily be adapted. From above observations, it is clear that the proofs of NORX directly carry over to

ICEPOLE. Without going into detail, we note that the same analysis can be generalized to the parallelized mode of ICEPOLE [65,66].

6.3. *Keyak*

Keyak v1 is a submission by Bertoni et al. [22]. The basic mode for the serial case is depicted in Fig. 4d, yet due to its hybrid character it is slightly more general in nature. It is built on top of SpongeWrap [19]. We remark that the discussion does not apply to Keyak v2, which is built on top of the full-state keyed Duplex [31,60].

Keyak initializes its state by 0^b, and concatenates K, N, and H using a special padding rule:

$$\mathsf{H}_{\mathrm{pad}}(K, N, H) = \mathsf{keypack}(K, 240) \parallel \mathsf{enc}_8(1) \parallel \mathsf{enc}_8(0) \parallel N \parallel H,$$

where $\mathsf{enc}_8(x)$ is an encoding of x as a byte and $\mathsf{keypack}(K, \ell) = \mathsf{enc}_8(\ell/8) \| K \| 10^{-\kappa-1 \bmod (\ell-8)}$. The key-nonce-header combination $\mathsf{H}_{\mathrm{pad}}(K, N, H)$ and message M can be of arbitrary length, and are padded as follows: first, every block is appended with two frame bits, being `00` for header blocks $(\mathsf{H}_{\mathrm{pad}}(K, N, H))_1, \ldots, (\mathsf{H}_{\mathrm{pad}}(K, N, H))_{u-1}$ and `01` for $(\mathsf{H}_{\mathrm{pad}}(K, N, H))_u$, and `11` for message blocks $M_1, \ldots, M_{v-1}$ and `10` for M_v. Then, the blocks are padded to length a multiple of r bits using 10^*1-padding. In other words, every padded block of r bits contains at most $r - 2$ data bits. This form of domain separation using frame bits suffices for the proof to go through. Due to above observations, our proof readily generalizes to SpongeWrap [19] and DuplexWrap [22], and thus to Keyak. Without going into detail, we note that the same analysis can be generalized to the parallelized mode of Keyak [22]. Additionally, Keyak also supports sessions, where the state is re-used for a next evaluation. Our proof generalizes to this case, simply with a more extended description of (17).

6.4. *BLNK (CBEAM and STRIBOB)*

CBEAM and STRIBOB are submissions by Saarinen [81,83–86]. Minaud identified an attack on CBEAM [62], but we focus on the modes of operation. Both modes are based on the BLNK Sponge mode [82], which is depicted in Fig. 4b.

The BLNK mode initializes its state by 0^b, compresses K into the state (using one or two permutation calls, depending on κ), and does the same with N. Then, the mode is similar to SpongeWrap [19], though using a slightly more involved domain separation system similar to the one of NORX. Due to above observations, our proof readily generalizes to BLNK [82], and thus to CBEAM and STRIBOB.

6.5. *PRIMATEs: GIBBON and HANUMAN*

PRIMATEs is a submission by Andreeva et al. [2,3], and consists of three algorithms: APE, GIBBON, and HANUMAN. The APE mode is the more robust one, and significantly differs from the other two, and from the other CAESAR submissions discussed

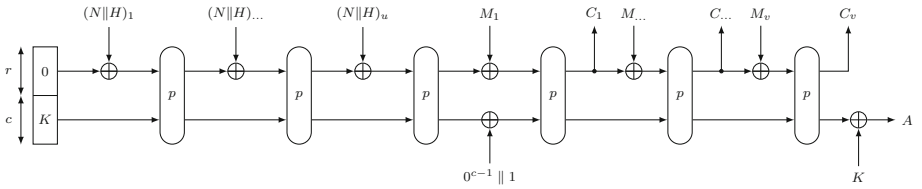

Fig. 5. APE (PRIMATEs) discussed in Sect. 7.

in this work, in the way that ciphertexts are derived and because the mode is secure against nonce-misusing adversaries up to common prefix [4]. (See Sect. 7 for a discussion on APE.) We now focus on GIBBON and HANUMAN, which are depicted in Fig. 4e, f. GIBBON is based on three related permutations $\mathbf{p} = (p_1, p_2, p_3)$, where the difference in p_2, p_3 is used as domain separation of the header compression and message encryption phases (the difference of p_1 from (p_2, p_3) is irrelevant for the mode security analysis). Similarly, HANUMAN uses two related permutations $\mathbf{p} = (p_1, p_2)$ for domain separation.[6]

GIBBON and HANUMAN initialize their state using init that maps (K, N) to $0^{b-\kappa-\nu} \| K \| N$. The header and message can be of arbitrary length, and are padded to length a multiple of r bits using 10^*-padding. In case the true header (or message) happens to be a multiple of r bits long, the 10^*-padding is considered to spill over into the capacity. From above observations, it is clear that the proofs of NORX directly carry over to GIBBON and HANUMAN. A small difference appears due to the usage of two different permutations: we need to make two RP-RF switches for each world. Concretely this means that the first term in Theorem 1 becomes $\frac{5(q_p+\sigma_\mathcal{E})^2}{2^{b+1}}$ and the first term in Theorem 2 becomes $\frac{3(q_p+\sigma_\mathcal{E}+\sigma_\mathcal{D})^2}{2^{b+1}}$.

