Full-State Keyed Duplex With Built-In Multi-User Support

Joan Daemen[1,2], Bart Mennink[1,3], and Gilles Van Assche[2]

[1] Digital Security Group, Radboud University, Nijmegen, The Netherlands
joan@cs.ru.nl, b.mennink@cs.ru.nl
[2] STMicroelectronics, Diegem, Belgium
gilles.vanassche@st.com
[3] CWI, Amsterdam, The Netherlands

Abstract. The keyed duplex construction was introduced by Bertoni et al. (SAC 2011) and recently generalized to full-state absorption by Mennink et al. (ASIACRYPT 2015). We present a generalization of the full-state keyed duplex that natively supports multiple instances by design, and perform a security analysis that improves over that of Mennink et al. in terms of a more modular security analysis and a stronger and more adaptive security bound. Via the introduction of an additional parameter to the analysis, our bound demonstrates a significant security improvement in case of nonce-respecting adversaries. Furthermore, by supporting multiple instances by design, instead of adapting the security model to it, we manage to derive a security bound that is largely independent of the number of instances.

Keywords: Duplex construction, full-state, distinguishing bounds, authenticated encryption.

1 Introduction

Bertoni et al. [8] introduced the sponge construction as an approach to design hash functions with variable output length (later called extendable output functions (XOF)). The construction faced rapid traction in light of NIST's SHA-3 competition, with multiple candidates inspired by the sponge methodology. Keccak, the eventual winner of the competition and now standardized as SHA-3 [27], internally uses the sponge construction. The sponge construction found quick adaptation in the area of lightweight hashing [19, 32]. Also beyond the area of hash functions various applications of the sponge construction appeared such as keystream generation and MAC computation [12], reseedable pseudorandom sequence generation [10, 30], and authenticated encryption [11, 14]. In particular, the ongoing CAESAR competition for the development of a portfolio of authenticated encryption schemes has received about a dozen sponge-inspired submissions.

At a high level, the sponge construction operates on a state of b bits. This is split into an inner part of size c bits and an outer part of size r bits, where $b =$

$c+r$. Data absorption and squeezing is done via the outer part, r bits at a time, interleaved with evaluations of a b-bit permutation f. Bertoni et al. [9] proved a bound on the security of the sponge construction in the indifferentiability framework of Maurer et al. [37]. While it was clear from the start that this birthday-type bound in the capacity is tight for the unkeyed use cases, i.e., hashing, for the keyed use cases of the sponge it appeared that a higher level of security could be achieved. This has resulted in an impressive series of papers on the generic security of keyed versions of the sponge, with bounds improving and the construction becoming more efficient.

1.1 Keyed Sponge and Keyed Duplex

Keyed Sponge. Bertoni et al.'s original keyed sponge [13] was simply the sponge with input $(K\|M)$ where K is the key. Chang et al. [21] suggested an alternative where the initial state of the sponge simply contains the key in its inner part. Andreeva et al. [2] generalized and improved the analyses of both the outer- and inner-keyed sponge, and also considered security of these functions in the multi-target setting. In a recent analysis their bounds were improved by Naito and Yasuda in [42]. All of these results, however, stayed relatively close to the (keyless) sponge design that absorbs input in blocks of r bits in the outer part of the state. It turned out that, thanks to the secrecy of part of the state after key injection, one can absorb data over the full state, and therewith achieve maximal compression. Full-state absorbing was first explicitly proposed in a variant of sponge for computing MACs: donkeySponge [14]. It also found application in various recent sponge-inspired designs, such as Chaskey [41].

Nearly tight bounds for the full-state absorbing keyed sponge were given by Gaži et al. [29] but their analysis was limited to the case of fixed output length. Mennink et al. [38] generalized and formalized the idea of the full-state keyed sponge and presented an improved security bound for the general case where the output length is variable.

Keyed Duplex. Whereas the keyed sponge serves message authentication and stream encryption, authenticated encryption is mostly done via the keyed duplex construction [11]. This is a stateful construction that consists of an initialization interface and a duplexing interface. Initialization resets the state and a duplexing call absorbs a data block of at most $r-1$ bits, applies the underlying permutation f and squeezes at most r bits. Bertoni et al. [11] proved that the output of duplexing calls can be simulated by calls to a sponge, a fortiori making duplex as strong as sponge.

Mennink et al. [38] introduced the full-state keyed duplex and derived a security bound on this construction with dominating terms:

$$\frac{\mu N}{2^k} + \frac{M^2}{2^c}\,. \tag{1}$$

Here M is the data complexity (total number of initialization and duplexing calls), N the computational complexity (total number of offline calls to f), $\mu \leq$

$2M$ is a term called the "multiplicity," and k the size of the key. This security bound was derived by describing the full-state keyed duplex in terms of the full-state keyed sponge. A naive bounding of μ (to cover the strongest possible adversary) yields a dominating term of the form $2MN/2^k$, implying a security strength erosion of $\log_2 M$ with respect to exhaustive key search.

The duplex construction finds multiple uses in the CAESAR competition [20] in the embodiment of the authenticated encryption mode SpongeWrap [11] or close variants of it. The recent line of research on improving bounds of sponge-inspired authenticated encryption schemes, set by Jovanovic et al. [35], Sasaki and Yasuda [46], and Reyhanitabar et al. [44], can be seen as an analysis of a specific use case of the keyed duplex. The Full-State SpongeWrap [38], an authenticated encryption scheme designed from the full-state keyed duplex, improves over these results. Particularly, the idea already found application in the Motorist mode of the CAESAR submission Keyak [16].

Trading Sponge for Duplex. As said, the duplex can be simulated by the sponge, but not the other way around. This is the case because duplex pads each input block and cannot simulate sponge inputs with, e.g., long sequences of zeroes. It is therefore natural that Mennink et al. [38] derived a security bound on the full-state keyed duplex *by viewing it as an extension to the full-state keyed sponge*. However, we observe that the introduction of full-state absorption changes that situation: the full-state keyed duplex can simulate the full-state keyed sponge. All keyed usages of the sponge can be described quite naturally as application of the keyed duplex and it turns out that proving security of keyed duplex is easier than that of keyed sponge. Therefore, in keyed use cases, the duplex is preferred as basic building block over the sponge.

1.2 Multi-Target Security

The problem of multi-target security of cryptographic designs has been acknowledged and analyzed for years. Biham [17] considered the security of blockciphers in the multi-target setting and shows that the security strength can erode to half the key size if data processed by sufficiently many keys is available. Various extensions have subsequently appeared [7,18,34]. It has been demonstrated (see, e.g., [5] for public key encryption and [22] for message authentication codes) that the security of a scheme in the multi-target setting can be reduced to the security in the single-target setting, at a security loss proportional to the number of keys used.

However, in certain cases, a dedicated analysis in the multi-target setting could render improved bounds. Andreeva et al. [2] considered the security of the outer- and inner-keyed sponge in the multi-target setting, a proof which internally featured a security analysis of the Even-Mansour blockcipher in the multi-target setting. The direction of multi-target security got subsequently popularized by Mouha and Luykx [40], leading to various multi-target security results [4,33] with security bounds (almost) independent of the number of targets involved.

1.3 Our Contribution

We present a generalization of the full-state keyed duplex that facilitates multiple instances by design (Section 2.2). This generalization is realized via the formalization of a state initialization function that has access to a key array $\mathbf{K}$ consisting of u keys of size k, generated following a certain distribution. Given as input a key index δ and an initialization vector iv, it initializes the state using iv and the δth key taken from $\mathbf{K}$. We capture its functional behavior under the name of an *extendable input function* (XIF) and explicitly define its idealized instance.

Unlike the approach of Mennink et al. [38], who viewed the full-state keyed duplex as an extension to the full-state keyed sponge, our analysis is a dedicated analysis of the full-state keyed duplex. To accommodate bounds for different use cases, we have applied a re-phasing to the definition of the keyed duplex. In former definitions of the (keyed) duplex, a duplexing call consisted of input injection, applying the permutation f, and then output extraction. In our new definition, the processing is as follows: first the permutation f, then output extraction, and finally input injection. This adjustment reflects a property present in practically all modes based on the keyed duplex, namely that the user (or adversary) must provide the input before knowing the previous output. The re-phasing allows us to prove a bound on keyed duplex that is tight even for those use cases. The fact that, in previous definitions, an adversary could see the output before providing the input allowed it to force the outer part of the state to a value of its choice, and as such gave rise to a term in the bound at worst $MN/2^c$ and at best $\mu N/2^c$, where μ is a term that reflects a property of the transcript that needs to be bound by out-of-band reasonings.

Alongside the re-phasing, we have eliminated the term μ and express the bound purely as a function of the adversary's capabilities. Next to the total offline complexity N, i.e., the number queries the adversary can make to f and the total online complexity M, i.e., the total number of construction queries (to keyed duplex or ideal XIF), we introduce two metrics: L and Ω, both reflecting the ability of the adversary to force the outer part of the state to a value of its choice. The metric L counts the number of construction queries with repeated path (intuitively, a concatenation of all data blocks up to a certain permutation call), which may typically occur in MAC functions and authenticated encryption schemes that do not impose nonce uniqueness. The metric Ω counts the number of construction queries where the adversary can overwrite the outer part of the state. Such a possibility may occur in authenticated encryption schemes that release unverified decrypted ciphertext (cf., [1]).

We prove in Section 4 a bound on the advantage of distinguishing a full-state keyed duplex from an ideal XIF in a multi-target setting. We here give the bound for several settings, all of which having multiple keys sampled uniformly at random without replacement. For adversaries with the ability to send queries with repeated paths and queries that overwrite the outer part of the state, the

dominating terms in our bound are:

$$\frac{q_{\mathrm{iv}}N}{2^k} + \frac{(L+\Omega)N}{2^c}\,. \tag{2}$$

The metric q_{iv} denotes the maximum number of sessions started with the same iv but different keys. For adversaries that cannot send queries with repeated paths or send queries that overwrite the outer part of the state, one of the dominating terms depends on the occurrence of multicollisions via a coefficient $\nu_{r,c}^{M}$ that is fully determined by the data complexity M and parameters r and c (see Section 6.5, and particularly Figure 4). For wide permutations we can have large rates (i.e., $r > 2\log_2(M) + c$) and the dominating terms in our bound are:

$$\frac{q_{\mathrm{iv}}N}{2^k} + \frac{N}{2^{c-1}}\,. \tag{3}$$

For relatively small rates the data complexity can be such that $M > 2^{r-1}$ and for that range the dominating terms are upper bounded by (assuming $\nu_{r,c}^{2M} \leq \frac{bM}{2^{r+1}}$):

$$\frac{q_{\mathrm{iv}}N}{2^k} + \frac{bMN}{2^b} + \frac{M^2}{2^b}\,. \tag{4}$$

For the case in-between where M is in the range $2^{(r-c)/2} < M \leq 2^{r-1}$, the bound becomes (assuming $\nu_{r,c}^{2M} \leq \min(b/\log\frac{2^r}{2M}, b/4)$):

$$\frac{q_{\mathrm{iv}}N}{2^k} + \frac{bN}{\max(4, r-1-\log_2 M)2^{c-1}}\,. \tag{5}$$

This bound is valid for permutation widths of 200 and above. These bounds are significantly better than that of [38].

Concretely, in implementations of duplex-based authenticated encryption schemes that respect the nonce requirement and do not release unverified plaintext, we have $L = \Omega = 0$. Assuming keys are randomly sampled without replacement, the generic security is governed by (3), (4), or (5). Depending on the parameters, a scheme is either in case (3), or case (4-5), where a transition happens for $M = 2^{r-1}$. Table 1 summarizes achievable security strengths for the duplex-based CAESAR contenders.

Our general security bound, covering among others a broader spectrum of key sampling distributions, is given in Theorem 1. It is noteworthy that, via the built-in support of multiple targets, we manage to obtain a security bound that is largely independent of the number of users u: the only appearance is in the key guessing part, $q_{\mathrm{iv}}N/2^k$, which shows an expected decrease in the security strength of exhaustive key search by a term $\log_2 q_{\mathrm{iv}}$. Note that security erosion can be avoided altogether by requiring iv to be a global nonce, different for each initialization call (irrespective of the used key).

