Key Prediction Security of Keyed Sponges

Bart Mennink

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

Abstract. The keyed sponge is a well-accepted method for message authentication. It processes data at a certain rate by sequential evaluation of an underlying permutation. If the key size k is smaller than the rate, currently known bounds are tight, but if it exceeds the rate, state of the art only dictates security up to $2^{k/2}$. We take closer inspection at the key prediction security of the sponge and close the remaining gap in the existing security analysis: we confirm key security up to close to 2^k, regardless of the rate. The result impacts all applications of the keyed sponge and duplex that process at a rate smaller than the key size, including the STROBE protocol framework, as well as the related constructions such as HMAC-SHA-3 and the sandwich sponge.

Keywords: outer-keyed sponge, full-keyed sponge, key prediction, graph-based proof

1 Introduction

Keyed cryptographic functions are desired to "behave like" a random function with the same interface, in such a way that an adversary cannot easily distinguish one from another. Almost all keyed cryptographic schemes have been analyzed in this so-called indistinguishability model, at least from a generic perspective where the underlying primitives are assumed to be sufficiently secure. The indistinguishability model is rather strong: an attack grants an adversary knowledge on some evaluations of the scheme, but not all. It does not produce a full break of the scheme, but rather indicates a non-random property.

Key recoveries are a stronger type of attack. A key recovery can be used to distinguish a scheme from random, but not the other way around. In some cases, the best distinguishability attack is (close to) a key recovery attack. For example, AES [20] supports keys of size 128, 192, or 256, and the best distinguishing attack on AES known to date is (close to) the generic key recovery attack that succeeds if the attacker makes a total amount of 2^k offline evaluations of AES, where k is the key size. In a bit more detail, it is generally believed that the strong pseudorandom permutation (SPRP) of AES is around $t/2^k$, where t is the number of offline evaluations of AES.

It is not always straightforward to achieve security against key recovery attacks up to 2^k. An earlier example of this is the authenticated encryption scheme McOE-X [22], for which Mendel et al. [37] described a simple key recovery attack in $2^{n/2}$ evaluations, where n equals both key and state size. Fuhr et al. [23]

described sophisticated birthday-type key recovery attacks on CAESAR candidates Marble [26] and AEZ v3 [30]. The designers of AEZ revised their scheme to AEZ v4 in order to mitigate the attack [31], but subsequently, Chaigneau and Gilbert [16] showed that AEZ v4.1 [32] is still vulnerable to a key recovery attack with similar complexity.

1.1 Keyed Sponges

We will focus on keyed versions of the sponge construction by Bertoni et al. [6]. The (keyless) sponge construction is a hash function mode that operates on a b-bit state, split into an inner part of capacity c and an outer part of rate r, where $c + r = b$. Data is absorbed, and the digest is extracted, block-by-block via the outer part of the state, interleaved with evaluations of a b-bit permutation π. The function is proven [7] to achieve $c/2$-bit security in the indifferentiability framework [36]. It has been adopted, among others, in the SHA-3 hashing standard [21].

The first, and perhaps simplest, approach of transforming the sponge into a pseudorandom function (PRF) is by Bertoni et al. [10], who suggested to simply prepend the message m with the key K, i.e., to evaluate the sponge on input $(K\|m)$. The scheme got renewed analysis by Andreeva et al. [2], who dubbed it the "outer-keyed sponge (OKS)," and later Naito and Yasuda [41].

Alternatively, one can design a PRF by just initializing the inner part with the key K instead of with 0^c. This construction, now known as the "inner keyed sponge (IKS)," was introduced by Chang et al. [17] and received improved analysis by Andreeva et al. [2] and Naito and Yasuda [41].

Finally, it appeared that secrecy of the state after key injection could be used to support *full-state absorption*: instead of absorbing data in the r-bit outer part only, one could absorb over the entire b-bit state. The idea appeared first in the donkeySponge [11]. An analysis for one output block only was given by Gaži et al. [24]. A complete security treatment was given by Mennink et al. [38] and Daemen et al. [18].

It is obvious that the full-keyed sponge is generically[1] more efficient than the outer-keyed and inner-keyed sponge: data is compressed b bits at a time rather than $r < b$ bits. From a security perspective, the schemes achieve approximately the same level of security. In detail, all achieve a security level of around

$$\frac{M^2}{2^c} + \frac{MN}{2^c} + \mathbf{Adv}_F^{\text{key-pre}}(N) \tag{1}$$

(omitting constants and details, see Section 4), where M denotes the number of queries to the construction, N the number of queries to the underlying permutation, and $\mathbf{Adv}_F^{\text{key-pre}}(N)$ is the probability that the adversary made queries to the underlying permutation that match the key input to $F \in \{\text{OKS}, \text{IKS}, \text{FKS}\}$. Intuitively, the first two fractions of (1) correspond to distinguishing F from a random function, and the last term represents a key prediction, an isolated event

[1] To get the same level of security, the underlying permutation may need more rounds.

in typical security analyses (there is a subtle difference between a key prediction and a key recovery, as we will explain in Section 5).

Note that all sponges achieve a security level of c bits at best, and there is no reason to take a key of size $k > c$. As such, FKS has made IKS obsolete: both require adaptation of the keyless sponge algorithm, both take one evaluation of π to compress the key, both are approximately equally secure, but the FKS is more efficient. FKS does not necessarily make OKS obsolete: although it is more efficient, OKS does not require an adaptation of the keyless sponge algorithm, and can evaluate it in a black-box manner. This happens, for example, in the STROBE protocol framework [28, 29], as we will elaborate upon in Section 1.4. We restrict our focus to OKS and FKS.

1.2 Key Prediction Security

For FKS, using that $k < b$ without loss of generality, it is easy to see that

$$\mathbf{Adv}_{\mathrm{FKS}}^{\text{key-pre}}(N) \leq N/2^k \,. \tag{2}$$

Indeed, the key is compressed using one evaluation of π, the adversary can make N attempts, and succeeds if one of those is performed for the corresponding key.

For OKS, the k-bit key is partitioned into r-bit blocks. If $k \leq r$, then the key prediction term is bounded as before (the key is compressed using one call to π). On the other hand, if $k > r$, then the key is processed using more than one evaluation of π, and the current state of the art suggests only $2^{k/2}$ security. In more detail, Gaži et al. [25, Lemma 12] (full version of [24]) derived the following bound for the key prediction term:

$$\mathbf{Adv}_{\mathrm{OKS}}^{\text{key-pre}}(N) \lesssim \begin{cases} N/2^k \,, \text{ if } k \leq r \,, \\ b^{\lambda/2} N/2^{k/2} \,, \text{ if } k > r \,, \end{cases} \tag{3}$$

where $\lambda := \lceil k/r \rceil$ denotes the number of permutation calls to process the key, and where negligible terms are omitted (refer to Proposition 1 for the details).

This bound shows a counter-intuitive and devastating loss in the key size: whereas one key block still yields the intuitively optimal bound of $N/2^k$ ("one needs to make around 2^k attempts to recover the key"), once the key size exceeds the rate one loses half the key! Stated differently, the bound suggests that OKS with a key of size $2r$ achieves the same level of security as OKS with a key of size r.

In Section 3, we derive an improved bound on the key prediction security, and demonstrate that

$$\mathbf{Adv}_{\mathrm{OKS}}^{\text{key-pre}}(N) \lesssim c^{\lambda-1} N/2^k \,, \tag{4}$$

where, again, $\lambda := \lceil k/r \rceil$, and negligible terms are omitted (refer to Theorem 1 for the details). For $k \leq r$ (or $\lambda = 1$), the earlier bound of Gaži et al. was already tight and the new bound matches it. For $k > r$ (or $\lambda > 1$), our bound demonstrates that close to k-bit security is achieved, with a logarithmic degradation in c. This innocent logarithmic loss, in turn, comes from the probability of a lucky multi-collision; it was already present in the bound of Gaži et al., but we slightly improved it. (We remark that the best attack against key prediction has a success probability of around $N/2^k$.)

The result is proven using a graph-based approach, wherein every edge corresponds to an r-bit key block guess and one considers the maximum number of paths of length λ departing from root node 0^b. We then consider *configurations* of paths of length λ, corresponding to the direction in which the λ individual queries to the underlying permutation are made. The proof is inspired by that of Gaži et al. [25, Lemma 12], who also adopted a graph-based approach and introduced the term of configurations, yet it differs in many aspects. Most importantly, Gaži et al. observed that in any configuration, at least half is in the same direction (either forward or inverse). They subsequently use a bad event based on multi-collisions to upper bound the number of possibilities over these $\geq \lambda/2$ edges, but ignore the problem of the adversary to find the connections over the remaining $\leq \lambda/2$ edges in reverse direction. This simplification leads to discarding half of the key (hence the denominator $2^{k/2}$ in (3)). In our bound, we perform an inductive argument on λ. We prove that for every added layer extension, the size of the *yield*, or equivalently the number of paths from 0^b, depends only on the presence of multi-collisions and is independent of the configuration of the query.