7. PRIMATEs: APE

Unlike GIBBON and HANUMAN, the APE authenticated encryption scheme follows a different design strategy. It is depicted in Fig. 5. APE is based on one permutation p, and characteristic to the design is the way the ciphertexts are derived and verified.

APE uses a key of size c bits, and the initialization init places K into the inner part of the state. In case of a present nonce N, in APE it is prepended to the header H, denoted $N \| H$. The nonce is of fixed length, and of suggested size $2r$ bits [2,3]. The header and message can be of arbitrary length and are padded to length a multiple of r bits using 10^*-padding. In case the true header (or message) happens to be a multiple of r bits long, the 10^*-padding is considered to spill over into the capacity. In case the message

[6] Vizár [94] pointed out an oversight in the domain separation of an earlier version of HANUMAN. In this work, we consider the latest version of HANUMAN, with fixed domain separation.

is not a multiple of r bits long, the last ciphertext is derived slightly differently, and we refer to [2,3].

The scheme is designed and proven to be $2^{c/2}$ secure against nonce-misusing adversaries up to common prefix [4]. We now consider the security of APE in the nonce-respecting setting, and present an adversary that breaks the privacy with a complexity of about $2^{c/2}$. We assume that the adversary can make blockwise queries to the scheme. In more detail, upon an authenticated encryption of $M_1, \ldots, M_v$, it only needs to input the jth message block after it receives the $j-1$ ciphertext block, for $j = 2, \ldots, v$.

Proposition 1. *Let $\Pi = (\mathcal{E}, \mathcal{D})$ be APE based on an ideal underlying primitive p. Then,*

$$\mathbf{Adv}^{\mathrm{priv}}_{\Pi}(0, q_{\mathcal{E}}, \lambda_{\mathcal{E}}) \geq 1 - 2/2^{r}\,, \tag{30}$$

where all $q_{\mathcal{E}}$ queries are of length $(2^{(c+1)/2}+1)/q_{\mathcal{E}} + \rho + 1$.

Proof. We first consider a simplified setting, where $q_{\mathcal{E}} = 1$ and $\lambda_{\mathcal{E}} \approx 2^{c/2}$, and will generalize the attack to arbitrary $q_{\mathcal{E}}$ afterward. Denote $\rho = \lceil c/r \rceil$. The adversary makes one query of length $\lambda_{\mathcal{E}} = 2^{(c+1)/2} + \rho + 2$ as follows. Let N be some nonce, the header H is absent. $\mathcal{A}$ puts $M_1 = 0$, and $M_i = C_{i-1}$ for $k \in \{2, \ldots, \lambda_{\mathcal{E}}\}$. If there exist distinct $k, k' \in \{2, \ldots, \lambda_{\mathcal{E}} - \rho\}$ such that

$$(C_k, \ldots, C_{k+\rho}) = (C_{k'}, \ldots, C_{k'+\rho})\,, \tag{31}$$

then it outputs 1; otherwise it outputs 0. Note that if $\mathcal{A}$ converses with $\mathcal{E}_K$, then (31) holds if the permutation calls for M_k and $M_{k'}$ are the same. As the outer parts are 0 for both, this holds with probability at least $1/2^c$. Therefore, any such $k \neq k'$ exist with probability at least $\binom{\lambda_{\mathcal{E}}-\rho-1}{2}/2^c$. On the other hand, if $\mathcal{A}$ converses with \$, then this would only hold with probability $\binom{\lambda_{\mathcal{E}}-\rho-1}{2}/2^{(\rho+1)r}$. Thus,

$$\mathbf{Adv}^{\mathrm{priv}}_{\Pi}(0, 1, \lambda_{\mathcal{E}}) \geq \binom{\lambda_{\mathcal{E}} - \rho - 1}{2}/2^{c} - \binom{\lambda_{\mathcal{E}} - \rho - 1}{2}/2^{(\rho+1)r}.$$

Putting $\lambda_{\mathcal{E}} = 2^{(c+1)/2} + \rho + 2$ gives $2^c \leq \binom{\lambda_{\mathcal{E}}-\rho-1}{2} \leq 2^{c+1}$, and subsequently $\mathbf{Adv}^{\mathrm{priv}}_{\Pi}(0, 1, 2^{(c+1)/2} + \rho + 2) \geq 1 - 2/2^r$.

