Leakage Resilience of the Duplex Construction

Christoph Dobraunig(✉) and Bart Mennink

Digital Security Group, Radboud University, Nijmegen, The Netherlands
{cdobraunig,b.mennink}@cs.ru.nl

Abstract. Side-channel attacks, especially differential power analysis (DPA), pose a serious threat to cryptographic implementations deployed in a malicious environment. One way to counter side-channel attacks is to design cryptographic schemes to withstand them, an area that is covered amongst others by leakage resilient cryptography. So far, however, leakage resilient cryptography has predominantly focused on block cipher based designs, and insights in permutation based leakage resilient cryptography are scarce. In this work, we consider leakage resilience of the keyed duplex construction: we present a model for leakage resilient duplexing, derive a fine-grained bound on the security of the keyed duplex in said model, and map it to ideas of Taha and Schaumont (HOST 2014) and Dobraunig et al. (ToSC 2017) in order to use the duplex in a leakage resilient manner.

Keywords: Duplex · Sponge · Security proof · Leakage resilience

1 Introduction

With the selection of Keccak [9] as SHA-3 [20], cryptography based on public permutations has become more and more popular. This is especially caused by the fact that the sponge [7] and the duplex [8] constructions provide a huge flexibility by enabling various cryptographic tasks besides hashing, such as encryption, authenticated encryption, and message authentication, by just relying on a public permutation. Keyed versions of the sponge and duplex constructions have been analyzed in a series of papers [1,8,10,12,15,21,26,30,31], however, this analysis has been done in a black-box scenario, not considering the leakage of information that occurs in applications where side-channel attacks are feasible.

Ever since the threat of side-channel attacks has become evident to the public [27,28], finding suitable protection mechanisms against this attack vector has become of increasing importance. One can identify two different ways to protect against side-channel attacks. The first one deals with hardening the implementation of cryptographic schemes by means of countermeasures like hiding [14] or masking [11,13,22,32,33]. The other one aims at developing dedicated schemes that provide easier protection against side-channel attacks in the first place,

S. D. Galbraith and S. Moriai (Eds.): ASIACRYPT 2019, LNCS 11923, pp. 225–255, 2019.
https://doi.org/10.1007/978-3-030-34618-8_8

like fresh re-keying [29] or leakage resilient cryptography [18]. With respect to the sponge and duplex constructions, there exist proposals of Taha and Schaumont [38] and Isap [16] that introduce dedicated algorithms that are claimed to provide protection against side-channel attacks.

Unfortunately, a closer look at the field of leakage resilient symmetric cryptography [6,17,19,34–36,41] reveals that the focus lies on constructions that can be instantiated with block ciphers. Hence, results regarding the leakage resilience of the keyed sponge, or more generally the keyed duplex construction that solely rely on unkeyed cryptographic permutations as building block are scarce. This particularly means that proposals such as those of [16,38] lack formal support regarding their leakage resilience.

1.1 Our Contribution

The contributions of this paper are manifold.

First, in Sect. 3, we describe a security model for leakage resilient duplexing. To do so, we start from the "ideal equivalent" of the keyed duplex of Daemen et al. [15], called an ideal extendable input function (IXIF), and present an adjusted version AIXIF. AIXIF is semantically equivalent to the IXIF if there is no leakage, but it allows to properly model leakage resilience of the keyed duplex. The model of leakage resilience of the duplex is now conceptually simple: as we argue in detail in Sect. 3.4, we consider a scheme leakage resilient if no attacker can distinguish a keyed duplex *that leaks for every query* from the random AIXIF. Here, we focus on non-adaptive leakage, where the leakage function is fixed in advance, akin to [17,19,35,37,41]. At this point our approach seems to be different from the typical models: the typical approach is to give a distinguisher access to a leaky version and a leak-free version of the cryptographic construction, and it has to distinguish the latter from a random function. The reason that we adopted a different model is that the duplex is just used as building block for encryption, authenticated encryption, or other types of functionalities. To prove that the use of a leakage resilient duplex gives rise to a leakage resilient construction with one of above-mentioned functionalities, the typical approach to give a distinguisher access to a leaky version and a leak-free version of the cryptographic construction has to be used again, as we will show later.

Second, in Sect. 5, we perform an in-depth and fine-grained analysis of the keyed duplex in the newly developed model. We take inspiration from Daemen et al. [15], who presented a detailed analysis of the keyed duplex in the black-box scenario, but the proof is not quite the same. To the contrary, due to various obstacles, it is not possible to argue similar to Daemen et al., nor to reduce the leakage resilience of a keyed duplex to its black-box security. Instead, we adopt ideas from the analysis of the NORX authenticated encryption scheme of Jovanovic et al. [26], and reason about the security of the keyed duplex in a sequential manner. One of the difficulties then is to determine the amount of min-entropy of a state in the duplex construction, given that the distinguisher may learn leakage from a duplex construction at different points in time. On the way, in Sect. 4 we give a detailed and accessible rationale of how leakage resilience proofs are performed in general and in our case.

Third, in Sect. 6, we interpret our results on the leakage resilience of the keyed duplex in the context of the proposals of Taha and Schaumont [38] and Isap [16]. In a nutshell, these proposals can be seen to consist of a sequential evaluation of two duplex constructions: one that "gains entropy" by absorbing a nonce with small portions at a time, and one that "maintains entropy" in the sense that after the nonce is absorbed any state that will be visited by the duplex has high entropy and will be visited only once. We will then have a closer look at one use case of such a keyed duplex, nonce-based stream encryption, in Sect. 7. We build this scheme using aforementioned ideas, and prove that it is leakage resilient in the conventional security model. The proof is hybrid and reduces security of the stream cipher to that of the underlying duplex.

1.2 Related Work

Guo et al. [23] independently considered leakage resilience of duplex based modes. Their work is more specifically targeted to authenticated encryption (rather than to the duplex as building block). A second difference is that it considers a more generous leakage assumption. We consider a bounded leakage model, that upper bounds the amount of information that an attacker learns by λ, whereas Guo et al. assume hard-to-invert leakages. As such, Guo et al. [23] follow a different approach that is complementary to ours, and that might likewise be relevant in many different use cases.

1.3 Notation

For $b \in \mathbb{N}$, the set of b-bit strings is denoted $\{0,1\}^b$ and the set of arbitrarily length strings is denoted $\{0,1\}^*$. We define by $\mathrm{func}(b)$ the set of all functions $\mathsf{f} : \{0,1\}^b \to \{0,1\}^b$ and by $\mathrm{perm}(b)$ the set of all permutations $\mathsf{p} : \{0,1\}^b \to \{0,1\}^b$. By $X \leftarrow Y$ we denote the assignment of the value Y to X, and by $X \xleftarrow{\$} \mathcal{X}$ we denote the uniformly random drawing of an element X from a finite set $\mathcal{X}$. For $X \in \{0,1\}^b$ and for $c \in \mathbb{N}$ with $c \leq b$, we denote by $\mathrm{left}_c(X)$ the c leftmost bits of X and by $\mathrm{right}_c(X)$ the c rightmost bits of X. We denote by $\mathsf{rot}_c(X)$ the right-rotation of X by c bits.

A random variable S has *min-entropy* at least h, denoted $H_\infty(S) \geq h$, if $\max_{s \in S} \mathbf{Pr}(S = s) \leq 2^{-h}$. The conditional min-entropy is straightforward to define: the probability term gets expanded by the condition.

2 Keyed Duplex Construction

Let $b, c, r, k, u, \alpha \in \mathbb{N}$, with $c + r = b$, $k \leq b$, and $\alpha \leq b - k$. We describe the keyed duplex construction KD in Algorithm 1. The keyed duplex construction gets as input a key array $\boldsymbol{K} = (K[1], \ldots, K[u]) \in (\{0,1\}^k)^u$ consisting of u keys, and it is instantiated using a b-bit permutation $\mathsf{p} \in \mathrm{perm}(b)$. The construction internally maintains a b-bit state S, and has two interfaces: $\mathsf{KD.init}$ and $\mathsf{KD.duplex}$.

The initialization interface gets as input a key index $\delta \in [1, u]$ and an initialization vector $IV \in \mathcal{IV} \subseteq \{0,1\}^{b-k}$, and initializes the state with the δ-th

Algorithm 1. Keyed duplex construction $\mathsf{KD}[\mathsf{p}]_{\boldsymbol{K}}$

Interface: KD.init
Input: $(\delta, IV) \in [1, u] \times \mathcal{IV}$
Output: $\varnothing$
  $S \leftarrow \mathsf{rot}_\alpha(\boldsymbol{K}[\delta] \parallel IV)$
  $S \leftarrow \mathsf{p}(S)$
  return $\varnothing$

Interface: KD.duplex
Input: $(\mathit{flag}, P) \in \{\mathit{true}, \mathit{false}\} \times \{0,1\}^b$
Output: $Z \in \{0,1\}^r$
  $Z \leftarrow \mathrm{left}_r(S)$
  $S \leftarrow S \oplus [\mathit{flag}] \cdot (Z \| 0^{b-r}) \oplus P$ ▷ if *flag*, overwrite outer part
  $S \leftarrow \mathsf{p}(S)$
  return Z

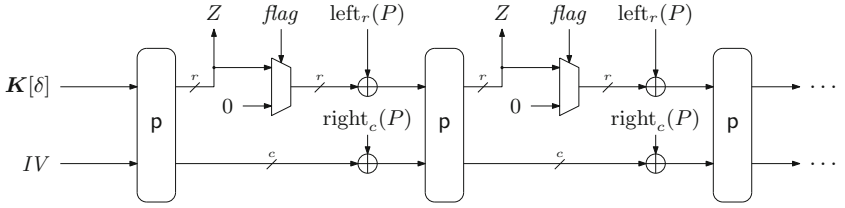

Fig. 1. The duplexing interface of KD.

key and the initialization vector IV as $S \leftarrow \mathsf{rot}_\alpha(\boldsymbol{K}[\delta] \parallel IV)$, followed by an evaluation of the underlying permutation p on the state S. It outputs nothing. Note that the constant α simply determines the bit positions where to place the key. We will see different examples of the value α in Sect. 6.

The duplexing interface gets as input a flag $\mathit{flag} \in \{\mathit{true}, \mathit{false}\}$ and a new data block $P \in \{0,1\}^b$. The interface outputs an r-bit block $Z \in \{0,1\}^r$ off the internal state S, transforms the state using the new data block P, and finally evaluates the underlying permutation p on the state. The flag *flag* describes how absorption is done on the r leftmost bits of the state that are squeezed: those r bits are either overwritten (if $\mathit{flag} = \mathit{true}$) or XORed with r bits of the input block P (if $\mathit{flag} = \mathit{false}$). See also Fig. 1, where the duplex is depicted for key offset $\alpha = 0$.

This description is another rephasing of how the duplex construction can be viewed compared to the original description used by Bertoni et al. [8], but also differs from the rephased description of Daemen et al. [15]. Compared to Daemen et al. the call of the underlying permutation is done at the end of the duplexing call instead of the beginning. This way of describing the duplex eases the proof in the leakage resilient setting, while at the same time empowers a leakage-aware attacker to adaptively react to the leakage of the permutation before providing new inputs. However, it still reflects the usage of the duplex in the same way as the description of Daemen et al. [15]. In particular, Daemen et al. also already considered multi-user security by default, and likewise had

two different types of duplexing calls (for $flag \in \{true, false\}$) to allow implementation of SpongeWrap and variants using the duplex construction. Indeed, whereas SpongeWrap encryption can be performed using $\mathsf{KD.duplex}(false, \cdot)$, the decryption function must be performed using evaluations of $\mathsf{KD.duplex}(true, \cdot)$.

3 Security Model

In this section, we will describe our leakage resilience security model for the keyed duplex. We consider sampling of keys in Sect. 3.1. We settle the basic notation of distinguishers in Sect. 3.2. For reference, the black-box duplex security model of Daemen et al. [15] is treated in Sect. 3.3. We lift the model to leakage resilience in Sect. 3.4.

3.1 Sampling of Keys

The duplex construction of Sect. 2 is based on an array of u k-bit keys. These keys may be generated uniformly at random, as $\boldsymbol{K} \xleftarrow{\mathcal{D}_{\boldsymbol{K}}} (\{0,1\}^k)^u$. In our analysis of leakage resilience, however, we will require the scheme to be still secure if the keys are not uniformly random but as long as they have sufficient min-entropy. Henceforth, we will adopt the approach of Daemen et al. [15] to consider keys sampled using a distribution $\mathcal{D}_{\boldsymbol{K}}$, that distributes the key independently[1] and with sufficient min-entropy, i.e., for which

$$H_\infty(\mathcal{D}_{\boldsymbol{K}}) = \min_{\delta \in [1,u]} H_\infty(\boldsymbol{K}[\delta])$$

is sufficiently high. Note that if $\mathcal{D}_{\boldsymbol{K}}$ is the random distribution, $H_\infty(\mathcal{D}_{\boldsymbol{K}}) = k$.

3.2 Distinguishers

A distinguisher D is an algorithm that is given access to one or more oracles O, denoted D^{O}, and that outputs a bit $b \in \{0,1\}$ after interaction with O. If O and P are oracles, we denote by $\Delta_{\mathsf{D}}(\mathsf{O}\ ;\ \mathsf{P})$ the advantage of a distinguisher D in distinguishing O from P. In our work, we will only be concerned with information-theoretic distinguishers: these have unbounded computational power, and their success probabilities are solely measured by the number of queries made to the oracles.

3.3 Black-Box Security

Daemen et al. [15] described the ideal extendable input function (IXIF) as ideal equivalent for the keyed duplex. We will also consider this function, modulo syntactical changes based on the changes we made on the keyed duplex in Sect. 2. The function is described in Algorithm 2.

[1] In Daemen et al. [15], the keys need not be mutually independent, but omitting this conditions will give various tricky corner cases in the analysis of leakage resilience.

