The Exact PRF Security of Truncation: Tight Bounds for Keyed Sponges and Truncated CBC

Peter Gaži[1], Krzysztof Pietrzak[1](✉), and Stefano Tessaro[2]

[1] IST, Klosterneuburg, Austria
{gazi,pietrzak}@ist.ac.at
[2] UC, Santa Barbara, USA
tessaro@cs.ucsb.edu

Abstract. This paper studies the concrete security of PRFs and MACs obtained by keying hash functions based on the sponge paradigm. One such hash function is KECCAK, selected as NIST's new SHA-3 standard.

In contrast to other approaches like HMAC, the exact security of keyed sponges is not well understood. Indeed, recent security analyses delivered concrete security bounds which are far from existing attacks.

This paper aims to close this gap. We prove (nearly) exact bounds on the concrete PRF security of keyed sponges using a random permutation. These bounds are tight for the most relevant ranges of parameters, i.e., for messages of length (roughly) $\ell \leqslant \min\{2^{n/4}, 2^r\}$ blocks, where n is the state size and r is the desired output length; and for $\ell \leqslant q$ queries (to the construction or the underlying permutation). Moreover, we also improve standard-model bounds.

As an intermediate step of independent interest, we prove tight bounds on the PRF security of the *truncated* CBC-MAC construction, which operates as plain CBC-MAC, but only returns a prefix of the output.

1 Introduction

Hash functions are popular building blocks for message-authentication codes (MACs) and pseudorandom functions (PRFs) [23]. The latter are *keyed* functions with the property that their outputs (under a secret key) are indistinguishable from random, except with a (small) distinguishing gap ε. PRFs are not only good MACs, but can also be used in a variety of other contexts, from symmetric encryption to key derivation. The to-date most widely used hash-based PRF construction is HMAC [4], and a large body of works has studied its concrete security under different assumptions [3,18,21,26].

It is very likely that hash-based MACs and PRFs will remain popular as the upcoming SHA-3 hash function will replace older designs like MD5, SHA-1 and SHA-256. In contrast to legacy functions, the SHA-3 [1] competition winner KECCAK [10] follows the *sponge* paradigm by Bertoni *et al.* [11]. A key property of sponges is that they resist *extension attacks*, and this enables much simpler approaches than HMAC to derive a PRF. For example, it is suggested (e.g. in [11]) that one may simply pre-prend the key to the message.

R. Gennaro and M. Robshaw (Eds.): CRYPTO 2015, Part I, LNCS 9215, pp. 368–387, 2015.
DOI: 10.1007/978-3-662-47989-6_18

Our contributions, in a nutshell. This paper studies the exact security (i.e., how large is the best distinguishing gap ε?) of keyed sponge constructions. The existing indifferentiability security proof [11], as well as recent targeted analyses [2,8] yield upper bounds on ε for several keying approaches. However, it is not clear that these bounds are the best possible ones. For example, they all degrade *quadratically* in the message length, yet no known generic attacks seem to exploit the message length at all.

In this work, we show that the concrete security of keyed sponges is far superior to what was previously proved, and in particular only minimally depends on the message length. We provide a nearly exact characterization of the PRF security of keyed sponges in the model where the underlying n-bit permutation is random and the adversary is allowed to issue queries to it. We consider both variants where the key is processed as part of the input (as in HMAC) or where the initialization value takes the role of key (akin to NMAC). Our bounds are *tight* for messages whose length does not exceed (roughly) $\min\{2^r, 2^{n/4}\}$ blocks, where r is the output length of the constructions and n is the underlying block length – a constraint satisfied in all envisioned application scenarios.[1]

The key to our results is a tight analysis of truncated CBC, the construction operating as plain CBC-MAC *without* prefix-free encoding, but only returning a subset of the output bits.

Security of Keyed Sponges. The sponge construction relies on an invertible permutation π on n-bit strings.[2] For a parameter $b < n$, it pads the message M into b-bit blocks $M[1], \ldots, M[\ell]$, and keeps a state $S_i \,\|\, T_i$, where $S_i \in \{0,1\}^b$ and $T_i \in \{0,1\}^{n-b}$. It outputs the first r bits of S_ℓ for some[3] $r \leqslant b$, where

$$S_0 \,\|\, T_0 \leftarrow 0^n \ , \quad S_i \,\|\, T_i \leftarrow \pi((S_{i-1} \oplus M[i]) \,\|\, T_{i-1}) \ \text{ for } i = 1, \ldots, \ell.$$

We first consider the keyed construction $\mathsf{GSponge}$ which sets $S_0 \,\|\, T_0$ to equal the n-bit key value. We prove that when this key is secret and random, no attacker making q_C queries of length at most $\ell < 2^{n/4}$ b-bit blocks to $\mathsf{GSponge}$ using a *random* permutation π, and q_π queries to π itself (and to its inverse π^{-1}), can distinguish it from a random function, except with distinguishing gap roughly

$$\varepsilon(q_C, q_\pi, \ell) = O\left(\frac{q_C^2 + q_C q_\pi + \ell q_C}{2^{n-r}} + \frac{\ell q_C^2 + \ell q_C q_\pi}{2^n}\right) .$$

[1] For SHA-3, we have $r \geqslant 224$ and $n = 1600$, and thus processing messages exceeding these lengths is practically impossible.

[2] Naming consistency with the TCBC setting below forces us to deviate from the usual naming in the literature on sponges.

[3] The sponge paradigm also allows for outputs of $r > b$ bits obtained by repeated application of π, an option that does not occur for any of the SHA-3 parameters, and that we will not consider for simplicity in the present paper.

The ideal-permutation model is common for sponge-based constructions, and was used in [2,8,11]. For comparison, the previously best known bound was dominated by a term of much larger magnitude $O((\ell^2 q_C^2 + \ell q_C q_\pi)/2^{n-r})$.[4]

The salient feature of our new bound is that *the length ℓ only affects terms with denominator 2^n, or appears in a term $\ell q_C/2^{n-r}$ linear in q_C*. Therefore, the terms with denominator 2^{n-r} are the dominating ones when $\ell \leqslant \min\{2^{n/4}, 2^r\}$, and in this case, our bound simply becomes of the order $O(\frac{q_C^2 + q_C q_\pi + \ell q_C}{2^{n-r}})$. We also show that this is tight for $\max\{q_C, q_\pi\} \geqslant \ell$, which is a very common scenario. We leave the question of proving tightness of the remaining terms (or, alternatively, of improving our bound) as a challenging open problem.

Our generalized analysis also shows that *with respect to PRF security*, we are not constrained to any block length $b < n$ – we could well XOR n-bit message blocks to the *whole* state. Shorter block lengths can then be enforced by the padding function setting some of the bits to 0 (e.g. the last $n - b$ bits). Note that full, n-bit blocks were already used in the design of the sponge-based MAC construction donkeySponge [9], which is implicitly covered by our result.

Black-box keying. In most scenarios, black-box keying by pre-pending a key to the message is more desirable than altering the initial value. We provide a complete analysis of key-prepending for arbitrary key-length $b \cdot w$ (for simplicity, we assume that the key length fits exactly in w blocks). Our results are in terms of the overall number of queries $q = q_\pi + \ell \cdot q_C$ made to the permutation. We distinguish two cases: If $q^2 \leqslant 2^{n-b}$, then the additional keying step is secure as long as $q \leqslant 2^{bw}$. In contrast, in the high-query regime, the keying step is secure as long as $q \leqslant 2^{bw/2}$, which effectively requires doubling the key length to achieve a similar security level as in the previous case. (This gap is due to the fact that the high-q regime enables meet-in-the-middle attacks.)

We note that a similar analysis was given in [2] concurrent to our work, but their initial proof was incorrect for $w \geqslant 2$. The current version of [2] uses the results from this paper to obtain a correct bound.

Standard-model bounds. We also show improved *standard-model* security of keyed sponges under an assumption on the permutation π introduced by Chang *et al.* [13] and further considered in [2]. The assumption is that a block cipher built from the permutation π as $E_K^\pi(X) = (0^b \,\|\, K) \oplus \pi(X \oplus (0^b \,\|\, K))$ for $X \in \{0,1\}^n$ and $K \in \{0,1\}^{n-b}$, where b is the block length, is a pseudorandom permutation. (Note that this construction is essentially a low-entropy single-key version of the Even-Mansour cipher [19,20]).

Our approach: Truncated CBC. Our analysis of keyed sponges builds on top of a result of independent interest – a tight analysis of truncated CBC. In particular,

[4] We note that the recently proved bound of Andreeva *et al.* [2] is slightly more general and modular, as discussed in the full version [22]. In particular, it uses a somewhat different parametrization of the attacker complexity for the second term $\ell q_C q_\pi/2^{n-r}$, which converges to the above in the worst case, but which can make the term smaller (and incomparable to ours) in some scenarios.

our standard-model bounds on sponges are a direct corollary of our truncated-CBC analysis, whereas our bounds in the random permutation model are obtained by a modification of the proof for truncated CBC.