1.3 Further Appearances of Key Prediction Security

The main functionality of the keyed sponge is authentication. It can be used for (authenticated) encryption via the duplex construction by Bertoni et al. [8]. The duplex is a stateful construction: a call to its initialization interface initializes a state, and a call to its duplexing interface absorbs a data block, applies the permutation π on the state, and squeezes at most r bits. Similar to the keyed sponges, the absorption of the data block can be performed over the outer part only (r bits) or over the entire state (b bits). We will refer to these two as the outer-keyed duplex (OKD) and the full-keyed duplex (FKD). It is important to note that these protocols do not extract data during absorption of the key; in other words, the key is absorbed as if the function is an outer-keyed, resp. full-keyed, sponge.

The security of keyed duplexes can be proven from keyed sponges and vice versa [18, 38]. The reduction makes use of the fact that one evaluation of the keyed duplex consists of ℓ evaluations of a keyed sponge, where ℓ denotes the number of blocks that are duplexed in-between two initialization calls. Even stronger, both the outer-keyed duplex and full-keyed duplex are proven to also achieve the bound of (1), up to constant, where the key prediction security is still for the keyed *sponge* function $F \in \{\mathrm{OKS}, \mathrm{FKS}\}$.

For both variants of the keyed sponges and keyed duplexes one can also consider security in the nonce-respecting setting, where a nonce is compressed into the state prior to the first data block. In this case, the general bound of (1)

Table 1: Suggested parameters in the STROBE protocol framework [28,29].

scheme	b	c	r	k
STROBE-128/1600	1600	256	1344	256
STROBE-256/1600	1600	512	1088	256
STROBE-128/800	800	256	544	256
STROBE-256/800	800	512	288	256
STROBE-128/400	400	256	144	256

can go down to around

$$\frac{M^2}{2^b} + \frac{N}{2^c} + \mathbf{Adv}_F^{\text{key-pre}}(N)\,, \tag{5}$$

using techniques of [18,34] (see also Section 4). Common factor in the bounds of (1) and (5) is the presence of the term $\mathbf{Adv}_F^{\text{key-pre}}(N)$ for $F \in \{\text{OKS}, \text{FKS}\}$. Our improved analysis of $\mathbf{Adv}_{\text{OKS}}^{\text{key-pre}}(N)$ in (4) immediately yields an improved bound for the outer-keyed duplex, as well as for the nonce-respecting variants of both the outer-keyed sponge and duplex.

Beyond the keyed sponges and keyed duplexes, key prediction security also appears in the analyses of HMAC-SHA-3 [40] and the sandwich sponge [39]. In these works, the authors adopted the old bound of Gaži et al. of (3). Our new bound of (4) directly improves the security bounds of these schemes.

1.4 Application

Despite that FKS generically improves over OKS both from a security as an efficiency perspective, there are still reasons to resort to OKS. First, the security results only focus on the generic construction: full-state compression allows the adversary more power, and the underlying permutation may likely need more rounds. Second, the usage of OKS is conceptually simpler: the PRF can be implemented using the keyless sponge algorithm as a black-box.

STROBE. This exact idea lies at the heart of the STROBE protocol framework [28, 29]. It is designed on top of the sponge construction, and extends the use of a single permutation to a lightweight framework for network protocols, that allows for (keyless) hashing, authenticated encryption, authentication, pseudorandom number generation, and many more. In order to allow for a simple framework with extremely small code size, all functionalities supported by STROBE operate on the outer part only. In particular, STROBE does not allow for full-state absorption but rather absorbs key and data in the outer part.

The STROBE protocol framework is based on SHAKE [42] and supports instantiations with various widths, as indicated in Table 1. Whereas the larger

Table 2: Parameters of the four sponge-based round 3 CAESAR candidates. For Ketje and Keyak, we have taken the recommended key length.

scheme	b	c	r	k
Ascon [19]	320	256	64	128
	320	192	128	96
Ketje [12]	200	184	16	92
	400	368	32	128
Keyak [13]	800	256	544	128..224
	1600	256	1344	128..224
NORX [4]	512	128	384	128
	1024	256	768	256

STROBEs have $k \leq r$, the more lightweight STROBE-128/400 has suggested key size larger than r. The same applies to "STROBE lite" on 200-bits state (which is not included in the table). Our analysis confirms that these instantiations do achieve the claimed level of security and that the effective key length is not halved (as suggested by the earlier bound of (3)). We remark that the 256-bit key size in the STROBE protocol framework is a mere suggestion by the author; implementers may opt to use a larger or smaller key.

CAESAR Competition. Multiple submissions to the CAESAR competition for the development of a portfolio of authenticated encryption schemes [15] adopted the keyed duplex. Focusing on the third round candidates, Table 2 lists the parameters (b, c, r, k) of the four sponge-based schemes Ascon [19], Ketje [12], Keyak [13], and NORX [4]. Keyak implements the full-keyed duplex construction of [18] and the original key prediction bound of (2) applies. The remaining three schemes evaluate the outer-keyed duplex, with one twist: despite that message is absorbed in the outer part only, the key is absorbed in a full-state fashion. For example, for Ascon-128, the 128-bit key is used (alongside the initialization vector and nonce) to initialize the 320-bit state, and departing from that state, the outer-keyed duplex (with rate $r = 64$) is evaluated. Therefore, also for these schemes the original key prediction bound of (2) applies. Our observations nevertheless demonstrate that if, for some reason, full-state absorption of the key is infeasible, multi-round outer part absorption does not degrade security.

Lightweight Permutations. Our analysis improves over the existing bounds for OKS in case the key size exceeds the rate. It becomes particularly relevant in the context of lightweight cryptography and the current abundance in "small" permutations, all with different features. Whereas most allow for a large enough capacity and rate so that key compression takes only one round (typically permutations of size ≥ 384 bits, including Keccak-f[400] [9], C-Quark [3], SPONGENT-256/256/128 [14], and Gimli [5]), some have a smaller state, such as 228-bit Photon-256/32/32 [27], 320-bit Ascon [19], 200- or 280-bits PRIMATEs [1], and 128- or 256-bits Prøst [35]. If the rate is too small and one does not opt to simply initialize the state using the key (as, e.g., described for Ascon above), key absorption is necessarily performed in multiple rounds.

2 Preliminaries

For a finite set $\mathcal{S}$, we denote by $s \xleftarrow{\$} \mathcal{S}$ the uniform random sampling of s from $\mathcal{S}$. For a natural number $b \in \mathbb{N}$, $\{0,1\}^b$ denotes the set of b-bit strings, and $\mathrm{perm}(b)$ denotes the set of all permutations $\pi : \{0,1\}^b \to \{0,1\}^b$. We denote by $\{0,1\}^*$ the set of all strings. For $m \in \{0,1\}^*$, we denote by $m_1, \ldots, m_\nu = ✂_b(m)$ the two-step process of (i) appending m with $10^{-|m|-1 \bmod b}$, and (ii) partitioning the resulting string into $\nu \geq 1$ b-bit strings. For a b-bit string $m \in \{0,1\}^b$ and values $c, r \in \mathbb{N}$ such that $c, r \leq b$, we denote by $\lfloor m \rfloor_c$ the c rightmost bits of m and by $\lceil m \rceil_r$ the r leftmost bits of m.

An adversary $\mathcal{A}^{\mathcal{O}}$ is a computationally unbounded algorithm that is given adaptive access to an oracle $\mathcal{O}$ and outputs certain data. It wins if its output fulfills a (possibly randomized) winning condition W.

2.1 General Keyed Sponge

This work is concerned with the outer-keyed sponge and the full-keyed sponge: the former absorbs data in the outer part only, the latter absorbs data over the entire state. We can therefore neatly describe both in one go by considering a general keyed sponge that operates with two capacities: c_{ab} for absorption and c_{sq} for squeezing. (Nevertheless, most of the results in the remainder of the work concern the outer-keyed sponge.)

Let $b, c_{\mathrm{ab}}, c_{\mathrm{sq}}, r_{\mathrm{ab}}, r_{\mathrm{sq}}, k \in \mathbb{N}$ be such that $b = c_{\mathrm{ab}} + r_{\mathrm{ab}} = c_{\mathrm{sq}} + r_{\mathrm{sq}}$. Define $\lambda := \lceil k / r_{\mathrm{ab}} \rceil$. The general keyed sponge is a pseudorandom function based on a permutation $\pi \in \mathrm{perm}(b)$ that takes as input a key $K \in \{0,1\}^k$, an arbitrary-sized message $M \in \{0,1\}^*$, and a natural number $\ell \in \mathbb{N}$, and outputs a value $z \in \{0,1\}^\ell$:

$$\mathrm{GKS}^\pi : (K, m, \ell) \mapsto z \in \{0,1\}^\ell \,. \tag{6}$$

The function is specified in Algorithm 1 and depicted (for integral $\lambda := k / r_{\mathrm{ab}}$) in Figure 1.