The analysis straightforwardly generalizes to $q_{\mathcal{E}}$ queries of total length $\lambda_{\mathcal{E}}$. Denote $\mu_{\mathcal{E}} = \lambda_{\mathcal{E}}/q_{\mathcal{E}}$, without loss of generality assuming that $\lambda_{\mathcal{E}}$ is a multiple of $q_{\mathcal{E}}$. For the jth query for $j \in \{1, \ldots, q_{\mathcal{E}}\}$, the adversary proceeds as follows. Let N_j be the unique nonce, the adversary does not query a header, as before. It takes $M_{j,1} = 0$, and sets $M_{j,k} = C_{j,k-1}$ for $k \in \{2, \ldots, \mu_{\mathcal{E}}\}$. If there exist $j, j' \in \{1, \ldots, q_{\mathcal{E}}\}$ and $k, k' \in \{2, \ldots, \mu_{\mathcal{E}} - \rho\}$ with $(j, k) \neq (j', k')$ such that

$$(C_{j,k}, \ldots, C_{j,k+\rho}) = (C_{j',k'}, \ldots, C_{j',k'+\rho})\,, \tag{32}$$

then it outputs 1; otherwise it outputs 0. The same analysis as before gives

$$\mathbf{Adv}^{\mathrm{priv}}_{\Pi}(0, q_{\mathcal{E}}, q_{\mathcal{E}}\mu_{\mathcal{E}}) \geq \binom{q_{\mathcal{E}}(\mu_{\mathcal{E}} - \rho - 1)}{2}/2^{c} - \binom{q_{\mathcal{E}}(\mu_{\mathcal{E}} - \rho - 1)}{2}/2^{(\rho+1)r}.$$

Consequently, for $q_{\mathcal{E}}\mu_{\mathcal{E}} = 2^{(c+1)/2} + (\rho+1)q_{\mathcal{E}} + 1$, we have $\mathbf{Adv}^{\mathrm{priv}}_{\Pi}(0, q_{\mathcal{E}}, 2^{(c+1)/2} + (\rho+1)q_{\mathcal{E}}+1) \geq 1 - 2/2^{r}$. Each of the $q_{\mathcal{E}}$ queries is of length approximately $(2^{(c+1)/2} + 1)/q_{\mathcal{E}} + \rho + 1$. □

8. Conclusions

In this work we analyzed one of the Sponge-based authenticated encryption designs in detail, NORX, and proved that it achieves security of approximately $\min\{2^{b/2}, 2^{c}, 2^{\kappa}\}$, significantly improving upon the traditional bound of $\min\{2^{c/2}, 2^{\kappa}\}$. Additionally, we showed that this proof straightforwardly generalizes to five other CAESAR modes, Ascon, BLNK (of CBEAM/STRIBOB), ICEPOLE, Keyak v1, and PRIMATEs. Our findings indicate an overly conservative parameter choice made by the designers, implying that some designs can improve speed by a factor of 4 at barely any security loss. It is expected that the security proofs also generalize to the modes of Artemia [1]. However, this mode is based on the JH hash function [96] and XORs data blocks in both the rate and inner part. It does not use domain separations, rather it encodes the lengths of the inputs into the padding at the end [9]. Therefore, a generalization of the proof of NORX to Artemia is not entirely straightforward.

The results in this work are derived in the ideal permutation model, where the underlying primitive is assumed to be ideal. We acknowledge that this model does not perfectly reflect the properties of the primitives. For instance, it is stated by the designers of Ascon, NORX, and PRIMATEs that non-random (but harmless) properties of the underlying permutation exist. Furthermore, it is important to realize that the proofs of security for the modes of operation in the ideal model do not have a direct connection with security analysis performed on the permutations, as is the case with block ciphers modes of operation. Nevertheless, we can use these proofs as heuristics to guide cryptanalysts to focus on the underlying permutations, rather than the modes themselves.

Acknowledgements

The authors would like to thank their co-designers of NORX and PRIMATEs and the designers of Ascon and Keyak for the discussions. In particular, we thank Samuel Neves for his useful comments. The authors furthermore thank the reviewers for their insightful comments. Bart Mennink is supported by a postdoctoral fellowship from the Netherlands Organisation for Scientific Research (NWO) under Veni grant 016.Veni.173.017.

Appendix: Proof Technique Used in Lemma 3

Formally, the proof technique used in Lemma 3 relies on the following paradigm. Note that there is an ordering of the $q_p + \sigma_{\mathcal{E}}$ primitive queries, and we can reformulate $\mathsf{guess}(\ell)$, $\mathsf{hit}(\ell)$, $\mathsf{key}(\ell)$, and $\mathsf{multi}(\ell)$ for $\ell = 1, \ldots, q_p + \sigma_{\mathcal{E}}$ analogously. Likewise define $\mathsf{event}(\ell) = \mathsf{guess}(\ell) \vee \mathsf{hit}(\ell)$ and $\mathsf{help}(\ell) = \mathsf{key}(\ell) \vee \mathsf{multi}(\ell)$, and define $\mathsf{event}(1 \ldots \ell) = \mathsf{event}(1) \vee \cdots \vee \mathsf{event}(\ell)$ and $\mathsf{help}(1 \ldots \ell) = \mathsf{help}(1) \vee \cdots \vee \mathsf{help}(\ell)$ for brevity. Then, we have