Algorithm 2. Ideal extendable input function $\mathsf{IXIF}[\mathsf{ro}]_{\boldsymbol{K}}$

Interface: IXIF.init
Input: $(\delta, IV) \in [1, u] \times \mathcal{IV}$
Output: $\varnothing$
  $path \leftarrow \mathsf{encode}[\delta] \parallel IV$
  return $\varnothing$

Interface: IXIF.duplex
Input: $(\mathit{flag}, P) \in \{\mathit{true}, \mathit{false}\} \times \{0,1\}^b$
Output: $Z \in \{0,1\}^r$
  $Z \leftarrow \mathsf{ro}(path, r)$
  $path \leftarrow path \parallel ([\mathit{flag}] \cdot (Z \| 0^{b-r}) \oplus P)$ ▷ if *flag*, overwrite outer part
  return Z

The IXIF has the same interface as the keyed duplex, but instead of being based on a key array $\boldsymbol{K} \in (\{0,1\}^k)^u$ and being built on primitive $\mathsf{p} \in \mathrm{perm}(b)$, it is built on a random oracle $\mathsf{ro} : \{0,1\}^* \times \mathbb{N} \to \{0,1\}^\infty$, that is defined as follows. Let $\mathsf{ro}_\infty : \{0,1\}^* \to \{0,1\}^\infty$ be a random oracle in the sense of Bellare and Rogaway [3]. For $P \in \{0,1\}^*$, $\mathsf{ro}(P, r)$ outputs the first r bits of $\mathsf{ro}(P)$. The IXIF maintains a path *path*, in which it unambiguously stores all data input by the user. It is initialized by $\mathsf{encode}[\delta] \parallel IV$ for some suitable injective encoding function $\mathsf{encode} : [1, u] \to \{0,1\}^k$, and upon each duplexing call, the new message block is appended to the path. Duplexing output is generated by evaluating the random oracle on *path*.

Let $b, c, r, k, u, \alpha \in \mathbb{N}$, with $c + r = b$, $k \leq b$, and $\alpha \leq b - k$. Let $\mathsf{p} \xleftarrow{\$} \mathrm{perm}(b)$ be a random transformation, ro be a random oracle, and $\boldsymbol{K} \xleftarrow{\mathcal{D}_K} (\{0,1\}^k)^u$ a random array of keys. In the black-box security model, one considers a distinguisher that has access to either $(\mathsf{KD}[\mathsf{p}]_{\boldsymbol{K}}, \mathsf{p}^\pm)$ in the real world or $(\mathsf{IXIF}[\mathsf{ro}], \mathsf{p}^\pm)$ in the ideal world, where "$\pm$" stands for the fact that the distinguisher has bi-directional query access:

$$\mathbf{Adv}^{\mathrm{bb}}_{\mathsf{KD}}(\mathsf{D}) = \Delta_{\mathsf{D}}\left(\mathsf{KD}[\mathsf{p}]_{\boldsymbol{K}}, \mathsf{p}^\pm \; ; \; \mathsf{IXIF}[\mathsf{ro}], \mathsf{p}^\pm\right). \tag{1}$$

This is the model explicitly considered by Daemen et al. [15].

3.4 Leakage Resilience

We consider non-adaptive leakage resilience of the keyed duplex construction. Non-adaptive leakage has been considered before in [17,19,35,37,41], among others, and we will use the description of $\mathcal{L}$-resilience of Dodis and Pietrzak [17]. These models, however, consider the underlying primitive to be a block cipher or weak PRF, whereas in our setting it is a public permutation. In addition, the duplex has its characteristic property that it allows variable length input *and* variable length output. A final, and technically more delicate difference (as becomes clear below), is that the duplex consists of two oracles init and duplex, which the distinguisher may call interchangeably at its own discretion.

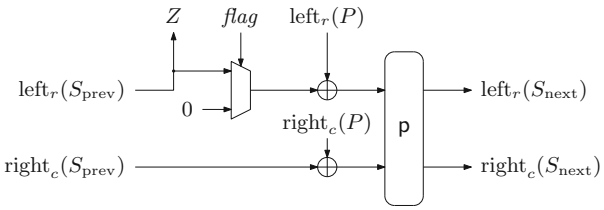

Fig. 2. An evaluation of KD.duplex, with its previous state S_{prev} and next state S_{next} are indicated. Intuitively, leakage occurs on both states, and the leakage function L returns λ bits of leakage.

We will assume that only values leak information that take part in the current computation, i.e., only leakage can occur from information that is used in calls to init and duplex. Note that this general way to describe what information can leak does not put restrictions on how this leakage occurs. For instance, this model covers even very strong attackers that can directly probe a limited amount of bits in a circuit, or that can get some limited amount of information about all values that are used in the current computation.

Recall from (1) that in the black-box model, one compares $(\mathsf{KD}[\mathsf{p}]_{\boldsymbol{K}}, \mathsf{p}^{\pm})$ with $(\mathsf{IXIF}[\mathsf{ro}], \mathsf{p}^{\pm})$, where $\mathsf{p} \xleftarrow{\$} \text{perm}(b)$ and ro is a random oracle. In order to prove leakage resilience of the construction, we have to demonstrate that "leakage does not help". For the real keyed duplex $\mathsf{KD}[\mathsf{p}]_{\boldsymbol{K}}$, modeling this is as simple as giving the distinguisher the leakage value $\ell \leftarrow \mathsf{L}(S_{\text{prev}}, \mathit{flag}, P, S_{\text{next}})$, where $\mathsf{L} : \{0,1\}^b \times \{\mathit{true}, \mathit{false}\} \times \{0,1\}^b \times \{0,1\}^b \to \{0,1\}^\lambda$ is the leakage function, S_{prev} the state before the call, and S_{next} the state after the call. See also Fig. 2.

For the ideal world $\mathsf{IXIF}[\mathsf{ro}]$, there is no such thing as a state, and simply generating random leakage allows for a trivial win for the distinguisher, as leaked bits may happen to coincide with the actual squeezed bits. For example, if L is defined as $\mathsf{L}(S_{\text{prev}}, \mathit{flag}, P, S_{\text{next}}) = \text{left}_\lambda(S_{\text{next}})$, in the real world, any leakage ℓ satisfies $\ell = \text{left}_\lambda(Z)$, whereas in the ideal world this equation holds with probability around $1/2^\lambda$, only. We resolve this by making a minor tweak to the duplexing interface of IXIF: the oracle maintains a dummy state S, and instead of $Z \leftarrow \mathsf{ro}(\mathit{path}, r)$, it gets Z from this dummy state $Z \leftarrow \text{left}_r(S)$ and updates the dummy state constantly by doing $S \leftarrow \mathsf{ro}(\mathit{path}, b)$. The dummy state is initialized as in the normal duplex (Algorithm 1). The resulting adjusted IXIF (AIXIF) is given in Algorithm 3.

It is important to note that the change from IXIF to AIXIF is purely administrative, in that for any distinguisher D,

$$\Delta_{\mathsf{D}}\left(\mathsf{IXIF}[\mathsf{ro}]\ ;\ \mathsf{AIXIF}[\mathsf{ro}]_{\boldsymbol{K}}\right) = 0.$$

The reason is that (i) an initialized state $S = \mathsf{rot}_\alpha(\boldsymbol{K}[\delta] \parallel IV)$ is never used for outputting data to the distinguisher, and (ii) later versions of the dummy state are always updated with b bits of ro-output of which only r bits are squeezed a

Algorithm 3. Adjusted ideal extendable input function $\mathsf{AIXIF}[\mathsf{ro}]_{\boldsymbol{K}}$

Interface: AIXIF.init
Input: $(\delta, IV) \in [1,u] \times \mathcal{IV}$
Output: $\varnothing$
  $path \leftarrow \mathsf{encode}[\delta] \,\|\, IV$
  $S \leftarrow \mathsf{rot}_\alpha(\boldsymbol{K}[\delta] \,\|\, IV)$
  $S \leftarrow \mathsf{ro}(path, b)$
  return $\varnothing$

Interface: AIXIF.duplex
Input: $(\mathit{flag}, P) \in \{\mathit{true}, \mathit{false}\} \times \{0,1\}^b$
Output: $Z \in \{0,1\}^r$
  $Z \leftarrow \mathrm{left}_r(S)$
  $path \leftarrow path \,\|\, ([\mathit{flag}] \cdot (Z \| 0^{b-r}) \oplus P)$ ▷ if *flag*, overwrite outer part
  $S \leftarrow \mathsf{ro}(path, b)$
  return Z

single time. Therefore, the original black-box security model could just as well be defined based on AIXIF. The good thing of AIXIF, now, is that it allows to easily formalize security in the leakage resilience setting where each construction call leaks.

Let $b, c, r, k, u, \alpha, \lambda \in \mathbb{N}$, with $c+r=b$, $k \le b$, $\alpha \le b-k$, and $\lambda \le 2b$. Let $\mathsf{p} \xleftarrow{\$} \mathrm{perm}(b)$ be a random permutation, ro be a random oracle, and $\boldsymbol{K} \xleftarrow{\mathcal{D}_K} (\{0,1\}^k)^u$ a random array of keys. Let $\mathcal{L} = \{\mathsf{L} : \{0,1\}^b \times \{\mathit{true}, \mathit{false}\} \times \{0,1\}^b \times \{0,1\}^b \to \{0,1\}^\lambda\}$ be a class of leakage functions, and for any leakage function $\mathsf{L} \in \mathcal{L}$, define by $\mathsf{KD}[\mathsf{p}]^{\mathsf{L}}_{\boldsymbol{K}}$ (resp., $\mathsf{AIXIF}[\mathsf{ro}]^{\mathsf{L}}_{\boldsymbol{K}}$) the keyed duplex (resp., adjusted ideal extendable input function) that for each construction call leaks $\mathsf{L}(S_{\mathrm{prev}}, \mathit{flag}, P, S_{\mathrm{next}})$, where S_{prev} is the state before the call and S_{next} the state after the call. In the leakage resilience security model, one considers a distinguisher that has access to either $(\mathsf{KD}[\mathsf{p}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{p}^{\pm})$ in the real world, and $(\mathsf{AIXIF}[\mathsf{ro}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{p}^{\pm})$ in the ideal world, maximized over all possible leakage functions $\mathsf{L} \in \mathcal{L}$:

$$\mathbf{Adv}^{\mathcal{L}\text{-naLR}}_{\mathsf{KD}}(\mathsf{D}) = \max_{\mathsf{L} \in \mathcal{L}} \Delta_{\mathsf{D}}\left(\mathsf{KD}[\mathsf{p}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{p}^{\pm} \;;\; \mathsf{AIXIF}[\mathsf{ro}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{p}^{\pm}\right). \tag{2}$$

Note that we indeed consider non-adaptive leakage resilience, as we maximize over all possible leakage functions L. Note furthermore that we do not consider future computation: the keyed duplex construction is based on the random permutation p and the set of allowed leakage functions is independent of p; the functions simply operate on the state right before and right after the transformation that leaks.

Remark 1. It is important to observe that, in our model, *any* duplex call leaks. In this way, our model seems to be conceptually different to the established models of, e.g., [17,19,35,37,41]. At a high level, in these models, the distinguisher has access to a leak-free version of the construction, which it has to distinguish from random, and a leaky version of the construction, which it may

use to gather information. The intuition is that, whatever the distinguisher may learn from leakage, any new evaluation of the construction still looks random. In comparison, in our model of (2), we simply assume that the construction *always leaks*: the real construction KD.duplex leaks actual data of the state, whereas AIXIF.duplex leaks random data. This can be tolerated in our model as, typically, the KD.duplex will be used as *building block* for constructions that enable functionalities like, e.g., encryption. When we realize leakage resilient encryption with the help of the keyed duplex in Sect. 7, we consider the established model where the distinguisher has access to a leaky and a leak-free version of the construction, and the latter has to be distinguished from random.

4 Proof Rationale

In this section, we outline the rationale of proving leakage resilience of the keyed duplex. The section is extensive, but should give a high-level overview of how the security analysis is performed. First, in Sect. 4.1, we detail how typically leakage resilience of sequential constructions is proven. Then, in Sect. 4.2, we explain to what degree these approaches apply to permutation based cryptography. In Sect. 4.3, we consider the keyed duplex construction in more detail, and explain at a high level how the security proof is performed and how it relies on existing research on the keyed duplex construction in the black-box model. The discussion will form a stepping stone to the formal analysis of the keyed duplex in Sect. 5 and of the application of the result in Sect. 6.

4.1 Proving Leakage Resilience

The rationale of leakage resilience security proofs is not straightforward, and the main cause of this is the delicate selection of entropy measure for a leaky state. First off, it is important to know that starting from the seminal work of Dziembowski and Pietrzak [18], almost all leakage resilient PRGs and PRFs in literature [5,6,17,19,34,35,40,41] are sequential: they maintain a state, and use a cryptographic primitive to evolve the state in a sequential manner and to output a random stream. The cryptographic primitive is, in most of these cases, a block cipher modeled as a weak PRF $\mathsf{F} : \{0,1\}^k \times \{0,1\}^m \rightarrow \{0,1\}^n$.

A measure to identify the amount of randomness of a value is the min-entropy. Informally, a value S has min-entropy $H_\infty(S) \geq h$ if the success probability of guessing S is at most $1/2^h$. Unfortunately, the min-entropy is not fully suited to deal with leakage in above-mentioned sequential constructions: each round, certain information of a state leaks, and the min-entropy will only *decrease* with the leakage over time. Dziembowski and Pietrzak [18] observed that one does not strictly need the min-entropy of the state to be high enough: all that is needed is that the state is *computationally indistinguishable from a state with sufficiently high min-entropy*, in the eye of the computationally bounded distinguisher. This

is formalized by the HILL-pseudoentropy [24] (or formally the conditional HILL-pseudoentropy [25], taking into account leakage data). The security proofs of above constructions now all exist of an iterative execution of the following steps:

(1) If the input to the wPRF F has sufficiently high min-entropy, then with high probability the output is an n-bit pseudorandom value S;
(2) If λ bits of the n-bit pseudorandom state S are leaked, then with high probability the state has HILL-pseudoentropy at least $n - 2\lambda$;
(3) By definition of the HILL-pseudoentropy, the state is computationally indistinguishable from a state with min-entropy at least $n - 2\lambda$;
(4) The resulting state will be (part of the) input to next round's wPRF.