In its *basic* form, the *cipher block-chaining* mode (or CBC, for short) [15,28] uses a block cipher E with n-bit block size. The input $M \in \{0,1\}^*$ is first padded into n-bit blocks $M = M[1] \dots M[\ell]$, and then for a key K, $\mathsf{CBC}_K(M)$ outputs the value Y_ℓ resulting from the following iterative computation: $Y_0 \leftarrow \mathsf{IV}$ and $Y_i \leftarrow E_K(Y_{i-1} \oplus M[i])$ for all $i \in [\ell]$, where IV is the initialization value, e.g., $\mathsf{IV} = 0^n$. The basic CBC construction is only secure for messages of *equal* length ℓ [5]. Otherwise, one can easily mount an extension attack.

Three (variants of) solutions prevent extension attacks: The first one is prefix-free encoding of messages [33]. The second outputs $E_{K'}(\mathsf{CBC}_K(M))$, under a key K' independent from K. (This has been used in EMAC, developed as part of the RACE project [35]). Also, combinations of these ideas have been used in other constructions, like XCBC [12], TMAC [27], and OMAC [24]. The third solution, considered in this paper, is to use *truncation*, i.e., to only output the first $r < n$ bits of the output. While the first two variants have been extensively analyzed [5–7,25,29–31,33,34,36], we are not aware of any explicit analysis of truncated CBC having ever been published,[5] let alone a tight one.

We prove that no attacker making q queries of length at most $\ell < 2^{n/4}$ to TCBC using a random permutation can distinguish it from a random function, except with distinguishing gap $\varepsilon(q,\ell) = O\left(\frac{q(q+\ell)}{2^{n-r}} + \frac{\ell q^2}{2^n}\right)$. This implies security when the random permutation is replaced by a secure block cipher which is a good PRP. The second term matches the one from the best known analysis of prefix-free CBC [6], whereas we prove that the first term is *tight* for $q \geqslant \ell$.

Our techniques. The analysis of TCBC immediately appears harder than that of related constructions. Existing proofs are based on "Bad event analyses": For example, for encrypted MAC (as in EMAC), one defines the bad event that for two distinct query messages M, M', $\mathsf{CBC}^\pi(M)$ and $\mathsf{CBC}^\pi(M')$ collide, where CBC^π denotes (plain) CBC-MAC using a random permutation π. It is not hard to prove that as long as no such collision occurs, the outputs $\pi'(\mathsf{CBC}^\pi(M))$ are indistinguishable from random for an independent permutation π', and the distinguishing advantage is upper-bounded by the probability of such collisions.[6] This implies indistinguishability when π and π' are replaced by E_K and $E_{K'}$, respectively, for a block cipher E and independent keys K and K'. Similarly, for prefix-free CBC the bad event is that in the evaluation of $\mathsf{CBC}^\pi(M)$, the *last* internal query to π is not *fresh*, i.e., it was already made within the same or an earlier evaluation of CBC^π.

For TCBC, however, if we make a query M, resulting into output Y (consisting of the first r bits of $\mathsf{CBC}^\pi(M)$), we cannot prevent a later query M',

[5] Implicitly, the techniques from sponge analyses [2,8,11] yield non-tight bounds of order $O(\ell^2 q^2/2^{n-r})$.

[6] This notwithstanding, proving bounds on the collision probability is far from trivial [6,34].

with output Y', where M' *is a prefix* of M. Previous machinery only tells us that $\mathsf{CBC}^{\pi}(M)$ and $\mathsf{CBC}^{\pi}(M')$ are unlikely to collide, but this is insufficient to argue randomness and independence of Y and Y'. Moreover, the last query to π within the evaluation of $\mathsf{CBC}^{\pi}(M')$ cannot be fresh, as the same query was made *earlier* within the evaluation of $\mathsf{CBC}^{\pi}(M)$. One cannot swap the order of these queries either, as the choice of M' may well depend *adaptively* on Y.

To deal with this, our proof will crucially use Patarin's H-coefficient technique [32], as recently revisited by Chen and Steinberger [14]: We fix a (deterministic) adversary $\mathcal{A}$ and a *compatible* transcript $(M_1, Y_1), \ldots, (M_q, Y_q)$ (i.e., $\mathcal{A}$ indeed would ask such queries $M_1, \ldots, M_q$ if fed with the corresponding answers $Y_1, \ldots, Y_q$) and then compare the probabilities that such a transcript would indeed occur with $\mathcal{A}$ in the real and in the ideal world, respectively. It is easy to see that the latter ideal-world probability is exactly 2^{-rq}, as all outputs of a random functions on (distinct) inputs $M_1, \ldots, M_q$ are random.

However, the real world (where TCBC is evaluated), is far more complex. We are going to show the probability that $\mathsf{Pr}\left[\mathsf{TCBC}^{\pi}(M_i) = Y_i\right]$ is at least $(1 - \varepsilon)2^{-rq}$, for some small ε, as long as π is uniformly distributed, conditioned on the following being true:

- For every message M_i, the value $Z_i \leftarrow \mathsf{CBC}^{\pi}(M_i)$ is *unique*. (This is equivalent to stating that the π-query leading to the value Z_i in the evaluation of M_i is unique). Recall that the actual output on input M_i consists of the first r bits of Z_i.
- For every message M_i, and every message M_j such that M_i is a prefix of M_j, the value $Z_{i,j} \leftarrow \mathsf{CBC}^{\pi}(M_i \,\|\, m)$ is unique, where m is the first n-bit block in M_j after the end of M_i.

It turns out that those conditions are satisfied also except with some small probability δ. The actual indistinguishability bound happens to be $\varepsilon + \delta$ by the H-coefficient method, but determining both values will be at the core of the proof. While an upper bound on δ follows by using techniques from [6,34], upper-bounding ε will require new techniques.

Our security proof for sponges is very similar, and will essentially rely on the argument that with good probability (roughly $\ell q_{\pi} q_C / 2^n$), queries to π made in the evaluation of the sponge queries and direct queries to π by the attacker are disjoint. However, while this is fairly simple to show when the sponge construction is keyed by setting the initial value (S_0, T_0) to be an n-bit secret key, proving the same statement when the key is input through several absorbing steps turns out to be more involved. We also give a security proof for this more complex setting using techniques inspired by [17].

Standard-model analysis. A recent paper by Chang *et al.* [13] also provides a security analysis of variants of sponge constructions in the *standard* model. We note that (a simple twist of) their very elegant trick reduces the security of the sponge construction with a random IV as the key (this is the construction $\mathsf{GSponge}$) to the security of TCBC for a random permutation *and* the PRP security against ℓq queries of the block cipher E^π described above. Our bounds for TCBC directly yield improved standard-model bounds.

Their technique was generalized further in the recent work of Andreeva *et al.* [2]. Beyond the modularity, the main technical contribution of their work is to reduce (in some contexts) the quantity ℓq in the reduction to the security of E^π. Their contribution is completely orthogonal to ours, and their techniques can be applied in our context.

2 Preliminaries

We denote $[n] := \{1, \ldots, n\}$. Moreover, for a finite set $\mathcal{S}$ (e.g., $\mathcal{S} = \{0,1\}$), we let $\mathcal{S}^n$, $\mathcal{S}^+$ and $\mathcal{S}^*$ be the sets of sequences of elements of $\mathcal{S}$ of length n, of arbitrary (but non-zero) length, and of arbitrary length, respectively (with ε denoting the empty sequence). We denote by $S[i]$ the i-th element of $S \in \mathcal{S}^n$ for all $i \in [n]$. Similarly, we denote by $S[i \ldots j]$, for every $1 \leqslant i \leqslant j \leqslant n$, the sub-sequence consisting of $S[i], S[i+1], \ldots, S[j]$, with the convention that $S[i \ldots i] = S[i]$. Moreover, we denote by $S \,\|\, S'$ the concatenation of two sequences in $\mathcal{S}^*$, and also, we let $S \mid T$ be the usual prefix-of relation: $S \mid T \Leftrightarrow (\exists S' \in \mathcal{S}^* \colon S \,\|\, S' = T)$.

We also let $\mathsf{Fcs}(m, n)$ be the set of functions mapping m-bit strings to n-bit strings, and let $\mathsf{Perm}(n) \subseteq \mathsf{Fcs}(n, n)$ be the set of *permutations* on the set of n-bit strings. We use the shorthand $\mathsf{Fcs}(*, n)$ to denote the set of functions from $\{0,1\}^*$ to $\{0,1\}^n$. Finally, we denote the event that an adversary $\mathcal{A}$, given access to an oracle O, outputs a value y, as $\mathcal{A}^{\mathsf{O}} \Rightarrow y$.