Our analysis improves over the one of [38] in multiple aspects. First, our security bound shows less security erosion for increasing data complexities. Whereas in (1) security strength erodes to $k - \log_2 M$, in (2) this is $c - \log_2(L + \Omega)$ with

Table 1: Application of our analysis to Ketje, Ascon, NORX, and Keyak. For the nonce misuse case, we consider $L + \Omega = M/2$. A "Strength" equal to s means that it requires a computational effort 2^s to distinguish. Here, $a = \log_2(Mr)$.

		Parameters			Respecting (Eqns. (3-5))		Misuse (Eqn. (2))
Scheme		b	c	r	Strength	Eqn.	Strength
Ketje [15]	Jr.	200	184	16	$\min\{196 - a, 177\}$	(4,5)	$189 - a$
	Sr.	400	368	32	$\min\{396 - a, 360\}$	(4,5)	$374 - a$
Ascon [24]	128	320	256	64	$\min\{317 - a, 248\}$	(4,5)	$263 - a$
	128a	320	192	128	$\min\{318 - a, 184\}$	(4,5)	$200 - a$
NORX [3]	32	512	128	384	127	(3)	$137 - a$
	64	1024	256	768	255	(3)	$266 - a$
Keyak [16]	River	800	256	544	255	(3)	$266 - a$
	Lake	1600	256	1344	255	(3)	$267 - a$

$L + \Omega < M$. By taking $c > k + \log_2 M_{\max}$ with $M_{\max}$ some upper bound on the amount of data an adversary can get its hands on, one can guarantee that this term does not allow attacks faster than exhaustive key search.

Second, via the use of parameters L and Ω our bound allows for a more flexible interpretation and a wide range of use cases. For example, in stream ciphers, $L = \Omega = 0$ by design. This also holds for most duplex-based authenticated encryption schemes in the case of nonce-respecting adversaries that cannot obtain unverified decrypted ciphertexts.

Third, even in the general case (with key size taken equal to c bits and no nonce restriction on iv), our bound still improves over the one of [38] by replacing the multiplicity metric, that can only be evaluated a posteriori, by the metrics L and Ω, that reflect what the adversary can do.

Fourth, in our approach we address the multi-key aspect natively. This allows us to determine the required set of properties on the joint distribution of all keys under attack. Theorem 1 works for arbitrary key sampling techniques with individual keys of sufficient min-entropy and the probability that two keys in the array collide is small enough, and demonstrates that the full-state keyed duplex remains secure even if the keys are not independently and uniformly randomly distributed.

Finally, we perform an analysis on the contribution of outer-state multi-collisions to the bound that is of independent interest. This analysis strongly contributes to the tightness of our bounds, as we illustrate in the Stairway to Heaven graph in Figure 4.

1.4 Notation

For an integer $n \in \mathbb{N}$, we denote $\mathbb{Z}_n = \{0, \ldots, n-1\}$ and by $\mathbb{Z}_2^n$ the set of bit strings of length n. $\mathbb{Z}_2^*$ denotes the set of bit strings of arbitrary length. For two bit strings $s, t \in \mathbb{Z}_2^n$, their bitwise addition is denoted $s + t$. The expression $\lfloor s \rfloor_\ell$

denotes the bitstring s truncated to its first ℓ bits. A random oracle [6] $\mathcal{RO} : \mathbb{Z}_2^* \to \mathbb{Z}_2^n$ is a function that maps bit strings of arbitrary length to bit strings of some length n. In this paper, the value of n is determined by the context. We denote by $(x)_{(y)}$ the falling factorial power $(x)_{(y)} = x(x-1)\cdots(x-y+1)$.

Throughout this work, b denotes the width of the permutation f. The parameters c and r denote the capacity and rate, where $b = c + r$. For a state value $s \in \mathbb{Z}_2^b$, we follow the general convention to define its outer part by $\overline{s} \in \mathbb{Z}_2^r$ and its inner part by $\widehat{s} \in \mathbb{Z}_2^c$, in such a way that $s = \overline{s}||\widehat{s}$. The key size is conventionally denoted by k, and the number of users by u. Throughout, we assume that $u \leq 2^k$, and regularly use an encoding function $\mathsf{Encode}(\delta) : \mathbb{Z}_u \to \mathbb{Z}_2^k$, mapping integers from $\mathbb{Z}_u$ to k-bit strings in some injective way.

2 Constructions

In Section 2.1, we will discuss the key sampling technique used in this work. The keyed duplex construction is introduced in Section 2.2, and we present its "ideal equivalent," the ideal extendable input function, in Section 2.3. To suit the security analysis, we will also need an in-between hybrid, the randomized duplex, discussed in Section 2.4.

2.1 Key Sampling

Our keyed duplex construction has built-in multi-user support, and we start with a formalization of the key sampling that we consider. At a high level, our formalization is not specific for the keyed duplex, and may be of independent interest for modeling multi-target attacks.

In our formalization the adversary can invoke a keyed object (block cipher, stream cipher, PRF, keyed sponge, ...) with a key selected from a key array $\mathbf{K}$ containing u keys, each of length k bits:

$$\mathbf{K} = (\mathbf{K}[0], \ldots, \mathbf{K}[u-1]) \in \left(\mathbb{Z}_2^k\right)^u.$$

These keys are sampled from the space of k-bit keys according to some distribution $\mathcal{D}_{\mathrm{K}}$. This distribution can, in theory, be anything. In particular, the distribution of the key with index δ may depend on the values of the δ keys sampled before.

Two plausible examples of the key distribution are *random sampling with replacement* and *random sampling without replacement*. In the former case, all keys are generated uniformly at random and pairwise independent, but it may cause problems in case of accidental collisions in the key array. The latter distribution resolves this by generating all keys uniformly at random from the space of values excluding the ones already sampled. A third, more extreme, example of $\mathcal{D}_{\mathrm{K}}$ generates $\mathbf{K}[0]$ uniformly at random and defines all subsequent keys as $\mathbf{K}[\delta] = \mathbf{K}[0] + \mathsf{Encode}(\delta)$.

Different distributions naturally entail different levels of security, and we define two characteristics of a distribution that are relevant for our analysis. Note

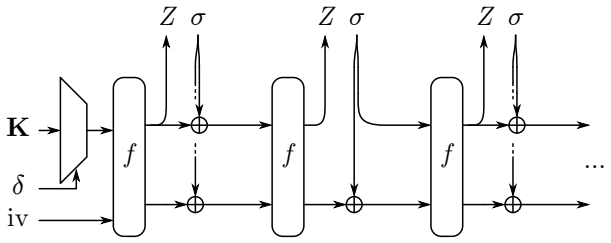

Fig. 1: Full-state keyed duplex construction $\mathrm{KD}^f_{\mathbf{K}}$. In this figure, the sequence of calls is $Z = \mathrm{KD.Init}(\delta, \mathrm{iv}, \sigma, \mathrm{false})$, $Z = \mathrm{KD.Duplexing}(\sigma, \mathrm{true})$, and $Z = \mathrm{KD.Duplexing}(\sigma, \mathrm{false})$.

that the characteristics take u as implicit parameter. The first characteristic is the min-entropy of the individual keys, defined as

$$H_{\min}(\mathcal{D}_{\mathrm{K}}) = -\log_2 \max_{\delta \in \mathbb{Z}_u, a \in \mathbb{Z}_2^k} \Pr(\mathbf{K}[\delta] = a)\,, \tag{6}$$

or in words, minus the binary logarithm of the probability of the key value to be selected with the highest probability. The three example samplings outlined above have min-entropy k, regardless of the value u.

The second characteristic is related to the maximum collision probability between two keys in the array:

$$H_{\mathrm{coll}}(\mathcal{D}_{\mathrm{K}}) = -\log_2 \max_{\substack{\delta, \delta' \in \mathbb{Z}_u \\ \delta \neq \delta'}} \Pr(\mathbf{K}[\delta] = \mathbf{K}[\delta'])\,. \tag{7}$$

Uniform sampling with replacement has maximum collision probability equal to 2^{-k} and so $H_{\mathrm{coll}}(\mathcal{D}_{\mathrm{K}}) = k$. Sampling without replacement and our third example clearly have collision probability zero, giving $H_{\mathrm{coll}}(\mathcal{D}_{\mathrm{K}}) = \infty$.

2.2 Keyed Duplex Construction

The full-state keyed duplex (KD) construction is defined in Algorithm 1, and it is illustrated in Figure 1.

It calls a b-bit permutation f and is given access to an array $\mathbf{K}$ consisting of u keys of size k bits. A user can make two calls: initialization and duplexing calls.

In an initialization call it takes as input a key index δ and a string $\mathrm{iv} \in \mathbb{Z}_2^{b-k}$ and initializes the state as $f(\mathbf{K}[\delta]||\mathrm{iv})$. In the same call, the user receives an r-bit output string Z and injects a b-bit string σ. A duplexing call just performs the latter part: it updates the state by applying f to it, returns to the user an r-bit output string Z and injects a user-provided b-bit string σ.

Both in initialization and duplexing calls, the output string Z is taken from the state prior to the addition of σ to it, but the user has to provide σ before

receiving Z. This is in fact a re-phasing compared to the original definition of the duplex [11] or of the full-state keyed duplex [38], and it aims at better reflecting typical use cases. We illustrate this with the SpongeWrap authenticated encryption scheme [11] and its more recent variants [38]. In this scheme, each plaintext block is typically encrypted by (i) applying f, (ii) fetching a block of key stream, (iii) adding the key stream and plaintext blocks to get a ciphertext block, and (iv) adding the plaintext block to the outer part of the state. By inspecting Algorithm 3 in [11], there is systematically a delay between the production of key stream and its use, requiring to buffer a key stream block between the (original) duplexing calls. In contrast, our re-phased calls better match the sequence of operations.

The flag in the initialization and duplexing calls is required to implement decryption in SpongeWrap and variants. In that case, the sequence of operations is the same as above, except that step (iii) consists of adding the key stream and ciphertext blocks to get a plaintext block. However, a user would need to see the keystream block before being able to add the plaintext block in step (iv). One can see, however, that step (iv) is equivalent to overwriting the outer part of the state with the ciphertext block. Switching between adding the plaintext block (for encryption) and overwriting with the ciphertext block (for decryption) is the purpose of the flag. The usage of the flag, alongside the re-phasing is depicted in Figure 1.

Note that in Algorithm 1 in the case that the flag is true, the outer part of the state is overwritten with $\overline{\sigma}$. For consistency with the algorithms of constructions we will introduce shortly, this is formalized as bitwise adding Z to $\overline{\sigma}$ before its addition to the state if flag is true. Alternatively, one could define an authenticated encryption mode that does not allow overwriting the state with the ciphertext block C. For example, encryption would be $C = P + \mathbf{M} \times Z$, with P the plaintext block and $\mathbf{M}$ a simple invertible matrix. Upon decryption, the outer part of the state then becomes $C + (\mathbf{M} + \mathbf{I}) \times Z$. If $\mathbf{M}$ is chosen such that $\mathbf{M} + \mathbf{I}$ is invertible, the adversary has no control over the outer part of the state after the duplexing call. This would require changing "$\overline{\sigma} \leftarrow \overline{\sigma} + Z$" into "$\overline{\sigma} \leftarrow \overline{\sigma} + \mathbf{M} \times Z$" in Algorithm 1.

2.3 Ideal Extendable Input Function

We define an ideal extendable input function (IXIF) in Algorithm 2. It has the same interface as KD, but instead it uses a random oracle $\mathcal{RO} : \mathbb{Z}_2^* \to \mathbb{Z}_2^r$ to generate its responses. In particular, every initialization call initializes a Path as $\mathsf{Encode}(\delta)||\text{iv}$. In both initialization and duplexing calls, an r-bit output is generated by evaluating $\mathcal{RO}(\mathsf{Path})$ and the b-bit input string σ is absorbed by appending it to the Path. Figure 2 has an illustration of IXIF (at the right).