A formalization of the first three steps can be found in [35, Lemma 2], [35, Lemma 6], and [35, Definition 3]. We note that the original introduction of leakage resilient cryptography of Dziembowski and Pietrzak [18] did not consider a weak PRF but a (stronger) PRG.

It is clear that an iterative execution of above steps allows to prove security of a sequential wPRF-based construction, provided that the state after step (4) has enough min-entropy to make the application of step (1) in next round go through. The iterative execution allows to prove security of the construction, with a security loss quantified by a sum of the individual losses in steps (1)–(3) for each of the rounds. More importantly, the security proof stands under the assumption that the block cipher is a weak PRF, or can be used to construct a weak PRF (see also Standaert et al. [37]). At this point, it requires cryptanalysts to investigate the weak PRF security of actual block ciphers.

4.2 Towards Permutation-Based Constructions

The focus in our work is on constructions based on cryptographic permutations. In the black-box model, both the keyed sponge [1,10,12,21,26,30,31] and the keyed duplex [8,15,30] have received thorough investigation.

The security analyses are different from black-box analyses of block cipher based constructions: whereas for the latter one argues security under the assumption that the block cipher is a (strong) pseudorandom permutation, in the former one assumes that the permutation is perfect and considers a distinguisher that is computationally unbounded and whose complexity is only measured by the online complexity (the amount of construction queries) and the offline complexity (the amount of primitive queries).

The approach is well-established, and in our analysis of the leakage resilience of the duplex, we adopt the approach. This gives two significant advantages in the analysis. First off, we consider computationally unbounded adversaries, and there is no need to make the HILL-detour. In other words, we can *directly* argue that an n-bit pseudorandom state S has min-entropy at least $n - \lambda$ after λ bits are leaked. Second, there is no issue with repeated min-entropy degradation: the state is transformed through a perfectly random permutation that outputs a random value (bar repetition) for each new input. We remark that concurrent

work [23] also builds upon the random permutation model (and, in addition, an ideal tweakable block cipher).

These two advantages clearly simplify the rationale and simplicity of the leakage resilience security analysis of the duplex, yet do not make the security analysis a trivial extension of earlier leakage resilience analyses: in the new setting, the amount of entropy of a state is not only dependent on the leakage, but *also* on the primitive queries that the distinguisher makes, recalling that the distinguisher has *direct access* to the primitive. Indeed, this is not the case in ordinary wPRF-based security proofs.

There is another complication in the analysis of our construction: the distinguisher can re-initialize the state and start over. This is in line with the particular application of the duplex: authenticated encryption, where different authenticated encryptions may start from the same state and even have identical first permutation calls. Even if we had the possibility to argue that the duplex primitive is a weak PRF, repeated or mutually related states would invalidate step (1) of above reasoning, as the query history would skew the distribution of the weak PRF. In detail, step (1) requires the inputs to be close-to-random, a condition that appears to be more delicate than one would expect (cf., [41]), and that is false for repeated states in the duplex.

In a nutshell, one can say that the main overlap in our leakage resilience analysis compared with earlier approaches [5,6,17,19,34,35,40,41] is that we use the min-entropy to express the amount of randomness that is left after leakage, and we argue security based on the assumption that all state values in a keyed duplex have enough entropy.

4.3 Proving Security of Duplex Construction

Our proof uses many ideas from the solid black-box research already performed on keyed sponges and duplexes [1,8,10,12,15,21,26,30,31]. However, not all techniques from this line of research are suited in the leakage resilience setting. Most importantly, a notable technique [1,12,15,30] is to view the keyed sponge/duplex as a mode based on an Even-Mansour construction on top of the permutation $\mathsf{p} \in \mathrm{perm}(b)$. The trick is to XOR two copies of a dummy key with the inner part in-between every two evaluations of the permutation p. The change is purely syntactical, and a distinguisher cannot note the difference. However, in the leakage resilience setting, the distinguisher may have chosen the leakage function L so as to leak part of the state that is keyed, and XORing dummy keys turns out to become tricky. In particular, adoption of the approach to the leakage resilience setting would require us to be able to "split" leakages into input leakages and output leakages, but this is not always possible, depending on the leakage function L.

Instead, the proof resembles much of the approach of Jovanovic et al. [26], who performed a direct security proof of the NORX nonce-based authenticated encryption scheme that also applied to other CAESAR candidates. At a high level, the proof of Jovanovic et al. consists of observing that the output states are always uniformly random (bar repetition, as a permutation is evaluated),

as long as no bad event occurs. A bad event, in turn, occurs if there are two construction queries with colliding states or if there is a construction query and a primitive query with colliding states. The absence of collisions is dealt with in the first phase by replacing the random permutation by a function that samples values from random at the cost of an RP-to-RF switch.

In our leakage resilience proofs, we follow the same approach. We also start by replacing the random permutation by a function f, that samples values from random and provides two-sided oracle access. Then, as long as the state of the keyed duplex has enough entropy, the result after applying f is random and also has enough entropy. Clearly, the entropy of the state reduces with the amount of leakage that occurs on the state, and consequently, bad events happen with a slightly larger probability as before. This also shows that estimating (formally, lower bounding) the amount of min-entropy of the states in the keyed duplex construction is important for deriving a tight security bound.

Focus on the keyed duplex (KD) of Algorithm 1, based on a function $\mathsf{f} \xleftarrow{\$} \mathrm{func}(b)$, and consider a duplex state $S_{\mathrm{prev}} \in \{0,1\}^b$. Assume that the interface $\mathsf{KD.duplex}$ is evaluated on this state for R different inputs,

$$\{(\mathit{flag}_i, P_i)\}_{i=1}^{R}.$$

As the previous state S_{prev} is the direct output of a call to a function f that samples b-bit values from random, S_{prev} is a value with min-entropy b minus the leakage occurred on this function call. Clearly, the R evaluations of the duplex in question are made for the same state S_{prev}, and hence, in total they reduce the entropy of S_{prev} further by at most $R \cdot \lambda$ bits due to the next function call. In addition, by regular squeezing, the distinguisher learns r bits of the state. In total, S_{prev} has conditional min-entropy at least

$$b - r - (R+1)\lambda.$$

If this entropy is sufficiently high, we get R new states S_{next} with min-entropy b minus the leakage occurred from one function call. The main lesson learned from this: a state that could be duplexed for different message blocks *should have small-rate absorption* (as this bounds R), and a unique state can be used for larger rates *even up to full-state absorption*.

5 Leakage Resilience of Keyed Duplex Construction

We will prove non-adaptive leakage resilience of the keyed duplex construction based on a cryptographic permutation $\mathsf{p} \xleftarrow{\$} \mathrm{perm}(b)$ in the model of Sect. 3.4 (see (2)). Although the generic construction and the model are based on the work of Daemen et al. [15], the security proof approach differs, as explained in Sect. 4.3. We quantify distinguishers in Sect. 5.1. The main security result is stated in Sect. 5.2, and an interpretation of it is given in Sect. 5.3. The proof is given in Sect. 5.4.

5.1 Distinguisher's Resources

We consider an information-theoretic distinguisher D that has access to either the real world $(\mathsf{KD}[\mathsf{p}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{p}^{\pm})$ or the ideal world $(\mathsf{AIXIF}[\mathsf{ro}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{p}^{\pm})$, where p is some permutation and L some leakage function. Two basic measures to quantify the distinguisher's resources are its online complexity M and offline complexity N:

- M: the number of distinct construction calls, either initialization or duplexing calls;
- N: the number of distinct primitive queries.

For each construction call, we define a path *path* that "registers" the data that got absorbed in the duplex up to the point that the cryptographic primitive (p in the real world and ro in the ideal world) is evaluated. For an initialization call $(\delta, IV) \mapsto \varnothing$, the associated path is defined as $path = \mathsf{encode}[\delta] \parallel IV$. For each duplexing call $(flag, P) \mapsto Z$, the value $[flag] \cdot (Z \| 0^{b-r}) \oplus M$ is appended to the path of the previous construction query. Not surprisingly, the definition matches the actual definition of *path* in the $\mathsf{AIXIF}[\mathsf{ro}]_{\boldsymbol{K}}$ construction of Algorithm 3, but defining the same thing for the real world will allow us to better reason about the security of the keyed duplex. Note that the value *path* contains no information that is secret to the distinguisher. In order to reason about duplexing calls, we will also define a *subpath* of a *path*, which is the path leading to the particular duplexing call. In other words, for a path *path*, it *subpath* is simply *path* with the last b bits removed.

In order to derive a detailed and versatile security bound, that in particular well-specifies how leakage influences the bound, we further parameterize the distinguisher as follows. For initialization calls:

- q: the number of initialization calls;
- q_{IV}: the maximum number of initialization calls for a single IV;
- q_δ: the maximum number of initialization calls for a single δ.

For duplexing calls:

- Ω: the number of duplexing queries with $flag = true$;
- L: the number of duplexing calls with repeated subpath, i.e., M minus the number of distinct subpaths;
- R: the maximum number of duplexing calls for a single non-empty *subpath*.

Note that these parameters can all be described as a function of the duplexing calls and the related *path*'s, and the distinguisher can compute these values based on the queries it made so far. The parametrization of the distinguisher is roughly as that of Daemen et al. [15], but we have added parameter R: it maximizes the number of occurrences of a path *subpath* for different inputs $(flag, P)$. The parameter will be used to determine, factually upper bound, the amount of leakage that the distinguisher learns on a state *after* the duplexing call. Indeed, if a certain path *subpath* occurs R times, this means that these R duplexing calls have the same input-state, and any evaluation of p in one of

these duplexing calls leaks information about that state. In total, this results in a maximum amount of $R+1$ leakages. The parameter R is related to parameter L, but it is not quite the same. The parameters Ω and L are, as in [15], used to upper bound the number of duplexing calls for which the distinguisher may have *set* the r leftmost bits of the input to the permutation in the duplexing call to a certain value of its choice. This brings us to the last parameter:

– ν_{fix}: the maximum number of duplexing calls for which the adversary has set the outer part to a single value $\mathrm{left}_r(T)$.

Note that $\nu_{\mathsf{fix}} \leq L + \Omega$, but it may be much smaller in specific use cases of the duplex, for example, if overwrites only happen for unique values.

5.2 Main Result

We will use a notion from Daemen et al. [15], namely that of the multicollision limit function.

Definition 1 (multicollision limit function). *Let $M, c, r \in \mathbb{N}$. Consider the experiment of throwing M balls uniformly at random in 2^r bins, and let μ be the maximum number of balls in a single bin. We define the multicollision limit function $\nu_{r,c}^M$ as the smallest natural number x that satisfies*

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

We derive the following result on the keyed duplex under leakage.

Theorem 1. *Let $b, c, r, k, u, \alpha, \lambda \in \mathbb{N}$, with $c + r = b$, $k \leq b$, $\alpha \leq b - k$, and $\lambda \leq 2b$. Let $\mathsf{p} \xleftarrow{\$} \mathrm{perm}(b)$ be a random permutation, and $\boldsymbol{K} \xleftarrow{\mathcal{D}_{\boldsymbol{K}}} (\{0,1\}^k)^u$ a random array of keys. Let $\mathcal{L} = \{\mathsf{L} : \{0,1\}^b \times \{0,1\}^b \to \{0,1\}^\lambda\}$ be a class of leakage functions. For any distinguisher D quantified as in Sect. 5.1,*

$$\begin{aligned}\mathbf{Adv}_{\mathsf{KD}}^{\mathcal{L}\text{-naLR}}(\mathsf{D}) &\leq \frac{\nu_{\mathsf{fix}} N}{2^{c-(R+1)\lambda}} + \frac{2\nu_{r,c}^M N}{2^{c-(R+1)\lambda}} + \frac{2\nu_{r,c}^M}{2^c} + \frac{\nu_{r,c}^M(L+\Omega) + \frac{\nu_{\mathsf{fix}}-1}{2}(L+\Omega)}{2^{c-R\lambda}} \\ &+ \frac{\binom{M-L-q}{2} + (M-L-q)(L+\Omega)}{2^{b-\lambda}} + \frac{\binom{M+N}{2} + \binom{N}{2}}{2^b} \\ &+ \frac{q(M-q)}{2^{H_\infty(\mathcal{D}_{\boldsymbol{K}})+\min\{c,\max\{b-\alpha,c\}-k\}-(R+q_\delta)\lambda}} + \frac{q_{IV} N}{2^{H_\infty(\mathcal{D}_{\boldsymbol{K}})-q_\delta\lambda}} + \frac{\binom{u}{2}}{2^{H_\infty(\mathcal{D}_{\boldsymbol{K}})}}.\end{aligned}$$

In addition, except with probability at most the same bound, the final output states have min-entropy at least $b - \lambda$.

The proof is given in Sect. 5.4; we first give an interpretation of the bound in Sect. 5.3.

5.3 Interpretation

By rephasing the duplex and by going over the duplex in a sequential manner (as [26]), and by only absorbing isolated concepts from Daemen et al. [15] (the quantification and the multicollision limit function), the proof is intuitively simpler to follow than the black-box variant. This is in part due to the fact that we start the proof with a transformation reminiscent of the RP-to-RF switch. This simplifies the proof at various aspects (for example, at the application of the multicollision limit function) but is not for free, as it induces an extra term of around $\binom{M+N}{2}/2^b$.