The general keyed sponge construction covers the outer-keyed sponge OKS^π for $(c_{\mathrm{ab}}, r_{\mathrm{ab}}) = (c_{\mathrm{sq}}, r_{\mathrm{sq}}) =: (c, r)$. It covers the full-keyed sponge FKS^π for $(c_{\mathrm{ab}}, r_{\mathrm{ab}}) = (0, b)$ and $(c_{\mathrm{sq}}, r_{\mathrm{sq}}) =: (c, r)$.

Note that the parameter λ indicates the number of invocations of π to process the key. As both the outer-keyed as the full-keyed sponge achieve a security level of roughly c_{sq} at best, we can w.l.o.g. assume that $k \leq c_{\mathrm{sq}}$ throughout. For FKS, this implies that we always have $\lambda = 1$. For OKS, λ may be larger.

Algorithm 1 General keyed sponge construction GKS^{π}

```
Input: $K \in \{0,1\}^k$, $m \in \{0,1\}^*$, $\ell \in \mathbb{N}$
Output: $z \in \{0,1\}^\ell$
1: $x_1, \ldots, x_\nu = ✂_{r_{\mathrm{ab}}}(K \parallel m)$
2: $t = 0^b$
3: for $i = 1, \ldots, \nu$ do
4:     $t = \pi(t \oplus (x_i \parallel 0^{c_{\mathrm{ab}}}))$
5: $z = \lceil t \rceil_{r_{\mathrm{sq}}}$
6: while $|z| < \ell$ do
7:     $t = \pi(t)$
8:     $z = z \parallel \lceil t \rceil_{r_{\mathrm{sq}}}$
9: return $\lceil z \rceil_\ell$
```

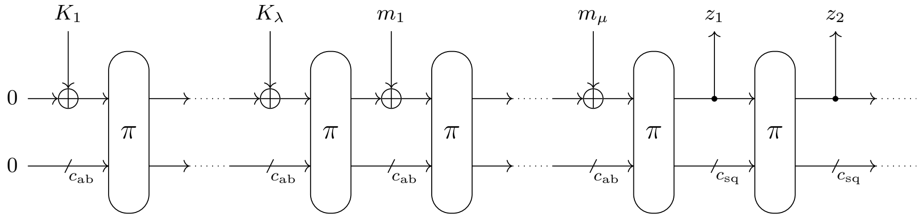


Fig. 1: General keyed sponge for λ integral key blocks and μ message blocks. The scheme covers the outer-keyed sponge for $c_{\mathrm{ab}} = c_{\mathrm{sq}} =: c$ and the full-keyed sponge for $c_{\mathrm{ab}} = 0$ and $c_{\mathrm{sq}} =: c$.

2.2 Yield

For proper understanding of the security of OKS and FKS against key prediction security as defined in Section 3, we will introduce the *yield* of a transcript of input-output tuples of π.

Definition 1. *Consider* GKS *based on* $\pi \in \mathrm{perm}(b)$*, and let* $\mathcal{Q}$ *be a finite list of input-output tuples of* π*. The* yield $\mathrm{yield}_{c_{\mathrm{ab}},\lambda}(\mathcal{Q})$ *of* $\mathcal{Q}$ *is defined as the set of all keys* $K \in \{0,1\}^k$ *for which there exists a message* $m \in \{0,1\}^*$ *such that the first* $\lambda = \lceil k/r_{\mathrm{ab}} \rceil$ *evaluations of* π *in line 4 of Algorithm 1 are in* $\mathcal{Q}$*.*

In other words, $\mathrm{yield}_{c_{\mathrm{ab}},\lambda}(\mathcal{Q})$ is the set of keys for which the absorption in GKS can be performed by only reading data from $\mathcal{Q}$, hence, without evaluating π.

The yield is closely related to a directed acyclic graph $G_{c_{\mathrm{ab}},\lambda}(\mathcal{Q}) = (V, A)$ defined as follows. Vertex set $V = \{V_0, V_1, \ldots, V_\lambda\}$ consists of $\lambda + 1$ layers, where

- $V_0 = \{0^b\}$;
- For $i = 1, \ldots, \lambda$: for any $(s, t) \in \mathcal{Q}$ and any L such that $s \oplus (L \parallel 0^{c_{\mathrm{ab}}}) \in V_{i-1}$, vertex t is added to V_i and arrow $s \oplus (L \parallel 0^{c_{\mathrm{ab}}}) \xrightarrow{L} t$ with label L is added to A.

For given c_{ab}, λ, and $\mathcal{Q}$, the size of the yield $\left|\mathrm{yield}_{c_{\mathrm{ab}},\lambda}(\mathcal{Q})\right|$ equals the number of paths from $V_0 = \{0^b\}$ to V_λ in $G_{c_{\mathrm{ab}},\lambda}(\mathcal{Q})$.

A visualization of the graph is depicted in Figure 2. By construction, $|V_i| \leq |\mathcal{Q}|$ for all $i = 1, \ldots, \lambda$. We highlight three further properties of the graph $G_{c_{\mathrm{ab}},\lambda}(\mathcal{Q})$:

- Two arrows departing from the same node, e.g., from u in Figure 2, happens if there exist distinct $L, L' \in \{0,1\}^{r_{\mathrm{ab}}}$ and distinct $t, t' \in \{0,1\}^b$ such that
$$\big(u \oplus (L \parallel 0^{c_{\mathrm{ab}}}), t\big), \big(u \oplus (L' \parallel 0^{c_{\mathrm{ab}}}), t'\big) \in \mathcal{Q}\,.$$
If the adversary makes a forward query, it can set such case with probability 1; if it makes an inverse query, the case happens only if the query response s matches that of an existing vertex u in the previous layer: $\lfloor s \rfloor_{c_{\mathrm{ab}}} = \lfloor u \rfloor_{c_{\mathrm{ab}}}$;
- Two arrows arriving at the same node, e.g., at v in Figure 2, happens if there exist distinct $L, L' \in \{0,1\}^{r_{\mathrm{ab}}}$ and distinct $s, s' \in \{0,1\}^b$ such that
$$\big(s \oplus (L \parallel 0^{c_{\mathrm{ab}}}), v\big), \big(s' \oplus (L' \parallel 0^{c_{\mathrm{ab}}}), v\big) \in \mathcal{Q}\,.$$
(These two queries are necessarily the same as π is a permutation.) If the adversary makes an inverse query, it can set such case with probability 1 (but the new edge will likely not appear in the tree, as the tree is built up from 0^b in V_0); if it makes a forward query, the case happens only if there exist two nodes in the previous layer with equal capacity: $\lfloor s \rfloor_{c_{\mathrm{ab}}} = \lfloor s' \rfloor_{c_{\mathrm{ab}}}$;
- As $\mathcal{Q}$ is the query history of a permutation, we can observe the following. In the first of the above properties, the two vertices at which the arrows end, t and t', must necessarily be distinct; in the second case, the two vertices from which the arrows depart, s and s', must necessarily be distinct. We can particularly conclude that there do not exist layer $i \in \{1, \ldots, \lambda\}$ and vertices $u \in V_{i-1}$ and $v \in V_i$ such that $G_{c_{\mathrm{ab}},\lambda}(\mathcal{Q})$ contains two arrows from u to v.

3 Key Prediction

For $F \in \{\mathrm{GKS}, \mathrm{OKS}, \mathrm{FKS}\}$, we define key prediction security by the following experiment: for a random permutation $\pi \xleftarrow{\$} \mathrm{perm}(b)$, consider an adversary $\mathcal{A}$ that has oracle access to $\pi^{\pm}$. The adversary can make a finite amount of queries to its oracle, which are summarized in a transcript $\mathcal{Q}$. Then, a key $K \xleftarrow{\$} \{0,1\}^k$ is uniformly randomly drawn, and the adversary wins if $K \in \mathrm{yield}_{c_{\mathrm{ab}},\lambda}(\mathcal{Q})$. Formally:

Definition 2. *The key prediction security of $F \in \{\mathrm{GKS}, \mathrm{OKS}, \mathrm{FKS}\}$ against an adversary $\mathcal{A}$ is defined as*

$$\mathbf{Adv}_F^{\text{key-pre}}(\mathcal{A}) = \mathbf{Pr}\Big(\pi \xleftarrow{\$} \mathrm{perm}(b)\,,\, \mathcal{Q} \leftarrow \mathcal{A}^{\pi^{\pm}}\,,\, K \xleftarrow{\$} \{0,1\}^k \;:\; K \in \mathrm{yield}_{c_{\mathrm{ab}},\lambda}(\mathcal{Q})\Big)\,. \quad (7)$$

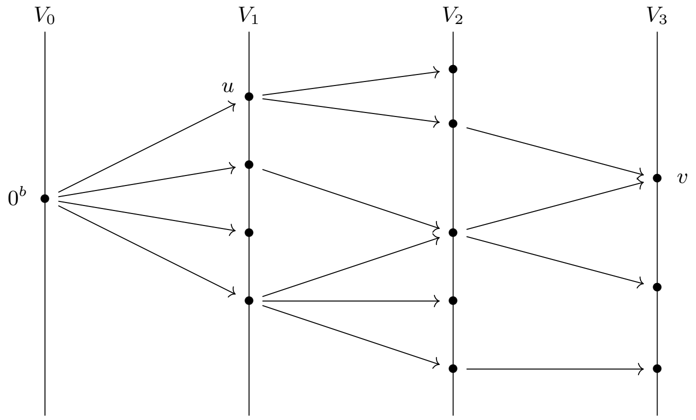


Fig. 2: Example graph $G_{c_{\mathrm{ab}},3}(\mathcal{Q})$ consisting of 4 vertex layers. Edge and vertex labels are omitted for clarity, except for vertices u and v for the sake of discussion.