Note that IXIF properly captures the random equivalent of the full-state keyed duplex: it simply returns random values from $\mathbb{Z}_2^r$ for every new path, and repeated paths result in identical responses. IXIF is in fact almost equivalent to the duplex as presented by Mennink et al. [38]: as a matter of fact, when (i) not considering multiple keys for our construction and (ii) avoiding overlap of the

Algorithm 1 Full-state keyed duplex construction $\mathrm{KD}^f_{\mathbf{K}}$

Require: $r < b$, $k \leq b$

Instantiation: $\mathrm{KD} \leftarrow \mathrm{KD}^f_{\mathbf{K}}$ with $\mathbf{K}$ an array of u keys of size k
Key array: $\mathrm{KD}.\mathbf{K} \xleftarrow{\mathcal{D}_\mathrm{K}} \mathbf{K}$

Interface: $Z = \mathrm{KD.Init}(\delta, \mathrm{iv}, \sigma, \mathrm{flag})$ with $\delta \in \mathbb{Z}_u$, $\mathrm{iv} \in \mathbb{Z}_2^{b-k}$, $\sigma \in \mathbb{Z}_2^b$, $\mathrm{flag} \in \{\mathrm{true}, \mathrm{false}\}$, and $Z \in \mathbb{Z}_2^r$
$s \leftarrow f(\mathbf{K}[\delta]||\mathrm{iv})$
$Z \leftarrow \lfloor s \rfloor_r$
if flag = true **then** $\overline{\sigma} \leftarrow \overline{\sigma} + Z$
$s \leftarrow s + \sigma$
return Z

Interface: $Z = \mathrm{KD.Duplexing}(\sigma, \mathrm{flag})$ with $\sigma \in \mathbb{Z}_2^b$, $\mathrm{flag} \in \{\mathrm{true}, \mathrm{false}\}$, and $Z \in \mathbb{Z}_2^r$
$s \leftarrow f(s)$
$Z \leftarrow \lfloor s \rfloor_r$
if flag = true **then** $\overline{\sigma} \leftarrow \overline{\sigma} + Z$
$s \leftarrow s + \sigma$
return Z

iv with the key (as possible in the construction of [38]), the ideal functionalities are the same. In our analysis, we do not consider overlap of the iv with the key as (i) it unnecessarily complicates the analysis and (ii) we discourage it as it may be a security risk if the keys in the key array $\mathbf{K}$ are not independently and uniformly randomly distributed.

2.4 Randomized Duplex Construction

To simplify our security analysis, we introduce a hybrid algorithm lying in-between KD and IXIF: the full-state randomized duplex (RD) construction. It is defined in Algorithm 3. It again has the same interface as KD, but the calls to the permutation f and the access to a key array $\mathbf{K}$ have been replaced by two primitives: a uniformly random injective mapping $\phi : \mathbb{Z}_u \times \mathbb{Z}_2^{b-k} \rightarrow \mathbb{Z}_2^b$, and a uniformly random b-bit permutation π. The injective mapping ϕ replaces the keyed state initialization by directly mapping an input (δ, iv) to a b-bit state value. The permutation π replaces the evaluations of f in the duplexing calls. In our use of RD, ϕ and π will be secret primitives. Figure 2 has an illustration of RD (at the left).

3 Security Setup

The security analysis in this work is performed in the *distinguishability framework* where one bounds the advantage of an adversary $\mathcal{A}$ in distinguishing a real system from an ideal system.

Algorithm 2 Ideal extendable input function $\mathrm{IXIF}^{\mathcal{RO}}$

Instantiation: $\mathrm{IXIF} \leftarrow \mathrm{IXIF}^{\mathcal{RO}}$
  Path: IXIF.$\mathsf{Path} \leftarrow$ empty string

Interface: $Z = \mathrm{IXIF.Init}(\delta, \mathrm{iv}, \sigma, \mathrm{flag})$ with $\delta \in \mathbb{Z}_u$, $\mathrm{iv} \in \mathbb{Z}_2^{b-k}$, $\sigma \in \mathbb{Z}_2^b$, $\mathrm{flag} \in \{\mathrm{true}, \mathrm{false}\}$, and $Z \in \mathbb{Z}_2^r$
  $\mathsf{Path} \leftarrow \mathsf{Encode}(\delta)||\mathrm{iv}$
  $Z \leftarrow \mathcal{RO}(\mathsf{Path})$
  if flag = true **then** $\overline{\sigma} \leftarrow \overline{\sigma} + Z$
  $\mathsf{Path} \leftarrow \mathsf{Path}||\sigma$
  return Z

Interface: $Z = \mathrm{IXIF.Duplexing}(\sigma, \mathrm{flag})$ with $\sigma \in \mathbb{Z}_2^b$ flag $\in \{\mathrm{true}, \mathrm{false}\}$, and $Z \in \mathbb{Z}_2^r$
  $Z \leftarrow \mathcal{RO}(\mathsf{Path})$
  if flag = true **then** $\overline{\sigma} \leftarrow \overline{\sigma} + Z$
  $\mathsf{Path} \leftarrow \mathsf{Path}||\sigma$
  return Z

Definition 1. *Let $\mathcal{O}, \mathcal{P}$ be two collections of oracles with the same interface. The advantage of an adversary $\mathcal{A}$ in distinguishing $\mathcal{O}$ from $\mathcal{P}$ is defined as*

$$\Delta_{\mathcal{A}}(\mathcal{O}\ ;\ \mathcal{P}) = \left|\Pr\left(\mathcal{A}^{\mathcal{O}} \to 1\right) - \Pr\left(\mathcal{A}^{\mathcal{P}} \to 1\right)\right| .$$

Our proofs in part use the H-coefficient technique from Patarin [43]. We will follow the adaptation of Chen and Steinberger [23]. Consider any information-theoretic deterministic adversary $\mathcal{A}$ whose goal is to distinguish $\mathcal{O}$ from $\mathcal{P}$, with its advantage denoted

$$\Delta_{\mathcal{A}}(\mathcal{O}\ ;\ \mathcal{P}) .$$

The interaction of $\mathcal{A}$ with its oracle, either $\mathcal{O}$ or $\mathcal{P}$, will be stored in a *transcript* τ. Denote by $D_{\mathcal{O}}$ (resp. $D_{\mathcal{P}}$) the probability distribution of transcripts that can be obtained from interaction with $\mathcal{O}$ (resp. $\mathcal{P}$). Call a transcript τ *attainable* if it can be obtained from interacting with $\mathcal{P}$, hence if $\Pr(D_{\mathcal{P}} = \tau) > 0$. Denote by $\mathcal{T}$ the set of attainable transcripts, and consider any partition $\mathcal{T} = \mathcal{T}_{\mathrm{good}} \cup \mathcal{T}_{\mathrm{bad}}$ of the set of attainable transcripts into "good" and "bad" transcripts. The H-coefficient technique states the following [23].

Lemma 1 (H-coefficient Technique). *Consider a fixed information-theoretic deterministic adversary $\mathcal{A}$ whose goal is to distinguish $\mathcal{O}$ from $\mathcal{P}$. Let ε be such that for all $\tau \in \mathcal{T}_{\mathrm{good}}$:*

$$\frac{\Pr(D_{\mathcal{O}} = \tau)}{\Pr(D_{\mathcal{P}} = \tau)} \geq 1 - \varepsilon . \tag{8}$$

Then, $\Delta_{\mathcal{A}}(\mathcal{O}\ ;\ \mathcal{P}) \leq \varepsilon + \Pr(D_{\mathcal{P}} \in \mathcal{T}_{\mathrm{bad}})$.

Algorithm 3 Full-state randomized duplex construction $\mathrm{RD}^{\phi,\pi}$

Require: $r < b$

Instantiation: $\mathrm{RD} \leftarrow \mathrm{RD}^{\phi,\pi}$
  State: $\mathrm{RD}.s \leftarrow 0^b$

Interface: $Z = \mathrm{RD.Init}(\delta, \mathrm{iv}, \sigma, \mathrm{flag})$ with $\delta \in \mathbb{Z}_u$, $\mathrm{iv} \in \mathbb{Z}_2^{b-k}$, $\sigma \in \mathbb{Z}_2^b$, $\mathrm{flag} \in \{\mathrm{true}, \mathrm{false}\}$, and $Z \in \mathbb{Z}_2^r$
  $s \leftarrow \phi(\delta, \mathrm{iv})$
  $Z \leftarrow \lfloor s \rfloor_r$
  if flag = true **then** $\overline{\sigma} \leftarrow \overline{\sigma} + Z$
  $s \leftarrow s + \sigma$
  return Z

Interface: $Z = \mathrm{RD.Duplexing}(\sigma, \mathrm{flag})$ with $\sigma \in \mathbb{Z}_2^b$ $\mathrm{flag} \in \{\mathrm{true}, \mathrm{false}\}$, and $Z \in \mathbb{Z}_2^r$
  $s \leftarrow \pi(s)$
  $Z \leftarrow \lfloor s \rfloor_r$
  if flag = true **then** $\overline{\sigma} \leftarrow \overline{\sigma} + Z$
  $s \leftarrow s + \sigma$
  return Z

The H-coefficient technique can thus be used to neatly bound a distinguishing advantage in the terminology of Definition 1, and a proof typically goes in four steps: (i) investigate what transcripts look like, which gives a definition for $\mathcal{T}$, (ii) define the partition of $\mathcal{T}$ into $\mathcal{T}_{\mathrm{good}}$ and $\mathcal{T}_{\mathrm{bad}}$, (iii) investigate the fraction of (8) for good transcripts and (iv) analyze the probability that $D_{\mathcal{P}}$ generates a bad transcript.

4 Security of Keyed Duplex Construction

We prove that the full-state keyed duplex construction (KD) is sound. We do so by proving an upper bound for the advantage of distinguishing the KD calling a random permutation f from an ideal extendable input function (IXIF). Both in the real and ideal world the adversary gets additional query access to f and f^{-1}, simply denoted as f.

The main result is stated in Section 4.2, but before doing so, we specify the resources of the adversary in Section 4.1.

4.1 Quantification of Adversarial Resources

We will consider information-theoretic adversaries that have two oracle interfaces: a construction oracle, $\mathrm{KD}_{\mathbf{K}}^{f}$ or $\mathrm{IXIF}^{\mathcal{RO}}$, and a primitive oracle f. For the construction queries, it can make initialization queries or duplexing queries. Note that, when querying $\mathrm{IXIF}^{\mathcal{RO}}$, every query has a *path* Path associated to it. To unify notation, we also associate a Path to each query (initialization or

duplexing) to $\mathrm{KD}^f_{\mathbf{K}}$. This Path is defined the straightforward way: it simply consists of the concatenation of $\mathsf{Encode}(\delta), \mathrm{iv}$ of the most recent initialization call and all σ-values that have been queried after the last initialization but before the current query. Using this formalization, every initialization *or* duplexing call that the adversary makes to $\mathrm{KD}^f_{\mathbf{K}}$ or $\mathrm{IXIF}^{\mathcal{RO}}$ can be properly captured by a tuple

$$(\mathsf{Path}, Z, \sigma)\,,$$

where, intuitively, Path is all data that is used to generate response $Z \in \mathbb{Z}_2^r$, and $\sigma \in \mathbb{Z}_2^b$ is the input string (slightly abusing notation; $\sigma = \sigma$ if flag = false and $\sigma = \sigma + (Z||0^c)$ if flag = true).

Following Andreeva et al. [2], we specify adversarial resources that impose limits on the transcripts that any adversary can obtain. The basic resource metrics are quantitative: they specify the number of queries an adversary is allowed to make for each type.

- $\boldsymbol{N}$: the number of primitive queries. It corresponds to computations requiring no access to the (keyed) construction. It is usually called the **time or offline complexity**. In practical use cases, N is only limited by the computing power and time available to the adversary.
- $\boldsymbol{M}$: the number of construction queries. It corresponds to the amount of data processed by the (keyed) construction. It is usually called the **data or online complexity**. In many practical use cases, M is limited.

We remark that identical calls are counted only once. In other words, N only counts the number of primitive queries, and M only counts the number of unique tuples (Path, σ).

It is possible to perform an analysis solely based on these metrics, but in order to more accurately cover practical settings that were not covered before (such as the multi-key setting or the nonce-respecting setting), and to eliminate the multiplicity (a metric used in all earlier results in this direction), we define a number of additional metrics.

- $\boldsymbol{q}$: the total number of different initialization tuples $(\mathsf{Encode}(\delta), \mathrm{iv})$. Parameter q corresponds to the number of times an adversary can start a *fresh* initialization of KD or IXIF.
- $\boldsymbol{q_{\mathrm{iv}}}$: iv multiplicity, the maximum number of different initialization tuples $(\mathsf{Encode}(\delta), \mathrm{iv})$ with same iv, maximized over all iv values.
- $\boldsymbol{\Omega}$: the number of queries with flag = true.
- $\boldsymbol{L}$: equals the number of queries M minus the number of distinct paths. It corresponds to the number of construction queries that have the same Path as some prior query.