The proof is still fairly general, in part due to the presence of the term $\nu^M_{r,c}$. A naive bounding akin to the derivation of Jovanovic et al. [26] would give a bound

$$\nu^M_{r,c} \leq \max\left\{r, \left(\frac{2eM2^c}{2^r}\right)^{1/2}\right\},$$

but the bound is loose, in particular for small r. Daemen et al. [15] gave a more detailed analysis of the term, including two lemmas upper bounding it. Omitting details, one can think of the multicollision limit function to behave as follows [15]:

$$\nu^M_{r,c} \lesssim \begin{cases} b/\log_2\left(\frac{2^r}{M}\right), \text{ for } M \lesssim 2^r, \\ b \cdot \frac{M}{2^r}, \text{ for } M \gtrsim 2^r. \end{cases}$$

Beyond this multicollision term, the bound of Theorem 1 is complicated due to the multivariate quantification of the distinguisher's resources, and most importantly the terms L and Ω. In Sect. 6, we will consider how the duplex can be used to create leakage resilient cryptographic schemes, and see how the bound simplifies drastically for specific use cases.

5.4 Proof of Theorem 1

Let $\mathsf{L} \in \mathcal{L}$ be any leakage function. Consider any information-theoretic distinguisher D. Our goal is to bound

$$\Delta_{\mathsf{D}}\left(\mathsf{KD}[\mathsf{p}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{p}^{\pm}\ ;\ \mathsf{AIXIF}[\mathsf{ro}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{p}^{\pm}\right). \tag{3}$$

The first step is to replace p with a function $\mathsf{f} : \{0,1\}^b \to \{0,1\}^b$ that has the same interface as p. The function f maintains an initially empty list $\mathcal{F}$ of input/output tuples (X, Y). For a new query $\mathsf{f}(X)$ with $(X, \cdot) \notin \mathcal{F}$, it generates $Y \xleftarrow{\$} \{0,1\}^b$ and returns this value. For a new query $\mathsf{f}^{-1}(Y)$ with $(\cdot, Y) \notin \mathcal{F}$, it generates $X \xleftarrow{\$} \{0,1\}^b$ and returns this value. In both cases, the primitive adds (X, Y) to $\mathcal{F}$, and it aborts if this addition yields a collision in X or in Y. Clearly, as long as f does not abort, the function is perfectly indistinguishable from p, so we get:

$$\Delta_{\mathsf{D}}\left(\mathsf{KD}[\mathsf{p}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{p}^{\pm}\ ;\ \mathsf{KD}[\mathsf{f}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{f}^{\pm}\right) \leq \frac{\binom{M+N}{2}}{2^b}\,,$$

$$\Delta_{\mathsf{D}}\left(\mathsf{AIXIF}[\mathsf{ro}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{p}^{\pm}\ ;\ \mathsf{AIXIF}[\mathsf{ro}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{f}^{\pm}\right) \leq \frac{\binom{N}{2}}{2^b}\,,$$

as in the former there are $M+N$ evaluations of p and in the latter there are N. Note that this is a purely probabilistic case, and the switch does not involve/concern any leakage. From (3) we get

$$\Delta_{\mathsf{D}}\left(\mathsf{KD}[\mathsf{p}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{p}^{\pm}\ ;\ \mathsf{AIXIF}[\mathsf{ro}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{p}^{\pm}\right) \leq \Delta_{\mathsf{D}}\left(\mathsf{KD}[\mathsf{f}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{f}^{\pm}\ ;\ \mathsf{AIXIF}[\mathsf{ro}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{f}^{\pm}\right) + \frac{\binom{M+N}{2} + \binom{N}{2}}{2^b}. \quad (4)$$

We proceed with the remaining distance of (4).

The distinguisher makes M construction calls, each of which is either an initialization call $(\delta_i, IV_i) \mapsto (\varnothing, \ell_i)$ or a duplexing call $(\mathit{flag}_i, P_i) \mapsto (Z_i, \ell_i)$, where ℓ_i is the λ bits of leakages obtained in this i-th construction call. In addition, associated to each call is a path path_i as described in Sect. 5.1. Noting that for an initialization call, δ_i and IV_i are implicit in $\mathit{path}_i = \mathsf{encode}[\delta_i] \parallel IV_i$, we can unify the description as follows. For any initialization call, we define $(\mathit{flag}_i, P_i, Z_i) := (0, 0^b, 0^r)$; all M construction calls – either initialization or duplex – can be summarized in a transcript

$$\mathcal{Q}_c := ((\mathit{path}_i, \mathit{flag}_i, P_i, Z_i, \ell_i))_{i=1}^{M}\,. \quad (5)$$

For each construction call, we define a triplet of states (S_i, T_i, U_i). The state S_i is the previous or incoming state. For initialization queries it is defined as $\mathsf{rot}_\alpha(\boldsymbol{K}[\delta_i] \parallel IV_i)$. The state U_i is the next or outgoing state. These are properly defined for both the real and ideal world. The state T_i is an intermediate state, which is defined as $T_i := S_i \oplus [\mathit{flag}_i] \cdot (Z_i \| 0^{b-r}) \oplus P_i$. Note that the intermediate state is only meaningful for the real world, but the value we add to it is known to the adversary. Without loss of generality, each leakage satisfies $\ell_i = \mathsf{L}(T_i, U_i)$.

Furthermore, the distinguisher makes N primitive calls that are summarized in a transcript

$$\mathcal{Q}_p := ((X_j, Y_j))_{j=1}^{N}\,. \quad (6)$$

We define the following two collisions events, one that captures collisions between two construction calls and one that captures collisions between a construction call and a primitive call:

$$\mathsf{col}_{\mathsf{cc}}\ :\ \exists\ i, i' \text{ such that } \mathit{path}_i \neq \mathit{path}_{i'} \wedge T_i = T_{i'}\,, \quad (7)$$

$$\mathsf{col}_{\mathsf{cp}}\ :\ \exists\ i, j \text{ such that } T_i = X_j \vee U_i = Y_j. \quad (8)$$

We write $\mathsf{col} = \mathsf{col}_{\mathsf{cc}} \vee \mathsf{col}_{\mathsf{cp}}$. The bad events are comparable with those of Daemen et al. [15], but they are not the same. One notable difference: Daemen et al. consider (in our terminology) $\mathsf{col}_{\mathsf{cc}}$ for *both input and output collisions*. We do not need to do so, thanks to the RP-RF switch made before.

In Lemma 1 below, we will prove that $(\mathsf{KD}[\mathsf{f}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{f}^{\pm})$ and $(\mathsf{AIXIF}[\mathsf{ro}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{f}^{\pm})$ are identical until col is triggered in the real world. Lemma 2 subsequently derives an upper bound on the event that col is triggered in the real world. These two results, together with (4) above, complete the proof of Theorem 1. Note that from the result of Lemma 1, we can particularly conclude that the final states of the keyed duplex, i.e., all states before re-initializations, have min-entropy $b - \lambda$.

Lemma 1. *As long as* $\mathsf{D}^{\mathsf{KD}[\mathsf{f}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{f}^{\pm}}$ *does not set* col*, the worlds* $(\mathsf{KD}[\mathsf{f}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{f}^{\pm})$ *and* $(\mathsf{AIXIF}[\mathsf{ro}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{f}^{\pm})$ *are identical, or formally,*

$$\Delta_{\mathsf{D}}\left(\mathsf{KD}[\mathsf{f}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{f}^{\pm}\ ;\ \mathsf{AIXIF}[\mathsf{ro}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{f}^{\pm}\right) \leq \mathbf{Pr}\left(\mathsf{D}^{\mathsf{KD}[\mathsf{f}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{f}^{\pm}}\ \textit{sets}\ \mathsf{col}\right). \tag{9}$$

Proof. By the fundamental lemma of game playing [4], it suffices to prove that, as long as the real world $(\mathsf{KD}[\mathsf{f}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{f}^{\pm})$ does not set col, the real and ideal world are indistinguishable.

Clearly, in the ideal world $(\mathsf{AIXIF}[\mathsf{ro}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{f}^{\pm})$, the construction oracle is independent of the primitive oracle $\mathsf{f}^{\pm}$. Also in the real world, the construction oracle $\mathsf{KD}[\mathsf{f}]^{\mathsf{L}}_{\boldsymbol{K}}$ is independent of $\mathsf{f}^{\pm}$, by exclusion of duplex-primitive collisions $\mathsf{col}_{\mathsf{cp}}$ and as each new query to $\mathsf{f}^{\pm}$ is replied with a uniformly generated value. Therefore, we can drop the primitive oracle, and focus on proving that $\mathsf{KD}[\mathsf{f}]^{\mathsf{L}}_{\boldsymbol{K}}$ is indistinguishable from $\mathsf{AIXIF}[\mathsf{ro}]^{\mathsf{L}}_{\boldsymbol{K}}$ under the assumption that $\neg\mathsf{col}_{\mathsf{cc}}$ holds.

We will not only consider the output values (Z_i, ℓ_i), but we will rather prove a stronger result, namely that output states are identically distributed in both worlds. Note that in the real world, the output state is computed as $U_i \leftarrow \mathsf{f}(T_i)$, whereas in the ideal world, it is computed as $U_i \leftarrow \mathsf{ro}(\mathit{path}_i, b)$. Consider the i-th construction call. Clearly, $\mathit{path}_i \neq \mathit{path}_{i'}$, as otherwise the query would be a repeated call. By $\neg\mathsf{col}_{\mathsf{cc}}$, also $T_i \neq T_{i'}$ for all $i' < i$. This means that in both worlds, U_i is a uniformly randomly generated value from $\{0,1\}^b$. □

Lemma 2. *The probability that* $\mathsf{D}^{\mathsf{KD}[\mathsf{f}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{f}^{\pm}}$ *sets* col *satisfies:*

$$\begin{aligned}
&\mathbf{Pr}\left(\mathsf{D}^{\mathsf{KD}[\mathsf{f}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{f}^{\pm}}\ \textit{sets}\ \mathsf{col}\right) \\
&\quad\leq \frac{\nu_{\mathsf{fix}} N}{2^{c-(R+1)\lambda}} + \frac{2\nu^M_{r,c} N}{2^{c-(R+1)\lambda}} + \frac{2\nu^M_{r,c}}{2^c} + \frac{\nu^M_{r,c}(L+\Omega) + \frac{\nu_{\mathsf{fix}}-1}{2}(L+\Omega)}{2^{c-R\lambda}} \\
&\quad+ \frac{\binom{M-L-q}{2} + (M-L-q)(L+\Omega)}{2^{b-\lambda}} \\
&\quad+ \frac{q(M-q)}{2^{H_\infty(\mathcal{D}_{\boldsymbol{K}})+\min\{c,\max\{b-\alpha,c\}-k\}-(R+q_\delta)\lambda}} + \frac{q_{IV} N}{2^{H_\infty(\mathcal{D}_{\boldsymbol{K}})-q_\delta\lambda}} + \frac{\binom{u}{2}}{2^{H_\infty(\mathcal{D}_{\boldsymbol{K}})}}.
\end{aligned}$$

Proof. Consider any distinguisher D that has query access to $(\mathsf{KD}[\mathsf{f}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{f}^{\pm})$, and is bound to the parameters $(M, N, q, q_{IV}, q_\delta, \Omega, L, R, \nu_{\mathsf{fix}})$ listed in Sect. 5.1. Our goal is to bound

$$\mathbf{Pr}\left(\mathsf{col}\right) := \mathbf{Pr}\left(\mathsf{D}^{\mathsf{KD}[\mathsf{f}]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{f}^{\pm}}\ \text{sets}\ \mathsf{col}\right). \tag{10}$$

Additional Notation. One can consider duplexing-calls to occur in a tree fashion, as long as $\mathsf{col}_{\mathsf{cc}}$ never happens. To proper reasoning about the probability that col is set, we will have to define parents, siblings, and children of a duplex call. Consider any construction query $(\mathit{path}_i, \mathit{flag}_i, P_i, Z_i, \ell_i)$.

The parent of this construction query, $\mathrm{parent}(i) \in \{\perp, 1, \ldots, i-1\}$, is defined as follows: if i corresponds to an initialization call, so if $|\mathit{path}_i| = b$, then $\mathrm{parent}(i) = \perp$; otherwise, $\mathrm{parent}(i)$ is the index of the unique duplexing call that satisfies

$$\mathit{path}_i = \mathit{path}_{\mathrm{parent}(i)} \parallel ([\mathit{flag}_{\mathrm{parent}(i)}] \cdot (Z_{\mathrm{parent}(i)} \| 0^{b-r}) \oplus P_{\mathrm{parent}(i)}). \tag{11}$$

If the i-th query is not an initialization call, its siblings $\mathrm{sibling}(i) \subseteq \{1, \ldots, i\}$ are the set of queries *up to the i-th one* (later siblings have yet to be born) with the same parent:

$$\mathrm{sibling}(i) = \left\{ l \in \{1, \ldots, i\} \mid \mathit{path}_{\mathrm{parent}(l)} = \mathit{path}_{\mathrm{parent}(i)} \right\}. \tag{12}$$

Note that we have $|\mathrm{sibling}(i)| \leq R$ for any $i \in \{1, \ldots, M\}$. The children of the i-th query are the set of all queries that have i as parent:

$$\mathrm{child}(i) = \{l \in \{i+1, \ldots, M\} \mid \mathrm{parent}(l) = i\}. \tag{13}$$

We define the type type_i of a construction query $(\mathit{path}_i, \mathit{flag}_i, P_i, Z_i, \ell_i)$:

$$\mathit{type}_i = \begin{cases} \mathit{init}\,, \text{ if } |\mathit{path}_i| = b\,, \\ \mathit{full}\,, \text{ if } |\mathit{path}_i| > b \wedge (|\mathrm{sibling}(i)| = 1 \wedge \mathit{flag}_i = \mathit{false})\,, \\ \mathit{fix}\,, \text{ if } |\mathit{path}_i| > b \wedge (|\mathrm{sibling}(i)| > 1 \vee \mathit{flag}_i = \mathit{true})\,. \end{cases} \tag{14}$$

Note that we have q queries of type *init*. Type *full* corresponds to duplex calls of which the input state S_i is a random value from $\{0,1\}^b$ of which the adversary may have learned the outer r bits, but it had no possibility to *set* the outer part to a certain value of its choice. By definition, there are at most $M - L - q$ queries of type *full*. Finally, type *fix* corresponds to duplex calls of which distinguisher might have set the outer part to a certain value of its choice; this happens if the preceding duplex call had siblings, or if the adversary has turned $\mathit{flag}_i = \mathit{true}$, i.e., enabled the overwrite functionality in the duplex. There are at most $L + \Omega$ queries of type *fix*.