For $N \geq 0$, we define by $\mathbf{Adv}_F^{\text{key-pre}}(N)$ the maximum over all adversaries making N queries to its oracle.

In above game, the key is only drawn *after* the adversary queries its oracle $\pi^{\pm}$. Its attack tools are mere combinatorics: it maximizes its chances by maximizing the size of the yield of the query history.

In the remainder of the section, we will focus on $F = \mathrm{OKS}$, hence we write $(c_{\mathrm{ab}}, r_{\mathrm{ab}}) = (c_{\mathrm{sq}}, r_{\mathrm{sq}}) =: (c, r)$. We will recall the bound of Gaži et al. [24, 25] on $\mathbf{Adv}_{\mathrm{OKS}}^{\text{key-pre}}$ in Section 3.1, and present our improved result in Section 3.2. Sections 3.3 and 3.4 cover the rationale and proof of our result, respectively. We discuss the possibility to generalize our results to multi-user security in Section 3.5.

3.1 Bound of Gaži et al.

We briefly discuss the bound of Gaži et al. [25, Lemma 12] (full version of [24]) on $\mathbf{Adv}_{\mathrm{OKS}}^{\text{key-pre}}$ for integral $\lambda := k/r$, including a concise sketch of their proof in our convention and notation. Afterwards, we will highlight several aspects of their analysis that contribute to non-tightness of their bound.

Proposition 1 (Gaži et al. [25, Lemma 12]). *Consider $F = \mathrm{OKS}$ for parameters (b, c, r, k), where $\lambda := k/r$ is integral. We have*

$$\mathbf{Adv}_F^{\text{key-pre}}(N) \leq \begin{cases} \frac{N}{2^k}\,, & \text{if } \lambda = 1\,, \\ \min\left\{ \frac{N^2}{2^c} + \frac{N}{2^k}, \frac{1}{2^b} + \frac{N 2^{\lambda}(3b-1)^{\lambda/2}}{2^{k/2}} \right\}\,, & \text{if } \lambda > 1\,. \end{cases} \tag{8}$$

Proof (sketch). In case of $\lambda = 1$, the adversary can make at most N evaluations of π for different key guesses $K_1, \ldots, K_N$, hence can obtain yield of size at most

$\left|\mathrm{yield}_{c,\lambda}(\mathcal{Q})\right| \leq N$. Likewise, the claim follows from the fact that $|V_1| \leq |\mathcal{Q}|$ and that $G_{c,\lambda}(\mathcal{Q})$ contains no two arrows from $0^b \in V_0$ to one and the same node in V_1 (property 3 in Section 2.2). The probability that a randomly selected key $K \xleftarrow{\$} \{0,1\}^k$ is in the yield is thus at most $N/2^k$.

We sketch Gaži et al.'s approach for $\lambda \geq 2$, assuming for simplicity that $k = \lambda \cdot r$. Consider any adversary making N queries, stored in a query history $\mathcal{Q}$. For $\alpha \in \mathbb{N}$, define by mc_α the event that there exist no $\alpha+1$ queries $(s_i, t_i)_{i=1}^{\alpha+1}$ to π for which either all queries are in forward direction and $\lfloor t_1 \rfloor_c = \cdots = \lfloor t_{\alpha+1} \rfloor_c$, or all are in inverse direction and $\lfloor s_1 \rfloor_c = \cdots = \lfloor s_{\alpha+1} \rfloor_c$. Obviously, $\mathbf{Pr}\left(\neg\mathsf{mc}_1\right) \leq N^2/2^c$. In addition, Gaži et al. prove using the Chernoff bound that[2]:

$$\mathbf{Pr}\left(\neg\mathsf{mc}_{3b-1}\right) \leq 1/2^b \,.$$

Gaži et al. subsequently prove that

$$\mathsf{mc}_1 \implies \left|\mathrm{yield}_{c,\lambda}(\mathcal{Q})\right| \leq N \,, \tag{9}$$

$$\mathsf{mc}_{3b-1} \implies \left|\mathrm{yield}_{c,\lambda}(\mathcal{Q})\right| \leq N 2^\lambda (3b-1)^{\lambda/2} 2^{k/2} \,, \tag{10}$$

which in turn concludes the proof as the probability that a randomly selected key $K \xleftarrow{\$} \{0,1\}^k$ is in the yield is at most $\left|\mathrm{yield}_{c,\lambda}(\mathcal{Q})\right|/2^k$. It thus remains to prove (9) and (10).

Regarding (9), mc_1 implies that $G_{c,\lambda}(\mathcal{Q})$ is a tree (cf. [25, Lemma 12]), and $\left|\mathrm{yield}_{c,\lambda}(Q)\right|$ is at most the number of nodes at layer λ: N. Regarding (10), as the adversary only makes N queries, there are at most N possible nodes in V_λ at distance λ from 0^b. For any such node v_λ, there are λ primitive queries connecting

$$0^b \longleftrightarrow \cdots \longleftrightarrow v_\lambda \,,$$

and each of these primitive queries could have been made in forward or in inverse direction. Let $C \in \{0,1\}^\lambda$ be any configuration, where $C_i = 0$ means that the arrow from V_{i-1} to V_i corresponds to a forward query and $C_i = 1$ that it corresponds to an inverse query. There are 2^λ possible configurations. Starting from the end node, by mc_{3b-1} there are at most $3b-1$ arrows into v_λ that correspond to forward queries to π, but we do not know anything about inverse queries, other than that the in-degree of v_λ is at most 2^r. This gives the following upper bound on the number of paths from 0^b to (fixed) v_λ for fixed configuration S:

$$(3b-1)^{\lambda - |S|}(2^r)^{|S|} \,.$$

For $|S| \leq \lambda/2$ this term is at most $(3b-1)^{\lambda/2}(2^r)^{\lambda/2}$. For $|S| > \lambda/2$ one can revert the reasoning, i.e., start from the first node 0^b instead of end node v_λ.

[2] There's a small gap in the reasoning, namely that the Chernoff bound considers a sum of *independent* events whereas in the current case a sum of inherently *dependent* events is considered.

This likewise gives upper bound

$$(3b-1)^{|S|}(2^r)^{\lambda-|S|}\,,$$

which for $|S| > \lambda/2$ is also at most$(3b-1)^{\lambda/2}(2^r)^{\lambda/2}$. Summing over all possible configurations S and all possible end nodes, we obtain

$$\left|\text{yield}_{c,\lambda}(\mathcal{Q})\right| \leq N2^\lambda(3b-1)^{\lambda/2}(2^r)^{\lambda/2} = N2^\lambda(3b-1)^{\lambda/2}2^{k/2}\,,$$

as $k = \lambda \cdot r$. □

Besides the minor and insignificant glitch that the Chernoff bound is applied to the sum of dependent events, the proof has two shortcomings that yield non-tightness of the bound:

(i) The selection of the end node (N possibilities) already fixes the last vertex. A single query may connect multiple elements from $V_{\lambda-1}$ with V_λ, hence one cannot just assume that fixing the end node gives one arrow "for free," but it still fixes the inner part of the node at layer $V_{\lambda-1}$;

(ii) Assuming mc_{3b-1}, there is no $3b$-fold inner collision, and no more than $3b-1$ forward queries that map to the same node in a layer V_i; yet for the inverse queries Gaži et al. cannot rely on mc_{3b-1} and resort to the fact that any node has at most 2^r incoming arrows. It is intuitively appealing to say that also for inverse queries the number of *useful* arrows is at most $3b-1$, where *useful* refers to the fact that the path should lead to root $0^b \in V_0$.

3.2 Improved Key Prediction Security

We derive the following improved bound for key prediction security. The bound for $\lambda = 1$ of Proposition 1 is already tight, and we restrict ourselves to the case of $\lambda > 1$.