In many practical use cases, q is limited, but as it turns out re-initialization queries give the adversary more power. The metric q_{iv} is relevant in multi-target attacks, where clearly $q_{\mathrm{iv}} \leq u$. The relevance of Ω and L is the following. In every query with flag equal to true, the adversary can force the outer part of the

input to f in a later query to a chosen value α by taking $\overline{\sigma} = \alpha$. Note that, as discussed in Section 2.2, by adopting authenticated encryption schemes with a slightly non-conventional encryption method, Ω can be forced to zero. Similarly, construction queries with the same path return the same value Z, and hence allow an adversary to force the outer part of the input to f in a later query to a chosen value α by taking σ such that $\overline{\sigma} = Z + \alpha$. An adversary can use this technique to increase the probability of collisions in $f(s) + \sigma$ and to speed up inner state recovery. By definition, $L \leq M - 1$ but in many cases L is much smaller. In particular, if one considers KD in the nonce-respecting setting, where no $(\mathsf{Encode}(\delta), \mathrm{iv})$ occurs twice, the adversary never generates a repeating path, and $L = 0$.

4.2 Main Result

Our bound uses a function that is defined in terms of a simple balls-into-bins problem.

Definition 2. *The* multicollision limit function $\nu_{r,c}^{M}$*, with M a natural number, returns a natural number and is defined as follows. Assume we uniformly randomly distribute M balls in 2^r bins. If we call the number of balls in the bin with the highest number of balls μ, then $\nu_{r,c}^{M}$ is defined as the smallest natural number x that satisfies:*

$$\Pr\left(\mu > x\right) \leq \frac{x}{2^c} \,.$$

In words, when uniformly randomly sampling M elements from a set of 2^r elements, the probability that there is an element that is sampled more than x times is smaller than $x2^{-c}$.

Theorem 1. *Let f be a random permutation and $\mathcal{RO}$ be a random oracle. Let $\mathbf{K}$ be a key array generated using a distribution $\mathcal{D}_{\mathrm{K}}$. Let $\mathrm{KD}_{\mathbf{K}}^{f}$ be the construction of Algorithm 1 and $\mathrm{IXIF}^{\mathcal{RO}}$ be the construction of Algorithm 2 and let $\nu_{r,c}^{M}$ be defined according to Definition 2. For any adversary $\mathcal{A}$ with resources as discussed in Section 4.1,*

$$\begin{aligned}\Delta_{\mathcal{A}}(\mathrm{KD}_{\mathbf{K}}^{f}, f \;;\; \mathrm{IXIF}^{\mathcal{RO}}, f) \leq\; & \frac{(L+\Omega)N}{2^c} + \frac{2\nu_{r,c}^{2(M-L)}(N+1)}{2^c} + \frac{\binom{L+\Omega+1}{2}}{2^c} + \\ & \frac{(M-q-L)q}{2^b - q} + \frac{M(M-L-1)}{2^b} + \\ & \frac{(M-q-L)q}{2^{H_{\min}(\mathcal{D}_{\mathrm{K}})+\min\{c,b-k\}}} + \frac{q_{\mathrm{iv}}N}{2^{H_{\min}(\mathcal{D}_{\mathrm{K}})}} + \frac{\binom{u}{2}}{2^{H_{coll}(\mathcal{D}_{\mathrm{K}})}} \,.\end{aligned}$$

The proof is given in Section 4.3.

For the case where $k + c \leq b - 1$, and where $\mathcal{D}_{\mathrm{K}}$ corresponds to uniform sampling without replacement, the bound simplifies to

$$\begin{aligned}\Delta_{\mathcal{A}}(\mathrm{KD}_{\mathbf{K}}^{f}, f \;;\; \mathrm{IXIF}^{\mathcal{RO}}, f) \leq\; & \frac{(L+\Omega)N}{2^c} + \frac{2\nu_{r,c}^{2(M-L)}(N+1)}{2^c} + \frac{\binom{L+\Omega+1}{2}}{2^c} + \\ & \frac{q_{\mathrm{iv}}N}{2^k} + \frac{(M-q-L)q}{2^{k+c-1}} + \frac{M(M-L-1)}{2^b} \,.\end{aligned}$$

The behavior of the function $\nu_{r,c}^{M}$ is discussed in Section 6.5 and illustrated in the Figure 4, which we refer to as the *Stairway to Heaven* graph.

4.3 Proof of Theorem 1

Let $\mathcal{A}$ be any information-theoretic adversary that has access to either, in the real world $(\mathrm{KD}_{\mathbf{K}}^{f}, f)$, or in the ideal world $(\mathrm{IXIF}^{\mathcal{RO}}, f)$. Note that, as $\mathcal{A}$ is information-theoretic, we can without loss of generality assume that it is deterministic, and we can apply the technique of Section 3. By the triangle inequality,

$$\begin{aligned} &\Delta_{\mathcal{A}}(\mathrm{KD}_{\mathbf{K}}^{f}, f \; ; \; \mathrm{IXIF}^{\mathcal{RO}}, f) \\ &\leq \Delta_{\mathcal{B}}(\mathrm{KD}_{\mathbf{K}}^{f}, f \; ; \; \mathrm{RD}^{\phi,\pi}, f) + \Delta_{\mathcal{C}}(\mathrm{RD}^{\phi,\pi}, f \; ; \; \mathrm{IXIF}^{\mathcal{RO}}, f)\,, \end{aligned} \tag{9}$$

where $\mathrm{RD}^{\phi,\pi}$ for random injection function ϕ and random permutation π is the construction of Algorithm 3, and where $\mathcal{B}$ and $\mathcal{C}$ have the same resources $(N, M, q, q_{\mathrm{iv}}, L, \Omega)$ as $\mathcal{A}$.

In the last term of (9), RD calls an ideal injective function ϕ and a random permutation π, both independent of f, and IXIF calls a random oracle $\mathcal{RO}$, also independent of f. The oracle access to f therefore does not "help" the adversary in distinguishing the two, or more formally,

$$\Delta_{\mathcal{C}}(\mathrm{RD}^{\phi,\pi}, f \; ; \; \mathrm{IXIF}^{\mathcal{RO}}, f) \leq \Delta_{\mathcal{D}}(\mathrm{RD}^{\phi,\pi} \; ; \; \mathrm{IXIF}^{\mathcal{RO}})\,. \tag{10}$$

where $\mathcal{D}$ is an adversary with the same construction query parameters as $\mathcal{A}$, but with no access to f.

The two remaining distances, i.e, the first term of (9) and the term of (10), will be analyzed in the next lemmas. The proof of Theorem 1 directly follows.

Lemma 2. *For any adversary $\mathcal{D}$ with resources as discussed in Section 4.1 but with no access to f,*

$$\Delta_{\mathcal{D}}(\mathrm{RD}^{\phi,\pi} \; ; \; \mathrm{IXIF}^{\mathcal{RO}}) \leq \frac{\binom{L+\Omega+1}{2}}{2^c} + \frac{M(M-L-1)}{2^b}\,. \tag{11}$$

Lemma 3. *For any adversary $\mathcal{B}$ with resources as discussed in Section 4.1,*

$$\begin{aligned} \Delta_{\mathcal{B}}(\mathrm{KD}_{\mathbf{K}}^{f}, f \; ; \; \mathrm{RD}^{\phi,\pi}, f) \leq \frac{(L+\Omega)N}{2^c} + \frac{2\nu_{r,c}^{2(M-L)}(N+1)}{2^c} + \frac{(M-q-L)q}{2^b - q} + \\ \frac{(M-q-L)q}{2^{H_{\min}(\mathcal{D}_{\mathrm{K}})+\min\{c,b-k\}}} + \frac{q_{\mathrm{iv}}N}{2^{H_{\min}(\mathcal{D}_{\mathrm{K}})}} + \frac{\binom{u}{2}}{2^{H_{coll}(\mathcal{D}_{\mathrm{K}})}}\,. \end{aligned} \tag{12}$$

The proof of Lemma 2 is given in Section 5, and the proof of Lemma 3 is given in Section 6.

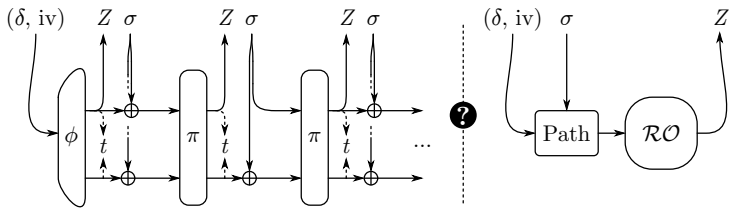

Fig. 2: Distinguishing experiment of RD and IXIF.

5 Distance Between RD and IXIF

In this section we bound the advantage of distinguishing the randomized duplex from an ideal extendable input function, (11) of Lemma 2. The distinguishing setup is illustrated in Figure 2. The derivation is performed using the H-coefficient technique.

Description of transcripts. The adversary has only a single interface, $\mathrm{RD}^{\phi,\pi}$ or $\mathrm{IXIF}^{\mathcal{RO}}$, but can make both initialization and duplexing queries. Following the discussion of Section 4.1, we can unify the two different types of queries, and summarize the conversation of $\mathcal{D}$ with its oracle in a transcript of the form

$$\tau_{\mathrm{C}} = \{(\mathsf{Path}_j, Z_j, \sigma_j)\}_{j=1}^{M} .$$

The values Z_j correspond to the outer part of the state just before σ_j gets injected. To make the analysis easier, we give at the end of the experiment for each query the inner value of the state at the moment Z_j is extracted (in the real world). We denote this as $t_j = \overline{t}_j || \widehat{t}_j$ with $\overline{t}_j = Z_j$. In the IXIF, $\widehat{t}_j$ is a value that is randomly generated for each path Path and can be expressed as $\mathcal{RO}'(\mathsf{Path})$ for some random oracle $\mathcal{RO}'$ with c-bit output. We integrate those values in the transcript, yielding:

$$\tau = \{(\mathsf{Path}_j, t_j, \sigma_j)\}_{j=1}^{M} .$$

Definition of good and bad transcripts. We define a transcript τ as *bad* if it contains a t-collision or an s-collision, where $s = t + \sigma$. A t-collision is defined as equal t values despite different Path values:

$$\exists(\mathsf{Path}, t, \sigma), (\mathsf{Path}', t', \sigma') \in \tau \text{ with } \big(\mathsf{Path} \neq \mathsf{Path}'\big) \text{ AND } \big(t = t'\big) . \tag{13}$$

An s-collision is defined as equal s values despite different (Path, σ') values:

$$\exists(\mathsf{Path}, t, \sigma), (\mathsf{Path}', t', \sigma') \in \tau \text{ with } \big((\mathsf{Path}, \sigma) \neq (\mathsf{Path}', \sigma')\big) \text{ AND } \big(t + \sigma = t' + \sigma'\big) . \tag{14}$$

In case the oracle is $\mathrm{RD}^{\phi,\pi}$, a t-collision is equivalent to two different inputs to π with identical outputs; an s-collision corresponds to the case of two identical inputs to f where the outputs are expected to be distinct. By considering these transcripts as bad, all queries properly define input-output tuples for ϕ and π.

Bounding the H-coefficient ratio for good transcripts. Denote $\mathcal{O} = \mathrm{RD}^{\phi,\pi}$ and $\mathcal{P} = \mathrm{IXIF}^{\mathcal{RO}}$ for brevity. Consider a good transcript $\tau \in \mathcal{T}_{\mathrm{good}}$. For the real world $\mathcal{O}$, the transcript defines exactly q input-output pairs for ϕ and exactly $M - q - L$ input-output pairs for π. It follows that $\Pr(D_{\mathcal{O}} = \tau) = 1/((2^b)_{(q)}(2^b)_{(M-q-L)})$. For the ideal world $\mathcal{P}$, every different Path defines exactly one evaluation of $\mathcal{RO}(\mathsf{Path})||\mathcal{RO}'(\mathsf{Path})$, so $\Pr(D_{\mathcal{P}} = \tau) = 2^{-(M-L)b}$. We consequently obtain that $\dfrac{\Pr(D_{\mathcal{O}} = \tau)}{\Pr(D_{\mathcal{P}} = \tau)} \geq 1$.