$$\begin{aligned}
&\mathbf{Pr}\,(\mathsf{event}) \\
&\quad \leq \mathbf{Pr}\left(\mathsf{event}(q_p + \sigma_{\mathcal{E}}) \mid \neg\mathsf{event}(1 \ldots q_p + \sigma_{\mathcal{E}} - 1) \wedge \neg\mathsf{help}(1 \ldots q_p + \sigma_{\mathcal{E}})\right) \\
&\qquad + \mathbf{Pr}\left(\mathsf{event}(1 \ldots q_p + \sigma_{\mathcal{E}} - 1) \vee \mathsf{help}(1 \ldots q_p + \sigma_{\mathcal{E}})\right),
\end{aligned}$$

and inductively $\mathbf{Pr}\,(\mathsf{event}) \leq \sum_{\ell=1}^{q_p+\sigma_{\mathcal{E}}} \mathbf{Pr}(\mathsf{event}(\ell) \mid \neg\mathsf{event}(1 \ldots \ell - 1) \wedge \neg\mathsf{help}(1 \ldots \ell)) + \mathbf{Pr}\,(\mathsf{help}(\ell) \mid \neg\mathsf{help}(1 \ldots \ell - 1))$. This formulation would however merely reduce the readability of the proof.

References

[1] J. Alizadeh, M. Aref, N. Bagheri, Artemia v1 (2014), submission to CAESAR competition

[2] E. Andreeva, B. Bilgin, A. Bogdanov, A. Luykx, F. Mendel, B. Mennink, N. Mouha, Q. Wang, K. Yasuda, PRIMATEs v1 (2014), submission to CAESAR competition

[3] E. Andreeva, B. Bilgin, A. Bogdanov, A. Luykx, F. Mendel, B. Mennink, N. Mouha, Q. Wang, K. Yasuda, PRIMATEs v1.1 (2016), submission to CAESAR competition

[4] E. Andreeva, B. Bilgin, A. Bogdanov, A. Luykx, B. Mennink, N. Mouha, K. Yasuda, APE: authenticated permutation-based encryption for lightweight cryptography, in C. Cid, C. Rechberger, (eds.) Fast Software Encryption—21st International Workshop, FSE 2014, London, UK, March 3–5, 2014. Revised Selected Papers. Lecture Notes in Computer Science, vol. 8540 (Springer, 2014), pp. 168–186

[5] E. Andreeva, A. Bogdanov, A. Luykx, B. Mennink, E. Tischhauser, K. Yasuda, Parallelizable and authenticated online ciphers, in K. Sako, P. Sarkar, (eds.) Advances in Cryptology—ASIACRYPT 2013—19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1–5, 2013, Proceedings, Part I. Lecture Notes in Computer Science, vol. 8269 (Springer, 2013), pp. 424–443

[6] E. Andreeva, J. Daemen, B. Mennink, G. Van Assche, Security of keyed sponge constructions using a modular proof approach, in G. Leander, (ed.) Fast Software Encryption—22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8–11, 2015, Revised Selected Papers. Lecture Notes in Computer Science, vol. 9054 (Springer, 2015), pp. 364–384

[7] J. Aumasson, P. Jovanovic, S. Neves, NORX v1 (2014), submission to CAESAR competition

[8] J. Aumasson, P. Jovanovic, S. Neves, NORX v2.0 (2015), submission to CAESAR competition

[9] N. Bagheri, Padding of Artemia (2014), CAESAR mailing list

[10] M. Bellare, V.T. Hoang, Identity-based format-preserving encryption, in B.M. Thuraisingham, D. Evans, T. Malkin, D. Xu, (eds.) Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30–November 03, 2017 (ACM, 2017), pp. 1515–1532

[11] M. Bellare, C. Namprempre, Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. J. Cryptol. 21(4), 469–491 (2008)

[12] M. Bellare, P. Rogaway, Code-based game-playing proofs and the security of triple encryption. Cryptology ePrint Archive, Report 2004/331 (2004)