Analyzing Bad Events. We define three additional collision events. The first two correspond to multicollisions among the construction queries exceeding an threshold $\nu := \nu^M_{r,c}$, and the third one corresponds to plain key collisions in the key array $\boldsymbol{K}$:

$$\mathsf{mc}_{\mathsf{in}} : \exists \text{ distinct } i_1, \ldots, i_{\nu+1} \text{ with } \mathit{type}_{i_j} = \mathit{full} \text{ such that } \mathrm{left}_r(T_{i_1}) = \cdots = \mathrm{left}_r(T_{i_{\nu+1}})\,, \tag{15}$$

$$\mathsf{mc}_{\mathsf{out}} : \exists \text{ distinct } i_1, \ldots, i_{\nu+1} \text{ such that } \mathrm{left}_r(U_{i_1}) = \cdots = \mathrm{left}_r(U_{i_{\nu+1}})\,, \tag{16}$$

$$\mathsf{key} : \exists \text{ distinct } \delta, \delta' \text{ such that } \boldsymbol{K}[\delta] = \boldsymbol{K}[\delta']. \tag{17}$$

We define $\mathsf{mc} = \mathsf{mc}_{\mathsf{in}} \vee \mathsf{mc}_{\mathsf{out}}$. By basic probability theory,

$$\mathbf{Pr}\left(\mathsf{col}\right) = \mathbf{Pr}\left(\mathsf{col}_{\mathsf{cc}} \vee \mathsf{col}_{\mathsf{cp}}\right) \leq \mathbf{Pr}\left(\mathsf{col}_{\mathsf{cc}} \vee \mathsf{col}_{\mathsf{cp}} \mid \neg(\mathsf{mc} \vee \mathsf{key})\right) + \mathbf{Pr}\left(\mathsf{mc} \vee \mathsf{key}\right).$$

Note that key is an event independent of the number of queries, whereas $\mathsf{col}_{\mathsf{cc}}$, $\mathsf{col}_{\mathsf{cp}}$, and mc are. The distinguisher can make $M+N$ queries, which it makes in a certain order. For $l \in \{1, \ldots, M+N\}$, denote by $\mathsf{col}_{\mathsf{cc}}(l)$, $\mathsf{col}_{\mathsf{cp}}(l)$, and $\mathsf{mc}(l)$ the event that the l-th query sets the respective event. For brevity of notation, write $\mathsf{col}(l) = \mathsf{col}_{\mathsf{cc}}(l) \vee \mathsf{col}_{\mathsf{cp}}(l)$. By basic probability theory,

$$\mathbf{Pr}\left(\mathsf{col}\right) \leq \sum_{l=1}^{M+N} \mathbf{Pr}\left(\mathsf{col}_{\mathsf{cc}}(l) \mid \neg\mathsf{col}(1\ldots l-1) \wedge \neg\mathsf{mc}(1\ldots l) \wedge \neg\mathsf{key}\right) \tag{18a}$$

$$+ \sum_{l=1}^{M+N} \mathbf{Pr}\left(\mathsf{col}_{\mathsf{cp}}(l) \mid \neg\mathsf{col}(1\ldots l-1) \wedge \neg\mathsf{mc}(1\ldots l) \wedge \neg\mathsf{key}\right) \tag{18b}$$

$$+ \mathbf{Pr}\left(\mathsf{mc}\right) \tag{18c}$$

$$+ \mathbf{Pr}\left(\mathsf{key}\right). \tag{18d}$$

Based on this, we will proceed as follows. We will consider any query made by the distinguisher and consider the probability that *this query* sets either of the events $\mathsf{col}_{\mathsf{cc}}$, $\mathsf{col}_{\mathsf{cp}}$, and mc under the assumption that no earlier query set the event. Note that $\mathsf{col}_{\mathsf{cc}}$ and mc may only be set by a construction query; $\mathsf{col}_{\mathsf{cp}}$ may be set by a construction or a primitive query.

Probability of $\mathsf{col}_{\mathsf{cc}}$ of Eq. (18a). The event can only be set in duplex queries. Consider any two $i \neq i'$, and assume that at the point that the latest of the two queries is made, the events col, mc, and key are still false. We will make a distinction depending on the type of queries of i and i'.

- $type_i = type_{i'} = init$. Note that $T_i = \mathsf{rot}_\alpha(\boldsymbol{K}[\delta_i] \parallel IV_i)$, where δ_i and IV_i can be deduced from $path_i$, and $T_{i'} = \mathsf{rot}_\alpha(\boldsymbol{K}[\delta_{i'}] \parallel IV_{i'})$, where $\delta_{i'}$ and $IV_{i'}$ can be deduced from $path_{i'}$. As $path_i \neq path'_i$, a collision $T_i = T_{i'}$ implies that necessarily $\delta_i \neq \delta_{i'}$ and $\boldsymbol{K}[\delta_i] = \boldsymbol{K}[\delta_{i'}]$. This is impossible under the assumption that $\neg\mathsf{key}$ holds;
- $type_i = init$ and $type_{i'} \neq init$. Note that $T_i = \mathsf{rot}_\alpha(\boldsymbol{K}[\delta_i] \parallel IV_i)$, where δ_i and IV_i can be deduced from $path_i$. Also, $T_{i'} = U_{\mathrm{parent}(i')} \oplus [flag_{i'}] \cdot (Z_{i'} \| 0^{b-r}) \oplus P_{i'}$.
  - $i < i'$. The conditional min-entropy of bits $\alpha \ldots \alpha + k$ of T_i is at least $H_\infty(\mathcal{D}_{\boldsymbol{K}}) - q_\delta\lambda$ and the conditional min-entropy of $\mathrm{right}_c(T_{i'})$ is at least $c - |\mathrm{sibling}(i')|\lambda$. The value T_i hits $T_{i'}$ with probability at most $1/2^{H_\infty(\mathcal{D}_{\boldsymbol{K}}) + \min\{c, \max\{b-\alpha, c\}-k\} - (|\mathrm{sibling}(i')| + q_\delta)\lambda}$;
  - $i' < i$. The conditional min-entropy of bits $\alpha \ldots \alpha + k$ of T_i is at least $H_\infty(\mathcal{D}_{\boldsymbol{K}}) - (q_\delta - 1)\lambda$ and the conditional min-entropy of $\mathrm{right}_c(T_{i'})$ is at least $c - (|\mathrm{sibling}(i')| + 1)\lambda$. The value T_i hits $T_{i'}$ with probability at most $1/2^{H_\infty(\mathcal{D}_{\boldsymbol{K}}) + \min\{c, \max\{b-\alpha, c\}-k\} - (|\mathrm{sibling}(i')| + q_\delta)\lambda}$.

  Note that $|\mathrm{sibling}(i')| \leq R$. There are at most q queries i with $type_i = init$, and at most $M - q$ with $type_{i'} \neq init$. By the union bound, $\mathsf{col}_{\mathsf{cc}}$ is set in this case with probability at most $q(M-q)/2^{H_\infty(\mathcal{D}_{\boldsymbol{K}}) + \min\{c, \max\{b-\alpha, c\}-k\} - (R+q_\delta)\lambda}$;

– $type_i \neq init$ and $type_{i'} \neq init$. We will argue based on the randomness generated in any query l, which generates a random output state $U_l \xleftarrow{\$} \{0,1\}^b$. The probability bound will follow through a union bound, as any query i with $type_i \neq init$ is the child any such query.
  - Consider any $i \in \text{child}(l)$ with $type_i = full$. So far, the distinguisher learned λ bits of leakage on state S_i in query l. Thus, T_i has conditional min-entropy at least $b - \lambda$. It hits any other $T_{i'}$ with probability at most $1/2^{b-\lambda}$. There are at most $M-L-q$ queries i, i' with $type_i = type_{i'} = full$, and furthermore, there are at most $L + \Omega$ queries i' with $type_{i'} = fix$. By the union bound, omitting duplicate counting:
$$\frac{\binom{M-L-q}{2} + (M-L-q)(L+\Omega)}{2^{b-\lambda}};$$
  - Consider any $i \in \text{child}(l)$ with $type_i = fix$. So far, the distinguisher learned λ bits of leakage on state S_i in query l, and $(|\text{sibling}(i)|-1)\lambda$ bits of leakage on state S_i from its sibling queries. Thus, T_i has conditional min-entropy at least $c - |\text{sibling}(l)|\lambda \geq c - R\lambda$. It hits any other $T_{i'}$ with probability at most $1/2^{c-R\lambda}$.
    There are at most $L+\Omega$ queries i with $type_i = fix$. By $\neg\mathsf{mc}_{\mathsf{in}}$, there are at most ν out of at most $M-L-q$ queries i' with $type_{i'} = full$ whose outer part equals $\text{left}_r(T_l)$. There are at most $\nu_{\mathsf{fix}} - 1$ queries i' with $type_{i'} = fix$ whose outer part equals $\text{left}_r(T_l)$. By the union bound, omitting duplicate counting:
$$\frac{\nu(L+\Omega) + \frac{\nu_{\mathsf{fix}}-1}{2}(L+\Omega)}{2^{c-R\lambda}}.$$

  $\mathsf{col}_{\mathsf{cc}}$ is set in this case with probability the sum of above two bounds.

By the union bound,

$$\begin{aligned}(18a) \;&\leq \frac{q(M-q)}{2^{H_\infty(\mathcal{D}_{\boldsymbol{K}})+\min\{c,\max\{b-\alpha,c\}-k\}-(R+q_\delta)\lambda}} \\ &+\frac{\binom{M-L-q}{2} + (M-L-q)(L+\Omega)}{2^{b-\lambda}} + \frac{\nu(L+\Omega) + \frac{\nu_{\mathsf{fix}}-1}{2}(L+\Omega)}{2^{c-R\lambda}}. \quad (19)\end{aligned}$$

Probability of $\mathsf{col}_{\mathsf{cp}}$ of Eq. (18b). The event can be set in duplex and in primitive queries. Consider any duplex query i or any primitive query j, and assume that at the point of querying, the events col, mc, and key are still false. Note that the bad event consists of two parts, namely input collisions $T_i = X_j$ and output collisions $U_i = Y_j$. For both cases, we will make a distinction depending on the type of query of i.

– Event $T_i = X_j$.
  - $type_i = init$. Note that $T_i = \mathsf{rot}_\alpha(\boldsymbol{K}[\delta_i] \parallel IV_i)$, where δ_i and IV_i can be deduced from $path_i$. For fixed primitive query, regardless of whether it is in forward or inverse direction, there are at most q_{IV} possible duplexing calls

with matching rightmost $b-k$ bits, i.e., for which $IV_i = \mathrm{right}_{b-k}(X_j)$. In addition, the conditional min-entropy of $\boldsymbol{K}[\delta_i]$ is at least $H_\infty(\mathcal{D}_{\boldsymbol{K}}) - q_\delta\lambda$, and a collision $T_i = X_j$ happens with probability at most $1/2^{H_\infty(\mathcal{D}_{\boldsymbol{K}})-q_\delta\lambda}$. Summing over all queries, $\mathsf{col_{cp}}$ is set in this case with probability at most $q_{IV}N/2^{H_\infty(\mathcal{D}_{\boldsymbol{K}})-q_\delta\lambda}$;

- $type_i = full$. As query i is of the type *full*, its preceding duplexing call parent(i) generated $U_{\mathrm{parent}(i)} = S_i$ uniformly at random from $\{0,1\}^b$. However, the distinguisher has learned $\mathrm{left}_r(T_i)$, where $T_i = S_i \oplus [flag_i] \cdot (Z_i \| 0^{b-r}) \oplus P_i$, and it may have learned leakage on the other part. For fixed primitive query, regardless of whether it is in forward or inverse direction, by $\neg\mathsf{mc_{in}}$ there are at most ν possible duplexing calls with matching leftmost r bits, i.e., for which $\mathrm{left}_r(T_i) = \mathrm{left}_r(X_j)$. In addition, the conditional min-entropy of $\mathrm{right}_c(T_i)$ is at least $c-(R+1)\lambda$, and a collision $T_i = X_j$ happens with probability at most $1/2^{c-(R+1)\lambda}$. Summing over all queries, $\mathsf{col_{cp}}$ is set in this case with probability at most $\nu N/2^{c-(R+1)\lambda}$;
- $type_i = fix$. As query i is of the type *fix*, the earliest sibling of its preceding duplex call $\min(\mathrm{sibling}(\mathrm{parent}(i)))$ generated $T_{\min(\mathrm{sibling}(\mathrm{parent}(i)))}$ uniformly at random from $\{0,1\}^b$, but in duplexing call i the distinguisher might have set the outer part to a certain value of its choice, and the distinguisher may have learned leakage on the other part. For fixed primitive query, regardless of whether it is in forward or inverse direction, there are at most ν_{fix} possible duplexing calls with matching leftmost r bits, i.e., for which $\mathrm{left}_r(T_i) = \mathrm{left}_r(X_j)$. In addition, the conditional min-entropy of $\mathrm{right}_c(T_i)$ is at least $c-(R+1)\lambda$, and a collision $T_i = X_j$ happens with probability at most $1/2^{c-(R+1)\lambda}$. Summing over all queries, $\mathsf{col_{cp}}$ is set in this case with probability at most $\nu_{\mathsf{fix}}N/2^{c-(R+1)\lambda}$;