PSEUDORANDOM FUNCTIONS. We consider *keyed* functions $\mathsf{F} : \{0,1\}^\kappa \times \{0,1\}^* \to \{0,1\}^r$ taking a κ-bit key, arbitrary long messages $M \in \{0,1\}^*$ as inputs, and returning an r-bit output. In particular, we denote as F_K the map such that $\mathsf{F}(K, \cdot) = \mathsf{F}_K(\cdot)$. We are typically interested in the security of F as a *pseudorandom function* (or PRF, for short) [23]. This is defined via the following advantage measure, involving an adversary $\mathcal{A}$, such that

$$\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{F}}(\mathcal{A}) := \left| \mathsf{Pr}\left[K \xleftarrow{\$} \{0,1\}^\kappa : \mathcal{A}^{\mathsf{F}_K} \Rightarrow 1\right] - \mathsf{Pr}\left[f \xleftarrow{\$} \mathsf{Fcs}(*, n) : \mathcal{A}^f \Rightarrow 1\right] \right|.$$

We consider constructions $\mathsf{C}[\pi] : \{0,1\}^* \to \{0,1\}^r$ invoking a permutation $\pi \in \mathsf{Perm}(n)$ (we sometimes write C^π instead of $\mathsf{C}[\pi]$), and denote by C the resulting keyed function where the key is a permutation $\pi \in \mathsf{Perm}(n)$ (i.e., there are $2^n!$ key values).

For our analysis of keyed sponges, we are also going to consider constructions $\mathsf{F}^\pi : \{0,1\}^\kappa \times \{0,1\}^* \to \{0,1\}^r$ invoking a *public* randomly chosen permutation $\pi \xleftarrow{\$} \mathsf{Perm}(n)$, i.e., one that can be evaluated *directly* by the adversary. For this case, we use the following notation to express the PRF advantage of $\mathcal{A}$ in the so-called *ideal permutation model*:

$$\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{F},\pi}(\mathcal{A}) := \Bigg| \mathsf{Pr}\left[K \xleftarrow{\$} \{0,1\}^\kappa, \pi \xleftarrow{\$} \mathsf{Perm}(n) : \mathcal{A}^{\mathsf{F}^\pi_K, \pi, \pi^{-1}} \Rightarrow 1\right] - \\ - \mathsf{Pr}\left[f \xleftarrow{\$} \mathsf{Fcs}(*, r), \pi \xleftarrow{\$} \mathsf{Perm}(n) : \mathcal{A}^{f, \pi, \pi^{-1}} \Rightarrow 1\right] \Bigg|.$$

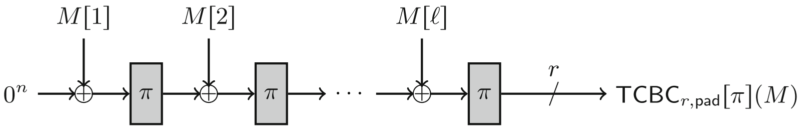

Fig. 1. Truncated CBC $\mathsf{TCBC}_{r,\mathsf{pad}}[\pi]$. Here, $M[1],\ldots,M[\ell]$ are n-bit blocks resulting from applying the padding scheme pad to the input message $M \in \{0,1\}^*$.

MACs and Unpredictability. It is appropriate to note that good PRFs also yield good message-authentication codes (MACs). A concrete security bound for unforgeability can be obtained from our PRF bounds via a standard argument.

3 Truncated CBC and its Security

This first part of the paper deals with the concrete security of truncated CBC (TCBC). On top of being of independent interest, the TCBC analysis of this section will be instrumental to analyze the security of keyed sponges in Sect. 5 below. First off, our analysis of keyed sponges in the ideal permutation model will rely on a modification of the proof for TCBC. Second, our standard-model proofs for keyed sponges will directly apply the TCBC result in a black-box way.

Truncated CBC. We fix two parameters $r < n$ and a *padding scheme* $\mathsf{pad} : \{0,1\}^* \to (\{0,1\}^n)^+$, uniquely encoding arbitrary strings into non-empty sequences of n-bit blocks. (We are *not* requiring the padding to be prefix-free.) The canonical approach computes $\mathsf{pad}(M)$ by appending a single 1-bit to M, and then sufficiently many 0's to reach a length which is a multiple of n.[7]

The (plain) CBC construction for padding scheme pad, using $\pi \in \mathsf{Perm}(n)$, computes $\mathsf{CBC}^{\pi}_{\mathsf{pad}}(M)$ by first producing n-bit blocks $M[1],\ldots,M[\ell] \leftarrow \mathsf{pad}(M)$, and then outputs S_ℓ, where

$$S_0 \leftarrow \mathsf{IV}\ , \quad S_i \leftarrow \pi(M[i] \oplus S_{i-1}) \qquad \text{for all } i = 1,\ldots,\ell. \tag{1}$$

Then, truncated CBC (or TCBC, for short) on input $M \in \{0,1\}^*$, outputs the first $r < n$ bits of CBC evaluated on input M, i.e.,

$$\mathsf{TCBC}^{\pi}_{r,\mathsf{pad}}(M) = \left(\mathsf{CBC}^{\pi}_{\mathsf{pad}}(M)\right)[1\ldots r].$$

Also cf. Fig. 1 for a pictorial representation.

Security analysis. The following theorem characterizes the concrete PRF security of the TCBC construction in the case where π is randomly sampled from $\mathsf{Perm}(n)$. By a standard argument, this implies that TCBC is a secure PRF when π is instantiated with a block cipher which is secure as a pseudorandom permutation (PRP).

[7] In this case, $\mathsf{pad}(M)$ consist of $\ell = \lceil \frac{|M|+1}{n} \rceil$ n-bit blocks.

Theorem 1 (Security of TCBC). *Let $\mathcal{A}$ be a prf-adversary making at most q queries, each of length at most $\ell < 2^{n/4}$ n-bit blocks (after padding). Let $\mathsf{TCBC} = \mathsf{TCBC}_{r,\mathsf{pad}}[\pi]$ for a random permutation $\pi \in \mathsf{Perm}(n)$. Then, for any $t \geqslant 1$,*

$$\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{TCBC}}(\mathcal{A}) \leqslant (6t+17)\,\frac{\ell q^2}{2^n} + \frac{8n \cdot q^2}{2^{n-r}} + \frac{8q\ell}{2^{n-r}} + \frac{2q}{2^n} + \frac{136\ell^4 q^2}{2^{2n}} + \frac{2q^{t+1}\ell^{t+1}}{2^{nt}}\ . \quad (2)$$

The proof of Theorem 1 is found below in Sect. 4, where we also give high-level overviews of the individual components of the proof. Here, we first discuss the bound and its tightness.

Discussion of the bound. First off, note that $q < 2^{(n-r)/2}$ for the above bound to be negligible. We stress in particular that under the constraints $\ell < 2^{n/4}$, the first three terms are the leading ones: Indeed, $2q/2^n$ is always negligible if the other terms are, and the second last term is for sure negligible as long as $\ell < 2^{n/4}$. For the final term, note that $q\ell < 2^{3n/4}$ for the previous terms to be negligible, and the term becomes negligible for $t \geqslant 4$.

Given this, the most important point is that when additionally $\ell < 2^r$, the bound is of the order $O((q^2 + q\ell)/2^{n-r})$, and thus only mildly depends on the length. In the full version, we also show how to break TCBC with a q-query prf-adversary achieving distinguishing advantage roughly $\Omega(q^2/2^{n-r})$. The attack works regardless of the permutation π used to instantiate TCBC. Therefore, the bound is tight when additionally $q \geqslant \ell$. We leave it as an open question to determine tightness for other parameter cases.

4 Proof of Theorem 1

We start with the high level overview of the proof of Theorem 1, which relies on Patarin's H-coefficient technique [32], for which we give a self-contained introduction below. (The notation we use is consistent with the recent revisited version of the framework by Chen and Steinberger [14]).

Roadmap. Sections 4.1 and 4.2 first introduce the notational framework to precisely describe interactions between $\mathcal{A}$ and the given system – i.e., *either* $\mathsf{TCBC}[\pi]$ for $\pi \xleftarrow{\$} \mathsf{Perm}(n)$ *or* a truly random function $f \xleftarrow{\$} \mathsf{Fcs}(*, n)$. Then, Sect. 4.3 will review the H-coefficient method, and apply it to our setting. Finally, Sect. 4.4 will state and explain the individual probabilistic lemmas composing the rest of the proof, and combine them into the theorem.