Bounding the probability of bad transcripts in the ideal world. In the ideal world, every t is generated as $\mathcal{RO}(\mathsf{Path})||\mathcal{RO}'(\mathsf{Path})$. As the number of distinct Path's in τ is $M - L$, there are $\binom{M-L}{2}$ possibilities for a t-collision, each occurring with probability 2^{-b}. The probability of such a collision is hence $\frac{\binom{M-L}{2}}{2^b}$.

There are $\binom{M}{2}$ occasions for an s-collision. Denote by S the size of the subset of these occasions for which the adversary can (in the worst case) force the outer part of $s = t + \sigma$ to be a value of its choice. Note that $S \leq \binom{L+\Omega+1}{2}$. In the worst case, in these S occasions the outer part of s always has the same value and s-collision probability is 2^{-c}. For the $\binom{M}{2} - S$ other occasions the s-collision probability is 2^{-b}. Thus, the probability of an s-collision is upper bound by (using our bound on S):

$$\frac{\binom{M}{2} - S}{2^b} + \frac{S}{2^c} \leq \frac{\binom{M}{2} - \binom{L+\Omega+1}{2}}{2^b} + \frac{\binom{L+\Omega+1}{2}}{2^c} \leq \frac{\binom{M}{2} - \binom{L+1}{2}}{2^b} + \frac{\binom{L+\Omega+1}{2}}{2^c}\,.$$

The total probability of having a bad transcript is hence upper bound by:

$$\frac{\binom{M-L}{2}}{2^b} + \frac{\binom{M}{2} - \binom{L+1}{2}}{2^b} + \frac{\binom{L+\Omega+1}{2}}{2^c} = \frac{M(M-L-1)}{2^b} + \frac{\binom{L+\Omega+1}{2}}{2^c}\,.$$

As the H-coefficient ratio is larger than 1, this is the bound on the distinguishing advantage and we have proven Lemma 2.

6 Distance Between KD and RD

In this section we bound the advantage of distinguishing the keyed duplex from a randomized duplex, (12) of Lemma 3. The analysis consists of four steps. In Section 6.1, we revisit the KD-vs-RD setup, and exclude the case where the queries made by the adversary result in a forward multiplicity that exceeds a certain threshold T_{fw}. Next, in Section 6.2 we convert our distinguishing setup

to a simpler one, called the *permutation setup* and illustrated in Figure 3. In this setup, the adversary has direct query access to the primitives ϕ and π of the randomized duplex, and at the keyed duplex side, we define two constructions on top of f that turn out to be hard to distinguish from ϕ and π. We carefully translate the resources of the adversary $\mathcal{B}$ in the KD-vs-RD setup to those of the adversary $\mathcal{C}$ in the permutation setup. In Section 6.3 we subsequently prove a bound in this setup. This analysis in part depends on a threshold on backward multiplicities T_{bw}. In Section 6.4 where we return to the KD-vs-RD setup and blend all results. Finally, in Section 6.5 and Section 6.6 we analyze the function $\nu_{r,c}^{M}$ that plays an important role in our analysis.

We remark that forward and backward multiplicity appeared before in Bertoni et al. [10] and Andreeva et al. [2], but we resolve them internally in the proof. There is a specific reason for resolving forward multiplicity *before* the conversion to the permutation setup and backward multiplicity *after* this conversion. Namely, in the permutation setup, an adversary could form its queries so that the forward multiplicity equals $M-q$, leading to a non-competitive bound, while the backward multiplicity cannot be controlled by the adversary as it cannot make inverse queries to the constructions. It turns out that, as discussed in Section 6.4, we can bound the thresholds as functions of M, L, and Ω.

6.1 The KD-vs-RD Setup

As in Section 4.1, we express the conversation that $\mathcal{B}$ has with $\mathrm{KD}_{\mathbf{K}}^{f}$ or $\mathrm{RD}^{\phi,\pi}$ in a transcript of the form:

$$\tau_{\mathrm{C}} = \{(\mathsf{Path}_j, Z_j, \sigma_j)\}_{j=1}^{M}\,.$$

We denote by μ_{fw} the maximum number of occurrences in this transcript of a value $Z_j + \overline{\sigma}_j$ over all possible values:

$$\mu_{\mathrm{fw}} = \max_{\alpha} \#\{(\mathsf{Path}_j, Z_j, \sigma_j) \in \tau_{\mathrm{C}} \mid Z_j + \overline{\sigma}_j = \alpha\}\,. \tag{15}$$

We now distinguish between two cases: μ_{fw} above some threshold T_{fw} and below it. Denoting $\mathcal{O} = (\mathrm{KD}_{\mathbf{K}}^{f}, f)$ and $\mathcal{P} = (\mathrm{RD}^{\phi,\pi}, f)$, we find using a hybrid argument,

$$\begin{aligned}
\Delta_{\mathcal{B}}(\mathcal{O}\,;\,\mathcal{P}) = {} & \left|\Pr\left(\mathcal{B}^{\mathcal{O}} \to 1\right) - \Pr\left(\mathcal{B}^{\mathcal{P}} \to 1\right)\right| \\
\leq {} & \left|\Pr\left(\mathcal{B}^{\mathcal{O}} \to 1 \wedge \mu_{\mathrm{fw}} \leq T_{\mathrm{fw}}\right) - \Pr\left(\mathcal{B}^{\mathcal{P}} \to 1 \wedge \mu_{\mathrm{fw}} \leq T_{\mathrm{fw}}\right)\right| + \\
& \left|\Pr\left(\mathcal{B}^{\mathcal{O}} \to 1 \wedge \mu_{\mathrm{fw}} > T_{\mathrm{fw}}\right) - \Pr\left(\mathcal{B}^{\mathcal{P}} \to 1 \wedge \mu_{\mathrm{fw}} > T_{\mathrm{fw}}\right)\right| \\
\leq {} & \left|\Pr\left(\mathcal{B}^{\mathcal{O}} \to 1 \wedge \mu_{\mathrm{fw}} \leq T_{\mathrm{fw}}\right) - \Pr\left(\mathcal{B}^{\mathcal{P}} \to 1 \wedge \mu_{\mathrm{fw}} \leq T_{\mathrm{fw}}\right)\right| + \\
& \max\left\{\Pr\left(\mu_{\mathrm{fw}} > T_{\mathrm{fw}} \text{ for } \mathcal{O}\right), \Pr\left(\mu_{\mathrm{fw}} > T_{\mathrm{fw}} \text{ for } \mathcal{P}\right)\right\}.
\end{aligned} \tag{16}$$

As we will find out (and explicitly mention) in Section 6.4, the bound we will derive on $\Pr\left(\mu_{\mathrm{fw}} > T_{\mathrm{fw}}\right)$ in fact applies to both $\mathcal{O}$ and $\mathcal{P}$, and for brevity denote the maximum of the two probabilities by $\Pr_{\mathcal{O},\mathcal{P}}\left(\mu_{\mathrm{fw}} > T_{\mathrm{fw}}\right)$.

6.2 Entering the Permutation Setup

To come to our simplified setup we define two constructions: the Even-Mansour construction and a "state initialization construction." The original Even-Mansour construction builds a b-bit block cipher from a b-bit permutation f and takes two b-bit keys K_1 and K_2 [25,26], and is defined as $f(x+K_1)+K_2$. We consider a variant, where $K_1 = K_2 = 0^r||\kappa$ with κ a c-bit key, and define

$$E^f_\kappa(x) = f(x + (0^r||\kappa)) + (0^r||\kappa) \ . \tag{17}$$

The state initialization construction is a dedicated construction of an injective function that maps an iv and a key selected from a key array $\mathbf{K}$ to a b-bit state and that takes a c-bit key κ.

$$I^f_{\kappa,\mathbf{K}}(\delta, \mathrm{iv}) = f(\mathbf{K}[\delta]||\mathrm{iv}) + (0^r||\kappa) \ . \tag{18}$$

Now, let $\kappa \xleftarrow{\$} \mathbb{Z}_2^c$ be any c-bit key. We call κ the *inner masking key*. Using the idea of bitwise adding the inner masking key twice in-between every two primitive evaluations [2, 21, 38], we obtain that: $\mathrm{KD}^f_{\mathbf{K}} = \mathrm{RD}^{I^f_{\kappa,\mathbf{K}}, E^f_\kappa}$. We thus obtain for (16), leaving the condition $\mu_{\mathrm{fw}} \le T_{\mathrm{fw}}$ implicit:

$$\begin{aligned} \Delta_{\mathcal{B}}(\mathrm{KD}^f_{\mathbf{K}}, f \ ; \ \mathrm{RD}^{\phi,\pi}, f) &= \Delta_{\mathcal{B}}(\mathrm{RD}^{I^f_{\kappa,\mathbf{K}}, E^f_\kappa}, f \ ; \ \mathrm{RD}^{\phi,\pi}, f) \\ &\le \Delta_{\mathcal{C}}(I^f_{\kappa,\mathbf{K}}, E^f_\kappa, f \ ; \ \phi, \pi, f) \, . \end{aligned} \tag{19}$$

Clearly an adversary $\mathcal{B}$ can be simulated by an adversary $\mathcal{C}$ as any construction query can be simulated by queries to the initialization function $\mathcal{O}_{\mathrm{i}}$ ($I^f_{\kappa,\mathbf{K}}$ in the real world and ϕ in the ideal world) and the duplexing function $\mathcal{O}_{\mathrm{d}}$ (E^f_κ in the real world and π in the ideal world). Hence, we can quantify the resources of adversary $\mathcal{C}$ in terms of the resources of adversary $\mathcal{B}$, making use of the threshold T_{fw} on the multiplicity (cf., (16)). This conversion will be formally performed in Section 6.4.

6.3 Distinguishing Bound for the Permutation Setup

We now bound $\Delta_{\mathcal{C}}(I^f_{\kappa,\mathbf{K}}, E^f_\kappa, f \ ; \ \phi, \pi, f)$. The permutation setup is illustrated in Figure 3. The derivation is performed using the H-coefficient technique.

Description of transcripts. The adversary has access to either $(I^f_{\kappa,\mathbf{K}}, E^f_\kappa, f)$ or (ϕ, π, f). The queries of the adversary and their responses are assembled in three transcripts $\tau_{\mathrm{f}}, \tau_{\mathrm{d}}$, and τ_{i}.

$\tau_{\mathrm{f}} = \{(x_j, y_j)\}_{j=1}^N$ The queries to f and f^{-1}. The transcript does not code whether the query was $y = f(x)$ or $x = f^{-1}(y)$.

$\tau_{\mathrm{i}} = \{(\delta_i, \mathrm{iv}_i, t_i)\}_{i=1}^{q'}$ The queries to the initialization function $\mathcal{O}_{\mathrm{i}}$, $I^f_{\kappa,\mathbf{K}}$ in the real world and ϕ in the ideal world.

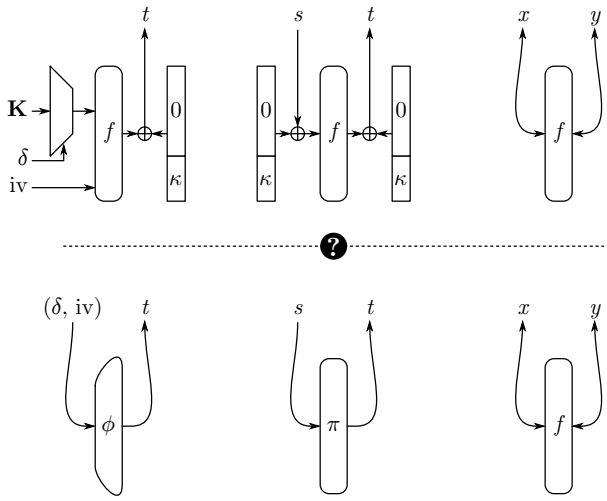

Fig. 3: Permutation setup.

$\tau_{\mathrm{d}} = \{(s_i, t_i)\}_{i=1}^{M'}$ The queries to the duplexing function $\mathcal{O}_{\mathrm{d}}$, E_κ^f in the real world and π in the ideal world.