[13] M. Bellare, P. Rogaway, The security of triple encryption and a framework for code-based game-playing proofs, in Vaudenay [93], pp. 409–426
[14] M. Bellare, P. Rogaway, D. Wagner, The EAX mode of operation, in B.K. Roy, W. Meier, (eds.) Fast Software Encryption, 11th International Workshop, FSE 2004, Delhi, India, February 5–7, 2004, Revised Papers. Lecture Notes in Computer Science, vol. 3017 (Springer, 2004), pp. 389–407
[15] J. Benaloh, (ed.), Topics in Cryptology—CT-RSA 2014—The Cryptographer's Track at the RSA Conference 2014, San Francisco, CA, USA, February 25–28, 2014, in Proceedings, Lecture Notes in Computer Science, vol. 8366 (Springer, 2014)
[16] G. Bertoni, J. Daemen, M. Peeters, G. Van Assche, Sponge Functions. ECRYPT Hash Function Workshop (2007)
[17] G. Bertoni, J. Daemen, M. Peeters, G. Van Assche, On the indifferentiability of the sponge construction, in N.P. Smart, (ed.) Advances in Cryptology—EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13–17, 2008. Proceedings. Lecture Notes in Computer Science, vol. 4965 (Springer, 2008), pp. 181–197
[18] G. Bertoni, J. Daemen, M. Peeters, G. Van Assche, Sponge-based pseudo-random number generators, in S. Mangard, F. Standaert, (eds.) Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17–20, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6225 (Springer, 2010), pp. 33–47
[19] G. Bertoni, J. Daemen, M. Peeters, G. Van Assche, Duplexing the sponge: Single-pass authenticated encryption and other applications, in A. Miri, S. Vaudenay, (eds.) Selected Areas in Cryptography—18th International Workshop, SAC 2011, Toronto, ON, Canada, August 11–12, 2011, Revised Selected Papers. Lecture Notes in Computer Science, vol. 7118 (Springer, 2011), pp. 320–337
[20] G. Bertoni, J. Daemen, M. Peeters, G. Van Assche, On the security of the keyed sponge construction. Symmetric Key Encryption Workshop (2011)
[21] G. Bertoni, J. Daemen, M. Peeters, G. Van Assche, Permutation-based encryption, authentication and authenticated encryption. Directions in Authenticated Ciphers (2012)
[22] G. Bertoni, J. Daemen, M. Peeters, G. Van Assche, R. Van Keer, Keyak v1 (2014), submission to CAESAR competition
[23] G. Bertoni, J. Daemen, M. Peeters, G. Van Assche, R. Van Keer, Keyak v2 (2015), submission to CAESAR competition
[24] A. Bogdanov, M. Knezevic, G. Leander, D. Toz, K. Varici, I. Verbauwhede, spongent: A lightweight hash function, in B. Preneel, T. Takagi, (eds.) Cryptographic Hardware and Embedded Systems—CHES 2011—13th International Workshop, Nara, Japan, September 28–October 1, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6917 (Springer, 2011), pp. 312–325
[25] CAESAR, Competition for Authenticated Encryption: Security, Applicability, and Robustness (2014). http://competitions.cr.yp.to/caesar.html
[26] D. Chang, M. Dworkin, S. Hong, J. Kelsey, M. Nandi, A Keyed Sponge Construction with Pseudorandomness in the Standard Model. NIST's 3rd SHA-3 Candidate Conference 2012 (2012)
[27] D. Chang, M. Nandi, Improved indifferentiability security analysis of chopmd hash function, in K. Nyberg, (ed.) Fast Software Encryption, 15th International Workshop, FSE 2008, Lausanne, Switzerland, February 10–13, 2008, Revised Selected Papers. Lecture Notes in Computer Science, vol. 5086 (Springer, 2008), pp. 429–443
[28] H. Chernoff, A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on the sum of Observations. Ann. Math. Stat. 23(4), 493–507 (1952)
[29] B. Cogliati, R. Lampe, Y. Seurin, Tweaking even-mansour ciphers, in Gennaro and Robshaw [40], pp. 189–208
[30] R.M. Corless, G.H. Gonnet, D.E.G. Hare, D.J. Jeffrey, D.E. Knuth, On the Lambert W function. Adv. Comput. Math. 5(1), 329–359 (1996)
[31] J. Daemen, B. Mennink, G. Van Assche, Full-state keyed duplex with built-in multi-user support, in T. Takagi, T. Peyrin, (eds.) Advances in Cryptology—ASIACRYPT 2017—23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3–7, 2017, Proceedings, Part II. Lecture Notes in Computer Science, vol. 10625 (Springer, 2017), pp. 606–637
[32] I. Dinur, O. Dunkelman, N. Keller, A. Shamir, Cryptanalysis of iterated even-mansour schemes with two keys, in Sarkar and Iwata [87], pp. 439–457. http://dx.doi.org/10.1007/978-3-662-45611-8_23