Simplifying assumption. Throughout the proof, we assume that (1) $\mathcal{A}$ is deterministic, (2) it makes *exactly* q queries, and (3) it never repeats the same query twice. All these assumptions are without loss of generality for an information-theoretic indistinguishability analysis, since any (possibly randomized) adversary making at most q queries can be transformed into one satisfying these constraints and achieving advantage which is at least as large.

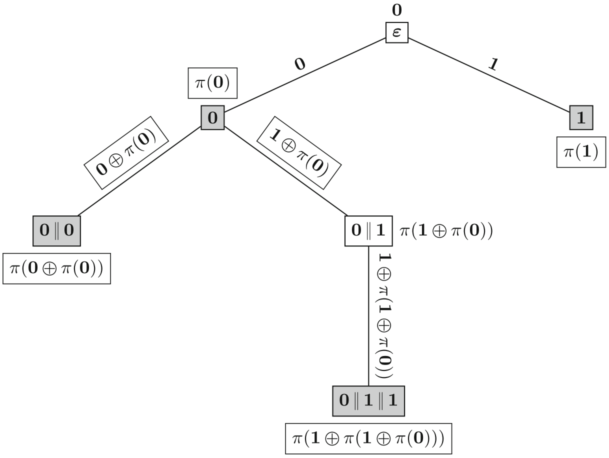

Fig. 2. (Reduced) message tree. Message tree for permutation $\pi \in \mathsf{Perm}(n)$ and four messages $M_1 = \mathbf{0}$, $M_2 = \mathbf{0} \,\|\, \mathbf{0}$, $M_3 = \mathbf{0} \,\|\, \mathbf{1} \,\|\, \mathbf{1}$, and $M_4 = \mathbf{1}$, where $\boldsymbol{b} = b^n$ for $b \in \{0,1\}$. The gray vertices correspond to these four messages. Labels are represented in proximity of the vertices and the edges they are assigned to (as a function of π) and we let $\lambda(\boldsymbol{\varepsilon}) = \mathbf{0} = \mathsf{IV}$. The *boxed* labels are omitted in the reduced message tree.

4.1 Message Trees

We start by introducing some graph-theoretic concepts – the *message tree*, and its reduced version – which capture the inherent combinatorial structure of any q messages $M_1, \ldots, M_q$ queried by the attacker, as well as the internal values computed while these messages are processed by TCBC. Then, we will put these concepts to work to define transcripts describing the adversary's interaction with either of TCBC or a random function f.

We stress that our transcripts will release more information than what is actually seen by the adversary $\mathcal{A}$: This information will make the proof simpler, and will not help substantially in distinguishing TCBC from random.

<u>The message tree.</u> Let $q \geqslant 1$, $\pi \in \mathsf{Perm}(n)$, and let $M_1, \ldots, M_q \in (\{0,1\}^n)^+$ represent the padded versions of the messages. These q messages induce a labeled tree $T^\pi(M_1, \ldots, M_q) = (V, E, \lambda, \gamma)$ – called the *message tree*, and often simply denoted as T or T^π, whenever parameters are clear from the context – defined as follows:

- The set V of vertices of the tree is $V := \left\{M' \in (\{0,1\}^n)^* \,:\, \exists i \in [q] : M' \mid M_i\right\}$, where $\mid$ is the prefix-of partial ordering of strings. In particular, note that the empty string $\boldsymbol{\varepsilon}$ is a vertex.
- The set $E \subseteq V \times V$ of edges is $E := \{(M, M') \,:\, \exists m \in \{0,1\}^n : M' = M \,\|\, m\}$.
- We label vertices and edges recursively. Concretely, we define $\lambda : V \to \{0,1\}^n$ and $\gamma : E \to \{0,1\}^n$. We start with $\lambda(\boldsymbol{\varepsilon}) = \mathsf{IV}$. Then, for every vertex $M \,\|\, m \in$

V where $M \in V$ and $m \in \{0,1\}^n$, we set

$$\lambda(M \,\|\, m) = \pi(\lambda(M) \oplus m).$$

Moreover, we let $\gamma((M, M \,\|\, m)) = \lambda(M) \oplus m$.

An example of a message tree is given in Fig. 2. Note that the vertex labels $\lambda(M)$ are exactly the values of $\mathsf{CBC}[\pi](M)$ while the edge labels correspond to the inputs on which π is invoked. Strictly speaking, edge labels are redundant as they can be reconstructed from the vertex labels and V, but their explicit definition will occasionally simplify descriptions.

For every vertex $M \in V$ (where possibly $M \notin \{M_1, \ldots, M_q\}$), we let $\mathcal{M}_M$ be the set of n-bit blocks m such that $(M, M \,\|\, m) \in E$ and $D_M = |\mathcal{M}_M|$ be the out-degree of vertex M. It is convenient to denote $D_i = D_{M_i}$ and $\mathcal{M}_i = \mathcal{M}_{M_i}$ for all $i \in [q]$. A very useful fact we repeatedly use below is that

$$\sum_{i=1}^{q} D_i < q. \tag{3}$$

This is because every edge $(M_i, M_i \,\|\, m)$ can be uniquely mapped to the shortest message M_j such that $M_i \,\|\, m$ is a prefix of M_j.

The reduced message tree. An abridged version of the above tree, called the *reduced* message tree and denoted $\overline{T}^{\pi} = \overline{T}^{\pi}(M_1, \ldots, M_q)$, will be used in the definition of transcripts below. In particular, $\overline{T}^{\pi}$ is obtained from $T^{\pi}(M_1, \ldots, M_q) = (V, E, \lambda, \gamma)$ as follows. First, we check whether the following condition is true for the given labels λ and γ, and if so, we let $\overline{T}^{\pi} = \bigstar$:

- There exists $i \in [q]$ and $M \in V \backslash \{M_i\}$ such that $\lambda(M_i) = \lambda(M)$; or
- For some $i \in [q]$ and $m \in \mathcal{M}_i$, there exists $M \in V \backslash \{M_i \,\|\, m\}$ such that $\lambda(M_i \,\|\, m) = \lambda(M)$.

This condition is met when a label of an actual message in $\{M_1, \ldots, M_q\}$, or of one of its successor vertices, collides with some other label. (Labels not associated with messages are allowed to collide with each other).

If the above condition is not true, we are going to selectively delete some labels from T (setting them to $\bot$) to obtain a new vertex- and edge-labeled tree, which is the value taken by $\overline{T}$. Specifically,

- For all $i \in [q]$, we let $\lambda(M_i) = \bot$.
- For all $i \in [q]$ and all $m \in \mathcal{M}_i$, we let $\gamma(M_i, M_i \,\|\, m) = \bot$.

In other words, we remove the information necessary to recover the values $\lambda(M_i)$ for all $i \in [q]$.[8]

In Fig. 2, we explicitly show what is omitted when computing the reduced message tree in the case where the tree is not reduced to equal $\bigstar$.

[8] Note, however, that some information about these values can be deduced from the rest of the labels using the fact that π is a permutation. As we will implicitly see below, this information is irrelevant.

4.2 Interactions and Transcripts

We call a sequence of query/answer pairs $(M_1, Y_1), \ldots, (M_q, Y_q)$ *valid* if the adversary $\mathcal{A}$ asks indeed queries $M_1, \ldots, M_q$ when fed with answers $Y_1, \ldots, Y_q$ to its queries. (Since $\mathcal{A}$ is deterministic, the first query M_1 only depends on $\mathcal{A}$, the second query only depends on $\mathcal{A}$ and the first answer Y_1, etc.) Moreover, a valid *transcript* has the form

$$\tau = ((M_1, Y_1), \ldots, (M_q, Y_q), \overline{T}^{\pi}(M_1, \ldots, M_q)),$$

where $(M_1, Y_1), \ldots, (M_q, Y_q)$ is valid, $\pi : \{0,1\}^n \to \{0,1\}^n$ is a permutation, and $\overline{T}^{\pi}(M_1, \ldots, M_q)$ is the reduced message tree for $M_1, \ldots, M_q$ with respect to π. We differentiate between the ways in which such valid transcripts are generated in the real and in the ideal worlds, respectively, by defining corresponding distributions $\mathsf{T}_{\mathsf{real}}$ and $\mathsf{T}_{\mathsf{ideal}}$ over the set of valid transcripts:

Real World. The transcript $\mathsf{T}_{\mathsf{real}}$ for the adversary $\mathcal{A}$ is obtained by sampling $\pi \xleftarrow{\$} \mathsf{Perm}(n)$, and letting

$$\mathsf{T}_{\mathsf{real}} = ((M_1, Y_1), \ldots, (M_q, Y_q), \overline{T}^{\pi}(M_1, \ldots, M_q)),$$

where we execute $\mathcal{A}$, which asks queries $M_1, \ldots, M_q$ answered with $Y_i = \mathsf{TCBC}[\pi](M_i)$ for all $i \in [q]$, and we let $\overline{T}^{\pi}(M_1, \ldots, M_q)$ be the corresponding reduced message tree. Note that because $\mathcal{A}$ is fixed and deterministic, $\mathsf{T}_{\mathsf{real}}$ only depends on π, and thus we occasionally write $\mathsf{T}_{\mathsf{real}}(\pi)$ for the corresponding map.