Theorem 1. *Consider $F = \text{OKS}$ for parameters (b, c, r, k), where $\lambda := \lceil k/r \rceil > 1$. Let $\alpha > \lambda$ be a natural number. We have*

$$\mathbf{Adv}_F^{\text{key-pre}}(N) \leq \frac{(2\alpha)^{\lambda-1}N}{2^k} + \lambda 2^\lambda 2^{2c}\left(\frac{2eN^*}{2^c\alpha}\right)^{\alpha/\lambda}\,, \qquad (11)$$

where $N^* = \max\{N, \lambda 2^{(\lambda-1)r}\}$.

The proof will be given in Section 3.4, preceded by its rationale in Section 3.3.

Parameter α is a threshold: the first term increases whereas the second term decreases for increasing α. By taking $\alpha/\lambda \geq c$, above bound simplifies to

$$\mathbf{Adv}_F^{\text{key-pre}}(N) \leq \frac{(2\alpha)^{\lambda-1}N}{2^k} + \lambda\left(\frac{16eN^*}{2^c\alpha}\right)^{\alpha/\lambda}\,,$$

using that $\lambda \leq c$ as $\lambda \leq k$ and $k \leq c$. For N close to 2^k, $N^* = N$ and the first term dominates the equation. For smaller N, $N^* = \lambda 2^{(\lambda-1)r}$, and the second term dominates.

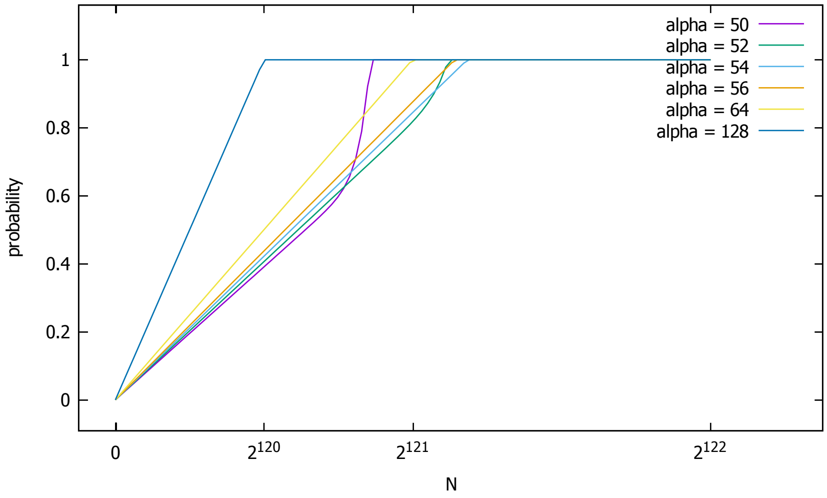


Fig. 3: For parameter set $(b, c, r, k) = (192, 128, 64, 128)$, the bound of Theorem 1 for $\alpha = 50, 52, 54, 56, 64, 128$ (from right to left).

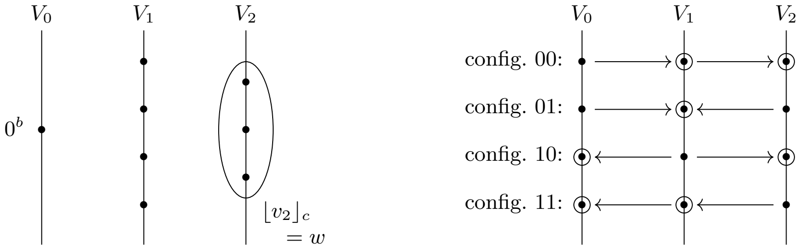
For the case of $(b, c, r, k) = (192, 128, 64, 128)$, with $\lambda = 2$, the graph in Figure 3 plots the bound of Theorem 1 for various choices of α. One sees that taking $\alpha \geq 2c$ is convenient for making the bound simple, but it is a rough estimate. The first term dominates as long as $\alpha \geq 54$. For $\alpha \leq 53$ the second term in (11) starts to dominate. The graph in Figure 3 depicts it well for $\alpha = 52$: it follows the linear behavior of the first term, until at some point the exponential term of (11) becomes dominating. Numerical computation shows that for $\alpha = 54$, the bound of (11) equals 1 for $N \approx 2^{121.25}$. For comparison, the bound of Gaži et al. of Proposition 1 equals 1 for $N \approx 2^{64}$.

The best known attack is the generic one, that fixes $\lambda - 1$ blocks $K_1, \dots, K_{\lambda-1}$ and varies K_λ: this procedure renders a yield of size exactly $N - (\lambda - 1)$ and succeeds in predicting the key with probability $(N - (\lambda - 1))/2^k \approx N/2^k$. The bound of Theorem 1 permits a small loss in comparison with this attack, which comes from the accidental event of multi-collisions on the inner part of π. The term increases exponentially in λ, which is due to the fact that multi-collisions could occur at every evaluation of π, and they can amplify each other. In the end, however, it only yields a small loss.

3.3 Rationale Behind Theorem 1

The proof, at a high level, relies on the observation that any path from V_0 to a fixed vertex $v_\lambda \in V_\lambda$ must contain at least one collision on the capacity: either between a query and $0^b \in V_0$ or v_λ, or between two queries at any layer in $V_1, \dots, V_{\lambda-1}$. In addition, it uses that for a fixed inner part, there are at most $\min\{N, 2^r\}$ queries.

The proof is easiest understood by considering $\lambda = 3$. In this case, we consider graph $G_{c,3}(\mathcal{Q})$ consisting of four layers V_0, V_1, V_2, V_3. The first step in the proof will be to *fix* the last vertex $v_3 \in V_3$. There are at most N choices, as this choice is equivalent to fixing $(s, v_3) \in \mathcal{Q}$. The choice also fixes the inner part of the nodes in V_2 under consideration: the query links V_2 with V_3 *only for* nodes v_2 such that $\lfloor v_2 \rfloor_c = \lfloor s \rfloor_c$. Define the fixed capacity by $w := \lfloor s \rfloor_c \in \{0,1\}^c$. We henceforth have to focus on paths from $0^b \in V_0$ to any $v_2 \in V_2$, where $\lfloor v_2 \rfloor_c = w$. See also Figure 4a. Any of these paths fits one of the four configurations of Figure 4b.



(a) Targeted subgraph consisting of 2 layers, where $w \in \{0,1\}^c$ is fixed.

(b) Possible configurations for the directions of the queries. A circle ○ indicates that a fixed inner value must be hit.

Fig. 4: Graph and configurations for the rationale of Section 3.3.

Assume, for the sake of argument, that $\mathcal{Q}$ contains no $(\alpha+1)$-fold inner collision (in either forward or inverse direction). For configuration 00 of Figure 4b, we have at most α arrows from V_1 to V_2, each of which fixes the inner part of the node in V_1, yielding at most α arrows from V_0 to V_1; see also the circle indications in the figure. We have obtained that there are at most α^2 paths that obey to configuration 00. Likewise, there are at most α^2 paths for configurations 10 and 11, using that the node in V_0, 0^b, has its inner part fixed: 0^c. What remains is configuration 01: the inner parts of the values at layers V_0 and V_2 are fixed, and thus there are at most $\min\{N, 2^r\}$ queries in $\mathcal{Q}$ that could depart from V_0 in forward direction and V_2 in inverse direction. The expected numbers of collisions in the middle is

$$\frac{\min\{N, 2^r\}^2}{2^c} \,.$$

By Markov inequality, more than α^2 paths exist with probability at most

$$\frac{\min\{N, 2^r\}^2}{2^c \alpha^2} \,. \tag{12}$$

This gives $\left|\text{yield}_{c,3}(\mathcal{Q})\right| = 4\alpha^2 N$, except with probability (12) and the probability that there exists an $(\alpha+1)$-fold inner collision.

Unfortunately, above reasoning is not entirely correct: the evaluation of configuration 01 has to be performed for all possible $w \in \{0,1\}^c$, as the adversary can freely choose the query for the last layer. Multiplying (12) by 2^c does not help; to the contrary, the bound becomes meaningless. Instead, configuration 01 will also be analyzed by relying on the non-existence of $(\alpha+1)$-fold inner collisions: any hit adds at most α solutions.

Additional issues surface if we extend the reasoning to larger values of λ: configurations like 001 ($0^c \to \cdot \to \cdot \leftarrow w$) or 0101 ($0^c \to \cdot \leftarrow \cdot \to \cdot \leftarrow w$) appear. To resolve these issues, a recursive reasoning (in λ) will be performed.

3.4 Proof of Theorem 1

Consider $F = \text{OKS}$ for parameters (b,c,r,k), where $\lambda := \lceil k/r \rceil > 1$, and let $\pi \xleftarrow{\$} \text{perm}(b)$. Consider any adversary $\mathcal{A}$ with two-sided query access to π. We will record the query history in a transcript $\mathcal{Q}$, where we keep track of the query direction using a bit $C \in \{0,1\}$, i.e., a tuple $(C,s,t) \in \mathcal{Q}$ is made in forward direction if $C=0$ and inverse direction if $C=1$. Associated with $\mathcal{Q}$ is graph $G_{c,\lambda}(\mathcal{Q}) = (V,A)$ as defined in Section 2.2.