– Event $U_i = Y_j$. The duplex call generates U_i uniformly at random from $\{0,1\}^b$. However, the distinguisher may have learned $\mathrm{left}_r(U_i)$ in any subsequent call in child(i), and it may have learned leakage on the other part. For fixed primitive query, regardless of whether it is in forward or inverse direction, by $\neg\mathsf{mc_{out}}$ there are at most ν possible duplexing calls with matching leftmost r bits, i.e., for which $\mathrm{left}_r(U_i) = \mathrm{left}_r(Y_j)$. In addition, the conditional min-entropy of $\mathrm{right}_c(U_i)$ is at least $c-(R+1)\lambda$, and a collision $U_i = Y_j$ happens with probability at most $1/2^{c-(R+1)\lambda}$. Summing over all queries, $\mathsf{col_{cp}}$ is set in this case with probability at most $\nu N/2^{c-(R+1)\lambda}$;

By the union bound,

$$(18b) \le \frac{q_{IV}N}{2^{H_\infty(\mathcal{D}_{\boldsymbol{K}})-q_\delta\lambda}} + \frac{2\nu N}{2^{c-(R+1)\lambda}} + \frac{\nu_{\mathsf{fix}}N}{2^{c-(R+1)\lambda}}. \tag{20}$$

Probability of mc of Eq. (18c). For $\mathsf{mc_{in}}$, note that the state values T_i are randomly generated using a random function f and $M-L-q$ drawings are made (we only consider queries of the type *full*). For $\mathsf{mc_{out}}$, the state values U_i are randomly generated using a random function f and M drawings are made. The event $\mathsf{mc_{in}}$ is thus identical to a balls-and-bins experiment with $M-L-q$

balls that are uniformly randomly thrown into 2^r bins, and the event is set if there is a bin with more than ν balls. The event $\mathsf{mc}_{\mathsf{out}}$ is the same experiment but with M balls. By definition of $\nu := \nu^M_{r,c}$ (see Definition 1), any of the two happens with probability at most

$$(18c) \leq \frac{2\nu}{2^c}. \tag{21}$$

Probability of key of Eq. (18d). This is a simple birthday bound collision event for u randomly drawn k-bit values, as $\boldsymbol{K} = (K[1], \ldots, K[u]) \xleftarrow{\mathcal{D}_K} (\{0,1\}^k)^u$. As the keys are mutually independent, we obtain:

$$(18d) \leq \frac{\binom{u}{2}}{2^{H_\infty(\mathcal{D}_K)}}. \tag{22}$$

Conclusion. The proof is completed by plugging the individual bounds (19), (20), (21), and (22) into main inequality (18). □

6 Limiting Leakage of Keyed Duplex Construction

As it can be seen in Theorem 5.2, the advantage that an attacker can gain from the leakage rises by an increase of either the maximum number of duplexing calls for a single *path* R, or the maximum number of different initialization calls q_δ for a single key. Taha and Schaumont [38] and the developers of Isap [16] presented ways to limit R and q_δ. Their usage of the keyed duplex, generalized to our description of the keyed duplex, is shown in Fig. 3.

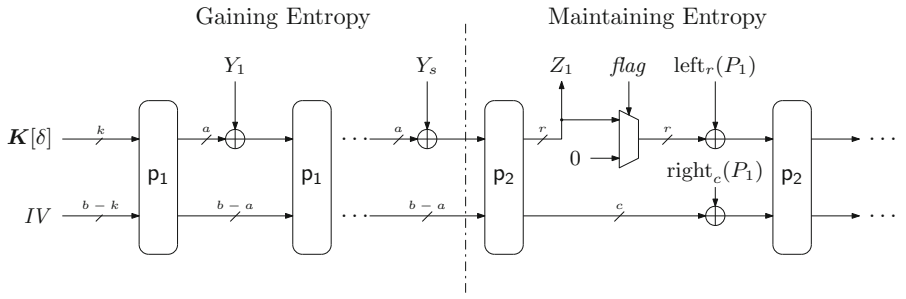

Fig. 3. The duplex as used by Taha and Schaumont [38] and Isap [16].

The limit on q_δ is simply put by limiting the number of different IV's to a small number, typically to one or two different IV's. The role of the IV is then emulated by a value Y, which is typically a nonce in the case of encryption. Y is absorbed directly after the initialization in a-bit portions, where $a \leq r$. Then, duplexing is performed the normal way, starting from the final state obtained after absorption of Y.

As becomes clear from Fig. 3, this approach splits the construction into two different keyed duplex constructions, KD_1 and KD_2, that use two different random permutations (p_1 and p_2) as well as different rate (a and r). The first part KD_1 is responsible for "gaining entropy", where the resulting output states are sufficiently random and mutually independent as long as no two values Y are the same. In the second part KD_2, entropy is "maintained" and used to perform cryptographic operations. In this separation, the last block Y_s is considered to be absorbed in KD_2.

The use of different permutations p_1 and p_2 may seem artificial, and to a certain extent it is: we will rely on mutual independence of the two permutations for easier composability. But also in practical scenarios different permutations for p_1 and p_2 would be used, yet, p_1 would often just be a permutation with a very small number of rounds and it could in a strict sense not be considered to be cryptographically strong.

In what follows, we will apply our general result of Theorem 1 to the construction of Fig. 3. For simplicity of reasoning, we will restrict our focus to the case of $a = 1$, where two different uniformly randomly generated keys are possible (so $u \leq 2$), and where two different IV's are possible (so $|\mathcal{IV}| \leq 2$). This matches the description of Isap [16]. We will consider a distinguisher that makes Q evaluations, each consisting of a unique s-bit Y and an arbitrary amount of duplexing calls in the second part. The distinguisher makes N offline evaluations of p_1 and N offline evaluations of p_2. The remaining parameters of Sect. 5.1 will be bounded by the specific use case of the two duplexes in the construction of Fig. 3.

6.1 Gaining Entropy

The keyed duplex construction KD_1 matches the construction of Sect. 2 with capacity $c = b - 1$, rate $r = 1$, and key offset $\alpha = 0$. The number of initialization calls is at most $q \leq 4$, as there are at most two keys and two IV's. Likewise, $q_{IV}, q_\delta \leq 2$. The number of overwrites satisfies $\Omega = 0$. For the number of repeated paths, note that if a query Y is made, and Y' is an older query with the longest common prefix, then the new query will add one new repeated path, namely the one that ends at the absorption of the bit where Y and Y' differ. In other words, $L \leq Q$, and thus also $\nu_{\mathsf{fix}} \leq Q$. The total number of duplexing calls is at most $M \leq q + Q \cdot s$, noting that each query consists of an initialization and s duplexing calls. We adopt a non-tight $\nu^M_{1,b-1} \leq M$ for simplicity. Finally, as the absorbed bits Y_i can be considered as b-bit blocks P_i where $b - 1$ bits are zero-padded, we obtain that R, the maximum number of duplexing calls for a single non-empty *subpath*, is at most 2.

We obtain the following corollary from Theorem 1, where we have simplified the bound by gathering some fractions with leakage in the denominator. Here, we have also assumed that there is at least 1 bit of leakage, and at least 3 bits of input, and at least 2 queries.

Corollary 1. *Let $b, k, s, \lambda \in \mathbb{N}$, with $k \leq b$, $s \geq 3$, and $1 \leq \lambda \leq 2b$. Let $\mathsf{p}_1 \xleftarrow{\$} \mathrm{perm}(b)$ be a random permutation, and $\boldsymbol{K} \xleftarrow{\$} (\{0,1\}^{k_1})^2$ a random array of keys. Let $\mathcal{L} = \{\mathsf{L} : \{0,1\}^b \times \{0,1\}^b \to \{0,1\}^\lambda\}$ be a class of leakage functions. For any distinguisher D making $Q \geq 2$ queries of length at most s bits, and making N primitive queries,*

$$\mathbf{Adv}^{\mathcal{L}\text{-naLR}}_{\mathsf{KD}_1}(\mathsf{D}) \leq \frac{4sQN + s^2Q^2}{2^{b-4\lambda}} + \frac{\binom{4+sQ+N}{2} + \binom{N}{2}}{2^b} + \frac{2N}{2^{k-2\lambda}} + \frac{1}{2^k}.$$

In addition, except with probability at most the same bound, all output states after absorption of the values Y have min-entropy at least $b - \lambda$.

6.2 Maintaining Entropy

For the keyed duplex construction KD_2, we consider Y_s to be not yet absorbed by KD_1, but instead, it forms the IV for KD_2. More detailed, KD_2 matches the construction of Sect. 2 with arbitrary c, r such that $c + r = b$, with $k = b - 1$, and key offset $\alpha = 1$ meaning that the key is in the bottom $b - 1$ bits. Note that, in fact, Y_s is XORed to the leftmost bit of the state, but for simplicity of reasoning, we simply consider it to *overwrite* it, making the key to KD_2 of size $b - 1$ bits. The number of initialization calls is Q, all of which may potentially be under different keys (so $u \leq Q$ and $q = Q$), one for every $Y \in \{0,1\}^s$ that goes through KD_1. The keys are not uniformly distributed, yet by Corollary 1 they are independent and all have min-entropy $b - 1 - \lambda$. The number of IV's is bounded by 2 (it corresponds to the single bit Y_s), so $q_\delta \leq 2$, but each IV may appear up to Q times, so $q_{IV} \leq q = Q$. The value R, the maximum number of duplexing calls for a single non-empty *subpath*, as it most the maximum number of repetitions of Y, so $R = 1$. There are no repeating paths, hence $L = 0$. As we make no a priori restriction on the choice of the *flag*'s, Ω is yet undetermined and $\nu_{\mathsf{fix}} \leq \Omega$.

We obtain the following corollary from Theorem 1, where we have simplified the bound by gathering some fractions with leakage in the denominator. Here, we have also assumed that there is at least 1 bit of leakage.

Corollary 2. *Let $b, c, r, \lambda \in \mathbb{N}$, with $c + r = b$ and $1 \leq \lambda \leq 2b$. Let $\mathsf{p}_2 \xleftarrow{\$} \mathrm{perm}(b)$ be a random permutation, and $\boldsymbol{K} \xleftarrow{\mathcal{D}_K} (\{0,1\}^b)^Q$ a random array of keys each with min-entropy at least $b - 1 - \lambda$. Let $\mathcal{L} = \{\mathsf{L} : \{0,1\}^b \times \{0,1\}^b \to \{0,1\}^\lambda\}$ be a class of leakage functions. For any distinguisher D making M construction queries, of which Q initialization calls, and N primitive queries,*

$$\mathbf{Adv}^{\mathcal{L}\text{-naLR}}_{\mathsf{KD}_2}(\mathsf{D}) \leq \frac{2\nu^M_{r,c}(N+1)}{2^{c-2\lambda}} + \frac{QN + 2M^2}{2^{b-4\lambda}} + \frac{\binom{M+N}{2} + \binom{N}{2}}{2^b} + \frac{(\nu^M_{r,c} + N + \Omega)\Omega}{2^{c-2\lambda}} + \frac{(M-Q)\Omega}{2^{b-\lambda}}.$$

The bound clearly reveals the impact of overwriting: if the distinguisher may make all its M duplexing calls with *flag* = *true*, the dominating term becomes $MN/2^{c-2\lambda}$.

7 Application to Encryption

We will put the results in practice, and show how Corollaries 1 and 2 guarantee leakage resilient nonce-based stream encryption in a modular manner. Let $b, c, r, k \in \mathbb{N}$ with $c + r = b$ and $k \leq b$. Consider the stream cipher encryption scheme $\mathcal{E}$ of Fig. 4, that gets as input a key K of k bits, a public nonce $\aleph$ of k bits, and an arbitrarily large plaintext P, and it outputs a ciphertext C. The ciphertext C is computed by adding $|P|$ bits of key stream generated by the duplex to P. The IV is a fixed constant.

7.1 Security of Stream Encryption

We consider security of $\mathcal{E}$ in the random permutation model. Let $\mathsf{p}_1, \mathsf{p}_2 \xleftarrow{\$} \mathrm{perm}(b)$ be two random permutations, and $K \xleftarrow{\$} \{0,1\}^k$. Let \$ be a function that for each $(\aleph, P)$ outputs a string of length $|P|$ bits (noting that a nonce should never be repeated). In the black-box security model, one would consider a distinguisher that has access to either $(\mathcal{E}[\mathsf{p}_1, \mathsf{p}_2]_K, \mathsf{p}_1^{\pm}, \mathsf{p}_2^{\pm})$ in the real world or $(\$, \mathsf{p}_1^{\pm}, \mathsf{p}_2^{\pm})$ in the ideal world, where again "$\pm$" stands for bi-directional query access:

$$\mathbf{Adv}_{\mathcal{E}}^{\text{bb-cpa}}(\mathsf{D}) = \Delta_{\mathsf{D}}\left(\mathcal{E}[\mathsf{p}_1, \mathsf{p}_2]_K, \mathsf{p}_1^{\pm}, \mathsf{p}_2^{\pm} \;;\; \$, \mathsf{p}_1^{\pm}, \mathsf{p}_2^{\pm}\right).$$

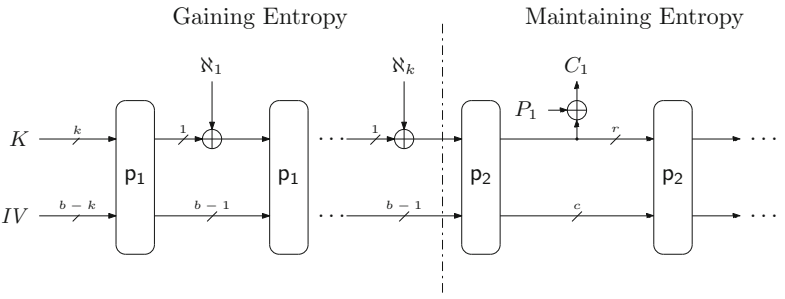

Fig. 4. Leakage-resilient stream encryption using the duplex.

In case of leakage resilience, we will stick to non-adaptive $\mathcal{L}$-resilience of Dodis and Pietrzak [17], as we did in Sect. 3.4. In the current case, however, we cannot simply consider *any* evaluation of the construction to leak, as this would allow for a trivial break of the scheme. Instead, we adopt the conventional approach of, e.g., [17,19,35,37,41], where the distinguisher has access to a leak-free version of the construction, which it has to distinguish from random, and a leaky version, which it may use to gather information. Formally, we obtain the following model, which follows Barwell et al. [2] with the difference that we consider security in the ideal permutation model. Let $\mathsf{p}_1, \mathsf{p}_2, K, \$$ be as above.