Ideal World. The transcript $\mathsf{T}_{\mathsf{ideal}}$ for the adversary $\mathcal{A}$ is obtained similarly to the above. However, here we sample *both* a random permutation $\pi \xleftarrow{\$} \mathsf{Perm}(n)$ and q independent random values $Y_1, \ldots, Y_q \xleftarrow{\$} \{0,1\}^r$, and let

$$\mathsf{T}_{\mathsf{ideal}} = \mathsf{T}_{\mathsf{ideal}}(Y_1, \ldots, Y_q, \pi) = ((M_1, Y_1), \ldots, (M_q, Y_q), \overline{T}^{\pi}(M_1, \ldots, M_q)),$$

where $M_1, \ldots, M_q$ are the queries asked when executing $\mathcal{A}$ and answering each query M_i with Y_i, for all $i \in [q]$. We stress that here we are *augmenting* the ideal world with an additional *independent* random permutation π which does not actually exist in the original prf distinguishing game. This is in order to make real- and ideal-world transcripts alike. In particular, the tree $\overline{T}^{\pi}$ is generated according to the permutation π.

Note that the range of $\mathsf{T}_{\mathsf{real}}$ is included in the range of $\mathsf{T}_{\mathsf{ideal}}$ by definition, and that the range of $\mathsf{T}_{\mathsf{ideal}}$ is easily seen to contain all valid transcripts.

4.3 The "H-Coefficient Method": Good and Bad Transcripts

We upper bound the advantage $\mathcal{A}$ in distinguishing $\mathsf{TCBC}[\pi]$ for $\pi \xleftarrow{\$} \mathsf{Perm}(n)$ from a random function using the statistical distance of the transcripts, i.e.,

$$\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{TCBC}}(\mathcal{A}) \leqslant \mathsf{SD}(\mathsf{T}_{\mathsf{real}}, \mathsf{T}_{\mathsf{ideal}}) = \frac{1}{2} \sum_{\tau} \left| \mathsf{Pr}\left[\mathsf{T}_{\mathsf{real}} = \tau\right] - \mathsf{Pr}\left[\mathsf{T}_{\mathsf{ideal}} = \tau\right] \right|, \quad (4)$$

where the sum is over all valid transcripts. This is because a distinguisher for $\mathsf{T}_{\mathsf{real}}$ and $\mathsf{T}_{\mathsf{ideal}}$, whose optimal advantage is exactly $\mathsf{SD}(\mathsf{T}_{\mathsf{real}}, \mathsf{T}_{\mathsf{ideal}})$, can always output the same decision bit as $\mathcal{A}$, ignoring any extra information provided by the transcript.

To this end, we are going to use Patarin's H-coefficient method [32], recently revisited in [14]. Concretely, this means that we need to partition the set of possible transcripts into *good* transcripts GT and *bad* transcripts BT to enable effective usage of the following lemma.

Lemma 1 (The *H*-Coefficient Method). *Let $\delta, \varepsilon \in [0,1]$ be such that:*

(a) $\mathsf{Pr}\left[\mathsf{T}_{\mathsf{ideal}} \in \mathsf{BT}\right] \leqslant \delta$.
(b) *For all $\tau \in \mathsf{GT}$, $\frac{\mathsf{Pr}[\mathsf{T}_{\mathsf{real}}=\tau]}{\mathsf{Pr}[\mathsf{T}_{\mathsf{ideal}}=\tau]} \geqslant 1 - \varepsilon$.*

Then, $\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{TCBC}}(\mathcal{A}) \leqslant \mathsf{SD}(\mathsf{T}_{\mathsf{real}}, \mathsf{T}_{\mathsf{ideal}}) \leqslant \varepsilon + \delta$.

More verbally, we require that with very high probability (i.e., $1-\delta$) a generated transcript *in the ideal world* is going to be in GT, and moreover, for each such good transcript, the probabilities that it occurs in the real and in the ideal worlds are *roughly* the same, i.e., at most a *multiplicative* factor $1 - \varepsilon$ apart.

Transcript-dependent quantities. Concretely, a transcript τ will be defined as "good" if the associated reduced message tree $\overline{T} = (V, E, \gamma, \lambda)$ is not $\bigstar$ and not "too degenerate". This requires introducing two relevant quantities. Before doing so, however, we first note that $\overline{T}$ defines a partial permutation $\overline{\pi}$ on the n-bit strings such that $\overline{\pi}(\gamma(e)) = \lambda(v)$ for every edge e with end-node v with $\gamma(e), \lambda(v) \neq \perp$, and $\overline{\pi}(x) = \perp$ for all other inputs.

We will make use of the following quantities, which connect the outputs $Y_1, \ldots, Y_q$ with $\overline{T}$.

Definition 1. *Let $\tau = ((M_1, Y_1), \ldots, (M_q, Y_q), \overline{T} = (V, E, \gamma, \lambda))$ be a valid transcript with associated partial permutation $\overline{\pi}$. Then, for all $i \in [q]$ we define:*

- *$N_i^{(1)}(\tau)$ is the number of $x \in \{0,1\}^n$ with $\overline{\pi}(x) \neq \perp$ and $\overline{\pi}(x)[1 \ldots r] = Y_i$.*
- *$N_i^{(2)}(\tau)$ is defined as*

$$N_i^{(2)}(\tau) := \left|\{z \in \{0,1\}^n \ : \ z[1 \ldots r] = Y_i \wedge \exists e \in E, m \in \mathcal{M}_i : \gamma(e) = z \oplus m\}\right| .$$

Moreover, for $a \in \{1,2\}$, let $N^{(a)} = \sum_{i=1}^{q} N_i^{(a)}$. If $\overline{T} = \bigstar$, then these values are set to 0.

Let us give some intuition on how the above quantities behave for an ideal-world transcript. Note that $\overline{\pi}$ is defined on at most $q \cdot \ell$ values, and the value $\overline{\pi}(x)$, when first defined, is obtained by sampling a (nearly) uniform random n-bit string. Thus the expectation of $N_i^{(1)}$ is roughly $q\ell/2^r$, and in turn, $N^{(1)}$ should be roughly $q^2\ell/2^r$.

Also, note that $N_i^{(2)}$ is the number of n-bit strings z which are consistent with Y_i in their first r bits which have additionally the property that for some message block $m \in \mathcal{M}_i$, $z \oplus m$ is the (non-$\perp$) label of an edge in the reduced message tree. Here, the intuition is that every edge label $\gamma(e)$ in the partial tree is uniform (this won't be quite true, but let us assume it is), and therefore the expectation of $N_i^{(2)}$ should be (roughly) $D_i q\ell/2^r$, and thus, the expectation of $N^{(2)}$ should also be roughly $q^2\ell/2^r$, using $\sum_i D_i \leqslant q$.

Good transcripts. We require that in a good transcript τ the actual values of $N^{(1)}$ and $N^{(2)}$ are not too far off their (heuristic) expected values we mentioned above. Moreover, we also want that the reduced message tree is not degenerate, i.e., even though we can't see them, we want the guarantee that the labels of the actual messages (and their successors) are unique – the failure to satisfy this would be signalled by $\overline{T} = \bigstar$ by definition.

Definition 2 (Good Transcripts). *Let $\tau = ((M_1, Y_1), \ldots, (M_q, Y_q), \overline{T})$ be a valid transcript. We say that the transcript is* good *(and thus $\tau \in \mathsf{GT}$) if the following properties are true (for $t \geq 1$ as in the theorem statement):*

(1) $\overline{T} \neq \bigstar$.
(2) $N^{(1)}(\tau) \leqslant 3q\,(qt\ell/2^r + n)$.
(3) $N^{(2)}(\tau) \leqslant (2n+1)q^2 + (3t+1)q^2\ell/2^r + 8q^2\ell^4/2^{n+r}$.

We denote as GT the set of all good transcripts, and BT the set of all *bad* transcripts, i.e., transcripts which can possibly occur (i.e., they are in the range of $\mathsf{T}_{\mathsf{ideal}}$) and are not good. More specifically, we denote by BT_i the set of all bad transcripts that do not satisfy the i-th property in the definition of a good transcript above, hence we have $\mathsf{BT} = \bigcup_{i=1}^{3} \mathsf{BT}_i$.

4.4 High-Level Lemmas and Putting Pieces Together

Bounding the ratio. In Sect. 4.5 below, we are going to prove the following lemma.