Defining Auxiliary Events. Let $\alpha > 0$ be an integral threshold. At the core of our proof is the event mc, that bounds the maximum size of a multi-collision in $\mathcal{Q}$. We define

$$\mathsf{mc} = \mathsf{mc}^+ \wedge \mathsf{mc}^- \,, \tag{13}$$

where

$$\mathsf{mc}^+ \;:\; \max_{w\in\{0,1\}^c} \left|\left\{(0,s,t)\in\mathcal{Q} \;\middle|\; \lfloor t\rfloor_c = w\right\}\right| \le \alpha\,, \tag{14}$$

$$\mathsf{mc}^- \;:\; \max_{v\in\{0,1\}^c} \left|\left\{(1,s,t)\in\mathcal{Q} \;\middle|\; \lfloor s\rfloor_c = v\right\}\right| \le \alpha\,. \tag{15}$$

The event mc corresponds to mc_α in the proof of Proposition 1, be it with α omitted as subscript.

The proof will, implicitly, be performed by induction on λ. For $i = 1,\ldots,\lambda-1$, we define the event ch_i, a condition on chains of length i, as follows:

$$\begin{split}\mathsf{ch}_i \;:\; \max_{C\in\{0,1\}^i} \max_{v,w\in\{0,1\}^c} \Big|\Big\{(C_1,s_1,t_1),\ldots,(C_i,s_i,t_i)\in\mathcal{Q} \;\Big|\\ \lfloor s_1\rfloor_c = v \wedge \left(\lfloor t_j\rfloor_c = \lfloor s_{j+1}\rfloor_c\right)_{j=1}^{i-1} \wedge \lfloor t_i\rfloor_c = w\Big\}\Big| \le \alpha^i\,.\end{split} \tag{16}$$

In other words, ch_i is the maximum number of solutions to

$$v \xleftrightarrow{C_1} \cdot \xleftrightarrow{C_2} \cdots \xleftrightarrow{C_i} w\,, \tag{17}$$

maximized over all possible configurations $C = (C_1,\ldots,C_i) \in \{0,1\}^i$ (the label on the arrow indicates the configuration of the query) and start and end nodes

$v, w \in \{0,1\}^c$. There is an important difference between ch_1 (or ch_i in general) on the one hand and mc on the other hand: in ch_1 the inner values of *both sides of the path* are fixed, whereas in mc only one side is fixed. Nevertheless, $\mathsf{mc} \Rightarrow \mathsf{ch}_1$ by definition. Furthermore, for $i = \lambda - 1$ the proof needs $\mathsf{ch}_{\lambda-1}$ only for $v = 0^c$, but we have opted to include the general event for simplicity of argument.

Bounding the Yield. Fix any vertex $v_\lambda \in V_\lambda$. There are at most N choices as this choice is equivalent to fixing $(s, v_\lambda) \in \mathcal{Q}$, and it also fixes the inner part $w := \lfloor s \rfloor_c \in \{0,1\}^c$ for the node at shore $V_{\lambda-1}$. We thus have to focus on paths from $0^b \in V_0$ to any $v_{\lambda-1} \in V_{\lambda-1}$ with $\lfloor v_{\lambda-1} \rfloor_c = w$. Let $C \in \{0,1\}^{\lambda-1}$ be any configuration. By $\mathsf{ch}_{\lambda-1}$, there are at most $\alpha^{\lambda-1}$ paths from 0^c to the fixed w for this particular configuration. Summing over all possible configurations and all possible choices v_λ, we obtain that

$$\mathsf{ch}_{\lambda-1} \implies \left|\mathrm{yield}_{c,\lambda}(\mathcal{Q})\right| \leq (2\alpha)^{\lambda-1} N \,. \tag{18}$$

As the key $K \xleftarrow{\$} \{0,1\}^k$ is randomly selected, the adversary succeeds with probability at most $\left|\mathrm{yield}_{c,\lambda}(\mathcal{Q})\right| / 2^k$ plus the probability that $\mathsf{ch}_{\lambda-1}$ is *not* satisfied.

Analyzing Auxiliary Events. It remains to analyze the probability that $\neg\mathsf{ch}_{\lambda-1}$. By basic probability theory,

$$\mathbf{Pr}\left(\neg\mathsf{ch}_{\lambda-1}\right) \leq \mathbf{Pr}\left(\neg\mathsf{mc}\right) + \sum_{i=1}^{\lambda-1} \mathbf{Pr}\left(\neg\mathsf{ch}_i \mid \mathsf{ch}_{i-1} \wedge \cdots \wedge \mathsf{ch}_1 \wedge \mathsf{mc}\right) . \tag{19}$$

Lemma 1. *We have*

$$\mathbf{Pr}\left(\neg\mathsf{mc}\right) \leq 2 \cdot 2^c \left(\frac{2eN}{2^c\alpha}\right)^\alpha . \tag{20}$$

Proof (of Lemma 1). Without loss of generality, consider mc^+. Fix any $w \in \{0,1\}^c$. Any forward query $(0, s, t)$ satisfies $\lfloor t \rfloor_c = w$ with probability at most $2^r/(2^b - N)$, as the response is randomly drawn from a set of size at least $2^b - N$ and at most 2^r of those fulfill the condition. More than α satisfy the condition with probability at most[3]

$$\binom{N}{\alpha}\left(\frac{2^r}{2^b - N}\right)^\alpha \leq \left(\frac{2eN}{2^c\alpha}\right)^\alpha ,$$

using Stirling's approximation and the fact that $N \leq 2^{b-1}$. The proof is completed by summing over all possible $w \in \{0,1\}^c$ and by taking into account mc^- as well (cf. (13)). □

[3] In the bound we could take $\alpha + 1$ instead of α, but have opted not to do so for simplicity.

Lemma 2.

$$\mathbf{Pr}\left(\neg \mathsf{ch}_i \mid \mathsf{ch}_{i-1} \wedge \cdots \wedge \mathsf{ch}_1 \wedge \mathsf{mc}\right) \leq \begin{cases} 0\,, & \textit{for } i = 1\,, \\ i2^i2^{2c}\left(\frac{2ei2^{ir}}{2^c\alpha}\right)^{\alpha/i}\,, & \textit{for } i > 1\,. \end{cases} \tag{21}$$

Proof (of Lemma 2). We have $\mathsf{mc} \Rightarrow \mathsf{ch}_1$ by definition, and focus on the case of arbitrary $i > 1$. Fix any configuration $C \in \{0,1\}^i$ and any $v, w \in \{0,1\}^c$, in total 2^i2^{2c} possible choices. We aim to prove that the number of solutions to (17) is at most α^i. To the contrary, assume it is more than α^i. Then, by the pigeonhole principle, there must be an index $j \in \{1, \ldots, i\}$ such that more than α^i/i solutions are constituted with the winning query, i.e., the query that completes the chain, occurring at position j.[4] Without loss of generality (the argument is fully symmetric), assume that $C_j = 0$, i.e., that it is a forward query.

As $v \in \{0,1\}^c$ is fixed, there are at most 2^{jr} possible values $s \in \{0,1\}^b$ at distance $j-1$ from v, i.e., the adversary can make at most $\min\{N, 2^{jr}\}$ attempts. As $w \in \{0,1\}^c$ is fixed, there are at most $2^{(i-j)r}$ possible inner parts x such that a path

$$x \xleftrightarrow{C_{j+1}} \cdots \xleftrightarrow{C_i} w$$

can be constituted from the query history. The new forward query hits any of these inner parts with probability at most $2^{(i-j)r} \cdot 2^r/(2^b - N)$. By ch_{j-1} and ch_{i-j}, any such hit adds at most $\alpha^{j-1} \cdot \alpha^{i-j} = \alpha^{i-1}$ solutions. In order to get more than α^i/i solutions, there must be more than α/i collisions, which happens with probability at most

$$\binom{\min\{N, 2^{jr}\}}{\alpha/i}\left(\frac{2^{(i-j)r} \cdot 2^r}{2^b - N}\right)^{\alpha/i} \leq \left(\frac{2ei2^{ir}}{2^c\alpha}\right)^{\alpha/i}\,, \tag{22}$$

again using Stirling's approximation and the fact that $N \leq 2^{b-1}$, and where we recall that the adversary has only $\min\{N, 2^{jr}\} \leq 2^{jr}$ shots to success. The proof is completed by summing over all possible configurations $C \in \{0,1\}^i$, start and end nodes $v, w \in \{0,1\}^c$, and positions of the winning query $j \in \{1, \ldots, i\}$. □

[4] It could be that this query occurs at multiple positions in the chain, but this does not invalidate the reasoning due to generous counting.