Let $\mathcal{L} = \{\mathsf{L} : \{0,1\}^b \times \{true, false\} \times \{0,1\}^b \times \{0,1\}^b \to \{0,1\}^\lambda\}$ be a class of leakage functions, and for any leakage function $\mathsf{L} \in \mathcal{L}$, define by $\mathcal{E}[\mathsf{p}_1, \mathsf{p}_2]^{\mathsf{L}}_K$ encryption such that for each call leaks $\mathsf{L}(S_{\text{prev}}, flag, P, S_{\text{next}})$, where S_{prev} is the state before the call and S_{next} the state after the call. In the leakage resilience security model, one considers a distinguisher that *in addition* to the oracles in the black-box model has access to $\mathcal{E}[\mathsf{p}_1, \mathsf{p}_2]^{\mathsf{L}}_K$:

$$\mathbf{Adv}^{\mathcal{L}\text{-naLR-cpa}}_{\mathcal{E}}(\mathsf{D}) = \max_{\mathsf{L}\in\mathcal{L}} \Delta_{\mathsf{D}}\left(\mathcal{E}[\mathsf{p}_1, \mathsf{p}_2]^{\mathsf{L}}_K, \mathcal{E}[\mathsf{p}_1, \mathsf{p}_2]_K, \mathsf{p}_1^{\pm}, \mathsf{p}_2^{\pm}; \mathcal{E}[\mathsf{p}_1, \mathsf{p}_2]^{\mathsf{L}}_K, \$, \mathsf{p}_1^{\pm}, \mathsf{p}_2^{\pm}\right). \quad (23)$$

The distinguisher is not allowed to make an encryption query (to the leaky or leak-free oracle) under a repeated nonce.

7.2 Security of $\mathcal{E}$

We will demonstrate that the stream cipher encryption is leakage resilient, by relying on Corollaries 1 and 2.

Theorem 2. *Let $b, c, r, k, \lambda \in \mathbb{N}$, with $c + r = b$, $4 \le k \le b$, and $1 \le \lambda \le 2b$. Let $\mathsf{p}_1, \mathsf{p}_2 \xleftarrow{\$} \text{perm}(b)$ be two random permutations, and $K \xleftarrow{\$} \{0,1\}^k$ a random key. Let $\mathcal{L} = \{\mathsf{L} : \{0,1\}^b \times \{0,1\}^b \to \{0,1\}^\lambda\}$ be a class of leakage functions. For any distinguisher making $Q \ge 2$ queries with unique nonces, with a total amount of M plaintext blocks, N primitive queries to p_1 and N primitive queries to p_2,*

$$\begin{aligned}\mathbf{Adv}^{\mathcal{L}\text{-naLR-cpa}}_{\mathcal{E}}(\mathsf{D}) &\le \frac{(16k+2)QN + 4M^2 + 4k^2Q^2}{2^{b-4\lambda}} + \frac{4\binom{4+kQ+N}{2} + 2\binom{M+N}{2} + 6\binom{N}{2}}{2^b} \\ &+ \frac{4\nu^M_{r,c}(N+1)}{2^{c-2\lambda}} + \frac{8N}{2^{k-2\lambda}} + \frac{4}{2^k}.\end{aligned}$$

Proof. Let $\mathsf{KD}_1[\mathsf{p}_1]$ and $\mathsf{KD}_2[\mathsf{p}_2]$ be the two duplexes described in Sects. 6.1 and 6.2, with the difference that $flag = false$ and no data is absorbed for all calls to $\mathsf{KD}_2[\mathsf{p}_2]$. One can equivalently describe $\mathcal{E}[\mathsf{p}_1, \mathsf{p}_2]_K$ based on $\mathsf{KD}_1[\mathsf{p}_1]_K$ and $\mathsf{KD}_2[\mathsf{p}_2]_{\boldsymbol{K}^\star}$ as in Algorithm 4, where $\boldsymbol{K}^\star$ is defined as the output states of $\mathsf{KD}_1[\mathsf{p}_1]_K$ (we use the * to remind of this fact).

Let $\mathsf{L} \in \mathcal{L}$ be any leakage and D be any distinguisher. Our goal is to bound

$$\begin{aligned}&\Delta_{\mathsf{D}}\left(\mathcal{E}[\mathsf{p}_1, \mathsf{p}_2]^{\mathsf{L}}_K, \mathcal{E}[\mathsf{p}_1, \mathsf{p}_2]_K, \mathsf{p}_1^{\pm}, \mathsf{p}_2^{\pm}; \mathcal{E}[\mathsf{p}_1, \mathsf{p}_2]^{\mathsf{L}}_K, \$, \mathsf{p}_1^{\pm}, \mathsf{p}_2^{\pm}\right) \\ &= \Delta_{\mathsf{D}}\left(\mathcal{E}[\mathsf{KD}_1[\mathsf{p}_1]^{\mathsf{L}}_K, \mathsf{KD}_2[\mathsf{p}_2]^{\mathsf{L}}_{\boldsymbol{K}^\star}], \mathcal{E}[\mathsf{KD}_1[\mathsf{p}_1]_K, \mathsf{KD}_2[\mathsf{p}_2]_{\boldsymbol{K}^\star}], \mathsf{p}_1^{\pm}, \mathsf{p}_2^{\pm}; \right. \\ &\qquad \left. \mathcal{E}[\mathsf{KD}_1[\mathsf{p}_1]^{\mathsf{L}}_K, \mathsf{KD}_2[\mathsf{p}_2]^{\mathsf{L}}_{\boldsymbol{K}^\star}], \$, \mathsf{p}_1^{\pm}, \mathsf{p}_2^{\pm}\right). \quad (24)\end{aligned}$$

Let $\mathsf{AIXIF}_1[\mathsf{ro}_1]$ be an AIXIF with the same parameter setting as $\mathsf{KD}_1[\mathsf{p}_1]$, and similarly for $\mathsf{AIXIF}_2[\mathsf{ro}_2]$.

We recall from Corollary 1 that, except with probability at most the bound stated in that corollary, the final output states of $\mathsf{KD}_1[\mathsf{p}_1]$ have min-entropy at

Algorithm 4. Equivalent description of $\mathcal{E}[\mathsf{p}_1, \mathsf{p}_2]$

Interface: $\mathcal{E}[\mathsf{KD}_1[\mathsf{p}_1], \mathsf{KD}_2[\mathsf{p}_2]]$
Input: $(K, \aleph, P) \in \{0,1\}^k \times \{0,1\}^k \times \{0,1\}^*$
Output: $C \in \{0,1\}^{|P|}$

$\mathsf{KD}_1.\mathsf{init}(1, IV)$ ▷ only one key K, only one IV
$\aleph_1 \| \ldots \| \aleph_k \leftarrow \aleph$
for $i = 1, \ldots, k-1$ **do**
  $Z \leftarrow \mathsf{KD}_1.\mathsf{duplex}(\mathit{false}, \aleph_i \| 0^{n-1})$ ▷ discard output
$\boldsymbol{K}^\star[\mathsf{encode}(\aleph_1 \ldots \aleph_{k-1})] \leftarrow \mathrm{right}_{b-1}(S)$ ▷ store state of KD_1 in key array of KD_2
$\mathsf{KD}_2.\mathsf{init}(\mathsf{encode}(\aleph_1 \ldots \aleph_{k-1}), Z \oplus \aleph_k)$ ▷ KD_2 has key offset $\alpha = 1$
$Z \leftarrow \varnothing$
$\ell \leftarrow \lceil |P|/r \rceil$
for $i = 1, \ldots, \ell$ **do**
  $Z \leftarrow Z \parallel \mathsf{KD}_2.\mathsf{duplex}(\mathit{false}, 0^b)$
return $\mathrm{left}_{|P|}(P \oplus Z)$

least $b - \lambda$. This means that we can replace the generation of $\boldsymbol{K}^\star$ in Algorithm 4 by a dummy $\boldsymbol{K} \xleftarrow{\mathcal{D}_K} (\{0,1\}^b)^Q$ consisting of keys with min-entropy $b - 1 - \lambda$ at negligible cost. Formally, denoting the resulting scheme by $\mathcal{E}^\star$, we have obtained:

$$\begin{aligned}
&\Delta_{\mathsf{D}}\left(\mathcal{E}[\mathsf{KD}_1[\mathsf{p}_1]^{\mathsf{L}}_K, \mathsf{KD}_2[\mathsf{p}_2]^{\mathsf{L}}_{\boldsymbol{K}^\star}], \mathcal{E}[\mathsf{KD}_1[\mathsf{p}_1]_K, \mathsf{KD}_2[\mathsf{p}_2]_{\boldsymbol{K}^\star}], \mathsf{p}_1^{\pm}, \mathsf{p}_2^{\pm}; \right.\\
&\qquad\left. \mathcal{E}[\mathsf{KD}_1[\mathsf{p}_1]^{\mathsf{L}}_K, \mathsf{KD}_2[\mathsf{p}_2]^{\mathsf{L}}_{\boldsymbol{K}^\star}], \$, \mathsf{p}_1^{\pm}, \mathsf{p}_2^{\pm}\right)\\
&\le \Delta_{\mathsf{D}}\left(\mathcal{E}^\star[\mathsf{KD}_1[\mathsf{p}_1]^{\mathsf{L}}_K, \mathsf{KD}_2[\mathsf{p}_2]^{\mathsf{L}}_{\boldsymbol{K}}], \mathcal{E}^\star[\mathsf{KD}_1[\mathsf{p}_1]_K, \mathsf{KD}_2[\mathsf{p}_2]_{\boldsymbol{K}}], \mathsf{p}_1^{\pm}, \mathsf{p}_2^{\pm}; \right.\\
&\qquad\left. \mathcal{E}^\star[\mathsf{KD}_1[\mathsf{p}_1]^{\mathsf{L}}_K, \mathsf{KD}_2[\mathsf{p}_2]^{\mathsf{L}}_{\boldsymbol{K}}], \$, \mathsf{p}_1^{\pm}, \mathsf{p}_2^{\pm}\right)\\
&+ 2 \cdot \Delta_{\mathsf{D}'}\left(\mathsf{KD}_1[\mathsf{p}_1]^{\mathsf{L}}_K, \mathsf{p}_1^{\pm}; \mathsf{AIXIF}_1[\mathsf{ro}_1]^{\mathsf{L}}_K, \mathsf{p}_1^{\pm}\right), \qquad (25)
\end{aligned}$$

where D' is some distinguisher making Q queries of length $k-1$ bits, and making N primitive queries. The factor 2 comes from the fact that we perform the change in both the real and ideal world.

For the remaining distance of (25), we can perform a hybrid argument:

$$\begin{aligned}
&\Delta_{\mathsf{D}}\left(\mathcal{E}^\star[\mathsf{KD}_1[\mathsf{p}_1]^{\mathsf{L}}_K, \mathsf{KD}_2[\mathsf{p}_2]^{\mathsf{L}}_{\boldsymbol{K}}], \mathcal{E}^\star[\mathsf{KD}_1[\mathsf{p}_1]_K, \mathsf{KD}_2[\mathsf{p}_2]_{\boldsymbol{K}}], \mathsf{p}_1^{\pm}, \mathsf{p}_2^{\pm}; \right.\\
&\qquad\left. \mathcal{E}^\star[\mathsf{KD}_1[\mathsf{p}_1]^{\mathsf{L}}_K, \mathsf{KD}_2[\mathsf{p}_2]^{\mathsf{L}}_{\boldsymbol{K}}], \$, \mathsf{p}_1^{\pm}, \mathsf{p}_2^{\pm}\right)\\
&\le \Delta_{\mathsf{D}}\left(\mathcal{E}^\star[\mathsf{AIXIF}_1[\mathsf{ro}_1]^{\mathsf{L}}_K, \mathsf{KD}_2[\mathsf{p}_2]^{\mathsf{L}}_{\boldsymbol{K}}], \mathcal{E}^\star[\mathsf{AIXIF}_1[\mathsf{ro}_1]_K, \mathsf{KD}_2[\mathsf{p}_2]_{\boldsymbol{K}}], \mathsf{p}_2^{\pm}; \right.\\
&\qquad\left. \mathcal{E}^\star[\mathsf{AIXIF}_1[\mathsf{ro}_1]^{\mathsf{L}}_K, \mathsf{KD}_2[\mathsf{p}_2]^{\mathsf{L}}_{\boldsymbol{K}}], \$, \mathsf{p}_2^{\pm}\right)\\
&+ 2 \cdot \Delta_{\mathsf{D}'}\left(\mathsf{KD}_1[\mathsf{p}_1]^{\mathsf{L}}_K, \mathsf{p}_1^{\pm}; \mathsf{AIXIF}_1[\mathsf{ro}_1]^{\mathsf{L}}_K, \mathsf{p}_1^{\pm}\right), \qquad (26)
\end{aligned}$$

where D' is some distinguisher making Q queries of length $k-1$ bits, and making N primitive queries. The distinguisher D' operates as follows: it generates a dummy key $\boldsymbol{K} \xleftarrow{\mathcal{D}_K} (\{0,1\}^b)^Q$ and dummy permutation $\mathsf{p}_2 \xleftarrow{\$} \mathrm{perm}(b)$ on its own; for each query $(\aleph, P)$ that D makes, D' pads $\aleph_1 \| \ldots \| \aleph_k \leftarrow \aleph$ and evaluates its own oracle for $\aleph_1 \| \ldots \| \aleph_{k-1}$; it uses the output value Z, the last nonce bit $\aleph_k$, and its freshly generated $\boldsymbol{K}$ and p_2 to simulate the encryption of P, and it

outputs the result. If D' is communicating with the real world $\mathsf{KD}_1[\mathsf{p}_1]_K$, this perfectly simulates $\mathcal{E}^\star[\mathsf{KD}_1[\mathsf{p}_1]_K, \mathsf{KD}_2[\mathsf{p}_2]_{\boldsymbol{K}}]$, and if it is communicating with the ideal world $\mathsf{AIXIF}_1[\mathsf{ro}_1]_K$, this perfectly simulates $\mathcal{E}^\star[\mathsf{AIXIF}_1[\mathsf{ro}_1]_K, \mathsf{KD}_2[\mathsf{p}_2]_{\boldsymbol{K}}]$.