[33] C. Dobraunig, M. Eichlseder, F. Mendel, M. Schläffer, Ascon v1 (2014), submission to CAESAR competition
[34] C. Dobraunig, M. Eichlseder, F. Mendel, M. Schläffer, Ascon v1.1 (2015), submission to CAESAR competition
[35] FIPS 202, SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions (2015)
[36] M. Fischlin, J. Coron, (eds.), Advances in Cryptology—EUROCRYPT 2016—35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8–12, 2016, Proceedings, Part I, Lecture Notes in Computer Science, vol. 9665 (Springer, 2016)
[37] E. Fleischmann, C. Forler, S. Lucks, Mcoe: A family of almost foolproof on-line authenticated encryption schemes, in A. Canteaut, (ed.) Fast Software Encryption—19th International Workshop, FSE 2012, Washington, DC, USA, March 19–21, 2012. Revised Selected Papers. Lecture Notes in Computer Science, vol. 7549 (Springer, 2012), pp. 196–215
[38] P. Gazi, K. Pietrzak, S. Tessaro, The exact PRF security of truncation: Tight bounds for keyed sponges and truncated CBC, in Gennaro and Robshaw [40], pp. 368–387
[39] P. Gazi, S. Tessaro, Provably robust sponge-based prngs and kdfs, in Fischlin and Coron [36], pp. 87–116
[40] R. Gennaro, M. Robshaw, (eds.), Advances in Cryptology—CRYPTO 2015—35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16–20, 2015, Proceedings, Part I, Lecture Notes in Computer Science, vol. 9215, (Springer, 2015)
[41] M. Girault, J. Stern, On the length of cryptographic hash-values used in identification schemes, in Y. Desmedt, (ed.) Advances in Cryptology—CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21–25, 1994, Proceedings. Lecture Notes in Computer Science, vol. 839 (Springer, 1994), pp. 202–215
[42] D. Gligoroski, H. Mihajloska, S. Samardjiska, H. Jacobsen, M. El-Hadedy, R. Jensen, π-Cipher v1 (2014), submission to CAESAR competition
[43] D. Gligoroski, H. Mihajloska, S. Samardjiska, H. Jacobsen, M. El-Hadedy, R. Jensen, π-Cipher v2.0 (2015), submission to CAESAR competition
[44] R. Granger, P. Jovanovic, B. Mennink, S. Neves, Improved masking for tweakable blockciphers with applications to authenticated encryption, in Fischlin and Coron [36], pp. 263–293
[45] J. Guo, T. Peyrin, A. Poschmann, The PHOTON family of lightweight hash functions, in P. Rogaway, (ed.) Advances in Cryptology—CRYPTO 2011—31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14–18, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6841 (Springer, 2011), pp. 222–239
[46] S. Hirose, K. Ideguchi, H. Kuwakado, T. Owada, B. Preneel, H. Yoshida, A lightweight 256-bit hash function for hardware and low-end devices: Lesamnta-lw, in K.H. Rhee, D. Nyang, (eds.) Information Security and Cryptology—ICISC 2010—13th International Conference, Seoul, Korea, December 1–3, 2010, Revised Selected Papers. Lecture Notes in Computer Science, vol. 6829 (Springer, 2010), pp. 151–168
[47] S. Hirose, H. Kuwakado, H. Yoshida, Compression functions using a dedicated blockcipher for lightweight hashing, in H. Kim, (ed.) Information Security and Cryptology—ICISC 2011—14th International Conference, Seoul, Korea, November 30–December 2, 2011. Revised Selected Papers. Lecture Notes in Computer Science, vol. 7259 (Springer, 2011), pp. 346–364
[48] V.T. Hoang, T. Krovetz, P. Rogaway, Robust authenticated-encryption AEZ and the problem that it solves, in E. Oswald, M. Fischlin, (eds.) Advances in Cryptology—EUROCRYPT 2015—34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26–30, 2015, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9056 (Springer, 2015), pp. 15–44
[49] V.T. Hoang, S. Tessaro, The multi-user security of double encryption, in J. Coron, J.B. Nielsen, (eds.) Advances in Cryptology—EUROCRYPT 2017—36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30–May 4, 2017, Proceedings, Part II. Lecture Notes in Computer Science, vol. 10211 (2017), pp. 381–411
[50] A. Hoorfar, M. Hassani, Inequalities on the Lambert W function and hyperpower function. J. Inequal. Pure Appl. Math. 9(2) (2008)
[51] T. Iwata, K. Ohashi, K. Minematsu, Breaking and repairing GCM security proofs, in R. Safavi-Naini, R. Canetti, (eds.) Advances in Cryptology—CRYPTO 2012—32nd Annual Cryptology Conference, Santa