Lemma 2. *For all good transcripts $\tau \in \mathsf{GT}$,*

$$\frac{\Pr[\mathsf{T}_{\mathsf{real}}=\tau]}{\Pr[\mathsf{T}_{\mathsf{ideal}}=\tau]} \geqslant 1 - \left(\frac{N^{(1)}+N^{(2)}}{2^{n-r}} + \frac{2q^2}{2^{n-r}}\right). \tag{5}$$

Bounding probability of bad transcripts. We now upper bound the probabilities that a transcript sampled according to $\mathsf{T}_{\mathsf{ideal}}$ is bad via the following lemmas, proved in the full version [22] for lack of space.

Lemma 3 (Bad-Transcript Analysis for BT_1). $\Pr[\mathsf{T}_{\mathsf{ideal}} \in \mathsf{BT}_1] \leqslant 16\ell q^2/2^n + 128\ell^4 q^2/2^{2n}$.

Lemma 4 (Bad-Transcript Analysis for BT_2). *For $t \geqslant 1$ as in the theorem statement,* $\Pr[\mathsf{BT}_2] \leqslant q/2^n + (q \cdot \ell)^{t+1}/2^{nt}$.

Lemma 5 (Bad-Transcript Analysis for BT_3). *For all $t \geqslant 1$ as in the theorem statement,* $\Pr[\mathsf{BT}_3] \leqslant q/2^n + 8q\ell/2^{n-r} + (q \cdot \ell)^{t+1}/2^{nt}$.

The proof of Lemma 3 above uses and extends techniques inherited from the work of [6] and in particular their analysis of prefix-free CBC. The proof requires some extra work, since we are considering non-prefix free messages.

One would expect that the proofs of Lemmas 4 and 5 follow by application of a simple Chernoff-like argument. Unfortunately, more work is required: First off, the sampled values are not uniform, but only close to uniform. But more importantly, Lemma 5 requires to prove a concentration bound on a series of random variables (the edge labels) which are defined adaptively by an iterative process when computing the reduced message tree. Our technique will essentially show that most of the edge labels will exhibit a high degree of independence, and only a small number of them will be defined by "recycled values" when generating the tree.

COMBINING PIECES. Therefore, we can apply Lemma 1 using ε and δ extracted from the above lemmas. In particular,

$$\varepsilon = \frac{N^{(1)} + N^{(2)}}{2^{n-r}} + \frac{2q^2}{2^{n-r}} \leqslant \frac{(6t+1)\ell q^2}{2^n} + \frac{8nq^2}{2^{n-r}} + \frac{8q^2\ell^4}{2^{2n}},$$

and

$$\delta = \frac{2q}{2^n} + \frac{8q\ell}{2^{n-r}} + \frac{16\ell q^2}{2^n} + \frac{128\ell^4 q^2}{2^{2n}} + 2\frac{(q\cdot\ell)^{t+1}}{2^{nt}}.$$

In particular, we simplify

$$\varepsilon + \delta \leqslant (6t+17)\,\frac{\ell q^2}{2^n} + \frac{8n\cdot q^2}{2^{n-r}} + \frac{8q\ell}{2^{n-r}} + \frac{2q}{2^n} + \frac{136\ell^4 q^2}{2^{2n}} + \frac{2q^{t+1}\ell^{t+1}}{2^{nt}}.$$

4.5 Lower Bounding the Probability Ratio (Proof of Lemma 2)

We fix a good transcript $\tau = ((M_1, Y_1), \ldots, (M_q, Y_q), \overline{T}) \in \mathsf{GT}$, where $\overline{T} = (V, E, \lambda, \gamma) \neq \bigstar$. To start with, we define the set $\Omega[\tau]$ of π's consistent with τ in the real world, i.e.,

$$\Omega[\tau] := \{\pi \in \mathsf{Perm}(n) \; : \; \mathsf{T}_{\mathsf{real}}(\pi) = \tau\}\,.$$

Moreover, let $\Omega'[\tau]$ be the set of permutations π which are consistent with the labels of the reduced message tree $\overline{T}$, however $\mathsf{TCBC}^{\pi}(M_i)$ does not need to equal Y_i for all i. More formally,

$$\Omega'[\tau] := \left\{\pi \in \mathsf{Perm}(n) \; : \; \overline{T}^{\pi}(M_1, \ldots, M_q) = \overline{T}\right\}.$$

Now, we define

$$\overline{p}(\tau) := \frac{|\Omega[\tau]|}{|\Omega'[\tau]|} = \mathsf{Pr}\left[\pi \xleftarrow{\$} \Omega'[\tau] : \pi \in \Omega[\tau]\right].$$

This is the probability that a random permutation π consistent with the constraints on the *reduced* message tree also yields $\mathsf{TCBC}^{\pi}(M_i) = Y_i$ for all $i \in [q]$.[9]

[9] Note that sampling such a π is *not* the same as sampling a random π which is consistent with $\overline{\pi}$. The latter may allow for some permutations which are not possibly generating a message tree which can be reduced to $\overline{T}$.

The following claim will reduce lower bounding the probability ratio to lower bounding $\overline{p}(\tau)$ for $\tau \in \mathsf{GT}$, and its proof is omitted here.

Claim (1). For all good transcripts $\tau \in \mathsf{GT}$, $\frac{\mathsf{Pr}[\mathsf{T}_{\mathsf{real}}=\tau]}{\mathsf{Pr}[\mathsf{T}_{\mathsf{ideal}}=\tau]} = 2^{r \cdot q} \cdot \overline{p}(\tau)$.

It is easy to see that the ordering of $(M_1, Y_1), \ldots, (M_q, Y_q)$ does not affect $\overline{p}(\tau)$, and we therefore assume without loss of generality that it is prefix-preserving, i.e., if $M_i \mid M_j$, then $i < j$. Let e_i be the edge leading to M_i.

To study $\overline{p}(\tau)$, we consider an *iterative process* where we extend $\overline{\pi}$ defined by $\overline{T}$ as above, setting the values of $\overline{\pi}(\gamma(e_i)) = \lambda(M_i)$ for $i = 1, \ldots, q$ one after the other in this order. Moreover, upon setting $\lambda(M_i) = \overline{\pi}(\gamma(e_i)) \leftarrow Z_i$, for all $m \in \mathcal{M}_i$, we do the following:

- We set $\gamma(M_i, M_i \,\|\, m) \leftarrow Z_i \oplus m$
- If we know the value $\lambda(M_i \,\|\, m)$, we set $\overline{\pi}(Z_i \oplus m) \leftarrow \lambda(M_i \,\|\, m)$.[10]

Note that depending on the choice of the Z_i's, the resulting $\overline{\pi}$ may or may not be a partial permutation, or we may overwrite values, etc. We will of course be only interested in sequences of Z_i's which maintain the permutation property.

To this end, let $\mathcal{L} = \mathcal{L}(\overline{T}, (M_1, Y_1), \ldots, (M_q, Y_q)))$ be the set of sequences $(z_1, \ldots, z_q)$ of *distinct* q values such that $z_i[1 \ldots r] = Y_i$ for all $i \in [q]$ and when assigning $\lambda(M_i) \leftarrow z_i$ for all $i \in [q]$ in the above process, at the end of the process the labels $\lambda(M_i) = z_i$ are unique (i.e., no other vertex has the same label) and moreover, for all $i \in [q]$ and all $m \in \mathcal{M}_i$, we also have that $\lambda(M_i \,\|\, m)$ is a unique label.

The following claim is proved in the full version [22] and reduces the problem of lower bounding $\overline{p}(\tau)$ to that of lower bounding the size of $\mathcal{L}$.

Claim (2). For all good transcripts $\tau \in \mathsf{GT}$, $\overline{p}(\tau) \geq \frac{|\mathcal{L}|}{2^{nq}}$.

THE LOWER BOUND ON $|\mathcal{L}|$. Here, to lower bound $|\mathcal{L}|$, we go through the above process, and assuming $z_1, \ldots, z_{i-1}$ have been fixed, we see how many ways we still have to fix z_i satisfying the invariant that it is still possible to reach sequence $(z_1, \ldots, z_q) \in \mathcal{L}$. In particular, at every step, we are going to exclude values z_i with the following properties:

(1) $z_i[1 \ldots r] \neq Y_i$
(2) There exists $1 \leqslant j < i$ such that $z_j = z_i$.
(3) There exists $M \notin \{M_1, \ldots, M_q\}$ with $\lambda(M) = z_i$.
(4) There exists $1 \leqslant j < i$, $m' \in \mathcal{M}_j$, $m \in \mathcal{M}_i$ such that $m \oplus z_i = m' \oplus z_j$.
(5) There exists a n-bit value $m \in \mathcal{M}_i$ and an edge $e \in E$ with tail node not in $\{M_1, \ldots, M_q\}$ such that $\gamma(e) = z_i \oplus m$.