The isolated distances on KD_1 in both (25) and (26) can be bounded directly by Corollary 1:

$$\Delta_{\mathsf{D}'}\left(\mathsf{KD}_1[\mathsf{p}_1]^{\mathsf{L}}_K, \mathsf{p}_1^{\pm}\ ;\ \mathsf{AIXIF}_1[\mathsf{ro}_1]^{\mathsf{L}}_K, \mathsf{p}_1^{\pm}\right) \le \frac{4kQN + k^2Q^2}{2^{b-4\lambda}} + \frac{\binom{4+kQ+N}{2} + \binom{N}{2}}{2^b} + \frac{2N}{2^{k-2\lambda}} + \frac{1}{2^k}. \quad (27)$$

We can proceed from the remaining distance in (26):

$$\begin{aligned}
&\Delta_{\mathsf{D}}\left(\mathcal{E}^\star[\mathsf{AIXIF}_1[\mathsf{ro}_1]^{\mathsf{L}}_K, \mathsf{KD}_2[\mathsf{p}_2]^{\mathsf{L}}_{\boldsymbol{K}}]\,,\ \mathcal{E}^\star[\mathsf{AIXIF}_1[\mathsf{ro}_1]_K, \mathsf{KD}_2[\mathsf{p}_2]_{\boldsymbol{K}}]\,,\ \mathsf{p}_2^{\pm}\ ;\right.\\
&\qquad\left.\mathcal{E}^\star[\mathsf{AIXIF}_1[\mathsf{ro}_1]^{\mathsf{L}}_K, \mathsf{KD}_2[\mathsf{p}_2]^{\mathsf{L}}_{\boldsymbol{K}}]\,,\ \$\,,\ \mathsf{p}_2^{\pm}\right)\\
&\le \Delta_{\mathsf{D}}\left(\mathcal{E}^\star[\mathsf{AIXIF}_1[\mathsf{ro}_1]^{\mathsf{L}}_K, \mathsf{AIXIF}_2[\mathsf{ro}_2]^{\mathsf{L}}_{\boldsymbol{K}}]\,,\ \mathcal{E}^\star[\mathsf{AIXIF}_1[\mathsf{ro}_1]_K, \mathsf{AIXIF}_2[\mathsf{ro}_2]_{\boldsymbol{K}}]\ ;\right.\\
&\qquad\left.\mathcal{E}^\star[\mathsf{AIXIF}_1[\mathsf{ro}_1]^{\mathsf{L}}_K, \mathsf{AIXIF}_2[\mathsf{ro}_2]^{\mathsf{L}}_{\boldsymbol{K}}]\,,\ \$\right)\\
&+ 2\cdot\Delta_{\mathsf{D}''}\left(\mathsf{KD}_2[\mathsf{p}_2]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{p}_2^{\pm}\ ;\ \mathsf{AIXIF}_2[\mathsf{ro}_2]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{p}_2^{\pm}\right)\,, \qquad (28)
\end{aligned}$$

where D'' is some distinguisher making M construction queries, of which Q initialization calls, and N primitive queries. Distinguisher D'' works symmetrically to distinguisher D' above, and its description is omitted.

The second distance of (28) can be bounded directly by Corollary 2 for $\Omega = 0$:

$$\Delta_{\mathsf{D}''}\left(\mathsf{KD}_2[\mathsf{p}_2]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{p}_2^{\pm}\ ;\ \mathsf{AIXIF}_2[\mathsf{ro}_2]^{\mathsf{L}}_{\boldsymbol{K}}, \mathsf{p}_2^{\pm}\right) \le \frac{2\nu^M_{r,c}(N+1)}{2^{c-2\lambda}} + \frac{QN + 2M^2}{2^{b-4\lambda}} + \frac{\binom{M+N}{2} + \binom{N}{2}}{2^b}. \quad (29)$$

It remains to consider the first distance of (28). As the adversary may never query its oracle (leaky nor leak-free) for the same nonce, the leaky and leak-free oracles are mutually independent, and we obtain:

$$\begin{aligned}
&\Delta_{\mathsf{D}}\left(\mathcal{E}^\star[\mathsf{AIXIF}_1[\mathsf{ro}_1]^{\mathsf{L}}_K, \mathsf{AIXIF}_2[\mathsf{ro}_2]^{\mathsf{L}}_{\boldsymbol{K}}]\,,\ \mathcal{E}^\star[\mathsf{AIXIF}_1[\mathsf{ro}_1]_K, \mathsf{AIXIF}_2[\mathsf{ro}_2]_{\boldsymbol{K}}]\ ;\right.\\
&\qquad\left.\mathcal{E}^\star[\mathsf{AIXIF}_1[\mathsf{ro}_1]^{\mathsf{L}}_K, \mathsf{AIXIF}_2[\mathsf{ro}_2]^{\mathsf{L}}_{\boldsymbol{K}}]\,,\ \$\right)\\
&= \Delta_{\mathsf{D}}\left(\mathcal{E}^\star[\mathsf{AIXIF}_1[\mathsf{ro}_1]_K, \mathsf{AIXIF}_2[\mathsf{ro}_2]_{\boldsymbol{K}}]\ ;\ \$\right) = 0. \qquad (30)
\end{aligned}$$

The proof is completed by combining (24)–(30). □

7.3 Towards Authentication

The stream cipher encryption construction considered in this section can be extended to cover authentication as well. One way of doing so is by absorbing the plaintext blocks P_i during streaming and outputting a tag at the end; another approach is by evaluating a MAC function (with a different key and IV, noting that Corollary 1 supports two keys and two IV's) after encryption has taken

place. Note that in the first case, authenticated decryption would require to turn *flag* = *true* (see Sect. 2). In either case, one must take care of the fact that, upon decryption, nonces may get reused. In terms of the general picture of Fig. 3, this means that a same nonce can be "tried" for different blocks P_i, leading to repeating paths (hence $L > 0$) and to a higher leakage per evaluation of p_2 (hence $R > 1$). An authenticated encryption scheme that prohibits such "trial" of the same nonce with different inputs is Isap [16].

Acknowledgments. We thank the ISAP team, the ESCADA team, and the authors of [23] for fruitful discussions. Christoph Dobraunig is supported by the Austrian Science Fund (FWF): J 4277-N38. 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., Daemen, J., Mennink, B., Van Assche, G.: Security of keyed sponge constructions using a modular proof approach. In: Leander, G. (ed.) FSE 2015. LNCS, vol. 9054, pp. 364–384. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48116-5_18
2. Barwell, G., Martin, D.P., Oswald, E., Stam, M.: Authenticated encryption in the face of protocol and side channel leakage. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017. LNCS, vol. 10624, pp. 693–723. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70694-8_24
3. 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.) CCS 1993, pp. 62–73. ACM (1993)
4. Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 409–426. Springer, Heidelberg (2006). https://doi.org/10.1007/11761679_25
5. Berti, F., Koeune, F., Pereira, O., Peters, T., Standaert, F.X.: Leakage-Resilient and Misuse-Resistant Authenticated Encryption. Cryptology ePrint Archive, Report 2016/996 (2016)
6. Berti, F., Pereira, O., Peters, T., Standaert, F.X.: On leakage-resilient authenticated encryption with decryption leakages. IACR Trans. Symmetric Cryptol. **2017**(3), 271–293 (2017)
7. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Sponge functions. In: Ecrypt Hash Workshop 2007, May 2007
8. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Duplexing the sponge: single-pass authenticated encryption and other applications. In: Miri, A., Vaudenay, S. (eds.) SAC 2011. LNCS, vol. 7118, pp. 320–337. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28496-0_19
9. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: The Keccak reference, January 2011
10. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: On the security of the keyed sponge construction. In: Symmetric Key Encryption Workshop, February 2011

11. Bloem, R., Gross, H., Iusupov, R., Könighofer, B., Mangard, S., Winter, J.: Formal verification of masked hardware implementations in the presence of glitches. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018. LNCS, vol. 10821, pp. 321–353. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78375-8_11
12. Chang, D., Dworkin, M., Hong, S., Kelsey, J., Nandi, M.: A keyed sponge construction with pseudorandomness in the standard model. In: NIST SHA-3 Workshop, March 2012
13. Chari, S., Jutla, C.S., Rao, J.R., Rohatgi, P.: Towards sound approaches to counteract power-analysis attacks. In: Wiener [39], pp. 398–412
14. Clavier, C., Coron, J.-S., Dabbous, N.: Differential power analysis in the presence of hardware countermeasures. In: Koç, Ç.K., Paar, C. (eds.) CHES 2000. LNCS, vol. 1965, pp. 252–263. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44499-8_20
15. Daemen, J., Mennink, B., Van Assche, G.: Full-state keyed duplex with built-in multi-user support. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017. LNCS, vol. 10625, pp. 606–637. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70697-9_21
16. Dobraunig, C., Eichlseder, M., Mangard, S., Mendel, F., Unterluggauer, T.: ISAP - towards side-channel secure authenticated encryption. IACR Trans. Symmetric Cryptol. **2017**(1), 80–105 (2017)
17. Dodis, Y., Pietrzak, K.: Leakage-resilient pseudorandom functions and side-channel attacks on feistel networks. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 21–40. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14623-7_2
18. Dziembowski, S., Pietrzak, K.: Leakage-resilient cryptography. In: FOCS 2008, pp. 293–302. IEEE Computer Society (2008)
19. Faust, S., Pietrzak, K., Schipper, J.: Practical leakage-resilient symmetric cryptography. In: Prouff, E., Schaumont, P. (eds.) CHES 2012. LNCS, vol. 7428, pp. 213–232. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33027-8_13
20. FIPS 202: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, August 2015
21. Gaži, P., Pietrzak, K., Tessaro, S.: The exact PRF security of truncation: tight bounds for keyed sponges and truncated CBC. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9215, pp. 368–387. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-47989-6_18
22. Goubin, L., Patarin, J.: DES and differential power analysis the "duplication" method. In: Koç, Ç.K., Paar, C. (eds.) CHES 1999. LNCS, vol. 1717, pp. 158–172. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48059-5_15
23. Guo, C., Pereira, O., Peters, T., Standaert, F.X.: Towards Lightweight Side-Channel Security and the Leakage-Resilience of the Duplex Sponge. Cryptology ePrint Archive, Report 2019/193 (2019)
24. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM J. Comput. **28**(4), 1364–1396 (1999)
25. Hsiao, C.-Y., Lu, C.-J., Reyzin, L.: Conditional computational entropy, or toward separating pseudoentropy from compressibility. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 169–186. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72540-4_10
26. Jovanovic, P., Luykx, A., Mennink, B.: Beyond $2^{c/2}$ security in sponge-based authenticated encryption modes. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT

2014. LNCS, vol. 8873, pp. 85–104. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45611-8_5
27. Kocher, P.C.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 104–113. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-68697-5_9
28. Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener [39], pp. 388–397
29. Medwed, M., Standaert, F.-X., Großschädl, J., Regazzoni, F.: Fresh re-keying: security against side-channel and fault attacks for low-cost devices. In: Bernstein, D.J., Lange, T. (eds.) AFRICACRYPT 2010. LNCS, vol. 6055, pp. 279–296. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12678-9_17
30. 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. LNCS, vol. 9453, pp. 465–489. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48800-3_19
31. 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, Heidelberg (2016). https://doi.org/10.1007/978-3-662-52993-5_1
32. Nikova, S., Rechberger, C., Rijmen, V.: Threshold implementations against side-channel attacks and glitches. In: Ning, P., Qing, S., Li, N. (eds.) ICICS 2006. LNCS, vol. 4307, pp. 529–545. Springer, Heidelberg (2006). https://doi.org/10.1007/11935308_38
33. Nikova, S., Rijmen, V., Schläffer, M.: Secure hardware implementation of nonlinear functions in the presence of glitches. J. Cryptol. **24**(2), 292–321 (2011)
34. Pereira, O., Standaert, F.X., Vivek, S.: Leakage-resilient authentication and encryption from symmetric cryptographic primitives. In: Ray, I., Li, N., Kruegel, C. (eds.) CCS 2015, pp. 96–108. ACM (2015)
35. Pietrzak, K.: A leakage-resilient mode of operation. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 462–482. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01001-9_27
36. Standaert, F.-X., Pereira, O., Yu, Y.: Leakage-resilient symmetric cryptography under empirically verifiable assumptions. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8042, pp. 335–352. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40041-4_19
37. Standaert, F.X., Pereira, O., Yu, Y., Quisquater, J.J., Yung, M., Oswald, E.: Leakage resilient cryptography in practice. In: Sadeghi, A.R., Naccache, D. (eds.) Towards Hardware-Intrinsic Security - Foundations and Practice. ISC, pp. 99–134. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14452-3_5
38. Taha, M.M.I., Schaumont, P.: Side-channel countermeasure for SHA-3 at almost-zero area overhead. In: HOST 2014, pp. 93–96. IEEE Computer Society (2014)
39. Wiener, M.J. (ed.): CRYPTO 1999. LNCS, vol. 1666. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48405-1
40. Yu, Y., Standaert, F.-X.: Practical leakage-resilient pseudorandom objects with minimum public randomness. In: Dawson, E. (ed.) CT-RSA 2013. LNCS, vol. 7779, pp. 223–238. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36095-4_15
41. Yu, Y., Standaert, F.X., Pereira, O., Yung, M.: Practical leakage-resilient pseudorandom generators. In: Al-Shaer, E., Keromytis, A.D., Shmatikov, V. (eds.) CCS 2010, pp. 141–151. ACM (2010)