Barbara, CA, USA, August 19–23, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7417 (Springer, 2012), pp. 31–49
[52] É. Jaulmes, A. Joux, F. Valette, On the security of randomized CBC-MAC beyond the birthday paradox limit: A new construction, in J. Daemen, V. Rijmen, (eds.) Fast Software Encryption, 9th International Workshop, FSE 2002, Leuven, Belgium, February 4–6, 2002, Revised Papers. Lecture Notes in Computer Science, vol. 2365 (Springer, 2002), pp. 237–251
[53] P. Jovanovic, A. Luykx, B. Mennink, Beyond 2 c/2 security in sponge-based authenticated encryption modes, in Sarkar and Iwata [87], pp. 85–104
[54] L.R. Knudsen, F. Mendel, C. Rechberger, S.S. Thomsen, Cryptanalysis of MDC-2, in A. Joux, (ed.) Advances in Cryptology—EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26–30, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5479 (Springer, 2009), pp. 106–120
[55] T. Krovetz, P. Rogaway, The software performance of authenticated-encryption modes, in A. Joux, (ed.) Fast Software Encryption—18th International Workshop, FSE 2011, Lyngby, Denmark, February 13–16, 2011, Revised Selected Papers. Lecture Notes in Computer Science, vol. 6733 (Springer, 2011), pp. 306–327
[56] U.M. Maurer, R. Renner, C. Holenstein, Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology, in M. Naor, (ed.) Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19–21, 2004, Proceedings. Lecture Notes in Computer Science, vol. 2951 (Springer, 2004), pp. 21–39
[57] D.A. McGrew, J. Viega, The security and performance of the galois/counter mode (GCM) of operation, in A. Canteaut, K. Viswanathan, (eds.) Progress in Cryptology—INDOCRYPT 2004, 5th International Conference on Cryptology in India, Chennai, India, December 20–22, 2004, Proceedings. Lecture Notes in Computer Science, vol. 3348 (Springer, 2004), pp. 343–355
[58] F. Mendel, S. Thomsen, An Observation on JH-512. Available online (2008)
[59] B. Mennink, XPX: generalized tweakable even-mansour with improved security guarantees, in Robshaw and Katz [76], pp. 64–94
[60] B. Mennink, R. Reyhanitabar, D. Vizár, Security of full-state keyed sponge and duplex: Applications to authenticated encryption, in T. Iwata, J.H. Cheon, (eds.) Advances in Cryptology—ASIACRYPT 2015—21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29–December 3, 2015, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9453 (Springer, 2015), pp. 465–489
[61] H. Mihajloska, B. Mennink, D. Gligoroski, π-Cipher with Intermediate Tags (2016), available online
[62] B. Minaud, Re: CBEAM Withdrawn as of today! (2014), CAESAR mailing list
[63] K. Minematsu, Parallelizable rate-1 authenticated encryption from pseudorandom functions, in P.Q. Nguyen, E. Oswald, (eds.) Advances in Cryptology—EUROCRYPT 2014—33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11–15, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8441 (Springer, 2014), pp. 275–292
[64] M. Mitzenmacher, E. Upfal, (eds.), Probability and Computing: Randomized Algorithms and Probabilistic Analysis. (Cambridge University Press, New York, 2005)
[65] P. Morawiecki, K. Gaj, E. Homsirikamol, K. Matusiewicz, J. Pieprzyk, M. Rogawski, M. Srebrny, M. Wójcik, ICEPOLE v1 (2014), submission to CAESAR competition
[66] P. Morawiecki, K. Gaj, E. Homsirikamol, K. Matusiewicz, J. Pieprzyk, M. Rogawski, M. Srebrny, M. Wójcik, ICEPOLE v2 (2015), submission to CAESAR competition
[67] R. Motwani, P. Raghavan, (eds.), Randomized Algorithms. (Cambridge University Press, New York, 1995)
[68] Y. Naito, Y. Sasaki, L. Wang, K. Yasuda, Generic state-recovery and forgery attacks on chopmd-mac and on NMAC/HMAC, in K. Sakiyama, M. Terada, (eds.) Advances in Information and Computer Security—8th International Workshop on Security, IWSEC 2013, Okinawa, Japan, November 18–20, 2013, Proceedings. Lecture Notes in Computer Science, vol. 8231 (Springer, 2013), pp. 83–98
[69] Y. Naito, K. Yasuda, New bounds for keyed sponges with extendable output: Independence between capacity and message length, in T. Peyrin, (ed.) Fast Software Encryption—23rd International Conference, FSE 2016, Bochum, Germany, March 20–23, 2016, Revised Selected Papers. Lecture Notes in Computer Science, vol. 9783 (Springer, 2016), pp. 3–22