It is clear that we reach a sequence in $\mathcal{L}$ if at every step we pick a non-excluded value. In particular, note that **(4)** and **(5)** are necessary for us to ensure that the edge labels leading to successor vertices of M_i are *fresh*, which is necessary to ensure that the sequence is in $\mathcal{L}$.

[10] Note that if for some $m \in \mathcal{M}_i$, we have $\lambda(M_i \,\|\, m) = \perp$, then $(M_i, m) = e_j$ for $j > i$, and will be set later in the process.

Now, for every i, note that due to condition **(1)** there are initially 2^{n-r} possible values for z_i, i.e., all strings with the first r bits equal to Y_i. However, we need to remove all strings satisfying any of **(2)**–**(5)** above. These can be counted as follows:

(2) There are at most $i \leqslant q$ such values.
(3) In order for M to be such that $\lambda(M) = z_i$, we need to have $\lambda(M)[1 \dots r] = Y_i$, but we know that there are at most $N_i^{(1)}$ such vertices by definition.
(4) Note that for every $j \in [i-1]$, there are exactly D_j possible values $m' \in \mathcal{M}_j$ which can be combined with a value $m \in \mathcal{M}_i$ (there are D_i of those) to get a possible "forbidden" value $z_i = z_j \oplus m \oplus m'$, and thus we need to exclude $D_i \cdot \sum_{j=1}^{i-1} D_j \leqslant q \cdot D_i$ possible values.
(5) This is exactly the definition of $N_i^{(2)}$.

Therefore, we can now lower bound $|\mathcal{L}|$ as

$$\begin{aligned} |\mathcal{L}| &\geqslant \prod_{i=1}^{q} (2^{n-r} - N_i^{(1)} - N_i^{(2)} - q - q \cdot D_i) \\ &\geqslant 2^{q \cdot (n-r)} \cdot \left(1 - \frac{N^{(1)} + N^{(2)}}{2^{n-r}} - \frac{2q^2}{2^{n-r}}\right), \end{aligned} \tag{6}$$

where we used the fact that $\prod_i (1 - x_i) \geqslant 1 - \sum_i x_i$, and that $\sum_{i=1}^{q} q \cdot D_i \leqslant q^2$.

5 Security Analysis of Sponge-Based PRFs

In this final section, we turn to discussing security of sponge-based PRFs. We first discuss the constructions considered in this section.

Sponge-Based MAC. As in the TCBC case above, we fix parameters n, r and an injective padding scheme $\mathsf{pad} : \{0,1\}^* \to (\{0,1\}^n)^+$. Then, the construction $\mathsf{Sponge} = \mathsf{Sponge}_{r,\mathsf{pad}}[\pi] : \{0,1\}^\kappa \times \{0,1\}^* \to \{0,1\}^r$, using a permutation $\pi \in \mathsf{Perm}(n)$, on input $M \in \{0,1\}^*$ and key $K \in \{0,1\}^\kappa$, first computes $K[1] \dots K[w] M[1] \dots M[\ell] \leftarrow \mathsf{pad}(K \,\|\, M)$. Then, it outputs $S_\ell[1 \dots r]$ (the first r bits of S_ℓ), where

$$\begin{aligned} V_0 \leftarrow 0^n \;, \quad & V_i \leftarrow \pi(K[i] \oplus V_{i-1}) \text{ for } i = 1, \dots, w, \\ S_0 \leftarrow V_w \;, \quad & S_i \leftarrow \pi(M[i] \oplus S_{i-1}) \text{ for } i = 1, \dots, \ell. \end{aligned}$$

We are explicitly assuming (for simplicity) that the (padded) keys and the actual message end up in different blocks, and hence our naming conventions.[11]

Different from the actual hash-function instantiations, the presented Sponge construction is *more general* in that it allows for processing *n-bit* input blocks in the absorption phase. We can retrieve the originals sponge construction and

[11] Our results can be extended to the more general case, but we avoid the notational overhead in this version of the paper.

SHA-3 instantiations as special case — shorter blocks can be enforced by the padding function pad, which we only require to be injective, but an added benefit of our analysis is that it shows that such shorter blocks are not necessary. The construction $\mathsf{Sponge}_{r,\mathsf{pad}}[\pi]$ using a customary padding pad_b that enforces b-bit blocks is depicted in Fig. 3.

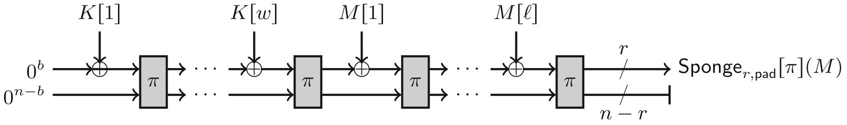

Fig. 3. Sponge construction. Representation of $\mathsf{Sponge}_{r,\mathsf{pad}_b}[\pi]$ used with a padding scheme pad_b that enforces b-bit blocks.

We also consider a variant of the construction – called GSponge – that takes an n-bit key K and differs from Sponge in that it directly sets $S_0 \leftarrow K$ instead of absorbing the key to obtain V_w. The construction is similar to some other MAC designs such as donkeySponge [9] and Pelican [16].

Security analysis of GSponge. We prove the following theorem:

Theorem 2 (Security of GSponge). *Let $\mathcal{A}$ be a* prf*-adversary in the ideal-permutation model, making at most q_π queries to π and at most q_C queries of length at most $\ell < 2^{n/4}$ blocks to the construction (either $\mathsf{GSponge}_{r,\mathsf{pad}}[\pi]$ for a random n-bit key K or a random function). Then, for all $t \geqslant 1$,*

$$\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{GSponge}_{r,\mathsf{pad}},\pi}(\mathcal{A}) \leqslant \frac{(6t+17)\ell q_C^2 + 7\ell q_\pi q_C + 2q_C}{2^n} + \frac{6nq_C^2 + 8\ell q_C + q_\pi q_C}{2^{n-r}} + \frac{136\ell^4 q_C^2}{2^{2n}} + \frac{2(\ell q_C)^{t+1}}{2^{nt}} . \qquad (7)$$

This bound substantially improves the previously known bound from [8], which was of the order $O(\frac{\ell^2 q_C^2 + \ell q_C q_\pi}{2^{n-r}})$. (We discuss the subtleties of the bound in [2] in detail in the full version [22]). For sufficiently large t and for $\ell < 2^{n/4}$, the first two terms are the leading terms. If we additionally assume that $\ell < 2^r$, then the bound becomes of order $O(\frac{q_C^2 + q_C q_\pi + \ell q_C}{2^{n-r}})$. In the full version [22], we prove that this is tight when additionally $\max\{q_\pi, q_C\} \geqslant \ell$.

The proof of Theorem 2 adapts the proof strategy of Theorem 1 to the setting of sponges. The GSponge and TCBC constructions are in fact the same, with the main difference that the initial value in GSponge is set to a random secret key and the underlying permutation π can be evaluated by the adversary. Intuitively, however, one can show that thanks to the random secret key, no internal permutation query in a construction query intersects with a direct permutation query by the attacker, except with probability $O(\ell \cdot q_C q_\pi / 2^n)$. Conditioned on the event that such intersection does not occur, the distinguishing bound (in terms of construction queries) is roughly the same as the one of TCBC.

Replacing the Uniform Key. We now address security of the Sponge construction when using the customary padding pad_b, where the ($\kappa = w \cdot b$)-bit key K is first split into w b-bit blocks as $K[1] \cdots K[w]$, each of them is padded with $n - b$ trailing zeroes and absorbed by the construction, as depicted in Fig. 3. The proof of the following theorem is found in the full version [22], and relies on a detailed analysis of the key absorption mechanism which shows that the behaviors of GSponge and Sponge are indistinguishable given enough key material.

Theorem 3 (Security of Sponge). *Let $\mathcal{A}$ be a* prf*-adversary in the ideal-permutation model, making at most q_π queries to π and at most q_C queries of length at most $\ell < 2^{n/4}$ blocks to the construction (either $\mathsf{Sponge}_{r,\mathsf{pad}_b}[\pi]$ with the padding pad_b and a random $(w \cdot b)$-bit key, or a random function). Then, for all $t \geqslant 1$, and $q = q_\pi + \ell q_C < 2^{n-b}$, we have*

$$\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{Sponge}_{r,\mathsf{pad}_b},\pi}(\mathcal{A}) \leqslant A_t(q_c, q_\pi, \ell) + \frac{wq}{2^n} + \min\left\{ \frac{q}{2^{\frac{b-\log(3n)-1}{2}w}} \; , \; \frac{q}{2^{bw}} + \frac{q^2}{2^{n-b}} \right\},$$

where A_t denotes the expression on the right-hand side of inequality (7). Moreover, if $w = 1$ then one can replace the whole min*-term by $q/2^{bw}$.*

We remark that our proof is highly involved for the case where $q^2 > 2^{n-b}$, where $q = q_\pi + q_C \cdot \ell$ is the overall number of queries to π in the experiment, and requires an adaptation of combinatorial techniques proposed in [17].