From (19) and Lemmas 1 and 2 we immediately obtain, using that $\alpha > \lambda$,

$$\begin{aligned}
\mathbf{Pr}\left(\neg\mathsf{ch}_{\lambda-1}\right) &\leq 2 \cdot 2^{c}\left(\frac{2eN}{2^{c}\alpha}\right)^{\alpha} + \sum_{i=2}^{\lambda-1} i2^{i}2^{2c}\left(\frac{2ei2^{ir}}{2^{c}\alpha}\right)^{\alpha/i} \\
&\leq 2 \cdot 2^{c}\left(\frac{2eN}{2^{c}\alpha}\right)^{\alpha} + \sum_{i=2}^{\lambda-1} \lambda2^{i}2^{2c}\left(\frac{2e\lambda2^{(\lambda-1)r}}{2^{c}\alpha}\right)^{\alpha/\lambda} \\
&\leq \sum_{i=1}^{\lambda-1} \lambda2^{i}2^{2c}\left(\frac{2eN^{*}}{2^{c}\alpha}\right)^{\alpha/\lambda} \\
&\leq \lambda2^{\lambda}2^{2c}\left(\frac{2eN^{*}}{2^{c}\alpha}\right)^{\alpha/\lambda}\,, \qquad (23)
\end{aligned}$$

where $N^{*} = \max\{N, \lambda2^{(\lambda-1)r}\}$. Slight improvements in the bound could be achieved, at the cost of readability penalties, in the derivation of Lemma 2 and in the bounding of (23) above.

3.5 Generalization to Multi-User Security

It is straightforward to generalize our analysis to multi-user security. First, to generalize Definition 2, one would consider $K_1, \ldots, K_u \xleftarrow{\$} \{0,1\}^k$, where $u \in \mathbb{N}$ is the number of users, and the adversary wins if $K_i \in \text{yield}_{c_{\text{ab}},\lambda}(\mathcal{Q})$ for some $i \in \{1, \ldots, u\}$. The core of the proof of Theorem 1 is about deriving an upper bound on $\left|\text{yield}_{c,\lambda}(\mathcal{Q})\right|$. Once this bound is derived (and the analysis in Section 3.4 carries over verbatim), any of the u keys is in the yield with probability at most $u \cdot \left|\text{yield}_{c,\lambda}(\mathcal{Q})\right| / 2^{k}$.

4 Application to Keyed Sponge and Duplex

Keyed sponges are evaluated in a PRF security model. In more detail, let $\mathcal{RO}^{\infty} : \{0,1\}^{*} \to \{0,1\}^{\infty}$ be a function that for every input m defines an infinitely large string of random bits, and define $\mathcal{RO} : \{0,1\}^{*} \times \mathbb{N} \to \{0,1\}^{\mathbb{N}}$ as a function that on input of $(m, \ell) \in \{0,1\}^{*} \times \mathbb{N}$ outputs $\lceil \mathcal{RO}^{\infty}(m) \rceil_{\ell}$. Abusing notation, write $\text{ro}(*, \mathbb{N})$ for the set of all such functions $\mathcal{RO}$. For a random permutation $\pi \xleftarrow{\$} \text{perm}(b)$, key $K \xleftarrow{\$} \{0,1\}^{k}$, and $\mathcal{RO} \xleftarrow{\$} \text{ro}(*, \mathbb{N})$, consider an adversary $\mathcal{A}$ that has oracle access to either $(F_K^{\pi}, \pi^{\pm})$ or $(\mathcal{RO}, \pi^{\pm})$. It succeeds if it manages to determine (with high probability) the world it is conversing with. Formally:

Definition 3. *The PRF security of $F \in \{\text{GKS}, \text{OKS}, \text{FKS}\}$ against an adversary $\mathcal{A}$ is defined as*

$$\begin{aligned}
\mathbf{Adv}_{F}^{\text{prf}}(\mathcal{A}) = \mathbf{Pr}&\left(\pi \xleftarrow{\$} \text{perm}(b)\,,\, K \xleftarrow{\$} \{0,1\}^{k} \;:\; 1 \leftarrow \mathcal{A}^{F_K^{\pi}, \pi^{\pm}}\right) \\
&- \mathbf{Pr}\left(\pi \xleftarrow{\$} \text{perm}(b)\,,\, \mathcal{RO} \xleftarrow{\$} \text{ro}(*, \mathbb{N}) \;:\; 1 \leftarrow \mathcal{A}^{\mathcal{RO}, \pi^{\pm}}\right)\,. \qquad (24)
\end{aligned}$$

For $M, N \geq 0$, we define by $\mathbf{Adv}_F^{\mathrm{prf}}(M, N)$ the maximum over all adversaries making M queries to its first oracle and N queries to its second oracle.

A bound on the PRF security of FKS was derived by Mennink et al. [38], but we are mostly concerned with the outer keyed sponge. For OKS, an earlier proof appeared by Bertoni et al. [10]. Andreeva et al. [2] improved the analysis and generalized it to multi-user security. Naito and Yasuda [41] derived a bound that is independent of the message length. As our security model considers an adversary whose complexity is solely measured by M and N, we discard most of the sophisticated improvements in [2,41] and consider a simplified bound. In addition, we modernize the bound using the key prediction security notion of Definition 2.

Theorem 2 (Andreeva et al. [2] and Naito and Yasuda [41], simplified). *Consider $F = \mathrm{OKS}$ for parameters (b, c, r, k). We have*

$$\mathbf{Adv}_F^{\mathrm{prf}}(M, N) \leq const_1 \cdot \frac{M^2}{2^c} + const_2 \cdot \frac{MN}{2^c} + \mathbf{Adv}_F^{\text{key-pre}}(N)\,. \tag{25}$$

The constant terms $const_1$ and $const_2$ are small (they equal 1 and 4 in [2]). The simplifications we have put through from [2,41] *only* affect the fractions in (25) and do not affect the point we are making with regard to the remaining term in (25). This term $\mathbf{Adv}_F^{\text{key-pre}}(N)$ in turn corresponds to a specific bad event in the analyses in [2,41]. Both Andreeva et al. [2, Lemma 2] and Naito and Yasuda [41, Theorem 2] rely on the bound of Gaži et al. [24,25] on $\mathbf{Adv}_F^{\text{key-pre}}(N)$: the bound expressed in Proposition 1. Our new result of Theorem 1 directly improves over the bounds from Andreeva et al. and Naito and Yasuda, and confirms that a shorter key can be taken to achieve the same level of security.

On passing, we remark that the result has comparable impact to the keyed duplex [8], a sponge-related construction well-suited for authenticated encryption. It is a stateful construction that has a "duplexing interface:" it gets as input a data block of size r bits, transforms the state using a permutation, and returns part of the outer part of the state. It comes with an outer-keyed flavor [8] as well as a full-keyed flavor [18,38], and the keyed sponge bounds are known to be transferable to the duplex (and vice versa) up to some degree. Crucial to this transition is that in the keyed duplex, the key absorption occurs with no intermediate output of the outer part, and thus occurs in a full-/outer-keyed sponge fashion. In this way, the key prediction term in the duplex bounds is also $\mathbf{Adv}_F^{\text{key-pre}}(N)$, where $F \in \{\mathrm{OKS}, \mathrm{FKS}\}$, i.e., the bound derived in Theorem 1.

In addition, generic security results on HMAC-SHA-3 [40] and the sandwich sponge [39] explicitly rely on the key prediction security term of Gaži et al., and our new bound immediately improves their results.

5 Note on Key Recovery

In a similar vein as in Section 3, one can define the key *recovery* security of $F \in \{\mathrm{GKS}, \mathrm{OKS}, \mathrm{FKS}\}$. For a random permutation $\pi \xleftarrow{\$} \mathrm{perm}(b)$ and key

$K \xleftarrow{\$} \{0,1\}^k$, consider an adversary $\mathcal{A}$ that has oracle access to $(F_K^\pi, \pi^\pm)$. The adversary can make a limited amount of queries to its oracles, summarized in a query transcript $\mathcal{Q}$, and afterwards it outputs a key $K' \in \{0,1\}^k$. It wins if $K' \in \mathrm{yield}_{c_{\mathrm{ab}},\lambda}(\mathcal{Q})$ and $F_K^\pi(\cdot) = F_{K'}^\pi(\cdot)$. Formally:

Definition 4. *The key recovery security of $F \in \{\mathrm{GKS}, \mathrm{OKS}, \mathrm{FKS}\}$ against an adversary $\mathcal{A}$ is defined as*

$$\mathbf{Adv}_F^{\text{key-rec}}(\mathcal{A}) = \mathbf{Pr}\Big(\pi \xleftarrow{\$} \mathrm{perm}(b)\,,\, K \xleftarrow{\$} \{0,1\}^k\,,\, (K', \mathcal{Q}) \leftarrow \mathcal{A}^{F_K^\pi, \pi^\pm} : \\ K' \in \mathrm{yield}_{c_{\mathrm{ab}},\lambda}(\mathcal{Q}) \wedge F_K^\pi(\cdot) = F_{K'}^\pi(\cdot)\Big)\,. \quad (26)$$

For $M, N \geq 0$, we define by $\mathbf{Adv}_F^{\text{key-rec}}(M,N)$ the maximum over all adversaries making M queries to its first oracle and N queries to its second oracle.