[70] I. Nikolic, L. Wang, S. Wu, Cryptanalysis of round-reduced \mathtttled, In S. Moriai, (ed.) Fast Software Encryption—20th International Workshop, FSE 2013, Singapore, March 11–13, 2013. Revised Selected Papers. Lecture Notes in Computer Science, vol. 8424 (Springer, 2013), pp. 112–129
[71] F.W.J. Olver, D.W. Lozier, R.F. Boisvert, C.W. Clark, (eds.), NIST Handbook of Mathematical Functions. (Cambridge University Press, New York, 2010)
[72] T. Peyrin, Y. Seurin, Counter-in-tweak: Authenticated encryption modes for tweakable block ciphers, in Robshaw and Katz [76], pp. 33–63
[73] B. Preneel, R. Govaerts, J. Vandewalle, On the power of memory in the design of collision resistant hash functions, in J. Seberry, Y. Zheng, (eds.) Advances in Cryptology—AUSCRYPT '92, Workshop on the Theory and Application of Cryptographic Techniques, Gold Coast, Queensland, Australia, December 13–16, 1992, Proceedings. Lecture Notes in Computer Science, vol. 718 (Springer, 1992), pp. 105–121
[74] M. Raab, A. Steger, "Balls into Bins"—A simple and tight analysis, in M. Luby, J.D.P. Rolim, M.J. Serna, (eds.) Randomization and Approximation Techniques in Computer Science, Second International Workshop, RANDOM'98, Barcelona, Spain, October 8–10, 1998, Proceedings. Lecture Notes in Computer Science, vol. 1518 (Springer, 1998), pp. 159–170
[75] R. Reyhanitabar, Do Sponge-based AE modes have beyond $2^{c/2}$ "Security"? (2014), CAESAR mailing list
[76] M. Robshaw, J. Katz, (eds.), Advances in Cryptology—CRYPTO 2016—36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14–18, 2016, Proceedings, Part I, Lecture Notes in Computer Science, vol. 9814 (Springer, 2016)
[77] P. Rogaway, Authenticated-encryption with associated-data, in V. Atluri, (ed.) Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, Washington, DC, USA, November 18–22, 2002. (ACM, 2002), pp. 98–107
[78] P. Rogaway, Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC, in P.J. Lee, (ed.) Advances in Cryptology—ASIACRYPT 2004, 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, December 5–9, 2004, Proceedings. Lecture Notes in Computer Science, vol. 3329 (Springer, 2004), pp. 16–31
[79] P. Rogaway, M. Bellare, J. Black, T. Krovetz, OCB: a block-cipher mode of operation for efficient authenticated encryption, in M.K. Reiter, P. Samarati, (eds.) CCS 2001, Proceedings of the 8th ACM Conference on Computer and Communications Security, Philadelphia, Pennsylvania, USA, November 6–8, 2001 (ACM, 2001), pp. 196–205
[80] P. Rogaway, T. Shrimpton, A provable-security treatment of the key-wrap problem, in Vaudenay [93], pp. 373–390
[81] M.J.O. Saarinen, Authenticated Encryption from GOST R 34.11-2012 LPS Permutation, in CTCrypt 2014 (2014)
[82] M.O. Saarinen, Beyond modes: Building a secure record protocol from a cryptographic sponge permutation, in Benaloh [15], pp. 270–285
[83] M.O. Saarinen, CBEAM: efficient authenticated encryption from feebly one-way ϕ functions, in Benaloh [15], pp. 251–269
[84] M.J.O. Saarinen, CBEAM r1 (2014), submission to CAESAR competition
[85] M.J.O. Saarinen, STRIBOB r1 (2014), submission to CAESAR competition
[86] M.J.O. Saarinen, B.B. Brumley, STRIBOB r2: "WHIRLBOB" (2015), submission to CAESAR competition
[87] P. Sarkar, T. Iwata, (eds.), Advances in Cryptology—ASIACRYPT 2014—20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7–11, 2014. Proceedings, Part I, Lecture Notes in Computer Science, vol. 8873 (Springer, 2014)
[88] Y. Sasaki, L. Wang, Generic attacks on strengthened HMAC: n-bit secure HMAC requires key in all blocks, in M. Abdalla, R.D. Prisco, (eds.) Security and Cryptography for Networks—9th International Conference, SCN 2014, Amalfi, Italy, September 3–5, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8642 (Springer, 2014), pp. 324–339
[89] Y. Sasaki, K. Yasuda, How to incorporate associated data in sponge-based authenticated encryption, in K. Nyberg, (ed.) Topics in Cryptology—CT-RSA 2015, The Cryptographer's Track at the RSA Conference 2015, San Francisco, CA, USA, April 20–24, 2015. Proceedings. Lecture Notes in Computer Science, vol. 9048 (Springer, 2015), pp. 353–370

[90] Y. Sasaki, K. Yasuda, Directly Evaluating Multi-Collisions and Improving Security Bounds. Symmetric Cryptography, Dagstuhl Seminar 16021 (2016)
[91] K. Suzuki, D. Tonien, K. Kurosawa, K. Toyota, Birthday paradox for multi-collisions, in M.S. Rhee, B. Lee, (eds.) Information Security and Cryptology—ICISC 2006, 9th International Conference, Busan, Korea, November 30–December 1, 2006, Proceedings. Lecture Notes in Computer Science, vol. 4296 (Springer, 2006), pp. 29–40
[92] K. Suzuki, D. Tonien, K. Kurosawa, K. Toyota, Birthday paradox for multi-collisions. IEICE Trans. 91-A(1), 39–45 (2008)
[93] S. Vaudenay, (ed.), Advances in Cryptology—EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28–June 1, 2006, Proceedings, Lecture Notes in Computer Science, vol. 4004 (Springer, 2006)
[94] D. Vizár, Ciphertext forgery on HANUMAN. Cryptology ePrint Archive, Report 2016/697 (2016)
[95] D. Whiting, R. Housley, N. Ferguson, AES Encryption and Authentication Using CTR Mode and CBC-MAC. IEEE 802.11-02/001r2 (2002)
[96] H. Wu, The Hash Function JH (2011), submission to NIST's SHA-3 competition