The standard-model bounds. We combine an approach by Chang *et al.* [13] (also used in [2]) with our improved bound for TCBC. In particular, we measure security of the underlying permutation π in terms of the advantage $\mathsf{Adv}^{(r,\oplus)\text{-}\mathsf{prp}}_{\pi}(\mathcal{B})$ of an adversary in distinguishing the map $M \mapsto (0^r \parallel K) \oplus \pi(M \oplus (0^r \parallel K))$ under a random secret key $K \overset{\$}{\leftarrow} \{0,1\}^{n-r}$ from $\tau \overset{\$}{\leftarrow} \mathsf{Perm}(n)$. In the full version [22], we prove and discuss the following theorem.

Theorem 4 (Standard-Model Security of GSponge). *Let $\pi \in \mathsf{Perm}(n)$ and $\mathsf{pad} : \{0,1\}^* \to (\{0,1\}^n)^+$ a padding scheme. Let $\mathcal{A}$ be a* prf*-adversary making at most q queries, each of length at most $\ell < 2^{n/4}$ n-bit blocks (after padding). Then, there exists an $(\oplus, r)$-*prp*-adversary $\mathcal{B}$ such that for any $t \geqslant 1$,*

$$\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{GSponge}_{r,\mathsf{pad}}[\mathsf{E}]}(\mathcal{A}) \leqslant \mathsf{Adv}^{(r,\oplus)\text{-}\mathsf{prp}}_{\pi}(\mathcal{B}) + B(q, \ell, n, r, t),$$

where $\mathcal{B}$ has $\mathsf{Time}(\mathcal{B}) = \mathsf{Time}(\mathcal{A}) + O(q \cdot \ell)$ and makes at most $q \cdot \ell$ permutation queries, and $B(q, \ell, n, r, t)$ is the term on the right-hand-side of Theorem 1.

Acknowledgments. We thank Mihir Bellare for insightful feedback, and Bart Mennink for technical comments. Gaži and Pietrzak's work was partly funded by the European Research Council under an ERC Starting Grant (259668-PSPC). Tessaro's research was partially supported by NSF grant CNS-1423566, by a gift from the Gareatis Foundation, and by the Culler Chair. Part of this work was done while the third author was visiting IST Austria.

References

1. SHA-3 standard. National Institute of Standards and Technology (NIST), Draft FIPS Publication 202, U.S. Department of Commerce, April 2014
2. Andreeva, E., Daemen, J., Mennink, B., Van Assche, G.: Security of keyed sponge constructions using a modular proof approach. In: FSE 2015. LNCS (2015, to appear)
3. Bellare, M.: New proofs for NMAC and HMAC: security without collision-resistance. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 602–619. Springer, Heidelberg (2006)
4. Bellare, M., Canetti, R., Krawczyk, H.: Keying hash functions for message authentication. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 1–15. Springer, Heidelberg (1996)
5. Bellare, M., Kilian, J., Rogaway, P.: The security of cipher block chaining. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 341–358. Springer, Heidelberg (1994)
6. Bellare, M., Pietrzak, K., Rogaway, P.: Improved security analyses for CBC MACs. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 527–545. Springer, Heidelberg (2005)
7. Bernstein, D.J.: A short proof of the unpredictability of cipher block chaining (2005). http://cr.yp.to/antiforgery/easycbc-20050109.pdf
8. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: On the security of the keyed sponge construction. In: Symmetric Key Encryption Workshop (SKEW), February 2011
9. Bertoni, G., Daemen, J., Peeters, M.: Permutation-based encryption, authentication and authenticated encryption. In: Directions in Authenticated Ciphers (2012)
10. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Keccak. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 313–314. Springer, Heidelberg (2013)
11. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: On the indifferentiability of the sponge construction. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 181–197. Springer, Heidelberg (2008)
12. Black, J., Rogaway, P.: CBC MACs for arbitrary-length messages: the three-key constructions. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 197–215. Springer, Heidelberg (2000)
13. Chang, D., Dworkin, M., Hong, S., Kelsey, J., Nandi, M.: A keyed sponge construction with pseudorandomness in the standard model. In: Proceedings of the Third SHA-3 Candidate Conference (2012)
14. Chen, S., Steinberger, J.: Tight security bounds for key-alternating ciphers. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 327–350. Springer, Heidelberg (2014)
15. Computer data authentication. National Bureau of Standards, NBS FIPS PUB 113, U.S. Department of Commerce, May 1985
16. Daemen, J., Rijmen, V.: The mac function pelican 2.0. Cryptology ePrint Archive, Report 2005/088 (2005). http://eprint.iacr.org/
17. Dai, Y., Lee, J., Mennink, B., Steinberger, J.: The security of multiple encryption in the ideal cipher model. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 20–38. Springer, Heidelberg (2014)
18. Dodis, Y., Ristenpart, T., Steinberger, J.P., Tessaro, S.: To hash or not to hash again? (In)differentiability results for h^2 and HMAC. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 348–366. Springer, Heidelberg (2012)

19. Dunkelman, O., Keller, N., Shamir, A.: Minimalism in cryptography: the even-mansour scheme revisited. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 336–354. Springer, Heidelberg (2012)
20. Even, S., Mansour, Y.: A construction of a cipher from a single pseudorandom permutation. J. Cryptology **10**(3), 151–162 (1997)
21. Gaži, P., Pietrzak, K., Rybár, M.: The exact PRF-security of NMAC and HMAC. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 113–130. Springer, Heidelberg (2014)
22. Gaži, P., Pietrzak, K., Tessaro, S.: The exact PRF security of truncation: Tight bounds for keyed sponges and truncated CBC. Cryptology ePrint Archive, Report 2015/053 (2015). Full version of this paper
23. Goldreich, O., Goldwasser, S., Micali, S.: On the cryptographic applications of random functions. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 276–288. Springer, Heidelberg (1985)
24. Iwata, T., Kurosawa, K.: OMAC: one-key CBC MAC. In: Johansson, T. (ed.) FSE 2003. LNCS, vol. 2887, pp. 129–153. Springer, Heidelberg (2003)
25. Iwata, T., Kurosawa, K.: Stronger security bounds for OMAC, TMAC, and XCBC. In: Johansson, T., Maitra, S. (eds.) INDOCRYPT 2003. LNCS, vol. 2904, pp. 402–415. Springer, Heidelberg (2003)
26. Koblitz, N., Menezes, A.: Another look at HMAC. Cryptology ePrint Archive, Report 2012/074 (2012). http://eprint.iacr.org/2012/074
27. Kurosawa, K., Iwata, T.: TMAC: two-key CBC MAC. In: Joye, M. (ed.) CT-RSA 2003. LNCS, vol. 2612, pp. 33–49. Springer, Heidelberg (2003)
28. Information technology security techniques message authentication codes (macs) part 1: Mechanisms using a block cipher. ISO/IEC 9797–1 (1999)
29. Maurer, U.M.: Indistinguishability of random systems. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 110–132. Springer, Heidelberg (2002)
30. Minematsu, K., Matsushima, T.: New bounds for PMAC, TMAC, and XCBC. In: Biryukov, A. (ed.) FSE 2007. LNCS, vol. 4593, pp. 434–451. Springer, Heidelberg (2007)
31. Nandi, M.: A simple and unified method of proving indistinguishability. In: Barua, R., Lange, T. (eds.) INDOCRYPT 2006. LNCS, vol. 4329, pp. 317–334. Springer, Heidelberg (2006)
32. Patarin, J.: The "Coefficients H" technique (invited talk). In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) SAC 2008. LNCS, vol. 5381, pp. 328–345. Springer, Heidelberg (2009)
33. Petrank, E., Rackoff, C.: CBC MAC for real-time data sources. J. Cryptology **13**(3), 315–338 (2000)
34. Pietrzak, K.: A tight bound for EMAC. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 168–179. Springer, Heidelberg (2006)
35. Vandewalle, J., Chaum, D., Fumy, W., Jansen, C.J.A., Landrock, P., Roelofsen, G.: A european call for cryptographic algorithms: RIPE; race integrity primitives evaluation. In: Quisquater, J.-J., Vandewalle, J. (eds.) EUROCRYPT 1989. LNCS, vol. 434, pp. 267–271. Springer, Heidelberg (1990)
36. Vaudenay, S.: Decorrelation over infinite domains: the encrypted CBC-MAC case. In: Stinson, D.R., Tavares, S. (eds.) SAC 2000. LNCS, vol. 2012, pp. 189–201. Springer, Heidelberg (2001)