In other words, key recovery security differs from key prediction security in that the adversary has access to the keyed construction F_K^π.

One may argue that key recovery is a more meaningful notion to consider than key prediction. However, close inspection at how a key recovery security proof would look like reveals that the key recovery security of OKS is very close to its PRF security. To wit, the core ingredients of the PRF security bound of OKS (Theorem 2) are (i) the event of two evaluations of π imposed by $F_K^\pm$ with the same inner part, (ii) a primitive query to $\pi^\pm$ and an evaluation of π imposed by $F_K^\pm$ with the same inner part, and (iii) guessing/predicting the key. These parts are represented by the three terms in the bound of Theorem 2 in equal order.

Obviously, one way to recover the key is to predict it (part (iii) of the above). Now, suppose the adversary makes a query to $\pi^\pm$ whose inner part is equal to the inner part of an evaluation of π imposed by $F_K^\pm$ (part (ii) of the above). In this case, the adversary can back-track the sponge to obtain t_λ, the state of the sponge after the compression of the last key block. Once it knows t_λ, depending on λ it can learn (part of) the key. Regarding part (i): if two evaluations of π imposed by $F_K^\pm$ have the same inner part, the adversary can use this information to distinguish the scheme from a random function but it has no means to use this information to recover the key.

To summarize, the key recovery security bound would be constituted of parts (iii) and (ii) of the PRF security bound. Part (i) is minor compared with (ii), as the offline complexity is typically higher than the online complexity. We can thus conclude that the key recovery security of OKS is very close to its PRF security.

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

References

1. Andreeva, E., Bilgin, B., Bogdanov, A., Luykx, A., Mendel, F., Mennink, B., Mouha, N., Wang, Q., Yasuda, K.: PRIMATEs v1.02 (September 2014)
2. 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. Lecture Notes in Computer Science, vol. 9054, pp. 364–384. Springer (2015)
3. Aumasson, J., Henzen, L., Meier, W., Naya-Plasencia, M.: Quark: A Lightweight Hash. In: Mangard, S., Standaert, F. (eds.) CHES 2010. Lecture Notes in Computer Science, vol. 6225, pp. 1–15. Springer (2010)
4. Aumasson, J., Jovanovic, P., Neves, S.: NORX v3.0 (September 2016)
5. Bernstein, D.J., Kölbl, S., Lucks, S., Massolino, P.M.C., Mendel, F., Nawaz, K., Schneider, T., Schwabe, P., Standaert, F., Todo, Y., Viguier, B.: Gimli : A Cross-Platform Permutation. In: Fischer, W., Homma, N. (eds.) CHES 2017. Lecture Notes in Computer Science, vol. 10529, pp. 299–320. Springer (2017)
6. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Sponge functions. ECRYPT Hash Workshop 2007 (May 2007)
7. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: On the Indifferentiability of the Sponge Construction. In: Smart, N.P. (ed.) EUROCRYPT 2008. Lecture Notes in Computer Science, vol. 4965, pp. 181–197. Springer (2008)
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. Lecture Notes in Computer Science, vol. 7118, pp. 320–337. Springer (2011)
9. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: The Keccak reference (January 2011), https://keccak.team/files/Keccak-reference-3.0.pdf
10. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: On the security of the keyed sponge construction. Symmetric Key Encryption Workshop (February 2011)
11. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Permutation-based encryption, authentication and authenticated encryption. Directions in Authenticated Ciphers (July 2012)
12. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G., Van Keer, R.: CAESAR submission: Ketje v2 (September 2016)
13. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G., Van Keer, R.: CAESAR submission: Keyak v2 (September 2016)
14. Bogdanov, A., Knezevic, M., Leander, G., Toz, D., Varıcı, K., Verbauwhede, I.: spongent: A Lightweight Hash Function. In: Preneel, B., Takagi, T. (eds.) CHES 2011. Lecture Notes in Computer Science, vol. 6917, pp. 312–325. Springer (2011)
15. CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness (January 2018), http://competitions.cr.yp.to/caesar.html
16. Chaigneau, C., Gilbert, H.: Is AEZ v4.1 Sufficiently Resilient Against Key-Recovery Attacks? IACR Trans. Symmetric Cryptol. 2016(1), 114–133 (2016)
17. Chang, D., Dworkin, M., Hong, S., Kelsey, J., Nandi, M.: A keyed sponge construction with pseudorandomness in the standard model. NIST SHA-3 Workshop (March 2012)
18. 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, Part II. Lecture Notes in Computer Science, vol. 10625, pp. 606–637. Springer (2017)
19. Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M.: Ascon v1.2 (September 2016), submission to CAESAR competition

20. FIPS 197: Advanced Encryption Standard (AES). National Institute of Standards and Technology (1999)
21. FIPS 202: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions (2015)
22. Fleischmann, E., Forler, C., Lucks, S.: McOE: A Family of Almost Foolproof On-Line Authenticated Encryption Schemes. In: Canteaut, A. (ed.) FSE 2012. Lecture Notes in Computer Science, vol. 7549, pp. 196–215. Springer (2012)
23. Fuhr, T., Leurent, G., Suder, V.: Collision Attacks Against CAESAR Candidates - Forgery and Key-Recovery Against AEZ and Marble. In: Iwata and Cheon [33], pp. 510–532
24. 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, Part I. Lecture Notes in Computer Science, vol. 9215, pp. 368–387. Springer (2015)
25. Gaži, P., Pietrzak, K., Tessaro, S.: Tight Bounds for Keyed Sponges and Truncated CBC. Cryptology ePrint Archive, Report 2015/053 (2015)
26. Guo, J.: Marble Specification Version 1.0 (March 2014)
27. Guo, J., Peyrin, T., Poschmann, A.: The PHOTON Family of Lightweight Hash Functions. In: Rogaway, P. (ed.) CRYPTO 2011. Lecture Notes in Computer Science, vol. 6841, pp. 222–239. Springer (2011)
28. Hamburg, M.: STROBE protocol framework (July 2017), https://strobe.sourceforge.io
29. Hamburg, M.: The STROBE protocol framework. Cryptology ePrint Archive, Report 2017/003 (2017)
30. Hoang, V., Krovetz, T., Rogaway, P.: AEZ v3: Authenticated Encryption by Enciphering (August 2014)
31. Hoang, V., Krovetz, T., Rogaway, P.: AEZ v4: Authenticated Encryption by Enciphering (August 2015)
32. Hoang, V., Krovetz, T., Rogaway, P.: AEZ v4.1: Authenticated Encryption by Enciphering (October 2015)
33. Iwata, T., Cheon, J.H. (eds.): ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II, Lecture Notes in Computer Science, vol. 9453. Springer (2015)
34. 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, Part I. Lecture Notes in Computer Science, vol. 8873, pp. 85–104. Springer (2014)
35. Kavun, E., Lauridsen, M., Leander, G., Rechberger, C., Schwabe, P., Yalçın, T.: Prøst v1.1 (June 2014)
36. Maurer, U.M., Renner, R., Holenstein, C.: Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology. In: Naor, M. (ed.) TCC 2004. Lecture Notes in Computer Science, vol. 2951, pp. 21–39. Springer (2004)
37. Mendel, F., Mennink, B., Rijmen, V., Tischhauser, E.: A Simple Key-Recovery Attack on McOE-X. In: Pieprzyk, J., Sadeghi, A., Manulis, M. (eds.) CANS 2012. vol. 7712, pp. 23–31. Springer (2012)
38. Mennink, B., Reyhanitabar, R., Vizár, D.: Security of Full-State Keyed Sponge and Duplex: Applications to Authenticated Encryption. In: Iwata and Cheon [33], pp. 465–489

39. Naito, Y.: Sandwich Construction for Keyed Sponges: Independence Between Capacity and Online Queries. In: Foresti, S., Persiano, G. (eds.) CANS 2016. Lecture Notes in Computer Science, vol. 10052, pp. 245–261 (2016)
40. Naito, Y., Wang, L.: Replacing SHA-2 with SHA-3 Enhances Generic Security of HMAC. In: Sako, K. (ed.) CT-RSA 2016. Lecture Notes in Computer Science, vol. 9610, pp. 397–412. Springer (2016)
41. Naito, Y., Yasuda, K.: New Bounds for Keyed Sponges with Extendable Output: Independence Between Capacity and Message Length. In: Peyrin, T. (ed.) FSE 2016. Lecture Notes in Computer Science, vol. 9783, pp. 3–22. Springer (2016)
42. NIST Special Publication 800-185: SHA-3 Derived Functions: cSHAKE, KMAC, TupleHash and ParallelHash (2016)