The resources of $\mathcal{C}$ are defined by the number of queries in each transcript: N, M', and q', as well as $q_{\mathrm{iv}} = \max_\alpha \#\{(\delta, \mathrm{iv}, t) \in \tau_{\mathrm{i}} \mid \mathrm{iv} = \alpha\}$. In addition, the resources of $\mathcal{C}$ are limited on τ_{d}, for which the *forward multiplicity* must be below the threshold T_{fw}:

$$\max_\alpha \#\{(s_i, t_i) \in \tau_{\mathrm{d}} \mid \bar{s}_i = \alpha\} \leq T_{\mathrm{fw}}\,.$$

To ease the analysis, we will disclose the full key array $\mathbf{K}$ and the inner masking key κ at the end of the experiment (in the ideal world, κ and the elements of $\mathbf{K}$ will simply be dummy keys). The transcripts are thus of the form $\tau = (\mathbf{K}, \kappa, \tau_{\mathrm{f}}, \tau_{\mathrm{i}}, \tau_{\mathrm{d}})$. Note that it is fair to assume that none of the transcripts contains duplicate elements (i.e., the adversary never queries f twice on the same value). Additionally, as we consider attainable transcripts only and ϕ, π, f are injective mappings, τ does not contain collisions.

We define the backward multiplicity as characteristic of the transcript τ:

Definition 3. *In the permutation setup, the backward multiplicity μ_{bw} is defined as:*

$$\mu_{\mathrm{bw}} = \max_\alpha \Big(\#\{(s_i, t_i) \in \tau_{\mathrm{d}} \mid \bar{t}_i = \alpha\} + \#\{(\delta, \mathrm{iv}, t_i) \in \tau_{\mathrm{i}} \mid \bar{t}_i = \alpha\}\Big)\,.$$

Definition of good and bad transcripts. In the real world, every tuple in $(\tau_{\mathrm{f}}, \tau_{\mathrm{i}}, \tau_{\mathrm{d}})$ defines exactly one evaluation of f. We define a transcript τ as *bad* if

it contains an input or output collision of f or if the backward multiplicity is above some limit T_{bw}. In other words, τ is bad if one of the following conditions is satisfied. Input collisions between:

$$\tau_{\mathrm{f}} \text{ and } \tau_{\mathrm{i}} : \exists (x,y) \in \tau_{\mathrm{f}}, (\delta, \mathrm{iv}, t) \in \tau_{\mathrm{i}} \text{ with } \left(x = \mathbf{K}[\delta] || \mathrm{iv}\right); \tag{20}$$

$$\tau_{\mathrm{f}} \text{ and } \tau_{\mathrm{d}} : \exists (x,y) \in \tau_{\mathrm{f}}, (s,t) \in \tau_{\mathrm{d}} \text{ with } \left(x = s + 0^r || \kappa\right); \tag{21}$$

$$\tau_{\mathrm{i}} \text{ and } \tau_{\mathrm{d}} : \exists (\delta, \mathrm{iv}, t) \in \tau_{\mathrm{i}}, (s',t') \in \tau_{\mathrm{d}} \text{ with } \left(\mathbf{K}[\delta] || \mathrm{iv} = s' + 0^r || \kappa\right); \tag{22}$$

$$\text{within } \tau_{\mathrm{i}} : \exists (\delta, \mathrm{iv}, t), (\delta', \mathrm{iv}', t') \in \tau_{\mathrm{i}} \text{ with } \left(\delta \neq \delta'\right) \text{ AND } \left(\mathbf{K}[\delta] || \mathrm{iv} = \mathbf{K}[\delta'] || \mathrm{iv}'\right). \tag{23}$$

Output collisions between:

$$\tau_{\mathrm{f}} \text{ and } \tau_{\mathrm{i}} : \exists (x,y) \in \tau_{\mathrm{f}}, (\delta, \mathrm{iv}, t) \in \tau_{\mathrm{i}} \text{ with } \left(y = t + 0^r || \kappa\right); \tag{24}$$

$$\tau_{\mathrm{f}} \text{ and } \tau_{\mathrm{d}} : \exists (x,y) \in \tau_{\mathrm{f}}, (s,t) \in \tau_{\mathrm{d}} \text{ with } \left(y = t + 0^r || \kappa\right); \tag{25}$$

$$\tau_{\mathrm{i}} \text{ and } \tau_{\mathrm{d}} : \exists (\delta, \mathrm{iv}, t) \in \tau_{\mathrm{i}}, (s',t') \in \tau_{\mathrm{d}} \text{ with } \left(t + 0^r || \kappa = t' + 0^r || \kappa\right). \tag{26}$$

Finally, τ is bad if the backward multiplicity μ_{bw} is above the threshold T_{bw}:

$$\mu_{\mathrm{bw}} > T_{\mathrm{bw}}\,. \tag{27}$$

Note that output collisions within τ_{i} are excluded by attainability of transcripts. Similarly, collisions (input or output) within τ_{f} as well as collisions within τ_{d} are excluded by attainability of transcripts.

Bounding the H-coefficient ratio for good transcripts. Denote $\mathcal{O} = (I^f_{\kappa,\mathcal{D}_{\mathrm{K}}}, E^f_\kappa, f)$ and $\mathcal{P} = (\phi, \pi, f)$ for brevity. Consider a good transcript $\tau \in \mathcal{T}_{\mathrm{good}}$.

In the real world $\mathcal{O}$, the transcript defines exactly $q' + M' + N$ input-output pairs of f, so $\Pr\left(D_{\mathcal{O}} = \tau\right) = 1/(2^b)_{(q'+M'+N)}$. In the ideal world $\mathcal{P}$, every tuple in τ_{f} defines exactly N input-output pairs for f, every tuple in τ_{i} defines exactly q' input-output pairs for ϕ, and every tuple in τ_{d} defines exactly M' input-output pairs for π. It follows that $\Pr\left(D_{\mathcal{P}} = \tau\right) = 1/((2^b)_{(N)}(2^b)_{(q')}(2^b)_{(M')})$. We consequently obtain that $\dfrac{\Pr\left(D_{\mathcal{O}} = \tau\right)}{\Pr\left(D_{\mathcal{P}} = \tau\right)} \geq 1$.

Bounding the probability of bad transcripts in the ideal world. In the ideal world, κ is generated uniformly at random. The key array $\mathbf{K}$ is generated according to distribution $\mathcal{D}_{\mathrm{K}}$, cf., Section 2.1. We will use the min-entropy and maximum collision probability definitions of (6) and (7).

For (20), fix any $(x,y) \in \tau_{\mathrm{f}}$. There are at most q_{iv} tuples in τ_{i} with iv equal to the last $b-k$ bits of x. For any of those tuples, the probability that the first k bits of x are equal to $\mathbf{K}[\delta]$ is at most $2^{-H_{\min}(\mathcal{D}_{\mathrm{K}})}$, cf., (6). The collision probability is hence at most $q_{\mathrm{iv}} N / 2^{H_{\min}(\mathcal{D}_{\mathrm{K}})}$.

For (21), fix any $(x,y) \in \tau_{\mathrm{f}}$. There are at most T_{fw} tuples in τ_{d} with $\overline{x} = \overline{s}$. For any of those tuples, the probability that $\widehat{x} = \widehat{s} + \kappa$ is 2^{-c}. The collision probability is hence at most $T_{\mathrm{fw}} N / 2^c$.

For (24) or (25), we will assume ¬(27). Fix any $(x, y) \in \tau_{\rm f}$. There are at most $T_{\rm bw}$ tuples in $\tau_{\rm i} \cup \tau_{\rm d}$ with $\overline{y} = \overline{t}$. For any of those tuples, the probability that $\widehat{y} = \widehat{t} + \kappa$ is 2^{-c}. The collision probability is hence at most $T_{\rm bw}N/2^c$.

For (22), fix any $(\delta, {\rm iv}, t) \in \tau_{\rm i}$ and any $(s', t') \in \tau_{\rm d}$. Any such combination sets (22) if $0^k||{\rm iv} + s' = \mathbf{K}[\delta]||0^{b-k} + 0^r||\kappa$. Note that the randomness of $\mathbf{K}[\delta]$ may overlap the one of κ. If $k + c \leq b$, the two queries satisfy the condition with probability at most $2^{-(H_{\min}(\mathcal{D}_{\rm K})+c)}$, cf., (6). On the other hand, if $k > b - c$, the first $b - c$ bits of $\mathbf{K}[\delta]$ has a min-entropy of at least $H_{\min}(\mathcal{D}_{\rm K}) - (k - (b - c))$. In this case, the two queries satisfy the condition with probability at most

$$2^{-(H_{\min}(\mathcal{D}_{\rm K})-(k-(b-c))+c)} = 2^{-(H_{\min}(\mathcal{D}_{\rm K})+b-k)} .$$

The collision probability is hence at most $\frac{M'q'}{2^{H_{\min}(\mathcal{D}_{\rm K})+\min\{c,b-k\}}}$, using that $\tau_{\rm i}$ contains q' elements and $\tau_{\rm d}$ contains M' elements.

For (26), fix any $(\delta, {\rm iv}, t) \in \tau_{\rm i}$ and any $(s', t') \in \tau_{\rm d}$. As ϕ and π are only evaluated in forward direction, and ϕ is queried at most q' times, the probability that $t = t'$ for these two tuples is at most $1/(2^b - q')$. The collision probability is hence at most $M'q'/(2^b - q')$.

For (23), a collision of this form implies the existence of two distinct δ, δ' such that $K[\delta] = K[\delta']$. This happens with probability at most $\binom{u}{2}/2^{H_{\rm coll}(\mathcal{D}_{\rm K})}$, cf., (7).

The total probability of having a bad transcript is at most:

$$\begin{aligned} &\frac{(T_{\rm fw} + T_{\rm bw})N}{2^c} + \Pr_{\mathcal{P}}\left(\mu_{\rm bw} > T_{\rm bw}\right) + \frac{M'q'}{2^b - q'} + \\ &\frac{M'q'}{2^{H_{\min}(\mathcal{D}_{\rm K})+\min\{c,b-k\}}} + \frac{q_{\rm iv}N}{2^{H_{\min}(\mathcal{D}_{\rm K})}} + \frac{\binom{u}{2}}{2^{H_{\rm coll}(\mathcal{D}_{\rm K})}} . \end{aligned} \tag{28}$$

As the H-coefficient ratio is larger than 1, Equation (28) is the bound on the distinguishing advantage.

6.4 Returning to the KD-vs-RD Setup

The resources of $\mathcal{C}$ can be computed from those of $\mathcal{B}$ (see Section 4.1) in the following way:

- $q' \leq q$: for every query to $\mathcal{O}_{\rm i}$ there must be at least one initialization query.
- $M' \leq M - q - L$: The minus L is there because queries with repeated paths just give duplicate queries to $\mathcal{O}_{\rm i}$ and the q initialization queries do not give queries to $\mathcal{O}_{\rm d}$.

The remaining resources have the same meaning for $\mathcal{B}$ and $\mathcal{C}$. Filling in these values in Equation (28) and combining with Equation (16) yields:

$$\Delta_{\mathcal{B}}(\mathcal{O}\ ;\ \mathcal{P}) \leq \left(\frac{T_{\mathrm{fw}}N}{2^c} + \Pr_{\mathcal{O},\mathcal{P}}\left(\mu_{\mathrm{fw}} > T_{\mathrm{fw}}\right)\right) + \tag{29a}$$

$$\left(\frac{T_{\mathrm{bw}}N}{2^c} + \Pr_{\mathcal{P}}\left(\mu_{\mathrm{bw}} > T_{\mathrm{bw}}\right)\right) + \tag{29b}$$

$$\frac{(M-q-L)q}{2^b-q} + \frac{(M-q-L)q}{2^{H_{\min}(\mathcal{D}_{\mathrm{K}})+\min\{c,b-k\}}} + \tag{29c}$$

$$\frac{q_{\mathrm{iv}}N}{2^{H_{\min}(\mathcal{D}_{\mathrm{K}})}} + \frac{\binom{u}{2}}{2^{H_{\mathrm{coll}}(\mathcal{D}_{\mathrm{K}})}}\,. \tag{29d}$$

Clearly $\mu_{\mathrm{fw}} \leq M-q-L$ and $\mu_{\mathrm{bw}} \leq M-L$. So by taking $T_{\mathrm{fw}} = T_{\mathrm{bw}} = M-L$, lines (29a-29b) reduce to $2(M-L)N/2^c$. However, much better bounds can be obtained by carefully tuning T_{fw} and T_{bw}.

Although the probabilities on the μ_{fw} and μ_{bw} are defined differently (the former in the KD-vs-RD setup, the latter in the permutation setup), in essence they are highly related and we can rely on multicollision limit function of Definition 2 for their analysis. There is one caveat. Definition 2 considers balls thrown uniformly at random into the 2^r bins, hence a bin is hit with probability $1/2^r$. In Lemma 6 in upcoming Section 6.6, we will prove that for non-uniform bin allocation where the probability that a ball hits any particular bin is upper bounded by $y2^{-r}$, the multicollision limit function is at most $\nu_{r,c}^{yM}$. In our case the states are generated from a set of size at least $2^b - M - N$ (for both $\mathcal{O}$ and $\mathcal{P}$) and thus its outer part is thrown in a bin with probability at most $2^c/(2^b-M-N)$, where we use that $M+N \leq 2^{b-1}$. Using the fact that $\nu_{r,c}^{M}$ is a monotonic function in M and that $2^b/(2^b-M-N) < 2$ for any reasonable value of $M+N$, we upper bound the multicollision limit function by $\nu_{r,c}^{2(M-L)}$

We first look at (29b) and treat μ_{bw}. As it is a metric of the responses of queries to π and ϕ, it is a stochastic variable. It corresponds to the multicollision limit function of Definition 2, where $M-L$ balls are distributed over 2^r bins, and each bin is hit with probability at most $2/2^r$. Using above observation, we take $T_{\mathrm{bw}} = \nu_{r,c}^{2(M-L)}$, and (29b) becomes

$$\frac{\nu_{r,c}^{2(M-L)}N}{2^c} + \frac{\nu_{r,c}^{2(M-L)}}{2^c} = \frac{\nu_{r,c}^{2(M-L)}(N+1)}{2^c}\,.$$

The case of μ_{fw} in (29c) is slightly more complex. As discussed in Section 4.1, the adversary can enforce the outer part $Z_j + \overline{\sigma}_j$ to match a value α in case Path_j is a repeating path. Moreover, for queries with flag = true, it can also enforce the outer part to any chosen value. These total to $L+\Omega$ queries. For the remaining queries, for simplicity upper bound by $M-L$ here, the adversary has no control over the outer part. Therefore, if take $T_{\mathrm{fw}} = L+\Omega+\nu_{r,c}^{2(M-L)}$, we have $\Pr\left(\mu_{\mathrm{fw}} > T_{\mathrm{fw}}\right) = \frac{\nu_{r,c}^{2(M-L)}}{2^c}$. Namely, this is the probability that in the (at most) $M-L$ queries where the adversary has no control over the outer part, the

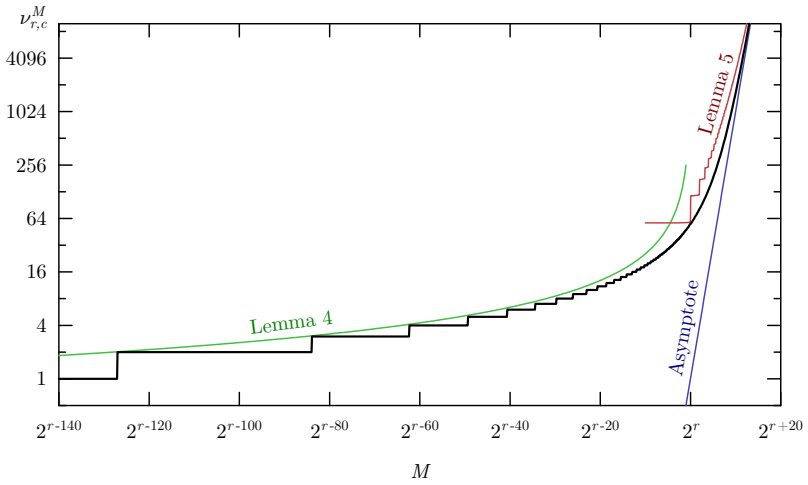

Fig. 4: Stairway to Heaven graph: $\nu_{r,c}^M$ computed with (33) for $r + c = 256$, with upper bounds and asymptote for $M \to \infty$.

multiplicity is above $\nu_{r,c}^{2(M-L)}$ assuming that the $L+\Omega$ queries are manipulated to hit the same outer value as those $\nu_{r,c}^{2(M-L)}$ queries. Equation (29a) now becomes:

$$\frac{(L+\Omega+\nu_{r,c}^{2(M-L)})N}{2^c} + \frac{\nu_{r,c}^{2(M-L)}}{2^c} = \frac{(L+\Omega)N}{2^c} + \frac{2\nu_{r,c}^{2(M-L)}(N+1)}{2^c}\,.$$

Plugging these two bounds into (29a-29b) yields the bound of Lemma 3.

6.5 Bounds on $\nu_{r,c}^M$

We will upper bound $\nu_{r,c}^M$ by approximating the term $\Pr(\mu > x)$ in Definition 2 by simpler expressions that are strictly larger.

In Definition 2, μ is the maximum of the number of balls over all 2^r bins. The probability that the maximum of a set of variables X is below some value x is the product of the probabilities that all variables are below x. Assuming all variables have the same distribution and that they are independent, we obtain:

$$\Pr(\mu > x) = 1 - \Pr(\mu \leq x) = 1 - (\Pr(X \leq x))^{2^r}\,. \tag{30}$$

The distributions that are of interest here are the number of balls in a bin, and they are not independent as they must sum to M. This means that if we know one distribution is high, the others are somewhat lower than if they would be independent. This makes that the value obtained by taking the product of factors $\Pr(X \leq x)$ slightly underestimates the probability $\Pr(\mu \leq x)$. Using the inequality $(1-\epsilon)^y \geq 1 - \epsilon y$, we obtain

$$(\Pr(X \leq x))^{2^r} = (1 - \Pr(X > x))^{2^r} \geq 1 - 2^r \Pr(X > x)\,,$$

and we obtain for (30):

$$\Pr(\mu > x) < 2^r \Pr(X > x)\,. \tag{31}$$

We will now upper bound $\Pr(X > x)$. The number of balls x in any particular bin has a binomial distribution. If the number of bins 2^r and the total number of balls M are large enough, for $x > \lambda$ this is (tightly) upper bounded by a Poisson distribution with $\lambda = M2^{-r}$. The probability that a Poisson-distributed variable X is larger than x is given by:

$$\Pr(X > x) = \sum_{i \geq x} \frac{e^{-\lambda}\lambda^i}{i!} = \frac{e^{-\lambda}\lambda^x}{x!} \sum_{i \geq 0} \frac{\lambda^i}{(i+x)_{(i)}} < \frac{e^{-\lambda}\lambda^x}{x!} \sum_{i \geq 0} \frac{\lambda^i}{x^i} = \frac{xe^{-\lambda}\lambda^x}{(x-\lambda)x!}\,.$$

This yields for (31):

$$\Pr(\mu > x) < 2^r \frac{xe^{-\lambda}\lambda^x}{(x-\lambda)x!}\,.$$

From Definition 2, we obtain that $\nu^M_{r,c}$ is upper bounded by the smallest value x that satisfies

$$\frac{2^b e^{-\lambda}\lambda^x}{(x-\lambda)x!} \leq 1\,, \tag{32}$$

with $\lambda = M2^{-r}$. Remarkably, the dependence of $\nu^M_{r,c}$ on r, c and M is only via $b = r + c$ and $\lambda = M2^{-r}$. Hence, it is a function in two variables b and λ rather than three. Taking the logarithm of (32), applying Stirling's approximation ($\ln(x!) \geq \frac{1}{2}\ln(2\pi x) + x(\ln(x) - 1)$) and rearranging the terms gives:

$$x\,(\ln(x) - \ln(\lambda) - 1) + \ln(x - \lambda) + \frac{1}{2}\ln(2\pi x) + \lambda \geq \ln(2)b\,. \tag{33}$$

We will now derive expressions from (32) and (33) that give insight in the behavior of this function for the full range of λ.

Case $\boldsymbol{\lambda < 1}$. If we consider Equation (33) with value of x given and we look for the maximum value of x such that it holds. This gives the value of λ where $\nu^M_{r,c}$ transitions from $x - 1$ to x. We can now prove the following lemma.

Lemma 4. *The value of λ where $\nu^M_{r,c}$ transitions from $x-1$ to x is lower bounded by $2^{-b/x}$.*

Proof. We need to prove that for $\lambda = 2^{-b/x}$, inequality (33) holds:

$$x(\ln(x) - 1) + \ln(x - 2^{-b/x}) + \frac{1}{2}\ln(2\pi x) + 2^{-b/x} \geq 0\,.$$

For $x > e$ all terms in the left hand side of this equation are positive and hence the equation is satisfied. The only other relevant value is $x = 2$ and it can be verified by hand that this is satisfied for all b. □

If we substitute λ by $M2^{-r}$, this gives bounds on M for which $\nu_{r,c}^M$ achieves a certain value. If we denote by M_x the value where $\nu_{r,c}^M$ transitions from $x-1$ to x, we have $M_x \geq 2^{r-b/x} = 2^{((x-1)r-c)/x}$. In particular $M_2 \geq 2^{(r-c)/2}$. It follows that $\nu_{r,c}^M$ is 1 for $M \leq 2^{(r-c)/2}$. Clearly, M must be an integer value, so the value of $\nu_{r,c}^M$ for $M = 1$ will be above 1 if $r < c + 2$.

Case $\lambda = 1$. Equation (33) for $\lambda = 1$ reads

$$x\,(\ln(x) - 1) + \ln(x-1) + \frac{1}{2}\ln(2\pi x) + 1 \geq \ln(2)b\,,$$

and $\nu_{r,c}^M$ is upper bounded by the smallest x such that this inequality holds, or equivalently, such that

$$x \geq \frac{\ln(2)b - 1 - \ln(x-1) - \frac{1}{2}\ln(2\pi x)}{\ln(x) - 1}\,.$$

The right hand side of this equation is upper bounded by $\frac{\ln(2)b}{\ln(x)-1}$. Therefore, $\nu_{r,c}^M$ is certainly upper bounded by the smallest x such that

$$x \geq \frac{\ln(2)b}{\ln(x) - 1}\,.$$

This expression can be efficiently evaluated for all values of b, and it turns out that the value of $\nu_{r,c}^{2^r}$ increases from $b/4$ for values of b close to 200 to values $b/6$ for values of b close to 2000.

Case $\lambda > 1$. For large λ, Equation (33) becomes numerically instable. We derive a formula for integer values of λ, or equivalently values of M that are a multiple of 2^r (w.l.o.g.). By a change of variable from x to $x = \lambda + y$ we obtain for the left hand side of (32):

$$\frac{2^b e^{-\lambda}\lambda^x}{(x-\lambda)x!} = \frac{2^b e^{-\lambda}\lambda^{\lambda+y}}{y(\lambda+y)!} = \frac{2^b\lambda^y}{y(\lambda+y)_y}\frac{(\lambda/e)^\lambda}{\lambda!} \leq \frac{2^b\lambda^y}{y\sqrt{2\pi\lambda}(\lambda+y)_y}$$

using Stirling's approximation. Now (32) holds provided that

$$\frac{2^b\lambda^y}{y\sqrt{2\pi\lambda}(\lambda+y)_y} = \frac{2^b}{y\sqrt{2\pi\lambda}\prod_{i=1}^{y}(1+\frac{i}{\lambda})} \leq 1\,.$$

Taking the logarithm:

$$\sum_{i=1}^{y}\ln\left(1+\frac{i}{\lambda}\right) + \ln(y) + \frac{1}{2}\ln(2\pi\lambda) \geq \ln(2)b\,. \tag{34}$$

This equation allows efficiently computing $\nu_{r,c}^M$ for $M > 2^r$ and also to prove a simple upper bound for the range $\lambda > 1$.

Lemma 5. *For M a nonzero integer multiple of 2^r, we have*

$$\nu_{r,c}^M \leq \frac{M}{2^r} + \nu_{r,c}^{2^r} \left\lceil \sqrt{\frac{M}{2^r}} \right\rceil .$$

Proof. First of all, note that for $\lambda = 1$, (34) is satisfied for $y = \nu_{r,c}^{2^r} - 1$. Therefore, we have

$$\Xi := \sum_{i=1}^{\nu_{r,c}^{2^r}-1} \ln(1+i) + \ln(\nu_{r,c}^{2^r} - 1) + \frac{1}{2}\ln(2\pi) - \ln(2)b \geq 0 .$$

Our goal is to prove that (34) holds for $y = \nu_{r,c}^{2^r}\lceil\sqrt{\lambda}\rceil$. Since $\Xi \geq 0$, we will in fact prove that

$$\sum_{i=1}^{\nu_{r,c}^{2^r}\lceil\sqrt{\lambda}\rceil} \ln\left(1 + \frac{i}{\lambda}\right) + \ln(\nu_{r,c}^{2^r}\lceil\sqrt{\lambda}\rceil) + \frac{1}{2}\ln(2\pi\lambda) - \ln(2)b \geq \Xi .$$

Note that

$$\begin{aligned} &\sum_{i=1}^{\nu_{r,c}^{2^r}\lceil\sqrt{\lambda}\rceil} \ln\left(1 + \frac{i}{\lambda}\right) + \ln(\nu_{r,c}^{2^r}\lceil\sqrt{\lambda}\rceil) + \frac{1}{2}\ln(2\pi\lambda) - \ln(2)b - \Xi \\ \geq &\sum_{i=1}^{\nu_{r,c}^{2^r}\lceil\sqrt{\lambda}\rceil} \ln\left(1 + \frac{i}{\lambda}\right) - \sum_{i=1}^{\nu_{r,c}^{2^r}-1} \ln(1+i) . \end{aligned}$$

This can be rewritten as

$$\sum_{i=0}^{\nu_{r,c}^{2^r}-1} \left(\sum_{j=1}^{\lceil\sqrt{\lambda}\rceil} \ln\left(1 + \frac{i\lceil\sqrt{\lambda}\rceil + j}{\lambda}\right) - \ln(1+i) \right) ,$$

and our claim holds if we can prove that the summand is at least 0 for all $i = 0, \ldots, \nu_{r,c}^{2^r} - 1$. This is easily verified as

$$\sum_{j=1}^{\lceil\sqrt{\lambda}\rceil} \ln\left(1 + \frac{i\lceil\sqrt{\lambda}\rceil + j}{\lambda}\right) \geq \sum_{j=1}^{\lceil\sqrt{\lambda}\rceil} \ln\left(1 + \frac{i\lceil\sqrt{\lambda}\rceil}{\lambda}\right) = \ln\left(\left(1 + \frac{i\lceil\sqrt{\lambda}\rceil}{\lambda}\right)^{\lceil\sqrt{\lambda}\rceil}\right) ,$$

which is at least

$$\ln\left(1 + \frac{i\lceil\sqrt{\lambda}\rceil^2}{\lambda}\right) \geq \ln(1+i) ,$$

as in general $(1+x)^y \geq 1 + xy$. □

Clearly, for large M, $\nu_{r,c}^M$ asymptotically converges to $M/2^r$.

6.6 Dealing with Non-Uniform Sampling

In this section we address the non-uniform balls-and-bins problem. We consider the balls-and-bins problems for some values r and c where the probability that a ball hits a particular bin (of the 2^r bins) is not 2^{-r}. In other words, the distribution is not uniform. In general the probability distribution for the n-th ball depends on how the previous $n-1$ balls were distributed. We denote this distribution by D and denote by $\nu_{r,c}^{D,M}$ the variant of the function with the same name with the given distribution.

Definition 4. *The* multicollision limit function *for some distribution D, $\nu_{r,c}^{D,M}$, with M a natural number, returns a natural number and is defined as follows. Assume we randomly distribute M balls in 2^r bins according to a distribution D. If we call the number of balls in the bin with the highest number of balls μ, then $\nu_{r,c}^{D,M}$ is defined as the smallest natural number x that satisfies:*

$$\Pr\left(\mu > x\right) < \frac{x}{2^c}\,.$$

We can now prove the following lemma.

Lemma 6. *If according to the distribution D the probability for a ball to end up in a bin is upper bounded by $y2^{-r}$, then $\nu_{r,c}^{D,M} \geq \nu_{r,c}^{yM}$, for yM an integer, $y \leq 2$ and $r \geq 8$.*

Proof. First we consider the cumulative distributions for the number of balls in a given bin for two experiments.

- Experiment 1: we drop yM balls and the distribution is uniform.
- Experiment 2: we drop M balls and the distribution is not uniform but the probability to land in a particular bin is upper bounded by $y2^{-r}$.

If the success probability for a ball to hit a bin is sufficiently small (e.g., below 2^{-6}) and the number of balls is sufficiently large (e.g., above 2^{10}) the number of balls in a bin has a Poisson distribution with λ the sum over all balls of the success probability that the ball hits the bin. In experiment 1 we have $\lambda = y2^{-r}M$, in experiment 2 it is upper bounded by $\lambda = y2^{-r}M$. It follows that the cumulative distribution of the bin in experiment 2 is not above that of the cumulative distribution of the bin in experiment 1. The cumulative distribution of μ in experiment 1 can be obtained by taking the cumulative distribution of a single bin to the power 2^r. In experiment 2, it would require taking the product of the cumulative distributions of all bins. As all these have a Poisson cumulative distribution with λ upper bounded by $y2^{-r}M$, it cannot be larger than that for experiment 1. □

Acknowledgments. Bart Mennink is supported by a postdoctoral fellowship from the Netherlands Organisation for Scientific Research (NWO) under Veni grant 016.Veni.173.017.

References

1. Andreeva, E., Bogdanov, A., Luykx, A., Mennink, B., Mouha, N., Yasuda, K.: How to securely release unverified plaintext in authenticated encryption. In: Sarkar and Iwata [45], pp. 105–125
2. Andreeva, E., Daemen, J., Mennink, B., Van Assche, G.: Security of Keyed Sponge Constructions Using a Modular Proof Approach. In: Leander [36], pp. 364–384
3. Aumasson, J., Jovanovic, P., Neves, S.: NORX v3.0 (2016), submission to CAESAR competition
4. Bellare, M., Bernstein, D.J., Tessaro, S.: Hash-Function Based PRFs: AMAC and Its Multi-User Security. In: Fischlin and Coron [28], pp. 566–595
5. Bellare, M., Boldyreva, A., Micali, S.: Public-Key Encryption in a Multi-user Setting: Security Proofs and Improvements. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 259–274. Springer (2000)
6. Bellare, M., Rogaway, P.: Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In: Denning, D.E., Pyle, R., Ganesan, R., Sandhu, R.S., Ashby, V. (eds.) ACM CCS '93. pp. 62–73. ACM (1993)
7. Bernstein, D.J.: The Poly1305-AES Message-Authentication Code. In: Gilbert, H., Handschuh, H. (eds.) FSE 2005. LNCS, vol. 3557, pp. 32–49. Springer (2005)
8. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Sponge functions. Ecrypt Hash Workshop 2007 (May 2007)
9. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: On the Indifferentiability of the Sponge Construction. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 181–197. Springer (2008)
10. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Sponge-Based Pseudo-Random Number Generators. In: Mangard, S., Standaert, F. (eds.) CHES 2010. LNCS, vol. 6225, pp. 33–47. Springer (2010)
11. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Duplexing the Sponge: Single-Pass Authenticated Encryption and Other Applications. In: Miri and Vaudenay [39], pp. 320–337
12. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: The Keccak reference (January 2011)
13. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: On the security of the keyed sponge construction. Symmetric Key Encryption Workshop (February 2011)
14. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Permutation-based encryption, authentication and authenticated encryption. Directions in Authenticated Ciphers (July 2012)
15. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G., Van Keer, R.: CAESAR submission: Ketje v2 (September 2016)
16. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G., Van Keer, R.: CAESAR submission: Keyak v2, document version 2.2 (September 2016)
17. Biham, E.: How to decrypt or even substitute DES-encrypted messages in 2^{28} steps. Inf. Process. Lett. 84(3), 117–124 (2002)
18. Biryukov, A., Mukhopadhyay, S., Sarkar, P.: Improved Time-Memory Trade-Offs with Multiple Data. In: Preneel, B., Tavares, S.E. (eds.) SAC 2005. LNCS, vol. 3897, pp. 110–127. Springer (2005)
19. Bogdanov, A., Knežević, M., Leander, G., Toz, D., Varıcı, K., Verbauwhede, I.: Spongent: A Lightweight Hash Function. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 312–325. Springer (2011)

20. CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness (November 2014), http://competitions.cr.yp.to/caesar.html
21. Chang, D., Dworkin, M., Hong, S., Kelsey, J., Nandi, M.: A keyed sponge construction with pseudorandomness in the standard model. NIST SHA–3 Workshop (March 2012)
22. Chatterjee, S., Menezes, A., Sarkar, P.: Another Look at Tightness. In: Miri and Vaudenay [39], pp. 293–319
23. Chen, S., Steinberger, J.P.: Tight Security Bounds for Key-Alternating Ciphers. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 327–350. Springer (2014)
24. Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M.: NORX v3.0 (2016), submission to CAESAR competition
25. Even, S., Mansour, Y.: A Construction of a Cipher From a Single Pseudorandom Permutation. In: Imai, H., Rivest, R.L., Matsumoto, T. (eds.) ASIACRYPT '91. LNCS, vol. 739, pp. 210–224. Springer (1991)
26. Even, S., Mansour, Y.: A Construction of a Cipher from a Single Pseudorandom Permutation. J. Cryptology 10(3), 151–162 (1997)
27. FIPS 202: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions (2015)
28. Fischlin, M., Coron, J. (eds.): EUROCRYPT 2016, Part I, LNCS, vol. 9665. Springer (2016)
29. Gazi, P., Pietrzak, K., Tessaro, S.: The Exact PRF Security of Truncation: Tight Bounds for Keyed Sponges and Truncated CBC. In: Gennaro and Robshaw [31], pp. 368–387
30. Gazi, P., Tessaro, S.: Provably Robust Sponge-Based PRNGs and KDFs. In: Fischlin and Coron [28], pp. 87–116
31. Gennaro, R., Robshaw, M. (eds.): CRYPTO 2015, Part I, LNCS, vol. 9215. Springer (2015)
32. Guo, J., Peyrin, T., Poschmann, A.: The PHOTON Family of Lightweight Hash Functions. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 222–239. Springer (2011)
33. Hoang, V.T., Tessaro, S.: Key-alternating ciphers and key-length extension: Exact bounds and multi-user security. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part I. LNCS, vol. 9814, pp. 3–32. Springer (2016)
34. Hong, J., Sarkar, P.: New Applications of Time Memory Data Tradeoffs. In: Roy, B.K. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 353–372. Springer (2005)
35. Jovanovic, P., Luykx, A., Mennink, B.: Beyond $2^{c/2}$ Security in Sponge-Based Authenticated Encryption Modes. In: Sarkar and Iwata [45], pp. 85–104
36. Leander, G. (ed.): FSE 2015, LNCS, vol. 9054. Springer (2015)
37. Maurer, U.M., Renner, R., Holenstein, C.: Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 21–39. Springer (2004)
38. Mennink, B., Reyhanitabar, R., Vizár, D.: Security of Full-State Keyed Sponge and Duplex: Applications to Authenticated Encryption. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015, Part II. LNCS, vol. 9453, pp. 465–489. Springer (2015)
39. Miri, A., Vaudenay, S. (eds.): SAC 2011, LNCS, vol. 7118. Springer (2012)
40. Mouha, N., Luykx, A.: Multi-key Security: The Even-Mansour Construction Revisited. In: Gennaro and Robshaw [31], pp. 209–223
41. Mouha, N., Mennink, B., Herrewege, A.V., Watanabe, D., Preneel, B., Verbauwhede, I.: Chaskey: An Efficient MAC Algorithm for 32-bit Microcontrollers. In:

Joux, A., Youssef, A.M. (eds.) SAC 2014. LNCS, vol. 8781, pp. 306–323. Springer (2014)
42. Naito, Y., Yasuda, K.: New bounds for keyed sponges with extendable output: Independence between capacity and message length. In: Peyrin, T. (ed.) FSE 2016. LNCS, vol. 9783, pp. 3–22. Springer (2016)
43. Patarin, J.: The "Coefficients H" Technique. In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) SAC 2008. LNCS, vol. 5381, pp. 328–345. Springer (2008)
44. Reyhanitabar, R., Vaudenay, S., Vizár, D.: Boosting OMD for Almost Free Authentication of Associated Data. In: Leander [36], pp. 411–427
45. Sarkar, P., Iwata, T. (eds.): ASIACRYPT 2014, Part I, LNCS, vol. 8873. Springer (2014)
46. Sasaki, Y., Yasuda, K.: How to Incorporate Associated Data in Sponge-Based Authenticated Encryption. In: Nyberg, K. (ed.) CT-RSA 2015. LNCS, vol. 9048, pp. 353–370. Springer (2015)